Caching at the network edge can significantly reduce users’ perceived latency and relieve backhaul pressure, hence invigorating a new set of innovations toward latency-sensitive applications. Nevertheless, the efficacy of caching policies relies on the users’ content preference to be 1) known a priori and 2) highly homogeneous, which is not always the case in the real world. In this article, we explore how artificial intelligence (AI) techniques and recommendation can be leveraged to address those core issues and reap the potentials of cache-enabled wireless networks. Specifically, we present the hierarchical, cache-enabled wireless network architecture, in which AI techniques and recommendation are utilized, respectively, to estimate users’ content requests in real time using historical data and to reshape users’ content preference. Through case studies, we further demonstrate the effectiveness of an AI-based predictor in estimating users’ content requests as well as the superiority of joint recommendation and caching policies over conventional caching policies without recommendation.

The Future of Cache-Enabled Wireless Networks
Along with the rapid upgrades of wireless communication technologies, a host of unprecedented services and innovative applications are emerging, ranging from the Internet of Things, virtual/augmented reality, and ubiquitous multimedia services to intelligent housing and transportation systems. These innovative applications have triggered the explosive growth of data traffic, which, in turn, imposes great pressure on current networks due to the depletable radio resource and scarce backhaul capacity. More precisely, the main crux of network burden stems from the heterogeneous users’ demands; that is, each user has an individual preference for the contents and will compete for the communication resource if more than one user requests content delivery simultaneously. To enhance network performance and users’ quality of experience, caching at the network edge, e.g., the base stations (BSs) and/or mobile devices, has been introduced as a promising solution [1]. Particularly, when provided with a priori information of each individual preference distribution, one can place the caching resource in an optimal way that simultaneously enhances energy efficiency, reduces service latency, and relieves backhaul load [1]–[3].

Despite such benefits, the gain from caching alone is pronounced, as noted previously, only when the users’ preference information is known a priori and highly homogeneous; i.e., the users tend to request the same contents. These two constraints, however, are less likely to be satisfied in next-generation wireless networks that possess a...
higher degree of heterogeneity. On one hand, users’ preferences for different contents vary drastically across time and space, thus making them extremely difficult to estimate and track, especially when the number of mobile devices becomes large. On the other hand, in practice, users’ preference distributions are highly diverse due to personality differences. These problems motivate a paradigm shift in the design of cache-enabled wireless networks.

Among the various solution candidates, we believe that AI techniques and recommendation mechanisms are arguably the most promising to resolve the aforementioned problems and further boost the performance of cache-enabled wireless networks. In fact, the burgeoning progress of AI technology and the increased capabilities of the network are setting the scene for continuously inferring and updating the key parameters related to users’ behaviors (e.g., content requests and download preferences). Data-driven machine learning and AI-enabled intelligent solutions become more crucial for real-world applications in the future generation of cellular networks. Additionally, recommendation mechanisms can stimulate users’ content consumption pattern. A typical recommendation mechanism recommends to users the most appealing content items so as to boost their engagement and satisfaction, which in return leads users to remain on the network as long as possible. In fact, it was shown by [4] that the viewing of videos on YouTube reaches 50% with recommendation. The percentage increases to be 80% for Netflix. Nevertheless, recommending to each user his/her corresponding top preferred contents is certainly not optimal in view of enhancing the performance of cache-enabled wireless networks, especially when users’ preference distributions are highly dispersed and flat. Seen from this viewpoint, the recommendation system shall be carefully designed to mitigate the heterogeneity in users’ preference distributions by recommending to each user the cached contents that not only align with his/her own preference but also drum up strong interest from many other users. As can be inferred from previous discussion, a full-stack perspective on comprising AI and recommendation in wireless caching networks is impending.

This article aims to provide an overarching understanding of why incorporating intelligent solutions and recommendations is an important aspect in cache-enabled wireless networks, together with specific elaborations on leading design principles and advanced technologies. To be more specific, we first introduce the hierarchical, edge-cloud, content-oriented wireless caching network architecture with recommendation consideration. We then elaborate on how AI techniques and recommendation can be effectively utilized to predict users’ content requests and reshape users’ content demand patterns, respectively. Through a pair of case studies, we explicitly demonstrate the effectiveness of the developed framework and then reveal potential design problems and challenges and outline possible solutions.

Hierarchical, Recommendation-Aware Wireless Caching Networks Architecture

In this section, we present the hierarchical, cache-enabled wireless network architecture with recommendations and discuss how AI techniques and the recommendation mechanisms work in the considered architecture.

Architecture of Hierarchical, Cache-Enabled Wireless Networks with Recommendations

Figure 1 illustrates our versatile recommendation-aware, hierarchical, cache-enabled wireless network architecture, wherein subscribers communicate with small-cell BSs (SBSs) via wireless cellular communication links, and the communications between SBSs and the core network are conducted through backhaul transmissions. We note that, beyond conventional cloud/fog caching-assisted networks, where the contents are cached at the BSs only, our architecture caches at the mobile devices as well, thanks to the surge of processing power and storage capability at the user terminals. Besides, in contrast to conventional wireless caching networks where users may not be aware of the
existence of numerous content items, in our recommendation-assisted scheme, BSs can recommend locally cached items to their associated subscribers. As such, popular content items become much closer and transparent to users, leading to an enhanced cache hit ratio that, in turn, reduces the service latency and backhaul load. We note that, although the total number of content items in wireless content caching network is very huge, a large amount of data traffic is induced by a tiny fraction of popular files. For instance, 1% of YouTube videos take up more than 90% of the total views [5]. To be more specific, in our hierarchical, cache-enabled wireless system architecture, a typical user requests its intended content from its local cache entity or neighboring users with a device-to-device (D2D) communication mechanism. When recommendation is conducted, the requests will be jointly determined by users’ personalized preferences as well as the recommendation mechanism, as elaborated in the “Joint Recommendation and Caching” section. When neither the local storage nor the D2D neighbors have stored the desired item proactively, the associated SBS will retrieve the requested content for this user if the intended item was cached at the SBS. Otherwise, the SBS will first download the content from the core network via the backhaul link and then transmit it to the typical user. It is worth mentioning that the foregoing hierarchical caching model, even with no recommendations, accommodating the conventional fog/cloud-caching-oriented networks, has proved capable of attaining significant improvement on cache hit ratio, energy efficiency, spectrum efficiency, and transmission reliability [1]–[3].

The Proposed AI- and Recommendation-Enabled Paradigm

To achieve high-effectiveness cache-enabled wireless networks, we incorporate AI with recommendations in the foregoing versatile framework, as shown in Figure 2. In this paradigm, the raw data containing users’ historical behaviors are first collected and pretreated to estimate users’ preference distributions and subsequently predict users’ future content requests, which have a significant influence on the performance of cache-enabled networks. Based on the inferred users’ preference distributions, the joint recommendation and caching (JRC) decision is then performed, in which we optimize both recommendation decisions and cache placement to enhance the effectiveness (e.g., the cache hit ratio, content access delay, and content delivery efficiency) of wireless networks. Details on AI-based prediction of users’ preferences and JRC are provided in “AI-Based Preference Estimation and JRC” section.

It is noteworthy that some research efforts have recently been devoted to explicitly examine the benefits of JRC for enhancing the effectiveness of cache-enabled
wireless networks compared to pure caching mechanisms [4], [6]–[8]. Specifically, Liu and Yang [6] proposed a JRC scheme that maximizes the successful offloading probability, which is defined as the probability that a requested content from a user can be downloaded from the content provider with certain quality of service (QoS). It was shown that the designed JRC strategy outperforms the scheme without significant recommendation. In [4] and [7], the cache hit ratio maximization problem in cache-enabled wireless networks with recommendation was investigated. In [8], revenue-maximization-assisted content pushing was studied for a recommendation-aware wireless caching network with repair consideration. However, the aforementioned works commonly assumed that users’ preference distributions are available as priori information. On preference estimation for cache-enabled wireless networks, the authors in [3] and [9] utilized deep learning (DL)-assisted approaches to predict the future content popularity distribution. Therein, the spatiotemporality regarding each individual subscriber’s preference information and how it can be attained is neglected, as investigated comprehensively in the next section.

AI-Based Preference Estimation and JRC

In this section, we first describe the proposed AI-oriented users’ preference distribution estimation and decision-making diagram. Then, we demonstrate how recommendation can reshape the users’ content requests in JRC.

AI-Based Preference Estimation

As illustrated in Figure 3, AI-driven preference prediction can be divided into three steps. In Step 1, the raw data that relate to users’ historical behavior are collected. Generally, it is the responsibility of the service providers to collect the historical behaviors data (e.g., contents ratings, requests, and downloads) related to subscribers under strict privacy guarantee. In this article, the telecom operators take the role of the service provider and, hence, are in charge of data collection and storing. Then, we perform the data pretreatment with the assistance of a data processing platform in Step 2. Finally, the prediction of users’ preferences and request probability distribution can be conducted based on the processed data to do the proactive JRC for achieving a high-effectiveness caching network. In the following, we elaborate on the detailed implementations for each step.

Data Collection and Processing

The historical behaviors of mobile users can be obtained via mirroring the Gn interface by telecom operators under strict privacy guarantee, wherein Gn is used as the outflux of the network packets per subscriber to the Internet. In reality, the data collection is generally conducted from several BSs collaboratively within specific regions during a period of time, e.g., several hours or half a day, because the total packet amount in the entire system is extremely huge. After collecting the raw data, data mining is adopted to extract useful information that will be applied to estimate users’ behaviors in the near future.

Among the various software platforms for data pretreatment/processing, we choose the crowning example as Hadoop, due to its openness and efficiency [10]. Hadoop comprises four major modules. Firstly, the collected raw data are parsed to obtain the cell ID (location information), user ID, hypertext transfer protocol, control packets (as well as the time information of data packets), and so on. Via analyzing the control information and data packets, the request information for each user can be obtained; this includes user ID, packet ID, data request time point, data transmission time endpoint, and so on. In the third module, the achieved data are further analyzed to obtain the final data set with the information of time-labeled content ID and content size for each mobile user. The processed data are stored in the database in the fourth step to do the estimation and decision making in the future.

**Figure 3** The details about AI-assisted prediction and decision making.
AI-Aided Prediction and Decision Making via Processed Data

Various techniques have been involved in the AI field, e.g., machine learning, optimization theory, control theory, and so on, wherein machine learning is one of the attractive ones and has been widely utilized in wireless communications to accelerate the upcoming intelligent beyond-5G era. Based on whether the learning needs supervision or not, machine learning can be divided into three categories [5], i.e., supervised learning, unsupervised learning, and reinforcement learning. In this following, we erase the differences among the traditional categories of machine learning and introduce several potentially applicable learning strategies that can be used to do estimation and decision making for our caching networks.

1) Generative adversarial network (GAN): GAN belongs to the category of generative network, which comprises two neural networks, where one acts as a data generator and the other plays as a discriminator [11]. By training two networks to compete against each other, GAN is capable of generating new data that have the same statistics as those of the training data set. From this point of view, one of the potential applications of GAN is to learn the content-requesting behavior of subscribers with the information of their historical content requests and downloads, generating possible content requests of users in the future.

2) Long short-term memory (LSTM): LSTM has the form of a chain with repeating modules of the neural networks; thus, it has the capability of learning long-term dependencies. In other words, LSTM is a special kind of recurrent neural network (RNN). Nevertheless, in contrast to standard RNNs, where the repeating module in general has a very simple structure, namely a single tanh layer, there are four neural network layers in LSTM that interact in a more sophisticated manner, enabling it to make a powerful selection for time-series prediction tasks [12]. As for our caching network, the individual information in terms of historical data accessing can be modeled as a temporal time-sequential record; i.e., the user demands different content items within different time intervals. This time-sequential data is used as the input of the LSTM network, with which future content requests for each user can be estimated.

3) DL: DL applies a multilayered and hierarchical architecture to determine the transformation relationship between inputs and outputs, which has the generalization capability of handling unseen data. This facilitates many optimization-based decision-making tasks, such as recommendation optimization and caching designs. In addition, DL provides automatic abstraction and feature extraction from the underlying data and, thus, is applicable to do the feature generation for content items in the system. Via learning the feature vectors of content items as well as the rating history of users, personalized future requests can be anticipated [13].

Before ending this subsection, we note that, beyond the application in recommendation and caching, the proposed AI-aided prediction and decision-making framework is also applicable for user-mobility and social-connectivity-aware usage scenarios, e.g., intelligent transportation systems, D2D communication networks, and so on. Therein, the historical data per subscriber refer to mobility as well as social activities.

Joint Recommendation and Caching

We now elaborate upon the impact on the users’ requests distributions of both the personalized inherent preference distribution and the recommendation mechanism. Let $I = \{1, 2, \ldots, I\}$ be the set of all contents, which can be classified into $M$ themes. Note that these $M$ themes can be seen as the feature set that distinguishes different items. For instance, the generalized notion “video” can be classified as different categories, e.g., variety entertainment, movie, talk show, music, and so on. Define $s_i = (s_{i,m})_{m \in M}$ as the feature vector of item $i$, whose $m$th element, $s_{i,m} \in [0, 1]$, expresses the score of content $i$ in theme $m$, i.e., how much content $i$ conforms to the $m$th theme. Based on a commonly used data set, i.e., MovieLens, there are in total 1,128 thematic tags for each movie-oriented content (namely, $M = 1,128$), and, hence, $s_i$ has the dimension of $1 \times 1,128$ for $i \in I$. Similarly, we define $s_k = (s_{k,m})_{m \in M}$ as the feature vector of subscriber $k$. In detail, $s_{k,m} \in [0, 1]$ describes how much user $k$ prefers the contents that fall into theme $m$.

Let $a_{k,i}^{\text{pref}} \in [0, 1]$ be the preference of user $k$ in terms of content $i$, which can be regarded as a metric to show how much user $k$ likes item $i$, and $\Sigma_{i \in I} a_{k,i}^{\text{pref}} = 1$. Then, the inherent preference distribution of user $k$ over all content items can be characterized by the cosine similarity between $s_i$ and $s_k$ [4], [12], [14]. Specifically, we have

$$a_{k,i}^{\text{pref}} = \frac{\sum_{i=1}^{M} a_{k,i}^{\text{pref}} s_i^2}{\sqrt{\sum_{i=1}^{M} s_i^2} \sqrt{\sum_{i=1}^{M} s_{k,i}^2}}, i \in I,$$

(1)

in which

$$a_{k,i}^{\text{pref}} = \frac{\sum_{m=1}^{M} s_{k,m} s_{i,m}}{\sqrt{\sum_{m=1}^{M} s_{k,m}^2} \sqrt{\sum_{m=1}^{M} s_{i,m}^2}}, i \in I.$$  

(2)

Note that the practicality of the considered preference distribution model, i.e., (1) and (2), has been well verified by [4] (as well as references [11] and [12] therein). The other measures of similarity can also be used in this connection. In the following, we characterize the impact of the recommendation mechanism on the content-requesting behavior of each user. The following two cases are considered, namely, without and with recommendation:
1) Without the recommendation mechanism: In this case, each user demands his/her intended contents following his/her intrinsic preference pattern (i.e., $a_{ki}^{\text{pref}}$). Specifically, if user $k$ requests content item $i$ with a higher probability than that of content $j$, it has $a_{ki}^{\text{pref}} > a_{kj}^{\text{pref}}$.

2) With the recommendation mechanism: In this case, the content request probability of each user will be impacted by both the recommendation mechanism and the personalized preference distribution, that is, $a_{ki}^{\text{pref}}$. In the literature, various methods were proposed to characterize such an impact. For example, in [4], the content request distribution of each user was captured as a convex combination of the inherent preference distribution and the recommendation probability distribution. In [15], a heuristic scheme was designed where the authors mapped the recommendations to a new distribution named $a_{ki}^{\text{rec}}$ for $i \in I$, and the request probability of user $k$ on item $i$ was set to be $\max(a_{ki}^{\text{rec}}, a_{ki}^{\text{pref}})$. In summary, via recommendation, users are aware of the contents locally cached by the nearby BSs; i.e., the recommended items have boosted request probabilities, resulting in an enhanced cache hit ratio.

From the previous discussions, we note that recommendation can reshape the content request probabilities of different users via a personalized recommendation strategy, which in turn influences the cache pushing decision. From this point of view, considering recommendation and caching independently cannot reap their benefits completely. Against this backdrop, the convergence among content providers and telecom operators becomes imperative, and considering the joint cache placement and recommendation decision is the necessary tendency for constructing user and network-friendly caching-assisted systems. Moreover, it is noteworthy that each individual user’s demand for contents is time varying, which grows for a period of time after the contents first become available for retrieving and then gradually fades out [4], [6]–[8]. Considering a time scale of a few hours within a day, the demand for each content can be treated as fixed [4], [6]–[8]. In other words, the individual user’s content request predictions, the caching decisions, and the recommendation decisions are done once for every such an interval, within which users’ content preference patterns change slowly.

Case Study I: AI-Based Preference Estimation

In this case, we focus on AI-based preference prediction. Specifically, we develop an artificial request generator based on the concept of GANs, as discussed in the “AI-Based Preference Estimation” section. Our request generator can learn the historical content requests of users distributed in a certain area and then generate the content requests as if they were actually generated by these users. In other words, the generated content requests can be treated as predicted future requests from users. More specifically, the GAN-based request generator consists of three main stages, namely, input generating (IG), feature generating (FG), and content mapping (CM). Generally, the role of IG is to guarantee that the generated items match the predetermined formations regarding users’ demands and rating scores for each individual subscriber. The information generated from IG is treated as a guide for FG to produce a set of fake feature vectors via alternatively training two separated neural networks, i.e., generator and discriminator, respectively. Finally, those fake vectors are processed by CM to output movie IDs, which are considered requests from users in the future. The detailed training procedure for each stage can be found in [11].

We now verify the effectiveness of our developed artificial request generator through experiments on the real data set. We use the classical MovieLens 20 M data set (https://grouplens.org/datasets/movielens/), which contains 20 million ratings for 10,381 movies generated by 1,38,000 users. On average, each individual user provides 145 rating records. Besides, each movie is associated with 1,128 tags, wherein each tag has a relevance score. These scores are used to form a 1,128-dimensional vector to represent the properties of each movie. The data set is separated into three parts with ratios 70%, 20%, and 10% for training, testing, and validating purposes, respectively. Our problem is to recognize the movies that are preferred by users among a given set. For performance evaluation, we compare our method (denoted by GAN) with four baseline methods, as follows:

- **Genie**: All future requests of users are revealed, which achieves the optimal cache placement. We note that this scheme is idealistic, and we provide it here as the upper bound on the prediction performance.
- **Local outlier factor (LOF)**: The local density of a data point is computed from the distance to their neighbors. The points with lower local density than their neighbors are marked as outliers.
- **Isolation forest (IF)**: Items are distributed to nodes of a tree by randomly selecting feature and splitting values. The scores are computed from the depth of each items. Low-score items will be marked as outliers.
- **One-class support vector machine (OSVM)**: A minimal-volume hyper-spherical that contains all the training data points is formed. Then, a new point is classified as an outlier if it stays outside the defined boundary.

**Case Studies**

In this section, we conduct case studies to validate the performance of our proposed AI- and recommendation-enabled wireless content caching framework. More precisely, we first verify the effectiveness of applying AI techniques in estimating users’ preference distributions. We then show how JRC can be utilized to minimize the total content delivery latency of multicell, cache-aware wireless networks.
Figure 4(a) depicts the prediction performance of the proposed requester during its training phase when normalized by the performance of the Genie scheme, which is the optimal, nevertheless unpractical baseline. We can see that the proposed GAN-based requester performs equally well on the training set, the testing set, and the validation set, showing that our proposed requester is not overfitted. In addition, we see that the proposed requester can achieve almost 80% of the cache hit ratio when compared to that of the optimal scheme. In Figure 4(b), we compare the cache hit rate of the proposed requester with those of LOF, IF, and OSVM methods. It can be seen that the cache hit rate of our GAN-based requester is superior to the other three approaches, indicating that the generated requests from our proposed requester can reliably serve as the predicted, future, preferred requests from the users.

Case Study II: JRC for Latency Minimization

So far, we have shown how the AI techniques, such as GAN, can be utilized to predict the users’ future content requests. Next, we exemplify the performance of JRC in cache-enabled wireless networks to minimize the total content delivery latency.

To be more specific, we consider a wireless content caching network with $N_c$ cells. Within cell $n$, there is one BS located at the center of the cell disk, serving $K_n$ mobile users. We assume that BS $n$ is equipped with $N_T$ antennas. Meanwhile, each user is equipped with a single antenna. It is assumed that BS $n$ has collocated with a content server, whose capacity is finite, denoted by $C_n$ (in bits). The users within each cell request their own desired contents from their associated BSs. The BS will transmit the required contents to the users, given that it cached the corresponding contents. Otherwise, the BS will first download the packets from the cloud server and then forward them to its attached users. We assume that the remote server has a relatively large cache capacity that can provide all kinds of the requested packets from the network users. Note that the communication between BSs and the remote server uses a wired backhaul link, while the users communicate with the BSs by the wireless radio resource.

There are $F$ content items in the file library. Denote by $L_f$ the volume of content item $f$. In addition, we assume that the statistical channel information is known by all of the BSs. We focus on minimizing the average transmission latency of the system taking into consideration the cache capacity budget and maximum transmit power requirement of each BS. The joint beamforming, cache decision, and recommendation decision optimization problem happens to be a mixed-integer, nonlinear programming, which is NP-hard in general. A joint-optimization method and a time-efficient alternative algorithm named JBCRD and TT-SOR, respectively, are proposed in our work [14]. In addition, we also compare the performance of the proposed methods with extensive benchmarks, which are distinguished next.

- **Top cache and top recommendation (TCTR):** In this method, each BS stores the aggregated top-ranked contents among its associated users and recommends the top preferred items for each user in accordance with his/her inherent preference distribution, i.e., $\alpha_{k,i}^{pref}$.
- **Top cache and no recommendation (TCNR):** In this case, the top-ranked contents are cached at each BS. Nevertheless, no recommendation is conducted.
- **No cache and random recommendation (NCRR):** There is no cache at the BS, and it randomly recommends content items to each user.
- **Random cache and random recommendation (RCRR):** Items are randomly cached at the BSs to reach their capacity budgets, and contents are randomly recommended to different users.

![Figure 4](image-url)
The simulation parameters are summarized as follows. We consider $N = 3$ cells with a cell radius of 200 m, and each cell serves $K_o = 2$ mobile users. The antenna numbers at each BS and user are set to be $N_r = 4$ and 1, respectively. The system bandwidth is assumed to be 10 MHz, while the noise power density is set to be $-174$ dBm/Hz. In addition, the content item number $F = 30$ and the size of each content, that is $L_i$, uniformly distribute within the interval of $(2 \times 10^7, 3.2 \times 10^8)$ in bits. Besides, each of the users can be recommended 10 items at most, and the recommendation probability is assumed to be 0.1. The corresponding numerical results are presented in Figure 5, wherein the content delivery latency versus backhaul transmission rate is demonstrated. From Figure 5, it is observed that our optimized JRC strategies can achieve better system performance when compared to their counterparts. For instance, when the backhaul transmission rate is set to be $10^8$ bits/s, JBCRD saves time by 118% when compared to the scheme without recommendation (i.e., TCNR). It is worth mentioning that the strategy TCNR is the commonly conducted caching strategy without a recommendation mechanism in prevalent wireless content caching networks. Moreover, we can see that the content delivery delay first decreases and then approaches a fixed value as the backhaul link data rate increases. This is because the content delivery delay consists of two components, namely, the backhaul delay and the content transmission delay. When the backhaul link rate is low, increasing the backhaul link data rate can reduce the backhaul delay and, consequently, the content delivery delay. However, when the backhaul link data rate is relatively high, the backhaul delay approaches a fixed small value. As such, continuing to increase the backhaul link rate cannot further reduce the content delivery delay.

Conclusions and Key Challenges

In this article, we introduced a new paradigm toward high-effectiveness cache-enabled wireless networks, particularly built around AI and recommendation. Specifically, we first presented the hierarchical, cache-enabled wireless network structure and then developed a novel framework that incorporates AI techniques and a recommendation mechanism to, respectively, estimate and reshape users’ preference distributions. The effectiveness of the proposed framework has been amply illustrated via a pair of case studies. Armed with the proposed models, a bright future perspective is an equally important problem yet receives much less attention.

2) Ethical nature-oriented recommendation mechanism: The generally conducted recommendation mechanism focuses on optimizing network-centric performance metrics (e.g., latency, cache hit ratio, and so on) and assumes that users should fully accept the recommendations, even though they do not know the quality of the recommended items. Exploring effective recommendation mechanisms from user-centric perspectives is an equally important problem yet receives much less attention.

3) JRC for D2D-aided wireless content caching networks: The rapid development of mobile terminal technology makes the caching capacity of users full of tremendous wealth in terms of achieving high-effectiveness wireless networks. The JRC optimization for D2D users is more challenging than that of pure-cache-aided D2D due to the binary recommendation decision, especially when the similarity and individual preference of users are taken into account. Thereby, how to design high-effectiveness and high-efficiency JRC algorithms for cache-enabled D2D users with various 5G techniques (e.g., non-orthogonal multiple access, multiple-input/multiple-output, and so on) becomes another essential problem.

4) Data security and individual privacy: To achieve a good proactive JRC decision for cache-enabled

![Figure 5](image-url)

**Figure 5** A comparison between the proposed JRC schemes and their various benchmarks, wherein the transmit power budget per BS is set to be 2 W. In addition, we assume that $C_{r_u} = 10^8$ bits and that the user preference, $a_{r_u}^{r_u}$, is generated in a random walk manner. TTSAO: two-time scale alternative optimization algorithm.
wireless networks, a large volume of personal data should be monitored, collected, and processed. Any data leakage may incur serious security and privacy concerns. Taking strong measures to enhance data security, secure personal privacy, and reduce disclosure risks becomes imperative.

Acknowledgments
This work was supported in part by the Hong Kong President’s Advisory Committee on Research and Development under project 2020/1.6, in part by the National Natural Science Foundation of China under grants 61971239 and 62001047, and in part by the Fundamental Research Funds for the Central Universities under grants 2020RC09 and 19LGPY79. The corresponding authors are Khai Nguyen Doan and Chenxi Liu.

Author Information
Yaru Fu (yfu@ouhk.edu.hk) is a research assistant professor with the School of Science and Technology, The Open University of Hong Kong, China. She received her Ph.D degree in electronic engineering from the City University of Hong Kong. She is a Member of IEEE.

Howard H. Yang (eehowardh@gmail.com) is an assistant professor with the Zhejiang University/University of Illinois at the Urbana-Champaign Institute, Zhejiang University, Haining, China, and is also with the Information System Technology and Design Pillar, Singapore University of Technology and Design (SUTD), Singapore. He received his Ph.D degree from SUTD. He is a Member of IEEE.

Khai Nguyen Doan (khai17014@gmail.com) is currently pursuing a Ph.D degree with the Singapore University of Technology and Design (SUTD), Singapore. He is with the Information Systems Technology and Design Pillar, SUTD, and is a Student Member of IEEE.

Chenxi Liu (chenxi.liu@bupt.edu.cn) is an associate professor with the Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China. He received his Ph.D degree from the University of New South Wales, Sydney, Australia. He is a Member of IEEE.

Xijun Wang (wangxijun@mail.sysu.edu.cn) is an associate professor with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. He received his Ph.D degree in electronic engineering from Tsinghua University, Beijing. He is a Member of IEEE.

Tony Q.S. Quek (tonyquek@sutd.edu.sg) is a tenured full professor with the Singapore University of Technology and Design (SUTD), Singapore. He also serves as the head of the Information Systems Technology and Design Pillar, SUTD, and deputy director of the SUTD-Zhejiang University Innovation, Design, and Entrepreneurship Alliance. He is a Fellow of IEEE.

References