TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Information & Networks Laboratory

'A Survey on the Cooperation of Network Content Caching with Recommendation Systems'

Emmanouil Sofikitis

Examining Committee:

Professor Michael Paterakis, Supervisor

Professor Athanasios Liavas

Professor Aggelos Bletsas

M.Sc. Diploma Thesis

March 2022

Acknowledgments

First and foremost, I would like to thank all the people involved in the department of Electrical & Computer Engineering of Technical University of Crete for making this journey an incredible experience.

I would like to express my sincere gratitude to my thesis advisor Prof. Michael Paterakis, for guiding me throughout this challenging process. I am thankful and indebted to him for sharing his valuable expertise, and for his support and unceasing encouragement. In addition, I would also like to thank the rest of my thesis committee: Prof. Athanasios Liavas and Prof. Aggelos Bletsas who toke the time and effort to examine my thesis.

Last but not least, I wish to thank my family for their support throughout my studies. Without them, this thesis and the completion of my studies would never be possible.

Abstract

The ease of access to the Internet and the fast pace, in which multimedia content is created and consumed by cellular and wired network users, has led to an unforeseen burst in network traffic. Content Caching has played a crucial role in providing a stable and high quality Internet service. However as cellular networks are becoming denser, with the increase in mobile users, and multimedia content catalogues are increasing in volume with a growing rate, it is unlikely that traditional approaches in caching techniques, are adequate to sustain the Quality of Service that users expect, while ensuring a reasonable profit for Network Operators. To this end, recent studies on network offloading techniques have shown that the cooperation of Recommendation Systems and Network Content Caching may prove fruitful. In this thesis we aspire to provide a guide through this line of research. These studies attempt to break new ground on how these two mechanisms can be combined, an approach that up until recently was not considered, since caching and recommendations aim by definition to different directions, regarding the popularity of contents that users consume.

Contents

Chapter 1
Introduction

Content Caching is widely used to alleviate network load and enhance the Quality of Experience (QoE) of users, by carefully choosing content that will potentially serve a large portion of user demand. Recommendation Systems (RS) on the other hand aim to personalize the content to be served according to the preferences of each individual user. Therefore, it is obvious that these two mechanisms have opposing goals, i.e., caching efficiency is increased when users' requests tend to be similar, whereas a Recommendation System helps users to explore content that better suits their unique taste, making content popularity distribution more volatile and harder to predict. In this thesis we discuss how recent studies combine these seemingly conflicting mechanisms in order to further improve the QoE of users while at the same time reduce resource usage in networks.

1.1. Network Content Caching

Network Content Caching (NCC) is an essential part of both Wired and Cellular Networks. Utilizing Content Delivery Networks (CDNs) and employing caches at the edge of Cellular Networks are common methods nowadays, aiming to cope with the increasing network traffic. Online caching strategies were first appeared in the 1970s and have been extensively studied and significantly improved since then. From traditional CDNs to newly proposed femtocaching techniques [7] and Device-to-Device (D2D) solutions, caching for telecommunication networks is rapidly evolving and has enabled the modern Internet landscape as we know it.

Recent studies, however, show that caching efficiency is reaching a point at which it cannot further cope with the rapid increase of network traffic and the number of cellular users. Reportedly, IP video traffic will grow four-fold from

2017 to 2022 and mobile data traffic is expected to increase seven-fold between 2017 and 2022 [1]. Common techniques, like increasing storage capacity of caches or making CDNs or edge caching access points denser, would not suffice to keep up with this trend mentioned above. Especially in cellular networks where users' mobility and the densification of network cells are creating an even more complicated setup.

Therefore, a new approach is starting to gain attention, i.e., blending Recommendation Systems, which are used by almost every content platform, and Content Caching to increase the potential gains of caching, in order to facilitate the envisioned increase in devices' density and the network traffic burst in the near future. This idea is relatively new, having appeared for the first time, to our best knowledge, in 2012 by M. Verhoeyen et al. Since then many studies have been published considering the interplay of Caching and Recommendation Systems. We will discuss the work of these studies in the following chapter.

1.2. Multimedia Content Recommendation Systems

With the burst of multimedia files streamed and uploaded online and the vast catalogues available to users, Content Providers started employing a Recommendation System into their platforms to enhance user experience. A well-designed RS helps users to better navigate through the enormous number of multimedia content, by identifying each user's unique preferences and providing suggestions that fit into these preferences. In addition, a user is more likely to consume more content when the recommendations are satisfactory; an important incentive for Content Providers. For example, Netflix reported that 80% of video requests in its platform originate from recommendation lists [2] and a significant percentage, of around 50%, is also reported by YouTube [3]. Considering these statistics, it is clear that Recommendation Systems can have a great impact on the popularity distribution of contents.

There is a significant interest in research for more efficient algorithms and techniques for Recommendation Systems. Creating a Recommendation System that can successfully meet the requirements mentioned above can be a rather complicated task, and often poor recommendations can drive users away from a multimedia content platform. The most common way to generate recommendations for a user is to create a user profile, which helps to identify which content is more likely to be interesting for this user. Depending on the application, the data used to form a user’s profile can be derived by the viewing history of the user, the ratings given to previously requested items, or even information from external sources like social or demographic factors.

1.3. Thesis Scope

The scope of the present thesis is to provide a map of the studies on the cooperation of Network Content Caching and Recommendation Systems for the interested reader and for future researchers. We present the work and briefly discuss the main point of each study. A significant difference can be noticed within these studies, regarding the extent in which the proposed model interferes with the quality of the RS output. In our opinion, this distinction plays a fundamental role in the design of future caching setups, therefore, we classify these studies in three categories. The first category refers to studies, in which the recommendations produced by the RS are altered in an effort to drive users’ requests to already cached content. The second category contains studies that do not interfere with the output of the RS, but use the information generated by the RS, in order to improve caching efficiency. Finally, the studies that fall into the third category propose a hybrid approach, where both caching and recommendations are subject to changing.

Chapter 2
Literature on the Interplay of Caching and Recommendation Systems

2.1. Cache-aware Recommendations

Having observed the strong effects of Recommendation Systems in shaping the users' demand, the studies that fall under the first category, aim to improve the efficiency of the cache by steering the preference of the users towards the cached contents. They manage to do so either implicitly by altering the recommendation list, offered to the users through the Content Provider's service, or either explicitly by informing the users of the cached contents available to them. In the latter case it is assumed that the users will prefer these contents over the originally requested ones because of the better streaming quality and the significantly lower delays that come along with the selection of cached contents (better Quality of Experience).

Motivated by the large disparity between the average and the peak traffic demand in the wireless mobile networks and the evident large-timescale user demand predictability, the authors in [4] propose a framework for optimal proactive resource allocation for wireless networks. The main goal is to even out the network traffic throughout the day. They construct a *demand profile* for each user which is used to determine proactive data downloads on users' equipment (UE) – which occur during the off-peak time periods of the network - in a way that minimizes the time-average expected cost. Furthermore, they investigate the improvements offered by slightly modifying the demand profiles of the users (*demand shaping*) - an equivalent of altering the recommendations of the system. In addition to the network offloading achieved by this approach there is a substantial promise for high Quality of Service (QoS) data delivery. Thus, it provides an incentive for users' cooperation with demand shaping.

The work in [5] focuses on the YouTube users' behavior regarding the recommendation lists offered by YouTube. The authors deducted a survey with YouTube traces and they observed that users prefer video files from the recommendation list over other sources, e.g. manual search, with a relatively high percentage, ranging from 33% to 46%, and that users usually select videos ranked higher on the recommendation list. More specifically 50% of the videos selected belong to the top 5 items on the list and 80% of the videos to the top 10 items. With these results in mind they propose a cache-based reordering of the items in the recommendation list, where cached videos are pushed to the top of the list while not cached videos are pushed lower. While this solution interferes with the original recommendations generated by YouTube, in this work it is shown that YouTube does not assign the videos in the related lists to specific positions for some internal purpose, and therefore, reordering the list does not interfere with the original goal of YouTube. In addition, the reordering of the recommendation list is a low-cost approach, with substantial benefits.

A more aggressive approach in this category is presented in [6]. The authors propose to "move away from trying to satisfy every possible user request, and instead try to satisfy the user", meaning, regarding the latter, to enhance to overall user's experience. Considering a setup of a cellular network where users are associated with a local cache, located on the edge of the network, the authors propose the *recommendation* or the *delivery* of an alternative - to the originally requested - content that will effectively reduce the cellular network traffic, and at the same time will improve the quality of experience of the users. They also extend the network setup to consider the complete problem with cache overlaps (referred to as "femtocaching" [7]). When a user requests a content that cannot be served locally by the associated caches, a cache-aware recommendation plugin recommends a set of related contents that are locally available. If the user chooses to accept one of the recommendations then a *soft cache hit* occurs. In a more extreme scenario the Recommendation System directly delivers a related content to the user that is cached, instead of the originally requested content. In both scenarios the authors monitor the satisfaction of the user regarding the quality of the recommendations. To maximize the Soft Cache Hit Ratio of their system they formulate optimization problems for the two different network setups and the two content recommendation scenarios, and they propose approximation algorithms to solve these problems efficiently.

The work in [8] proposes a model for recommendation-driven sequential user requests which captures the behavior of users observed in practice in a number of popular applications. The main motivation of the authors in [8] is to provide a practical software-based solution that can improve caching efficiency, at a low cost. Their proposal is to not try to further improve what is stored at each cache, but rather to better exploit the already cached content by using the existing Recommendation System - employed in numerous Content Providers' applications - to steer users' requests towards the cached contents. By assigning a fixed probability a, to whether a user chooses to request a file from the recommendation list or not, after the users has consumed a media file, they formulate a Markov chain that models the sequential request process of users. Considering the above request process and by imposing a simple binary cost model of providing each file, they formulate an optimization problem to minimize the total cost of content delivery. The optimization problem has a constraint to ensure that the quality of recommendations is above a desired threshold (defined by the authors). This constraint justifies the fact that probability a is fixed, an assumption that would be otherwise unrealistic, as low quality of recommendations would make users lose their trust in the Recommendation System.

In [9] the authors focus their work on the mobile content delivery setup with caches installed on the edge of the cellular network. They are motivated by the lack of results for the joint caching and recommendation paradigm, that are based on realistic setups with real users. They argue that, existing works on this subject have proven the multiple benefits of this cooperation in both network efficiency and user satisfaction, yet only on simulation environment by using synthetic or public datasets that do not originate from real delivery services.

Therefore, they perform all their experiments using the output of the YouTube Recommendation System provided by the YouTube API. Moreover, in order to tackle the "ethical" concerns, expressed by many researchers, regarding the manipulation of recommendations, and to establish in what extend the users are willing to settle for "lower quality" recommendations for better Quality of Experience, they conduct an experiment recruiting real users.

More specifically, the authors propose an algorithm, called *CABaRet*, first presented in [20]. The CABaRet algorithm takes as input a video from the YouTube catalogue and the recommendation list for that video, provided by the YouTube API. Using a Breath-First-Search (in this work the maximum Depth used was 2), the algorithm creates a small catalogue L with videos closely related to the initial video consumed by the user. The final output of the CABaRet algorithm is a *new* recommendation list N (where $|N| < |L|$), which consists of the cached videos that are included in the auxiliary catalogue L. The experiment conducted in this work show that the cooperation of caching and recommendations, has indeed the potentials to improve the efficiency of a realistic network setup.

2.2. Recommendation-based Caching

The studies in this category aim to improve the performance of the network without intervening with the original purpose of the Recommendation System, i.e. allow users to navigate a massive catalogue of content in an efficient and satisfying way. The caching policies are instead enhanced with information provided by the Recommendation System, e.g. the demand profiles of users, or with techniques commonly used for extracting recommendations, such as collaborative filtering, content-based filtering, etc.

The work in [10] focuses on the distinct interest in video demand observed in different regions, which leads to differences on the parameter values of the popularity distribution of items between regional and global level. Given a network setup in which Content Providers (CP) (e.g. Netflix) utilize Content Delivery Networks (CDN) or regional caches for an efficient content delivery, such locality of interest can be leveraged to improve caching efficiency, by determining which content to store in local caches. To this end the authors in [10] are developing a set of metrics and a methodology that a CP could use to decide whether to cache locally or globally. These two terms, i.e. locally and globally, refer to whether the content placement policy of a regional cache will be decided based on the user preference of this region or on the user preference of the global or national customer base, respectively. First the authors divide a large region (e.g. U.S.A.) into sub-regions and they employ the biased matrix factorization algorithm (BMF), to characterize the unique user preference of each sub-region, to quantify the similarity of preference between sub-regions and to generate predictions for the user demand in each sub-region. Note that the BMF algorithm is typically used by CPs to generate recommendations for users, and thus extracting the proposed model parameters would be straight forward for CPs. The proposed model can be then used to measure the efficiency of the caching policy for each sub-region, which depends on the parameters of the model, and to determine the best approach, i.e. store content based on local or global (nation-level) user preference.

Considering an infrastructure-based (server-provisioned) CDN setup with a uniform distribution of servers, the authors in [11] examine the potential benefits offered by the application of two widely used recommendation technologies, namely content based filtering and collaborative filtering, into the core of CDNs. The benefits of this enhancement come from the improvement of the content placement and content access policies of CDNs with the contribution of these recommendation techniques. More specifically, the authors describe a system in which a catalogue of items is available to be split and stored to several servers. When a user requests an item, he/she is connected to the server which can serve this request with the minimum possible cost, in terms of delay, bandwidth, packet loss, server load and other factors. Considering the above the authors formulate an optimization problem in order to minimize the overall serving cost by carefully choosing the appropriate content placement strategy.

The authors in [12] explore the benefits of a caching policy based on recommendations on the caching performance, considering a Device-to-Device (D2D) caching setup over a cellular network. Motivated by the observation that existing works, regarding D2D caching, neglect users' individual character and assume a common popularity skew for all users, they classify the users in geographical groups and choose an important user (IU) for each group, whose device will ultimately serve as a cache for this group. In each geographical region the users' social connection is represented by a graph, and the social importance of users in each group is decided by a convex combination of users' device capacity and betweenness centrality of the user node in the graph. The user with the higher social importance is then assigned to store files and distribute, in a D2D manner, these files to other users, with whom they share a social connection. The authors derive the issued recommendations using three operations consequently, namely pre-filtering, collaborative filtering and the latent factor model. Finally, using these recommendations the proposed system choose the top-X (depending on the IU's device capacity) recommended contents corresponding to all users in the group to be cached at the IUs device.

The main scope of the work in [13] is the exploitation of the observed time-correlation in video requests introduced by recommendations, and the exploration of the trade-off between bandwidth usage and the quality of service provided to the users, when pre-fetching is employed in a CDN. The authors use a Markovian model, derived by a small-world graph depicting the request pattern of users, which captures the time-correlation of requests. Through simulations in this study, it is shown that this model is consistent with empirically observed properties of request patterns, presented in existing works, which study the effects of Recommendation Systems on users' viewing patterns ([14], [15], [3]). This model is used to generate recommendations for users, which are essential for the realization of the proposed caching policy. More specifically, the authors propose a prefetching approach, referred to as PreFetch, according to which, as soon as a user requests a specific video, a number of the top recommendation for the requested video are pre-fetched in the dedicated CDN cache. Considering a high chance that the user will continue the viewing session with one of the recommended videos, this system would achieve high cache hit ratio, and thus better quality of experience (QoE) for the user (due to reduced chance of start-up delay). In order to examine the trade-off between the bandwidth consumption needed for the realization of the PreFetch policy and the QoE achieved, a simple cost model is formulated as the sum of the number of video file downloads (fetches) to the local cache and the number of cache misses (start-up delays).

The work in [16] is, to our best knowledge, the earliest study in this line of work, presented in 2012. It examines the benefits of Recommendation based algorithms that base caching decisions on predictions of individual end user behavior. The authors focus on the combined services of content delivery, through CDNs, and content discovery, i.e. content recommendations to users that modern Network and Content Providers offer. This study combines both key roles of content discovery and content delivery, and it analytically assesses how the output of the Recommendation system can be advantageously used to increase the CDN performance. Two caching algorithms are designed and compared to each other. The first one, called Popularity Tracking based Caching, ranks the items available to the user by popularity and stores the most popular to the local cache, according to the storage capacity of the cache. This method represents the commonly used policy in CDNs. The second algorithm, which is referred to as Recommendation Based Caching, chooses to store the items that maximize the overall score of the caching list, which is derived by a combination of the inherent popularity distribution of the items and the propensity of each user to request each item, based on his/her individual profile determined by the Recommendation.

2.3. Joint Caching and Recommendation

The potential gains of the interplay between content replication and Recommendation Systems to the efficiency of caching in CDNs motivate the authors in [17]. They define a model that captures the coupling between caching decision and recommendations to a set of users and formulate a joint optimization problem to maximize the cache hit ratio. A model is also proposed in order to measure the impact of recommendations on user content requests. The authors assume an inherent popularity distribution of items and argue that, the recommendations issued to users for a set of video files, provide all items in this set with an equal demand boost. On the contrary the demand for the files that are not recommended decreases. Combining the original popularity distribution and the effect of the Recommendation system on this distribution, the final demand distribution for all files is formed. In addition, in order to address the ethical concerns raised by the manipulation of users' preference via the Recommendation Systems, they introduce a measure called user preference distortion. This measure is embedded as a constraint to the optimization problem mention above, in order to assure the high quality of recommendations offered to users.

The authors in [18] rely on the strong effect of recommendations on the users' demand in online media content to make the cellular network systems with local caching at the Base Station (BS) more efficient and intelligent. They introduce a simple idea of informing the users about what is cached at the BS as a means to improve the cache hit ratio, which can be regarded as a form of recommendation. More specifically, an abstract of the currently cached files is broadcasted to the users, at regular time intervals. In this study a simplified cellular network model with one cell, incorporating a single cache at the BS, is considered. A random model is created to predict the demand patterns of the users before and after the recommendations, using a Zipf distribution to model the popularity of the files. In addition, it is assumed that the mobile users enter and leave the cell according to a Poisson process. In order to maximize the expected long-term system reward of the above setting, by carefully choosing which files to cache at the BS, the authors formulate an optimization problem and they use a Q-Learning algorithm to solve it. The use of online learning in this case is required in order to perceive the request probability and the statistics of random arrival and departure, which are otherwise unknown.

The authors in [19] argue that content caching and recommendations are strongly coupled, since content recommendations have a significant impact on users' demand, which consequently affects the optimal caching policy, and should therefore be jointly considered and optimized. Moreover, they have observed that most studies on this line of work often consider the impact of recommendations on users' demand, or the propensity of users to be influenced by recommendations, known; this data is however unavailable in practice. Driven by the above observations, the authors design a system which takes into consideration individual user preferences, to derive the inherent content popularity distribution for each user, and uses reinforcement learning practices to estimate a personalized threshold, that determines whether a user will accept a given recommendation or not. More specifically, according to the proposed model, when a user decides to watch a video file, he/she will either know which content he/she wishes to request or he/she will be influenced by the recommendations offered on the Content Provider's user interface. The extent to which each user accepts a given recommendation is determined by a user specific threshold θ_u; if the inherent preference (p_{uf}) of the user towards that recommended item is higher than θ_u, then p_{uf} increases, whereas if it is lower, then p_{uf} is unaffected by the recommendation. Finally, the authors formulate a joint caching and recommendation optimization problem and propose an ε-greedy algorithm to solve it.

The authors in [20] aim to bypass the difficulties arising by the need of collaboration between Content Providers and Network Operators, when the latter require sensitive information about the relations between content and users, owned and managed by the former. For example, such information could contain similarity scores between content and users, user preferences/history, etc., that maybe be necessary in order to extract recommendations to users, but is unlikely to be disclosed due to privacy and/or economic reasons. To this end they consider a popular CP service, i.e. YouTube, and design a system that obtains publicly available recommendation lists, based on which it builds extended lists of directly and indirectly related videos. These new recommendation lists are then used to carefully steer initial recommendations - and thus user demand - towards cached videos. More specifically after a user watches a video v, the proposed system obtains in a Breadth-First Search manner a list (L) of videos directly or indirectly related to video v, until a predefined depth. Then a new recommendation list is provided to the user containing videos from the list L that are also cached. By applying this scheme, the authors manage to boost caching efficiency, while at the same time retain high-quality recommendations, since using the YouTube recommendations ensures strong relations between videos (especially for low BFS depth values). To further improve caching efficiency, the authors formulate a joint caching and recommendations optimization problem and propose an approximation algorithm with provable performance guarantees.

While there are numerous studies on the benefits of the interplay between caching and recommendations in mobile networks, no such study has examined the impact of the time-varying connectivity of users on the joint caching and recommendation policy. Mainly driven by this observation, the work in [21] studies a cellular network where a single-antenna Base Station serves M single-antenna users, and considers three cases of user connectivity, namely noncausal, statistical and causal connectivity. Each user connects to the BS randomly and the three cases above refer to the information available to the BS about the users' connectivity. In order to increase the bandwidth availability of the cellular network, the authors propose a scheme with time slots, during which the BS pushes packets of content to users that are connected to it. These packets are stored to the users' personal device and at the beginning of the next timeslot the BS issues recommendations to the users based on the contents pushed. If a user requests an item, of which a packet is already pushed to his/her device, then there are obvious savings in the available network bandwidth. For all cases of connectivity, the authors formulate optimization problems in order to maximize the effective throughput, defined as the average size of contents read directly from the users' device.

The main scope of the work in [22] is the minimization of the cost of delivering a requested content to user through a mobile network with caches installed at the Base Stations of the network. The authors argue that, while existing works on this field consider the improvement of network congestion, users' experience or cache hit ratio, the equally important aspect of the cost of content delivery burdening the cellular network, receives much less attention. Thereof, they design a system setup where users request video content through the Content Providers' platforms, which can be served either from the designated cache installments or from a cloud server. In the later scenario, the delivery of the requested content utilizes the backhaul link, which strains the network and compromises the total efficiency of the network setup. The authors assume that there are time periods, of few hours, over which the users' preference over the available catalogue of videos, the composition of the caches, and the output of the Recommendation System are all constant.

Considering the above setup, they formulate an optimization problem in order to minimize the total cost of content delivery for each user. The control variables of the optimization problem are the caching decisions of the Caching Operator and the recommendations generated for each user.

In addition, in order to generate a detailed and accurate user preference distribution over the content catalogue, they apply a deep learning method, namely the Long Short-Term Memory architecture, over raw data of user rating, which they download from the MovieLens Database.

Chapter 3
Setup Analysis

In this chapter we will further analyze and discuss the setup of one study from each category.

3.1. Category 1: Show me the Cache: Optimizing Cache-Friendly Recommendations for Sequential Content Access

The authors in [8] assume a setup that can reflect the sequential request pattern of users, which is driven by recommendations. More specifically, as mentioned previously in this thesis, a significant proportion of user multimedia content requests originate from the recommendations that are suggested to them. Therefore, this request pattern can be expresses as a Markov Chain, where users request a random item *i*, with probability p_i, or they choose to request an item *j* from the recommendation list generated while consuming item *i*, with probability p_{ij}. Using this setup, they propose a recommendation system that will steer users' requests towards cached content. In order to achieve maximum performance of the proposed scheme, they formulate and solve an optimization problem.

Problem Setup

The authors consider a content catalogue K of cardinality K, and a similarity matrix $U \in \mathbb{R}^{K\times K}$, where $u_{ij} \in [0,1]$ denotes the similarity of content *i* to content *j*, calculated by a Recommendation System. After a user has consumed content *i*, the RS creates a list of *N* items with the highest u_{ij} value, and suggests that list to the user. In addition, they create a cost model, where fetching content *i* is associated with a cost $x_i \in \mathbb{R}$. This value is very low when the requested content can be served by the designated cache, and high otherwise. The exact values of x_i depend on the caching topology (e.g. hierarchical caching). For simplicity in this work x_i is considered binary, where $x_i = 0$ if the requested item is cached, and $x_i = 1$ otherwise. Considering the above setup, the following content request model is formulated.

Definition 1. After a user has consumed a content i**,** then

- (*recommender request*) with probability α the user picks one of the N recommended items with equal probability $\frac{1}{N}$.
- (*direct request*) with probability $1-a$ the user ignores the recommendations, and picks any content j from the catalogue K with probability p_j, where $p_j \in [0,1]$ and $\sum_{j=1}^{K} p_j = 1$. For short, they denote the vector $\boldsymbol{p_0} = [p_1, \cdots, p_K]^T$.

It is important to note that authors assume that the probability α is fixed, given that the recommendation quality is above a threshold. To this end, they use a parameter q, associated with the similarity parameter u_{ij}, as a quality constraint, and they assume that if a recommendation is above this threshold, then a will remain fixed throughout a user's browsing session.

In order to express the user request model, the authors define a Markov chain, whose transition matrix P is given by

$$P = a \cdot Y + (1-a) \cdot P_0, \tag{1}$$

where $P_0 = \mathbf{1} \cdot \boldsymbol{p_0}^{\boldsymbol{T}}$, and Y is a stochastic matrix, where $y_{ij} = \frac{z_{ij}}{N} \in \left[0, \frac{1}{N}\right]$. Variable z_{ij} denotes whether content j is in the list of N recommended items, after the user has consumed content i, with $z_{ij} \in [0,1]$ and $\sum_j z_{ij} = N, \forall i$.

Given the above setup, the authors goal is "to reduce the total cost of serving user requests by choosing matrix Y, while maintaining a required recommendation quality". Considering a user that starts a session by requesting a content $i \in K$ with probability p_i, and then proceeds to request a sequence of M contents according to the transition matrix P of Eq. (1), the associated access cost of this session would be given by

$$\sum_{m=0}^{M} \boldsymbol{p_0}^{T} \cdot P^m \cdot \mathbf{x} \approx \boldsymbol{\pi}^{\boldsymbol{T}} \boldsymbol{x} \cdot, \tag{2}$$

where $\mathbf{x} = [x_1, \cdots, x_K]^T$ is the vector of the costs per content.

Using the above expression, the authors formulate the following optimization problem.[1]

Optimization Problem 1. (Cache-Friendly Recommendations).

$$\underset{Y}{minimize} \quad \boldsymbol{p_o}^T \cdot (I - aY)^{-1} \cdot \mathbf{x} \qquad (\mathbf{3})$$

$$0 \leq y_{ij} \leq \frac{1}{N}, \forall\, i \; and \; j \; \in K \qquad (3a)$$

$$\sum_{j=1}^{K} y_{ij} = 1, \forall i \; \in K \qquad (3b)$$

$$y_{ii} = 0, \forall\, i \in K \qquad (3c)$$

$$\sum_{j=1}^{K} y_{ij} u_{ij} \geq q_i, \forall\, i \in K \qquad (3d)$$

For the above optimization problem, the variables y_{ij} are the control variables. Equations Eq. (3a) to Eq. (3d) are the constraints of the optimization problem. The first three constraints make sure that y_{ij} form a stochastic matrix. Eq. (3d) refers to the quality of the recommendations. More specifically, it ensures that the quality of the recommended contents for each i is above a desired threshold.

Optimization Algorithm

The above optimization problem is non-convex, because of the expression of the stationary distribution $\boldsymbol{\pi}$ of the Markov chain Eq. (3). Therefore, the authors suggest two heuristic approaches to solve the Optimization Problem 1, through relaxations of the objective function.

In first heuristic approach, referred to as "Myopic Cache-Friendly Recommendations", the authors assume that every user's browsing session

[1] Refer to **Lemma 1** of [8] for the conversion of Eq. (2) to Eq. (3).

consists of only two content requests. The user initiates the session by requesting an item i of the catalogue K, and then requests another item j, according to the transition matrix P of Eq. (1). In this case, the objective function becomes

$$(\boldsymbol{p}_0{}^T \cdot P) \cdot \mathbf{x} \tag{4}$$

As a result, Optimization Problem 1 is transformed in the following optimization problem.

Optimization Problem 2 (Myopic Cache-Friendly Recommendations).

$$\underset{Y}{minimize} \quad \boldsymbol{p}_0{}^T \cdot (a \cdot Y + (1-a) \cdot P_0).x, \tag{5}$$

$$s.t. \quad Eqs.(3a)-(3d)$$

The Optimization Problem 2 is now a Linear Problem with affine and box constraints and can be solved efficiently in polynomial time, using e.g. interior-point methods.

In the second heuristic approach, in order to discard the inverse of the control matrix Y in the objective of Optimization Problem 1, the authors propose an equivalent of Optimization Problem 1, by replacing the cost function with the Eq.2. As a result, a new optimization problem is formulated as follows

Optimization Problem 3 (Cache-Friendly Recommendations: Equivalent Problem).

$$\underset{\pi, Y}{minimize} \quad \boldsymbol{\pi}^T \cdot \mathrm{x}, \tag{6}$$

$$s.t. \quad Eqs.(3a)-(3d)$$

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}^T(a \cdot Y + (1-a) \cdot \boldsymbol{p_0}^T) \; (6a)$$

$$\sum_{j=1}^{K} \pi_j = 1 \qquad (6b)$$

$$\pi_j \geq 0, \forall j \in K \qquad (6c)$$

This optimization problem corresponds to the full structure of the Markov chain that describes the sequential request pattern of users.

The objective function is now linear in the control variables of vector $\boldsymbol{\pi}$. However, the constraint Eq. (6a) is a quadratic equality constraint, and although the problem is bi-convex, it can be shown that there are convergence issues. The authors propose an Augmented Lagrangian relaxation for this constraint, so that it can be moved to the objective function. Since the problem is bi-convex they propose an ADMM-like method, which solves the convex subproblems iteratively.[2] From the simulations conducted by the authors, it can be seen that their algorithm converges to its maximum achieved cache hit ratio in 5 iterations.

3.2. Category 2: Request Patterns and Caching for VoD Services with Recommendation Systems

The authors in [13] propose a pre-fetching policy to populate caches in a CDN, using information for the users' request patterns from a Recommendation System. The RS is modeled as a small-world network graph, which is built based on empirically observed properties of user request patterns effected by recommendations in Video on Demand (VoD) services. More specifically, when a user requests an item i to watch, a number (r) of videos from the recommendation list generated for item i, are stored in a dedicated cache. The proposed scheme introduces a trade-off between bandwidth consumption, caused by pre-fetching items and improved Quality of Service offered to users, due to reduced start-up delay (if the requested item is stored locally and can be served by the cache). To evaluate this scheme, the authors explore this trade-off as a function of the above two factors.

Users' Requests Pattern Properties

[2] Refer to **Algorithm 1.** of [8] for more details on the ADMM-like algorithm.

According to the literature on request patterns in VoDs services with recommendations, and more specifically in the YouTube platform, the following properties can accurately characterize the nature of the recommendations and the behavior of users in such services.

1) Small-World Network Recommendation Graph: The graph representing the YouTube recommendation network is observed to be small-world. Small-world networks are a class of networks that are highly clustered with small characteristic path lengths.
2) Content Popularity Profiles: Content popularity for such services with RS, can be well-fitted with the Zipf distribution.
3) Click Through Rate: The Click Through Rate (CTR) is a metric that indicates the frequency in which an item in the r position of a recommendation list is requested after a user consumes a video. It is shown that the mean of the CTR follows the Zipf distribution as a function of r.
4) Chain Count: Chain Count is defined as the number of consecutive requests generated by the recommendation list through one viewing session of a user. For YouTube, the Chain Count is estimated to be between 1.3 and 2.4.
5) Degree Distribution: The degree distribution of a recommendation graph has been found to follow the power law. More specifically, the number of nodes with degree k is approximately proportional to k^{-3}.

Proposed Model Definition

The authors construct a directed graph $G(V, E)$, where the set V represents the full catalogue of videos offer by the VoD service, and the set E represents the edges of the graph. An edge $e = \{i, j\} \in E$ implies that video j is one of the recommended videos for video i. They assign then weights to every edge of the graph. The process of weight assignment will be described later in this section. Considering this setup, each user's request process is a random walk on this weighted graph and therefore, the request arrival process can be modeled as a Markov Chain. In order to force the graph describing the request process, to comply with the empirically observed properties described in the previous section, the authors use the Barabasi-Albert model to generate a random small-world network graph.[3] In addition, in order to obtain a directed graph, all the edges of the graph are replaced with two directed edges.

The request process of a user is described as follows. After a user has finished consuming video i, he/she can either request a video from the recommendation list of video i, with probability P_{cont}, or a video from the entire catalogue of items according to a Zipf distribution describing the popularity of videos. In order to model the later, the authors add a 'dummy' node n_0 to the graph G, which is connected to all other nodes in G.

In order to complete the graph describing the request process of a user, the authors assign weights to every edge of the graph. They set $P_{i,j}$ to be the transition probability from node n_i to node n_j.

- By definition, $P_{i,j} = 0$ if $\{i, j\} \notin E$.
- By definition, $P_{i,0} = 1 - P_{cont}, \forall i > 0$. This is the probability that a user does not request a video from the list of videos that are recommended a given moment.
- As mentioned beforehand, content popularity of VoD services without recommendations follows the Zipf distribution. Therefore, the authors set the value of $P_{0,j} \propto j^{-\beta}$, where β is a positive constant that ranges between 0.6 and 2.
- In order to assign transition probabilities to edges between a video and its recommended videos, the authors use the distance between two nodes in the graph as a measure of similarity. For each $i, j \in E, P_{i,j} \propto P_{cont} \cdot \left(D(i,j)\right)^{-\kappa}$, where $D(i,j) = |i - j|$ and κ is a positive constant.

[3] Refer to **Figure 1.** of [13] for a formal definition of the Barabasi-Albert model

The model described above follows by definition the empirically observed properties 1, 4 and 5, as listed in the previous section. Through simulations, the authors prove that also properties 2 and 3 are satisfied by the proposed model.

CDN Setting

The authors assume a network setup with a central server which is connected with a CDN, consisting of a local cache, which serves requests from users in an area. Each request for a video is served by this cache, if the requested item is stored there, otherwise it is served by the central server, imposing a load in the network's backbone infrastructure. The users' requests arrive according to the Markovian process described in the previous section. The proposed scheme, pre-fetches the top r videos of the recommendation list generated for video i, when a user consumes video i. This process introduces a bandwidth usage cost for pre-fetching these videos. On the other hand, if the user requests a video that is not cached locally, then a cost of Delayed Startup occurs. The goal of this study is to create a cost function that can accurately describe the total cost of the proposed scheme, and to minimize this cost.

Considering all the above, the authors propose a caching policy refer to as the PreFetch policy.[4] The key idea of the PreFetch policy is to pre-fetch the top r recommended videos as soon as a user requests a specific video. The policy uses the Least Recently Used (LRU) metric to delete stored content from the local cache. The information about the recommendation lists is derived by the graph constructed by the authors. In any case, any CDN operator with knowledge of the Recommendation System employed by a VoD service could implement the proposed PreFetch policy. Through simulations, the authors deduct a value for r (the number of videos to pre-fetch after every user request) that achieves a good trade-off between Bandwidth Usage cost and Delayed Startup cost.

3.3. Category 3: Caching-aware Recommendations: Nudging User Preferences towards better Caching Performance

[4] Refer to **Figure 7.** of [13] for a detailed analysis of the proposed algorithm.

The authors in [17] assume a cellular network model, where there are multiple cells with local caches embedded in each cell. Mobile users are located within range of more than one cells in the network any given time. Considering the above setup, the authors argue that a coupling between content caching and recommender systems could prove profitable for both users (better QoE) and network operators (better network recourse allocation and reduction in bandwidth usage). More specifically, by jointly optimizing caching decisions and recommendations provided to users, they show that their proposed algorithm can achieve higher caching hit ratio than traditional caching methods. It is important to note that altering the initial recommendations that are generated for users, can significantly degrade users' experience and raises some ethical concerns, therefore the authors introduce a preference distortion measure, which is used as a constraint for the optimization problem mentioned above.

System Model

The authors assume a model which involves a set of caches C, a catalogue of content items I and a set of users U. Caches are of limited storage, $C_i, i \in C$ and are co-located with the wireless network microcells. Content items are characterized by one or more thematic categories, M. Namely, each item $i \in I$ has a finite size L_i and is represented by a feature vector f_i, whose j^{th} element $f_i(j), j \in [1, \cdots, M]$ denotes the score of item i in feature j. These relevance scores assume values in $[0,1]$ and are normalized so that $\sum_{j=1}^{M} f_i(j) = 1 \ \forall i \in I$. Users are described by similar feature vectors f_u as the content items. Each vector element $f_u(j), j \in [1, \cdots, M]$ expresses how much user u is interested in content classified under the thematic category j. Similarly, the user preference vectors are normalized so that $\sum_{j=1}^{M} f_u(j) = 1 \ \forall u \in U$.

The authors assume two different content preference distributions, namely, one distribution that describes the inherent content preferences of users and the one that will be formed eventually, due to the recommendations issued to users. The former distribution is denoted as $p_u^{pref}, \sum_{i \in I} p_u^{pref}(i) = 1$, and the later as $p_u^{req}, \sum_{i\in I} p_u^{req}(i) = 1$. In addition, it is safe to assume that $p_u^{pref} \neq p_u^{req}$, since recommendations have an impact on users' preferences. The inherent preference distribution, p_u^{pref}, is taken to be the cosine similarity index a_{ui} of the feature vectors f_i and f_u.

$$a_{ui} = \frac{\sum_{j=1}^{M} f_u(j) \cdot f_i(j)}{\sqrt{\sum_{j=1}^{M} f_u(j)}\sqrt{\sum_{j=1}^{M} f_i(j)}}$$

Normalizing these index values over all items for a given user u yields the content preference distribution p_u^{pref}, $p_u^{pref}(i) = \frac{a_{ui}}{\sum_{i\in I} a_{ui}}$.

The proposed scheme in this study is summarized as this. Typically, a Recommendation System seeks to recommend R items to user u, after consuming item i, with the higher rank from the preference distribution p_u^{pref}. Instead, the system selects R items out of a *recommendation window* W_u determined by the top K_u items, where $K_u > R$. This gives the opportunity to the entity that operates the RS, to suggest to users items that will have a positive impact on caching efficiency. On the other hand, with large values of K_u, there is a risk of an unwanted degradation of the quality of recommendations, as mention before. This, in result, will lead to dissatisfied users. In order to limit this risk, the authors introduce a measure referred to as the *user preference distortion* measure, Δ_u. It is defined as

$$\Delta_u(K_u, R) = 1 - \frac{\sum_{j=K_u-R}^{K_u} p_u^{pref}(j)}{\sum_{j=1}^{R} p_u^{pref}(j)}.$$

This measure expresses the worst-case deviation, in terms of original user preferences for the recommended items, that may result from the choices of the proposed scheme. Therefore, it captures an upper bound on the possible distortion of the original recommendations.

The authors argue that, the system recommendations affect the relative user demand for all content items. More specifically, they boost the demand for the recommended items and proportionately decrease the demand for the remaining items. In order to capture this effect, the authors work as follows. They assume that recommendation provide all R (the recommended items) with an equal boost, $p_u^{rec}(i) = {}^{1}\!/_{R}$. Thus, the ultimate content request distribution is a convex combination of these two distributions, p_u^{pref} and p_u^{rec}, so that

$$p_u^{req}(i) = w_u^r \cdot p_u^{rec}(i) + (1 - w_u^r) \cdot p_u^{pref}(i) \tag{1}$$

for the R items that are recommended to user u, and

$$p_u^{\sim req}(i) = (1 - w_u^r) \cdot p_u^{pref}(i) \tag{2}$$

for the $|I| - R$ items that are not recommended. The recommendation weights w_u^r express the importance user u assigns to recommendations.

The Joint Caching and Recommendations Problem

Considering the above setup, the authors formulate an optimization problem, in order to achieve the optimal caching and recommendation decisions. The coordination of caching decision, i.e., which items to store in the cell cache(s), with the recommendation decisions, i.e., which items to recommend to each user in the cell, aims at best serving both user- and network-centric performance measures. On the user side, this translates to increased QoE, whereas on the network side, to decreased backhaul recourses usage (since the caches are located at the edge of the cellular network).

The formulated optimization problem is show below.

$$\max_{\mathbf{y},\mathbf{x}} \quad \sum_{u \in U} \sum_{i \in W_u} y_i (x_{ui} p_u^{req}(i) + (1 - x_{ui}) p_u^{\sim req}(i))$$

$$s.t. \quad \sum_{i \in I} y_i L_i \leq C$$

$$\sum_{i \in W_u} x_{ui} = R \quad \forall u \in U$$

$$y_i x_{ui} \in \{0,1\} \quad u \in U, i \in W_u$$

W_u denotes the set of items within the recommendation window of user u. The cardinality of this set is K_u with

$$K_u = max\{k|\Delta_u(k, R) \leq r_d\}$$

where $r_d \in [0,1)$, is the upper bound on user preference distortion.

There are two sets of binary decision variables: $y_i = 1$ when item i is cached and $y_i = 0$, otherwise; $x_{ui} = 1$ when item i is recommended to user u and $x_{ui} = 0$ when it is not. The constraints of this problem ensure that the capacity of the caches is not exceeded, that every user is recommended exactly R items, and finally that the issued recommendations lie within the distortion limit.

This problem is a (non-linear) generalization of the 0-1 Knapsack problem (0-1 KSP) and, thus, it is NP-hard.

A Heuristic Algorithm for the Joint Caching and Recommendations Problem

The algorithm that the authors devised in order to solve the optimization problem of the previous section consists of three steps. In the first step, the initial recommendation set, RC_u^{in}, for every user is generated, using the inherent preference distribution, p_u^{pref}. These recommendations are not communicated to users, and instead they are used as input for the next phase of the algorithm. Then, in the *content placement* step, each content is assigned with a request probability, referred to as *utility*, derived from (1) and (2).

$$util(i) = \sum_{u \in U} p_u^{req}(i) \quad i \in I$$

In the final phase of the content placement step, the algorithm assigns content items into caches, using the utilities defined above. Since the capacity of each cache is finite, and the caching decisions are binary, i.e., the decision to be made is whether to cache a specific item in a specific cache, or not, this problem is an instance of the 0-1 KSP. The authors use the Dynamic Programming (DP) FTPAS algorithm to obtain an $1-\epsilon, \epsilon > 0$ approximation of the optimal solution.[5] The resulting placement is denoted as P. Finally, in the *recommendation amendment* step, the original recommendations to users are amended in an attempt to maximize the utility of the cached content items. For this step, the two sets P and RC_u^{in} are compared. Out of this comparison, two possibilities arise:

- $RC_u^{in} \subseteq P$: The initially recommended items are stored in the caches associated with the user, therefore these recommendations remain intact. The resulting user preference distortion is, thus, zero.
- $\left|RC_u^{in} \cap P\right| = F_1 < P$: There are cached items that are not recommended to users, therefore the initial recommendation set is altered to match the cached content items.

The final set of recommended items, RC_u^f, is formed. The items in this set are in general different from the provisional set RC_u^{in}.[6]

[5] Refer to [12], §8.2 of [17] for more information about the DP FPTAS algorithm.

[6] Refer to **Algorithm 1** of [17] for a more detailed analysis of the proposed algorithm.

Chapter 4
Conclusions

In this thesis, we consider the recently proposed cooperation of Content Caching in wired and cellular networks with the Recommendation Systems employed in most Content Providers’ platforms. We analyze the studies on this direction and we discuss the main point of interest in every study. An important feature in this line of research, in our opinion, is the interference with the initial purpose of the Recommendation System. Some researchers in their work choose to alter the recommendations provided to users, in order to improve the efficiency of the cache which serves the users’ requests in a system. This approach raises some important ethical concerns, as well as some performance issues, since altering the recommendation list of a platform leads to manipulation of the popularity of contents, and steers away the Recommendation System from its original purpose, which may cause the platform to lose clients-users. With this criterion, we categorize the studies in this thesis in three categories; studies that interfere with the output of the Recommendation System but not with the caching algorithm, studies that alter the caching algorithm but not the recommendation lists, and finally studies that modify both caches and recommendation lists. The most prevailing approach, in spite of the ethical and performance issues mentioned above, seems to be the Cache-aware Recommendations.

To sum up, the cooperation of Network Content Caching and Recommendation Systems is proven to be a promising way to enhance the performance of both wired and cellular networks. It offers significant benefits to Internet Providers in terms of saving network resources, and promises to provide high Quality of Experience to users. It is also important to note that this is a “software-based” solution, therefore there is no need for costly hardware enhancements in the network layout, in order to implement the proposed schemes. Consequently, we expect to that such solutions will be implemented by many network operators and Content Providers in the near future.

References

[1] Cisco White Paper, “Cisco Visual Networking Index: Forecast and Trends, 2017 – 2022”, *Cisco Public Information*. 2017.

[2] C.A. Gomez-Uribe and N. Hunt, “The Netflix recommender system: Algorithms, business value, and innovation,” ACM Transactions on Management Information Systems (TMIS), vol. 6, no. 4, p. 13, 2016.

[3] R. Zhou, S. Khemmarat, and L. Gao, “The impact of youtube recommendation system on video views,” in Proc. 10th ACM SIGCOMM Internet Measurement Conference (IMC), Melbourne, Australia, November 2010, pp. 404–410.

[4] J. Tadrous, A. Eryilmaz and H. El Gamal, "Proactive Content Download and User Demand Shaping for Data Networks," in IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 1917-1930, Dec. 2015, doi: 10.1109/TNET.2014.2346694.

[5] Dilip Kumar Krishnappa, Michael Zink, Carsten Griwodz, and Pål Halvorsen. 2015. Cache-Centric Video Recommendation: An Approach to Improve the Efficiency of YouTube Caches. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 4, Article 48 (April 2015), 20 pages. doi:https://doi.org/10.1145/2716310

[6] P. Sermpezis, T. Giannakas, T. Spyropoulos and L. Vigneri, "Soft Cache Hits: Improving Performance Through Recommendation and Delivery of Related Content," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 6, pp. 1300-1313, June 2018, doi: 10.1109/JSAC.2018.2844983.

[7] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch and G. Caire, "FemtoCaching: Wireless Content Delivery Through

Distributed Caching Helpers," in IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8402-8413, Dec. 2013, doi: 10.1109/TIT.2013.2281606.

[8] T. Giannakas, P. Sermpezis and T. Spyropoulos, "Show me the Cache: Optimizing Cache-Friendly Recommendations for Sequential Content Access," 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Chania, 2018, pp. 14-22, doi: 10.1109/WoWMoM.2018.8449731.

[9] S. Kastanakis, P. Sermpezis, V. Kotronis, D. S. Menasche and T. Spyropoulos, "Network-aware Recommendations in the Wild: Methodology, Realistic Evaluations, Experiments," in *IEEE Transactions on Mobile Computing*, doi: 10.1109/TMC.2020.3042606.

[10] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot and A. Ashkan, "Cache content-selection policies for streaming video services," IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, 2016, pp. 1-9, doi: 10.1109/INFOCOM.2016.7524619.

[11] Mohamed Ali Kaafar, Shlomo Berkovsky, and Benoit Donnet. 2013. “On the potential of recommendation technologies for efficient content delivery networks,” *SIGCOMM Comput. Commun. Rev.* 43, 3 (July 2013), 74–77. doi:https://doi.org/10.1145/2500098.2500109

[12] Y. Wang, M. Ding, Z. Chen and L. Luo, "Caching Placement with Recommendation Systems for Cache-Enabled Mobile Social Networks," in IEEE Communications Letters, vol. 21, no. 10, pp. 2266-2269, Oct. 2017, doi: 10.1109/LCOMM.2017.2705695.

[13] S. Gupta and S. Moharir, "Request patterns and caching for VoD services with Recommendation Systems," 2017 9th International Conference on Communication Systems and Networks (COMSNETS), Bangalore, 2017, pp. 31-38, doi: 10.1109/COMSNETS.2017.7945355.

[14] X. Cheng, J. Liu and C. Dale, "Understanding the Characteristics of Internet Short Video Sharing: A YouTube-Based Measurement Study," in IEEE Transactions on Multimedia, vol. 15, no. 5, pp. 1184-1194, Aug. 2013, doi: 10.1109/TMM.2013.2265531.

[15] X. Cheng and J. Liu, "NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing," IEEE INFOCOM 2009, Rio de Janeiro, 2009, pp. 1152-1160, doi: 10.1109/INFCOM.2009.5062028.

[16] M. Verhoeyen, J. De Vriendt and D. De Vleeschauwer, "Optimizing for video storage networking with Recommendation systems," in Bell Labs Technical Journal, vol. 16, no. 4, pp. 97-113, March 2012, doi: 10.1002/bltj.20536.

[17] L. E. Chatzieleftheriou, M. Karaliopoulos and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057031.

[18] K. Guo, C. Yang and T. Liu, "Caching in Base Station with Recommendation via Q-Learning," 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, 2017, pp. 1-6, doi: 10.1109/WCNC.2017.7925848.

[19] D. Liu and C. Yang, "A Learning-Based Approach to Joint Content Caching and Recommendation at Base Stations," 2018 IEEE

Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-7, doi: 10.1109/GLOCOM.2018.8647827.

[20] Savvas Kastanakis, Pavlos Sermpezis, Vasileios Kotronis, and Xenofontas Dimitropoulos. 2018. CABaRet: Leveraging Recommendation Systems for Mobile Edge Caching. In *Proceedings of the 2018 Workshop on Mobile Edge Communications* (*MECOMM'18*). Association for Computing Machinery, New York, NY, USA, 19–24. doi:https://doi.org/10.1145/3229556.3229563

[21] Z. Lin and W. Chen, "Joint Pushing and Recommendation for Susceptible Users with Time-Varying Connectivity," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6, doi: 10.1109/GLOCOM.2018.8647838.

[22] Y. Fu, Z. Yang, T. Q. S. Quek and H. H. Yang, "Towards Cost Minimization for Wireless Caching Networks with Recommendation and Uncharted Users' Feature Information," in *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6758-6771, Oct. 2021, doi: 10.1109/TWC.2021.3076495.