IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2019, accepted May 11, 2019, date of publication May 22, 2019, date of current version June 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918325

Enhancing Mobile App User Understanding and Marketing With Heterogeneous Crowdsourced Data: A Review

BIN GUO[1], (Senior Member, IEEE), YI OUYANG[1], TONG GUO[1], LONGBING CAO[2], (Senior Member, IEEE), AND ZHIWEN YU[1], (Senior Member, IEEE)

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2]Advanced Analytics Institute, University of Technology Sydney, Ultimo, NSW 2007, Australia

Corresponding author: Bin Guo (guob@nwpu.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1001800 and in part by the National Natural Science Foundation of China under Grant 61772428 and Grant 61725205.

ABSTRACT The mobile app market has been surging in recent years. It has some key differentiating characteristics which make it different from traditional markets. To enhance mobile app development and marketing, it is important to study the key research challenges such as app user profiling, usage pattern understanding, popularity prediction, requirement and feedback mining, and so on. This paper reviews CrowdApp, a research field that leverages heterogeneous crowdsourced data for mobile app user understanding and marketing. We first characterize the opportunities of the CrowdApp, and then present the key research challenges and state-of-the-art techniques to deal with these challenges. We further discuss the open issues and future trends of the CrowdApp. Finally, an evolvable app ecosystem architecture based on heterogeneous crowdsourced data is presented.

INDEX TERMS App marketing, user profiling, popularity prediction, app recommendation, usage pattern mining, mobile crowdsourcing.

I. INTRODUCTION

The number and popularity of mobile apps is rising dramatically as there is an accelerating rate of adoption of smartphones. By the end of 2017, there have been more than 5.6 million apps at Apple app store and Google Play. Moreover, the growth in the number of available apps has been accompanied by an exponential increase in downloads. As reported,[1] 150 billion mobile apps have been downloaded in 2015, and the number is predicted to reach 200 billion in 2018. With several million apps available, one of the most challenging problems faced by developers is to catch the attention of users. To this end, the app market has been defined as a ''hypercompetitive'' marketplace, as recently found in the free-floating shared bike app market (e.g., Mobike,[2] Ofo[3]) [1] and on-demand takeout ordering & delivery app market (e.g., ele.me,[4] Uber Eats[5]) [2].

It has been reported that mobile app store markets exhibit key characteristics of the ''long tail'' phenomenon [3], [4]. However, app store market structure has some characteristics which makes it distinguished from traditional long-tail markets such as books, music, and movies. First, unlike music creators, app developers have the opportunity to make changes to the app by new releases according to user feedbacks and reviews. Second, sellers in mobile app markets compete more directly with other developers. The business strategies (e.g., app rank charts, app lists by categories) used in app stores makes it easier to compare competing apps within a category than comparing music offerings within a genre. Finally, compared to other products, mobile apps are used in sensor-rich devices and we can collect various data about apps from both real-world usage and online social media.

Smartphones have become human companions, and it is possible for app developers to leverage the heterogeneous crowdsourced data about mobile apps, such as app usage, app reviews, user ratings, and app-related posts, to better understand the mobile app market and improve user experience. We term the usage of heterogeneous crowdsourced app data for mobile app user understanding and marketing as CrowdApp.

The associate editor coordinating the review of this manuscript and approving it for publication was Javier Medina.

[1]https://www.mobilemarketer.com/ex/mobilemarketer/cms/news/software-technology/20699.html
[2]http://www.mobike.com/
[3]http://www.ofo.so/
[4]http://www.ele.me/
[5]https://www.ubereats.com/

As a promising and rapidly developing research area, there have been limited literature reviews about mobile apps. Quite recently, [5], [6] review mobile apps from the perspective of software engineering, focus on technical attributes such as API usage, requirement analysis, code/library analysis, faults, release planning, and permissions. Moreover, the data studied are mainly limited to app stores. Different from them, our paper is presented via a human-centric manner, from the perspective of using heterogeneous crowdsourced data for app user understanding and marketing. The topics investigated include app popularity forecasting, usage pattern understanding, user profiling and app recommendation, usage prediction, and so on. It should be noted that we are not intended to provide a systematic literature review. The area of app marketing is still developing, but has not reached a level of maturity at which research questions can be chosen and asked of a well-defined body of literature.

The following gives a summary of the main contributions of our work.

(1) Characterizing the core concepts and features of CrowdApp. We first present the heterogeneous crowdsourced app data, and then characterize the cyber-physical-social and data-driven features of CrowdApp systems.

(2) Reviewing key research challenges and techniques of CrowdApp, including usage pattern understanding, review analysis and processing, requirement and feedback mining, popularity forecasting, app usage prediction, user profiling and app recommendation.

(3) Investigating the open issues and future trends of this research field, such as competitive intelligence and app evolution, personalized and context-aware app adaptation, fine-grained app popularity prediction, and heterogeneous crowdsourced data mining.

(4) Presenting the generic framework of the CrowdApp ecosystem, which is a crowdsourced data-driven app ecosystem that can better support the development and marketing of mobile apps.

The remaining paper is organized as follows. In section II, we characterize the unique features of mobile apps. Section III presents the key techniques and application areas. Our insights and future research directions are discussed in Section IV. Section V presents the generic framework of CrowdApp ecosystem. We conclude the paper in Section VI.

II. OPPORTUNITIES OF CROWDSOURCING FOR MOBILE APPS

A. HETEROGENEOUS CROWDSOURCED APP DATA

Crowd computing presents the usage of heterogeneous, multi-modal, and complementary crowd-contributed data for reaching a comprehensive picture of the interested target [7]. CrowdApp is a particular research field under the crowd computing perspective. For mobile apps, we have at least the following data sources that are contributed by app users and developers.

- *App stores*. There are different types of information in app stores, including the information from developers, such as app features and app description, and the information from app users, such as ratings, comments, downloads, etc. To facilitate the adoption of mobile apps and understand user experience with apps, many app stores provide the periodical app chart rankings and allow users to post ratings and reviews for apps. Such popularity information plays an important role in mobile app services.
- *Smartphone logs and app management tools*. At mobile clients, app usage data and additional contexts (e.g., time, location) can be recorded as smartphone logs or kept in app management tools (e.g., Wandoujia[6]).
- *Posts in social media*. People report their user experience, feedback, and comments about mobile apps. They also make comparisons over similar apps (e.g., Uber[7] and Didi[8]). App developers also publish and disseminate posts about their marketing activities in the social media.

B. CHARACTERIZING CROWDAPP SYSTEMS

Compared with traditional software systems, there are two significant features of CrowdApp systems, as presented below.

Cyber-physical-social systems (CPSS). CPSS refers to the incorporation of cyber-, physical-, and social-elements in the design and usage of computing systems [8]. Conventional software engineering processes are rather transactional and lack a common theory for the involvement of users and their communities. Smartphones have become human "companions" and mobile apps are closely adhered to human daily activities in an anytime, anywhere manner. We thus should extend the app ecosystem by incorporating physical and social elements. Maalej and Pagano [9] define the socialness of a software system as "*the degree of involvement of its users and their communities in the software lifecycle*". Software users can be involved either by actively working on a specific engineering activity (e.g., suggesting modifications and enhancements), or by influencing a management decision about the software (e.g., give feedback, influence the opinion of others). Guzman *et al.* [10] reports on the analysis and mining of app-relevant tweets to support the continuous evolution of the apps. Beyond social features, physical features are also important as mobile apps, e.g., user daily activities, usage behaviors, spatio-temporal contexts, and so on. Smartphones and wearable devices are equipped with various sensors that can understand the surrounding environment and learn the preferences of users.

Data-driven CrowdApp systems. The data-driven mobile app systems refer to the usage of heterogeneous crowdsourced data to understand human usage patterns and user feedback, which can enhance user experience and marketing. We characterize them as follows.

[6]http://www.wandoujia.com/
[7]http://www.uber.com/
[8]http://www.didichuxing.com/

TABLE 1. App usage pattern understanding.

Work	Problem and method	Dataset
Li *et al*. [11]	Exploring multiple patterns of app usage; Method: systematic descriptive analysis	0.26 million Android apps, app usage from 2 million users, one-month of app management activities from 0.8 million users, Source: Wandoujia
Liu *et al*. [12]	Exploring multiple patterns of app usage; Method: systematic descriptive analysis	0.28 million Android apps, app usage from 6 million users, five-month of app management activities from 17 million users, Source: Wandoujia
Li *et al*. [13]	Discovering app usage patterns from large-scale users; Method: correlation analysis	0.2 million Android apps, 0.7 million users, five-month of usage records, Source: Wandoujia
Jones *et al*. [14]	The revisitation analysis of app use; Method: micro-level analysis	1,527 Android apps, 165 users, three-month of app launch logs, 199,859 app launch events, Source: Google Play
Zhao *et al*. [15]	Discovering different kinds of smartphone users; Method: clustering, feature ranking	77,685 Android apps, 106,762 users, one-month of usage records, 52,872,129 app launch events, Source: A mobile Internet Company in China
Liu *et al*. [17] Li *et al*. [18]	Inferring user preferences from app management activities; Method: systematic descriptive analysis	1 million Android apps, 17 million users, five-month of usage records, 240 million app launch events, Source: Wandoujia
Xu *et al*. [20]	Identifying diverse usage behaviors of apps; Method: systematic descriptive analysis	Collecting anonymized IP-level networking traces in a large tier-1 cellular network in the U.S. for one week in Aug. 2010, covering different sources

- *Mobile app user understanding*. The benefits of mining app usage data are at least two folds. First, it allows us to learn app usage patterns of users, which can improve the app design and improve user experience by better rendering the apps or preloading them. Second, we can distill user profiles (e.g., age, job, personality, and interests) and better suggest highly-relevant apps to users.
- *Mobile app marketing*. By mining user feedback in app stores and their interaction with apps, app marketing can be enhanced. For example, by analyzing user comments and ratings, we can learn how users think about the app and their suggestions to improve the app. The downloading dynamics of apps enable us to predict their future popularity and define more intelligent marketing and releasing plans. With past installed apps and usage patterns, we can recommend apps to users according to their habits and preferences.

III. KEY RESEARCH CHALLENGES AND TECHNIQUES

A. APP USAGE PATTERN UNDERSTANDING

Investigating how people manage mobile apps in their everyday lives creates a unique opportunity to understand the behaviors and preferences of mobile device users, infer the quality of apps, and improve user experience. A summary of app usage pattern understanding is given in Table 1.

Usage pattern analysis. Recent studies [11], [12] characterize the popularity of apps with various metrics including the number of downloads, the number of unique users, the volume of data traffic, and the length of network-access time. They validate that the distribution of app downloads follows the ''Pareto-like'' principle and further indicate that the popularity of apps typically complies with the power law, i.e., only a small proportion of apps account for substantial downloads. Li and Lu [13] present the correlation analysis of choice of device models against the user behaviors of using Android apps. Some significant correlations between device models and app usage are derived, leading to important findings on the various user behaviors. For example, users with different device models show substantial diversity on competing app selection, and users of lower-end devices spend more money to purchase apps and spend more time under cellular networks. Jones *et al.* [14] cluster at least three different kinds of smartphone users based on their app re-visitation patterns. Zhao *et al.* [15] identify distinct types of users based on their app usage, and they also demonstrate a strong relationship between demographics and app usage. For instance, by linking users with the features such as category, time of day, workday versus weekend, they can category them into different groups, such as evening learners or screen checkers. Chen *et al.* [16] develop a matrix factorization approach that models speech and app usage patterns to predict user intents (e.g. launching a specific app).

Preference analysis. By mining app usage records, user preferences can be learned. For example, [17], [18] aim to understand the preferences of mobile users through mining app management activities. By analyzing a very-large dataset from an app management tool, they find that the metrics commonly used to rank apps in app stores do not truly reflect users' real preferences. Furthermore, they identify behavioral patterns from the app management activities that can more accurately indicate user preferences of an app even when no explicit rating is available. They also make statistical analysis to evaluate machine learning models that are trained to predict user preferences using the identified behavioral patterns.

TABLE 2. Review analysis and processing.

Work	Problem and method	Dataset
Khalid *et al.* [22]	Investigating the various types of complaints Method: manually tagging	iOS apps, 15,000 reviews, Source: Apple Store
Mujahid *et al.* [23]	Examining user complaints of wearable apps Method: manually analysis	6 Android wearable apps, 589 reviews, Source: Google Play
Vu *et al.* [24]	Mining user opinions in app reviews Method: keyword-based natural language processing	95 Android apps, 3 million reviews, Source: Google Play
Pagano *et al.* [25]	The empirical summary of user reviewing behavior Method: statistical analysis	1,100 iOS apps, 1,126,453 reviews, Source: Apple Store
Keertipati *et al.* [26]	Prioritizing features for app improvements Method: text mining, feature prioritization	1 Android app, 4,442 reviews, Source: Google Play
Chen *et al.* [27]	Extracting informative reviews Method: review prioritization	4 Android apps, five-month reviews, Source: Google Play
Park *et al.* [28]	Mining target users for app marketing Method: the random walk method	122,875 Android apps, 2 million users, One-year of user installed log, 45 million records Source: Huawei App Store
Palomba *et al.* [29]	Linking reviews to code changes Method: IR-based approaches	100 Android apps, 5,691 reviews, Source: Google Play
Villarroel *et al.* [30]	Releasing planning of apps Method: text classification, clustering	200 Android apps, 1,000 reviews, Source: Google Play
Li *et al.* [31]	Mining reviews for app comparisons Method: text processing, opinion mining	2,311 Android apps, 5 million reviews, Source: Google Play
Guzman *et al.* [32] Panichella *et al.* [33]	Classifying app reviews Method: text classification, sentiment analysis	7 Android apps, 4,550 reviews, Source: Google Play and Apple store
Mcllroy *et al.* [34]	Analyzing and labeling the types of user issues Method: automated labeling, opinion mining	24 Android and iOS apps, 230,277 reviews, Source: Google Play and Apple store
Maalej *et al.* [35]	Classifying app reviews Method: text processing, text classification	1180 Android and iOS apps 1,303,182 reviews, Source: Google Play and Apple store
Ciurumelea *et al.* [38]	Review summarization Method: IR-based, text classification	39 Android apps, 1,566 reviews, Source: Google Play

Context-related patterns. Many usage patterns are relevant to user contexts, such as time and location. For example, Hintze *et al.* [19] conduct a large scale and long-term analysis of mobile device usage characteristics such as session length, interaction frequency, and daily usage in locked/unlocked state with respect to location context and diurnal pattern. The results indicate that contexts have a highly significant effect on both frequency and extent of app usage. Xu *et al.* [20] characterize the usage pattern of smartphone apps in a 3G cellular network. They identify the correlations between spatial and temporal factors and usage, and observe that different types of apps have different diurnal and mobility patterns.

B. REVIEW ANALYSIS AND PROCESSING

Users who encounter issues will likely stop using an app. Even worse, negative reviews in early releases make recovering afterwards rather difficult. So, one of app developers' main goals is to detect and respond as quickly as possible to quality issues. User feedback plays a paramount role in the development and maintenance of mobile apps [21]. Timely and constructive feedback from users becomes extremely crucial for developers to fix bugs, implement new features, and improve user experience agilely. A summary of the representative studies on review analysis and processing is given in Table 2.

Review analysis. Khalid *et al.* [22] study the types of complaints and investigate how they affect app ratings. Their findings enable developers to better anticipate possible complaints and prioritize the resources to the most important ones. Mujahid *et al.* [23] analyze user reviews to understand user complaints of wearable apps. Their findings indicate that the most frequent complaints are related to functional errors, lack of functionality, and cost. Vu *et al.* [24] propose a keyword-based framework for semi-automated review analysis, which can summarize the reviews for referencing. Pagano and Maalej [25] collect a sample of 1.1 million reviews from the Apple App Store to investigate how and when users provide feedback. They find that positive feedback is often associated with highly downloaded apps.

Review selection and ranking. The reviews contributed by app users are often of varied quality. Review selection and prioritizing is thus important to choose the most important reviews for referencing. Keertipati *et al.* [26] select reviews with negative emotions (e.g., sadness, anger, fear) to prioritize features for improvements. Chen *et al.* [27] aim to extract the most informative reviews and placing weights on negative sentiment reviews. They first extract informative user reviews by filtering noisy and irrelevant ones, and further prioritize the informative reviews by an effective review ranking scheme. Park *et al.* [28] utilize user reviews to improve mobile app retrieval by designing a topic model, AppLDA, which discards review-only topics. Palomba *et al.* [29] study the effects of informative user reviews. They find that a mean of 49 percent of review requests were implemented in new releases, and that the apps with changes based on user reviews

TABLE 3. Requirement and feedback mining.

Work	Problem and method	Dataset
Sorbo *et al.* [39]	App review summarization Method: text summarization	17 Android apps, 1,390 reviews, Source: Google Play
Gu *et al.* [40]	App review summarization Method: text summarization	17 Android apps, 2,000 reviews, Source: Google Play
Fu *et al.* [41]	App review summarization and analysis Method: text mining, sentiment analysis, topic model	171,493 Android apps, 13 million reviews, Source: Google Play
Galvis *et al.* [42]	App review analysis Method: text mining, sentiment analysis, topic model	3 Android apps, 2,651 reviews, Source: Google Play
Mcilroy *et al.* [43]	Case studies on review value to app improvement Method: manual analysis, statistical analysis	12,000 Android apps, 0.6 million reviews, Source: Google Play
Palomba *et al.* [46]	User request recommendation Method: text processing, text mining	10 Android apps, 44,683 reviews, Source: Google Play
Jin *et al.* [47]	Feature request extraction Method: text mining, sentiment analysis	6 mobile phones, 5,730 reviews Source: Epinions.com
Iacob *et al.* [48]	Feature request extraction Method: text mining, text summarization, linguistic rules	169 Android apps, 3,279 reviews Source: Google Play
Sharma *et al.* [49]	Software relevant tweets identification and feature extraction Method: language modeling	66 million sentences Source: Twitter and StackOverflow
Guzman *et al.* [50]	Feature extraction Method: text mining, sentiment analysis	7 apps, 32,210 reviews Source: Google Play and Apple Store
Vu *et al.* [51]	Mining user opinions on app feature requests Method: phrase-based text mining, text processing	200,000 reviews Source: Amazon.com

can improve their ratings. Villarroel *et al.* [30] develop a tool that can automatically cluster and prioritize reviews. Li *et al.* [31] present a method to identify comparative reviews for mobile apps from app stores, which can be used to provide fine-grained app comparisons over different topics.

App review classification. App reviews are with different topics and thus the classification of app reviews can help better manage and understand the reviews. Guzman *et al.* [32] propose an ensemble of machine learning classifiers to classify user reviews. Panichella *et al.* [33] present a system that automatically classifies user reviews based on a predetermined taxonomy to support software maintenance and requirement evolution. McIlroy *et al.* [34] present an automated labeling scheme that can identify multiple elements or tags to reviews. For example, a review may contain a feature request and a bug report, and a label for each type will be tagged to the review. Maalej and Nabil [35] propose a classification method to classify app reviews into four types: bug reports, feature requests, user experiences, and ratings. Hosseini *et al.* [36] propose to utilize the power of the crowd to support richer, more powerful text mining by enabling the crowd to categorize and annotate feedback through a context menu. Similarly, Murukannaiah *et al.* [37] propose a sequential crowd requirement engineering process, where workers in one stage review requirements from the previous stage and produce additional requirements. To help developers deal with the large amount of available data, Ciurumelea *et al.* [38] propose an approach that can automatically organize app reviews according to predefined tasks (e.g., battery, performance, and memory). It can further recommend the related app parts that should be changed.

C. REQUIREMENT AND FEEDBACK MINING

User-contributed data in app stores and app-relevant microblogging posts usually present their feedback based on user experience about the interested apps. Such kinds of information are valuable to extract user requirements and app feature requests. A summary of the existing works on app requirement and feedback mining is given in Table 3.

App review summarization. User-contributed reviews in app stores are usually at a large-scale and thus review summarization becomes important to facilitate developers to understand diverse user requests and feelings. Sorbo *et al.* [39] introduce SURF (Summarizer of User Reviews Feedback) to capture user needs for developers performing maintenance and evolution tasks. Gu and Kim [40] present a novel review summarization framework, SURMiner, which can produce a visualization of the reviews to developers. A large-scale study is given by Fu *et al.* [41], where over 13 million app reviews are analyzed for summarization. They find that there is a large difference between free and paid apps, and that paid apps have an associated 'complaint' type about price. The work is useful for large-scale overviews of competitor apps and gathering information about the app market. Galvis Carreño and Winbladh [42] extract the main topics as well as some representative sentences of those topics from user reviews. This information is useful for requirement engineers to revise the requirements for next releases.

Requirement extraction. Requirement extraction focuses on capturing the needs and feedback from users. Traditionally requirement extraction is conducted through interviews, workshops, and focused groups. Recently, social media channels have become important venues for collecting user feedback. With the emergence of app stores as a software

TABLE 4. **Popularity forecasting.**

Work	Problem and method	Dataset
Petsas *et al*. [58]	App popularity analysis Method: statistical model	316,143 Android apps, app statistics data, Source: Google Play
Finkelstein *et al*. [59]	Correlation analysis of popularity and rating Method: text mining, statistical analysis	42,000 apps, app statistics data, Source: Blackberry App Store
Wang *et al*. [60]	Download patterns of mobile game apps Method: statistical analysis	75,000 game apps, app statistics data, Source: four app stores
Tian *et al*. [61]	High-rated app analysis Method: statistical analysis	1,492 Android apps, app statistics data, Source: Google Play
Zhu *et al*. [62]	App popularity prediction model Method: hidden Markov model	15,045 iOS apps, app statistics data, Source: Apple store
Lee *et al*. [4]	Feature measurement for app popularity prediction Method: hierarchical linear model	17,697 iOS app, app statistics data, Source: Apple Store

marketplace and user-participatory community, users can easily submit their feedback, review new releases, report bugs, rate apps, or request new features. They can be broadly categorized into the following types.

- *Bug reports*. It refers to the problems in an app that needs to be corrected, such as app crashes, erroneous behaviors, or performance-related issues.
- *User experiences*. It refers to the user experience related to certain app features, such as the app user interface, loading/processing speed, and so on.
- *Praises*. It indicates the reviews where users express general appreciation and positive feedback with the app.
- *Feature requests*. For feature requests, users ask for missing functionality or missing content. They are valuable information that helps developers improve the app by adding, augmenting or changing features [43].

Overall, we can extract various user requirements from the automatically collected usage data, logs, and interaction traces, which help developers understand user feedback and react to it [44], [45]. The varying review dynamics seen in different app stores also help design future app development strategies.

Feature mining. Each app usually contains numerous features and users often express their feelings on specific app features. User feedback and attributes on app features are particularly important for future app improvement. Palomba *et al*. [46] propose to link relevant user feedback extracted from user reviews onto source code elements. They analyze the semantics and sentiments of sentences contained in user reviews to extract useful feedback and recommend to developers potential changes. Jin *et al*. [47] study aspects of product features and detailed reasons which are extracted from online reviews to inform designers regarding what leads to unsatisfied opinions. Iacob and Harrison [48] present MARA, an automated system that extracts and analyzes app reviews to identify feature requests. Sharma *et al*. [49] automatically identify app-relevant tweets based on language modeling and statistical analysis. Guzman and Maalej [50] use natural language processing techniques to identify fine-grained app features in the reviews. They then extract user sentiments about the identified features and give them a general score across all reviews. Vu *et al*. [51] present a phrase-based approach to extract user opinions about feature requests from app reviews.

D. POPULARITY FORECASTING

Popularity modeling and forecasting is quite important in mobile Internet marketing. The worldwide prevalence of mobile apps leads to fierce competition, and many apps will die out as a result. To thrive in this competitive app market, it is vital for app developers to understand the popularity evolution of their mobile apps, and make strategic decisions for mobile app development. Therefore, it is significant and necessary to forecast the future popularity evolution of mobile apps.

Popularity prediction has been a trending research area in recent years, especially for social contents, such as: news [52], [53], microblogs [54], [55], videos [56], [57], and so on. However, mobile apps are different from the social contents in the following aspects. First, the lifecycle of app evolution is a long-term process, which will last one year or even several years, while the lifecycle of social contents will be shorter. During the long-term evolution, there may be various complex factors affecting the popularity. Second, update is a unique characteristic of mobile apps. Through updating the version, mobile apps can continuously improve themselves and prolong the lifecycle, while social contents can no longer change their properties since released. Therefore, it is crucial to study the popularity patterns and prediction methods for mobile apps. A summary of popularity forecasting is given in Table 4.

Popularity analysis. To predict app popularity, it is important to first understand the popularity patterns and impact factors. Various factors may affect the popularity of apps, such as: price [58], [59], rating [59], [60], review [60], meta-information [59], [61], and so on. Petsas *et al*. [58] present a systematic study on app popularity distribution and app pricing by collecting and analyzing data from four popular third-party Android app stores. They find that 10 percent of the apps accounted for at least 70 percent of the total downloads in the stores, and that popularity follows a power-law distribution against app price of paid apps.

TABLE 5. App usage prediction.

Work	Problem and method	Dataset
Lee *et al.* [65]	Usage pattern analysis Method: unsupervised learning, optimization	100 Android users, 1,057 days, app usage log, By crowdsourcing
Tan *et al.* [66]	Context-based app usage pattern prediction Method: unsupervised learning	38 users, 2 months ~ 1 year, app usage log, Source: Nokia MDC
Yan *et al.* [67]	App prediction & preloading Method: optimization, nearest neighbor	34 users, 14 months, app usage log, Open source dataset
Liao *et al.* [68, 69]	Low cost app usage prediction Method: temporal-based prediction, feature selection	15 users for 6 months, 80 users for 5 months, app usage log, Open source dataset
Parate *et al.* [70]	Low cost app usage prediction Method: no-prior-training based approach	34 users, 14 months, app usage log, Open source dataset
Zou *et al.* [71]	Low cost app usage prediction Method: lightweight Bayesian methods	80 users, app usage log, Open source dataset
Shin *et al.* [72]	Intelligent app rendering Method: naïve Bayes method	23 users, 1 month, sensory data, By crowdsourcing
Ricardo *et al.* [73]	Intelligent app rendering Method: Parallel Tree Augmented Naive Bayesian Network	480 users, 7 months, app usage log, By crowdsourcing
Wang *et al.* [74]	Context-based app usage prediction Method: contextual collaborative filtering	65 days, app usage log, Open source dataset
Huang *et al.* [75]	Context-based app usage prediction Method: Bayesian methods, linear model	38 users, 2 months ~ 1 year, app usage log, Source: Nokia MDC
Xu *et al.* [76]	Context-based app usage prediction Method: nearest neighbor classification	4,606 users, app usage log, Open source dataset

Finkelstein *et al.* [59] extract app descriptions, price, rating, and popularity information from the Blackberry World App Store. The findings reveal that there are strong correlations between customer rating and popularity (rank of app downloads). Wang *et al.* [60] collect information of over 75,000 mobile game apps in a period of three months. They investigate the download patterns of mobile game apps and the impacts brought by user comments and ratings. Tian *et al.* [61] extract API information and evaluate apps in terms of code complexity, API dependency, API quality, as well as a number of other factors, in order to train features to distinguish high from low rated apps. They use 28 factors to characterize the differences between high-rated apps and low-rated apps.

Popularity prediction. With the identified impact factors and patterns, we can then develop models and algorithms to predict the future popularity. Zhu *et al.* [62] use popularity information to construct a popularity-based Hidden Markov Model (PHMM), to encode trend and other latent factors. The authors stated that this can be used in a variety of ways, including app recommendation, review spam detection, and ranking fraud detection. Lee and Raghu [4] study the factors that affect an app's likelihood of staying in the top (most popular) charts in the Apple App Store. They find that free apps are more likely to 'survive' in the top charts, and that frequent feature updates are the most important factor in ensuring their survival, along with releasing in smaller categories. The authors also find that high volumes of positive reviews can improve an app's likelihood of survival. Wang *et al.* [63] consider the competition among different mobile apps, and they proposed an evolutionary hierarchical competition model to forecast the app downloads. Lu *et al.* [64] study the correlations between developer-controllable app properties (e.g., code-level metrics, textual descriptions, and so on) and large-scale user behaviors. Based on the correlations, they use the machine learning algorithms to predict user adoption of apps.

E. APP USAGE PREDICTION

Predicting app usage has become an important task due to the proliferation and complexity of apps. Such an app usage prediction framework is a crucial prerequisite for fast app launching, intelligent user experience, and power management of smartphones. A summary of the studies on app usage prediction is given in Table 5.

Usage analysis and prediction. Many studies have been done on app usage analysis collected from real-world usage. Lee *et al.* [65] find that the launching probabilities of mobile apps follow the Zipf's law, and inter-running and running times of apps conform to log-normal distributions. Based on these findings, they develop a novel context-aware application scheduling framework that adaptively unloads and preloads background applications in a timely manner. Tan *et al.* [66] propose to predict mobile app usage patterns in different contexts. They conduct experiments on the Nokia MDC dataset which contains a small group of 38 users and their experiments show promising results. Yan *et al.* [67] aim to predict app usage so that they can preload those apps to remedy the launch delay.

Low-cost prediction. App usage prediction algorithms are generally running on mobile clients in a continuous manner. The development of low-cost algorithms is thus important to have the wide acceptance of mobile users. To save the energy consumption for the task of predicting app usage, Liao *et al.* [68] propose a temporal-based app predictor to dynamically predict the apps that are most likely to be used.

Using the temporal information of the usage of collected apps, the system discovers three useful features: the global usage count of each app, the usage count regarding a specific temporal bucket, and the usage periods of each app. In a similar study [69], a framework to predict mobile apps that are most likely to be used regarding the current device status of a smartphone is proposed. The personalized feature selection algorithm is proposed to reduce the log size and the prediction time. Parate *et al.* [70] design an app prediction algorithm that requires no prior training and predicts which app will be used next and when. Zou *et al.* [71] propose some lightweight Bayesian methods to predict the next app based on the app usage history.

Intelligent app rendering. Given the large number of installed apps and the limited screen size of mobile devices, it is often tedious for users to search for the app they want to use. Intelligent app rendering is thus leveraged for optimizing device operation. Shin *et al.* [72] use smartphone traces to build models for predicting the apps used based on the current context. They collect sensory data to perform a comprehensive analysis of the context related to mobile app usage. Based on the prediction result, it will present icons for the most probable apps on the main screen of the smartphone and give highlights. Baeza-Yates *et al.* [73] study how to improve the usage experience of mobile apps based on the Parallel Tree Augmented Naive Bayesian Network model. The prediction technique is based on a set of features representing the real-time spatiotemporal contexts sensed by the apps.

Context-based prediction. Most works focus on predicting users' upcoming requirements at current contexts (location, time, etc.). However, they do not consider the long-term temporal co-evolution among apps and contexts. To address this, Wang *et al.* [74] propose a contextual collaborative forecasting (CCF) model through tensor decomposition for app usage forecasting. The model integrates contextual collaborative filtering with time series analysis, and simultaneously captures various components of temporal patterns, including trend, seasonality and stationarity. Huang *et al.* [75] study the problem of pre-loading the right apps in memory for faster execution by exploiting contextual information such as time, location and the user profile. They then pop the desired app on the mobile's home screen at the right moment. Xu *et al.* [76] present a prediction framework for smartphone app usage that incorporates three factors influencing user app behavior, including spatiotemporal contexts, community behaviors, and user preferences.

F. USER PROFILING AND APP RECOMMENDATION

Users of different attributes would like to choose different set of apps to install and have distinct usage patterns, which can be leveraged for user profiling. A summary of user profiling and app recommendation is given in Table 6.

App list-based profiling. The apps installed on one's smartphone can convey lots of information regarding his/her personal attributes, such as gender, occupation, income, and preferences. Achara *et al.* [77] collect the open-accessible lists of running apps on mobile devices over more than seven months. Their study shows that any four apps installed by a user are enough (more than 95% times) for the re-identification of the user in the dataset. Zhao *et al.* [78] develop an attribute-specific representation to describe user characteristics and then model the relationship between user attributes and app lists. Seneviratne *et al.* [79] investigate how user traits such as religion, relationship status, spoken languages, and age can be inferred by observing only a single snapshot of installed apps. They can predict user's gender and demographic attributes with the accuracy of around 70% [80]. Xu *et al.* [81] explore the influence of personality traits on mobile app adoption based on the installed apps. Similarly, Malmi and Weber [82] study the predictability of user demographics (e.g., age, race, and income) based on the list of a user's apps.

App usage-based profiling. Besides the differences of installed apps, user attributes are also closely associated with app usage patterns. For example, Qin *et al.* [83] investigate the correlations between user demographic information and their requests of network resources. Park *et al.* [84] study how to infer a user's intent based on the user's status and retrieve relevant apps that may satisfy the user's needs. Murnane *et al.* [85] connect patterns of mobile app usage with biological factors. They find that app usage patterns vary for individuals with different body clock types. Welke *et al.* [86] model a user as the set of apps that he/she uses. They further classify users of different attributes based on their app usage patterns.

By learning human profiles, we can recommend suitable apps to users based on their interests. Cao and Lin [87] make a comparison of app recommendation systems and conventional recommendation systems (e.g. recommending movies). They state that movies and apps are by nature very different products for users to consume. A movie is mostly got consumed on a one-time basis in the cinema. However, an app provides certain function to mobile users on a day-to-day basis. The functionality and interactivity are far more important for one to choose an app than for a movie. Therefore, movie recommendation can be supported by simple ways based on a few explicit features such as genre and producer, while for apps, user experiences can be more important than its description. Therefore, there are many other factors (e.g., contexts, privacy concerns) that should be considered.

Context-aware recommendation. In [88], a context-aware recommender system for mobile apps is proposed, which utilizes a binary tensor to represent the personal usage history. Liang *et al.* [89] also present an approach for context-aware app recommendation with tensor analysis, which integrates user's preferences, app category information and multi-view features to facilitate the performance of app rating prediction. Zhu *et al.* [90] propose an allocation-based probabilistic mechanism that considers multiple user-app factors for app recommendation.

Privacy concerns. Understanding user preferences may also leak user privacy information. Therefore, it is also

TABLE 6. User profiling and app recommendation.

Work	Problem and method	Dataset
Achara *et al*. [77]	App-list based user profiling Method: Markov Chain Monte Carlo method	54,893 users, 7 months, app usage log, Open source dataset
Zhao *et al*. [78]	App-list based user profiling Method: Markov Chain Monte Carlo method	100,281 users, 0.2 million records, app installation log, Open source dataset
Seneviratne *et al*. [79]	App-list based user profiling Method: Support vector machine	9,000 users, app installation log, Open source dataset
Seneviratne *et al*. [80]	App-list based user profiling Method: Supervised learning	200 users, app installation log, By crowdsourcing
Xu *et al*. [81]	App-list based user profiling Method: Supervised learning, adoption theory	2,043 users, app installation log, By crowdsourcing
Malmi *et al*. [82]	App-list based user profiling Method: Dimensionality Reduction, Singular Value Decomposition	3,760 users, 7 months, app usage log, Open source dataset
Qin *et al*. [83]	Network resource requests and user profiling Method: Supervised learning, Optimization	32,660 users, 4 months, app usage log, By crowdsourcing
Park *et al*. [84]	App recommendation Method: Topic model, Information retrieval	1,115,948 users, 1,609,894 tweets, By crowdsourcing
Murnane *et al*. [85]	App usage and biological profiling Method: Temporal analysis	20 users, 40 days, app usage log, By crowdsourcing
Welke *et al*. [86]	App usage pattern and user profiling Method: Statistic analysis	46,726 users, app usage log, By crowdsourcing
Karatzoglou *et al*. [88]	Context-aware app recommendation Method: Collaborative filtering, Tensor factorization	3,260 users, 3.7 million records, app usage log, Open source dataset
Liang *et al*. [89]	Context-aware app recommendation Method: Tensor Analysis	11,175 users, 8,103 apps, app statistics data, Google Play and Apple store
Zhu et al. [90]	Context-aware app recommendation Method: Topic model, Supervised learning	13,969 users, 485 days, app usage log, By crowdsourcing
Liu *et al*. [91]	Privacy-preserving app recommendation Method: Latent factorization model	16,344 users, 6,157 apps, app statistics data, Google Play
Zhu *et al*. [92]	Privacy-preserving app recommendation Method: modern portfolio theory, optimization	173 users, 170,753 apps, app statistics data, Source: Google Play
Lin *et al*. [93]	Cold start problem in app recommendation Method: Collaborative filtering, Topic model	10,133 users, 7,116 apps, app statistics data, Source: Apple store

important to protect user privacy in app recommendation. For example, Liu *et al*. [91] propose to incorporate both app functionality and user privacy preferences as features for app recommendation. Zhu *et al*. [92] propose a flexible approach based on modern portfolio theory for recommending apps by striking a balance between the apps' popularity and the users' security concerns.

The cold-start problem. For newly released apps, there does not exist any user ratings for them and can lead to the cold-start problem. Traditional recommender systems (i.e., collaborative filtering) cannot well address this problem. It is believed that collecting information from complementary and associated data sources can be an effective way to address this issue. For example, Lin *et al*. [93] describe a method that collects nascent information from Twitter followers to provide relevant recommendation in cold-start situations. They apply the latent Dirichlet allocation (LDA) method to generate latent groups based on the posted information in Twitter.

IV. OPEN ISSUES AND FUTURE TRENDS

By exploring existing studies on mobile apps, we find that there are open issues that are not well addressed. They have high potential to be future trends in mobile app data mining.

A. COMPETITIVE INTELLIGENCE AND APP EVOLUTION

Existing studies mainly provide suggestions to app evolution based on the history data of apps. However, as a fierce competitive market, the behaviors and performance of similar apps also impact the development of apps. As reported in [94], app markets generally organize apps in a hierarchical taxonomy, and apps with similar functionalities are usually placed together. It facilitates users to compare similar apps in app markets and thus the competition is easy to find among them. Figure 1 gives the statistical information about two competitive apps on bike sharing, Mobike and ofo [1]. It is evident that there exist significant correlations among the download trends and activities among them. Thereby, an interesting research direction is that we should study app evolution based on competitive learning, that is, learning and transfer knowledge from similar "buddies" [95] or a competitive group to improve the evolution of each app. There are several interesting topics about competitive learning, such as download forecasting, requirement learning, business strategy learning, and so on. We make a discussion of them below.

Download forecasting. Understanding competition status can help predict future app downloads. For example,

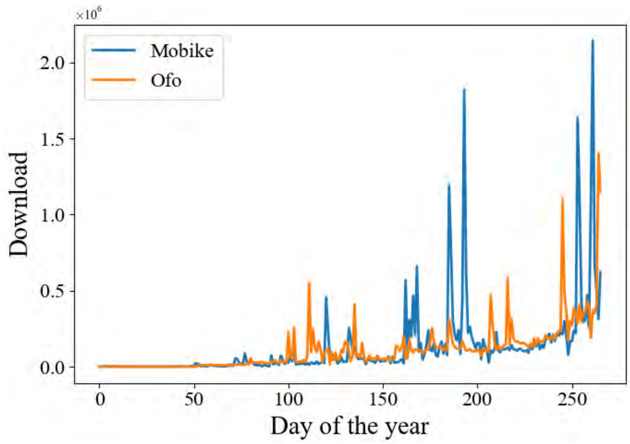


FIGURE 1. The daily downloads of Mobike and Ofo [1].

Wang *et al.* [63] propose the evolutionary hierarchical competition model (EHCM), which considers the time-evolving multi-level competition among apps for download forecasting. Liu *et al.* [96] present a framework that automatically labels apps with a richer and more detailed categorization and use the labeled apps to study the app market competition.

Requirement learning. App stores usually use categories to group similar apps. The combination of official information about similar apps (such as the list of features, the release history, and prices) with reviews and rating comments can help identify and prioritize new app requirements. It also helps determine the new app's optimal set of features by mining user experience from similar apps. As reported in [97], examining similar apps in a category or across categories is a great way to define potential features.

Business strategy learning. Monitoring the behaviors and activities of competitors help app developers to make decisions on strategies, advertising, promotion, and planning offline activities. For example, Mobike often plans offline activities to increase the attraction of its products. Accordingly, Ofo can keep track of the activities of Mobike and design its marketing plans.

App recommendation with sparse data. Studying apps in groups can help deal with the cold-start problem in app recommendation. For instance, Liu *et al.* [94] develop a structural user choice model (SUCM) to learn fine-grained user preferences by exploiting the hierarchical taxonomy of apps as well as the competitive relationships among apps. Zhu *et al.* [98] illustrate how to extract personal context-aware preferences from context-rich device logs and how to use the identified preferences for personalized, context-aware recommendation. To address the data scarcity problem in individual context logs, they propose to first learn common context-aware preferences from the context logs of many users, and the preference of each user can be represented as a distribution of these common context-aware preferences.

B. PERSONALIZED AND CONTEXT-AWARE APP ADAPTATION

Large-scale app usage data has been leveraged for understanding app usage patterns. In existing studies, the learned app usage patterns can be applied for fast launching of next apps to be used [65], [71], [73] and user profiling [79], [80], [83]. However, there are still many things that can be exploited.

Context-aware app adaptation. In general, app usage pattern mining is categorized to user behavior understanding [16], [20]. By mining user behaviors, we can develop context-aware mobile apps that can intelligently adapt to user contexts. For mobile apps, user contexts refer to location, time, behaviors, environment dynamics, emotion, and so on. There have been previously numerous studies on mobile user context learning [99], [100]. This, however, should be combined with app usage patterns to support more personalized app adaptation.

Recommendation with usage mining. App recommendation cannot only rely on what apps have been installed but how users use them (e.g., the frequency of openings and daily usage patterns), considering that certain apps are rarely used since installed. In other words, precision app marketing and recommendation can be enabled with the mining of app usage data. Yan and Chen [101] develop a mobile app recommender system, which is based on users' app usage records to build a preference matrix instead of using explicit user ratings. To solve the scarcity problem of app usage records, Shi and Ali [102] leverage a combination of content and usage data and extend the collaborative filtering model to recommend apps.

Above all, to better fit user requirements in real-world scenarios, the future app management system should be enhanced with context learning and personalized adaptation modules.

C. FINE-GRAINED APP POPULARITY PREDICTION

Besides the factors studied by existing works on app popularity prediction, there are other important factors that may impact app popularity, such as app updates and app rank charts.

Release planning. App developers have an urgent requirement to optimize their release strategy to maximize user adoption of their apps. Shen *et al.* [103] introduce an approach that can assist developers to select the proper release opportunity based on the purpose of the update and current condition of the app. They reveal important characteristics of update intervals and how these factors can influence updates. They also find that it is important for app developers to understand both the impact of their own and their competitors' software releases. Martin *et al.* [104] mine 38,858 apps from Google Play over 12 months. They find that 33% of these releases cause a statistically significant change in user ratings.

Top-ranked app charts. Several scholars have shown that app rank charts promote downloads of apps [105], [106].

Garg and Telang [105] present an innovative method that uses public data to infer the rank-demand relationship for the paid apps on Apple App Store. To climb the top-ranked charts and improve the visibility of apps, Comino *et al.* [107] assume that the release of an update stimulates the "buzz" surrounding the app in social networks, and thus can enhance the potential influence and downloads of the app. They find that publishers might have incentives to release new versions of their app even if they offer little improvement to the technical quality of it.

More impact factors. The download users at release days consist of new installers and existing updaters, while at other moments are mainly based on new installers. Therefore, the prediction of app downloading should consider this factor and make predictions at the inter-release phase level, or make predictions by taking consideration of the release day factor. Furthermore, we identify that different types of apps (e.g., education, game, news, travelling) have different fall and rise patterns. Therefore, app type should also be considered in popularity modeling.

D. HETEROGENEOUS CROWDSOURCED DATA MINING

Existing approaches are based on statistical analysis methods and matured machine learning models. In recent years, Artificial Intelligence (AI) has rapid development with the rise of deep learning techniques. More advanced data mining and machine learning techniques should thus be introduced to improve app marketing.

Future app systems should leverage crowdsourced information about apps, devices, and users to help developers improve users' experience with the delivered mobile apps [21]. A wide range of data sources related to mobile apps should be considered, including app usage data, app store data, app development communities (e.g., Stack Overflow,)[9] and the relevant posts in social networks (e.g., Twitter). As stated in [45], by systematically observing user communities, forums, social media channels, and review platforms, a range of information that supports requirement decisions is available. For example, Nayebi *et al.* [108] study how Twitter can provide complementary information to support mobile app development. By analyzing a total of 70 apps over a period of six weeks, they show that 22.4% more feature requests and 12.9% more bug reports could be found on Twitter. Above all, the integration of complementary information from different data sources can draw a comprehensive picture about mobile app market and development.

Embracing economic and AI models. As a new type of business, the empirical economic models can be applied for app data analysis. Econometrics is an often-used approach in business intelligence that supports sale or demand prediction [109]. Ghose and Han [106] build a structural econometric model to estimate consumer preferences toward different mobile app characteristics. They find that app demand increases with the in-app purchase option wherein a user can complete transactions within the app. On the contrary, app demand decreases with the in-app advertisement option where consumers are shown ads while they are engaging with the app. Furthermore, novel AI and deep learning-based techniques, such as Denoising Auto-Encoders [110], Neural Collaborative Filtering [111], Neural Tensor Factorization [112], Generative adversarial networks [113] have been successfully used for object categorization/classification, time-series prediction, and recommendation. As in different research communities, new AI techniques are yet to be studied in addressing the issues in app development and marketing. With large-scale, heterogeneous crowdsourced app data, deep learning techniques can be applied to attain better performance in addressing the challenges in mobile app systems.

Cross-space data mining. One of the major issues regarding app mining is cross-space data mining [7], which presents the association analysis regarding online/offline behaviors, the investigation of the complementary nature of online/offline features, as well as the integration of them for enhanced user understanding and app evolution. As stated by Gomez *et al.* [21], the "wisdom of the crowds" can be combined, one augmenting the other, the sum being more powerful than each one in isolation. Cao *et al.* [114] tackle the problem of cross-platform app recommendation, aiming at leveraging users' and apps' data on multiple platforms to enhance the recommendation accuracy. Pan *et al.* [115] propose a model to predict the most likely mobile application that a user will install, where a composite network computed from different networks sensed by phones is leveraged.

V. CROWDAPP: TOWARD AN EVOLVABLE APP ECOSYSTEM

The surge of mobile apps, app stores, and crowd-app interactions is leading to a new mobile app ecosystem that constitutes stakeholders such as app developers, marketplace operators, end-users, and advertisers. Having presented the major features and open issues regarding the usage of heterogeneous crowdsourced data in mobile app systems, we propose the CrowdApp framework, which is a crowdsourced data-driven app ecosystem that can better support the development and marketing of mobile apps. The architecture of it is given in Fig. 2.

One of the biggest features of CrowdApp is that it applies the mobile edge computing concept [116]. There are several benefits of mobile edge computing for app systems. First, processing data on the mobile device provides better privacy guarantees to users, and reduces the dependency on cloud connectivity. Second, it reduces data transmission cost considering that only extracted semantic information while not large-scale original data is transmitted. As an important step in this direction, Srinivasan *et al.* [117] develop a novel general-purpose service called MobileMiner that runs on the phone and discovers frequent co-occurrence patterns indicating which context events frequently occur together.

[9] https://stackoverflow.com/

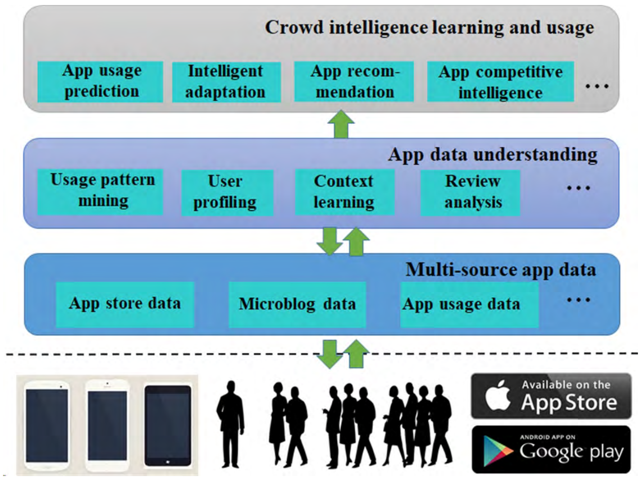


FIGURE 2. **The CrowdApp Ecosystem.**

CrowdApp is a layered framework and in the following we make a description of the major layers and components of it.

(1) *Multi-source app data*. Based on the existing studies, we observe that heterogeneous crowdsourced app data (e.g., app store data, microblog data, app usage data, smartphone sensing data, developer community data, user profiling data) is complementary and correlated with each other. CrowdApp collects and integrates all these app-related data sources and leverages the aggregated power of them for app development and marketing.

(2) *App data understanding*. It is about data analysis and semantic extraction based on raw collected data. As discussed, app data processing can be performed at either mobile clients or the cloud server. Due to the privacy and data transmission cost concerns, the first three components described below can be placed at mobile clients.

- *Usage pattern mining*. It supports the understanding of the usage patterns and habits of mobile device users, which can be used to intelligent adaptation of mobile apps.
- *User profiling*. It extracts users' demographic attributes (e.g., age, affiliation, interests, etc.) and preferences from app installation and usage data.
- *Context learning*. The mobile app ecosystem should equip the capability on user context learning, which is helpful for providing more personalized services for app users and meet their real-time requests.
- *Review analysis*. The reviews contain a lot of user feedback, which plays an important role in the development of mobile apps. It refers to important review classification and review selection/ranking.
- *Heterogeneous crowdsourced data mining*. It involves the new techniques on multi-modal crowdsourced data mining, such as the usage of novel economic and AI models, and cross-space data mining.

(3) *Crowd intelligence learning and usage*. The ecosystem should support the building of unified crowd intelligence model, including developer intelligence, user intelligence, and app intelligence. The main research challenge is how to explore the complementary nature of various types of crowd intelligence and fuse them for intelligent app marketing and development. References [6], [118] report the various ways in which developers can use crowds throughout the app development lifecycle. Following is a summary of the main components of this layer.

- *App requirement/feedback mining*. User feedback plays a paramount role in the development and maintenance of mobile applications. Timely and constructive feedback from users becomes extremely crucial for developers to fix bugs, implement new features, and improve user experience agilely. It refers to review summarization, requirement extraction, and feature mining.
- *App usage prediction*. Predicting app usage is a crucial prerequisite for fast app launching, intelligent user experience, and power management of smartphones.
- *Intelligent adaptation*. Next generation app stores should more focus on local usage and user context, which can better understand user context and adapt to diversified user behaviors and usage scenarios.
- *App recommendation*. App recommendation should rely on not only what apps have been installed, but also how users use them, which can better meet user requests and preferences.
- *App competitive intelligence*. Apps with similar functionalities are competing. Competitive learning techniques should be studied for precise app download forecasting, strategy planning, requirement/feature learning, and so on.
- *App popularity prediction*. There are many important factors that affect the popularity of apps, including release planning, app rank charts, reviews, ratings, categories, etc. To predict the future popularity of mobile apps, we should incorporate these factors in popularity modeling.

VI. CONCLUSION

This paper has reviewed the current state and future directions of mobile app user understanding and marketing, which is based on a crowdsourced data driven perspective. First, we investigate the opportunities of crowdsourcing for mobile apps, i.e., the so-called CrowdApp. We then present the key research challenges and techniques of this research field, including usage pattern understanding, review analysis and processing, requirement and feedback mining, popularity forecasting, app usage prediction, user profiling and recommendation, and so on. We further discuss the open issues and future directions, such as competitive intelligence and app evolution, personalized and context-aware app evolution, fine-grained app popularity prediction, and heterogeneous crowdsourced data mining. The CrowdApp framework is finally presented to better support the development and marketing of apps under an evolvable app ecosystem vision.

REFERENCES

[1] Y. Ouyang, B. Guo, X. Lu, Q. Han, T. Guo, and Z. Yu, "CompetitiveBike: Competitive prediction of bike-sharing apps using heterogeneous crowdsourced data," 2018, *arXiv:1802.05568*. [Online]. Available: https://arxiv.org/abs/1802.05568

[2] Y. Liu, B. Guo, H. Du, Z. Yu, D. Zhang, and C. Chen, "Poster: FooDNet: Optimized on demand take-out food delivery using spatial crowdsourcing," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 564–566.

[3] C. Anderson, "The long tail," *Wired Mag.*, vol. 12, no. 10, pp. 170–177, 2004.

[4] G. Lee and T. S. Raghu, "Determinants of mobile apps' success: Evidence from the app store market," *J. Manage. Inf. Syst.*, vol. 31, no. 2, pp. 133–170, 2014.

[5] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE Trans. Softw. Eng.*, vol. 43, no. 9, pp. 817–847, Sep. 2017.

[6] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *J. Syst. Softw.*, vol. 126, pp. 57–84, Apr. 2017.

[7] B. Guo, C. Chen, D. Zhang, A. Chin, and Z. Yu, "Mobile crowd sensing and computing: When participatory sensing meets participatory social media," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 131–137, Feb. 2016.

[8] B. Guo, Z. Yu, and X. Zhou, "A data-centric framework for cyber-physical-social systems," *It Prof.*, vol. 17, no. 6, pp. 4–7, Nov./Dec. 2015.

[9] W. Maalej and D. Pagano, "On the socialness of software," in *Proc. IEEE 9th Int. Conf. Dependable, Autonomic Secure Comput. (DASC)*, Dec. 2011, pp. 864–871.

[10] E. Guzman, R. Alkadhi, and N. Sey, "An exploratory study of twitter messages about software applications," *Requirements Eng.*, vol. 22, no. 3, pp. 387–412, 2017.

[11] H. Li, X. Lu, X. Liu, T. Xie, K. Bian, F. X. Lin, Q. Mei, and F. Feng, "Characterizing smartphone usage patterns from millions of android users," in *Proc. Internet Meas. Conf.*, 2015, pp. 459–472.

[12] X. Liu, H. Li, X. Lu, T. Xie, Q. Mei, F. Feng, and H. Mei, "Understanding diverse usage patterns from large-scale appstore-service profiles," *IEEE Trans. Softw. Eng.*, vol. 44, no. 4, pp. 384–411, Apr. 2018.

[13] H. Li and X. Lu, "Mining device-specic apps usage patterns from large-scale android users," 2017, *arXiv:1707.09252*. [Online]. Available: https://arxiv.org/abs/1707.09252

[14] S. L. Jones, D. Ferreira, S. Hosio, J. Goncalves, and V. Kostakos, "Revisitation analysis of smartphone app use," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 1197–1208.

[15] S. Zhao, J. Ramos, J. Tao, Z. Jiang, S. Li, Z. Wu, G. Pan, and A. K. Dey, "Discovering different kinds of smartphone users through their application usage behaviors," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 498–509.

[16] Y.-N. Chen, M. Sun, A. I. Rudnicky, and A. Gershman, "Leveraging behavioral patterns of mobile applications for personalized spoken language understanding," in *Proc. ACM Int. Conf. Multimodal Interact.*, 2015, pp. 83–86.

[17] X. Liu, W. Ai, H. Li, J. Tang, G. Huang, F. Feng, and Q. Mei, "Deriving user preferences of mobile apps from their management activities," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, p. 39, 2017.

[18] H. Li, W. Ai, X. Liu, J. Tang, G. Huang, F. Feng, and Q. Mei, "Voting with their feet: Inferring user preferences from app management activities," in *Proc. 25th Int. Conf. World Wide Web. Int. World Wide Web Conf. Steering Committee*, 2016, pp. 1351–1362.

[19] D. Hintze, P. Hintze, R. D. Findling, and R. Mayrhofer, "A large-scale, long-term analysis of mobile device usage characteristics," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, p. 13, 2017.

[20] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman, "Identifying diverse usage behaviors of smartphone apps," in *Proc. ACM SIGCOMM Conf. Internet Meas. Conf.*, 2011, pp. 329–344.

[21] M. Gomez, B. Adams, W. Maalej, M. Monperrus, and R. Rouvoy, "App store 2.0: From crowdsourced information to actionable feedback in mobile ecosystems," *IEEE Softw.*, vol. 34, no. 2, pp. 81–89, Mar./Apr. 2017.

[22] H. Khalid, E. Shihab, M. Nagappan, and A. E. Hassan, "What do mobile App users complain about?" *IEEE Softw.*, vol. 32, no. 3, pp. 70–77, May/Jun. 2014.

[23] S. Mujahid, G. Sierra, R. Abdalkareem, E. Shihab, and W. Shang, "Examining user complaints of wearable apps: A case study on android wear," in *Proc. IEEE/ACM 4th Int. Conf. Mobile Softw. Eng. Syst. (MOBILESoft)*, May 2017, pp. 96–99.

[24] P. M. Vu, H. V. Pham, and T. T. Nguyen, "Mining user opinions in mobile app reviews: A keyword-based approach (T)," in *Proc. 30th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2015, pp. 749–759.

[25] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *Proc. 21st IEEE Int. Requirements Eng. Conf. (RE)*, 2013, pp. 125–134.

[26] S. Keertipati, B. T. R. Savarimuthu, and S. A. Licorish, "Approaches for prioritizing feature improvements extracted from app reviews," in *Proc. 20th Int. Conf. Eval. Assessment Softw. Eng.*, 2016, p. 33.

[27] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, "Ar-miner: Mining informative reviews for developers from mobile app marketplace," in *Proc. 36th Int. Conf. Softw. Eng.*, 2014, pp. 767–778.

[28] D. H. Park, M. Liu, C. Zhai, and H. Wang, "Leveraging user reviews to improve accuracy for mobile app retrieval," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 533–542.

[29] F. Palomba, M. Linares-Vasquez, G. Bavota, R. Oliveto, M. Di Penta, D. Poshyvanyk, and A. De Lucia, "User reviews matter! tracking crowdsourced reviews to support evolution of successful apps," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2015, pp. 291–300.

[30] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, and M. Di Penta, "Release planning of mobile apps based on user reviews," in *Proc. 38th Int. Conf. Softw. Eng.*, 2016, pp. 14–24.

[31] Y. Li, B. Jia, Y. Guo, and X. Chen, "Mining user reviews for mobile app comparisons," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, p. 75, 2017.

[32] E. Guzman, M. EL-Haliby, and B. Bruegge, "Ensemble methods for app review classification: An approach for software evolution (N)," in *Proc. 30th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2015, pp. 771–776.

[33] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? Classifying user reviews for software maintenance and evolution," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep./Oct. 2015, pp. 281–290.

[34] S. McIlroy, N. Ali, H. Khalid, and A. E. Hassan, "Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews," *Empirical Softw. Eng.*, vol. 21, no. 3, pp. 1067–1106, 2016.

[35] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 116–125.

[36] M. Hosseini, E. C. Groen, A. Shahri, and R. Ali, "CRAFT: A crowd-annotated feedback technique," in *Proc. IEEE 25th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2017, pp. 170–175.

[37] P. K. Murukannaiah, N. Ajmeri, and M. P. Singh, "Acquiring creative requirements from the crowd: Understanding the influences of personality and creative potential in crowd RE," in *Proc. IEEE 24th Int. Requirements Eng. Conf. (RE)*, Sep. 2016, pp. 176–185.

[38] A. Ciurumelea, A. Schaufelbühl, S. Panichella, and H. C. Gall, "Analyzing reviews and code of mobile apps for better release planning," in *Proc. IEEE 24th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Feb. 2017, pp. 91–102.

[39] A. Di Sorbo, S. Panichella, C. V. Alexandru, J. Shimagaki, C. A. Visaggio, G. Canfora, and H. C. Gall, "What would users change in my app? summarizing app reviews for recommending software changes," in *Proc. 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, 2016, pp. 499–510.

[40] X. Gu and S. Kim, "'What parts of your apps are loved by users?' (T)," in *Proc. 30th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2015, pp. 760–770.

[41] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, "Why people hate your app: Making sense of user feedback in a mobile app store," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1276–1284.

[42] L. V. G. Carreño and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," in *Proc. Int. Conf. Softw. Eng.*, May 2013, pp. 582–591.

[43] S. McIlroy, W. Shang, N. Ali, and A. E. Hassan, "Is it worth responding to reviews? Studying the top free apps in Google play," *IEEE Softw.*, vol. 34, no. 3, pp. 64–71, May/Jun. 2017.

[44] S. Mcilroy, W. Shang, N. Ali, and A. E. Hassan, "User reviews of top mobile apps in apple and Google app stores," *Commun. ACM*, vol. 60, no. 11, pp. 62–67, 2017.

[45] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, "Toward data-driven requirements engineering," *IEEE Softw.*, vol. 33, no. 1, pp. 48–54, Nov. 2016.
[46] F. Palomba, P. Salza, A. Ciurumelea, S. Panichella, H. Gall, F. Ferrucci, and A. De Lucia, "Recommending and localizing change requests for mobile apps based on user reviews," in *Proc. 39th Int. Conf. Softw. Eng.*, 2017, pp. 106–117.
[47] J. Jin, P. Ji, and C. K. Kwong, "What makes consumers unsatisfied with your products: Review analysis at a fine-grained level," *Eng. Appl. Artif. Intell.*, vol. 47, pp. 38–48, Jan. 2016.
[48] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in *Proc. 10th Working Conf. Mining Softw. Repositories*, 2013, pp. 41–44.
[49] A. Sharma, Y. Tian, and D. Lo, "NIRMAL: Automatic identification of software relevant tweets leveraging language model," in *Proc. IEEE 22nd Int. Conf. Softw. Anal., Reeng. (SANER)*, Mar. 2015, pp. 449–458.
[50] E. Guzman and W. Maalej, "How do users like this feature? A fine grained sentiment analysis of app reviews," in *Proc. IEEE 22nd Int. Requirements Eng. Conf. (RE)*, Aug. 2014, pp. 153–162.
[51] P. M. Vu, H. V. Pham, T. T. Nguyen, and T. T. Nguyen, "Phrase-based extraction of user opinions in mobile app reviews," in *Proc. 31st IEEE/ACM Int. Conf. Automated Softw. Eng.*, Sep. 2016, pp. 726–731.
[52] R. Bandari, S. Asur, and B. A. Huberman, "The pulse of news in social media: Forecasting popularity," in *Proc. 6th Int. AAAI Conf. Weblogs Social Media*, May 2012, pp. 26–33.
[53] X. Wang, B. Fang, and H. Zhang, "Predicting the popularity of news based on competitive matrix," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2017, pp. 151–155.
[54] M. Mazloom, R. Rietveld, S. Rudinac, M. Worring, and W. Van Dolen, "Multimodal popularity prediction of brand-related social media posts," in *Proc. ACM Multimedia Conf.*, 2016, pp. 197–201.
[55] B. Wu, T. Mei, W.-H. Cheng, and Y. Zhang, "Unfolding temporal dynamics: Predicting social media popularity using multi-scale temporal decomposition," in *Proc. Assoc. Adv. Artif. Intell.*, 2016, pp. 272–278.
[56] T. Trzciński and P. Rokita, "Predicting popularity of online videos using support vector regression," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2561–2570, Nov. 2017.
[57] F. Figueiredo, "On the prediction of popularity of trends and hits for user generated videos," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 741–746.
[58] T. Petsas, A. Papadogiannakis, M. Polychronakis, E. P. Markatos, and T. Karagiannis, "Rise of the planet of the apps: A systematic study of the mobile app ecosystem," in *Proc. Conf. Internet Meas. Conf.*, 2013, pp. 277–290.
[59] A. Finkelstein, M. Harman, Y. Jia, W. Martin, F. Sarro, and Y. Zhang, "Investigating the relationship between price, rating, and popularity in the blackberry world app store," *Inf. Softw. Technol.*, vol. 87, pp. 119–139, Jul. 2017.
[60] T. Wang, D. Wu, J. Zhang, M. Chen, and Y. Zhou, "Measuring and analyzing third-party mobile game app stores in China," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 793–805, Dec. 2016.
[61] Y. Tian, M. Nagappan, D. Lo, and A. E. Hassan, "What are the characteristics of high-rated apps? A case study on free android applications," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep./Oct. 2015, pp. 301–310.
[62] H. Zhu, C. Liu, Y. Ge, H. Xiong, and E. Chen, "Popularity modeling for mobile apps: A sequential approach," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1303–1314, Jul. 2015.
[63] Y. Wang, N. J. Yuan, Y. Sun, C. Qin, and X. Xie, "App download forecasting: An evolutionary hierarchical competition approach," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2978–2984.
[64] X. Lu, Z. Chen, X. Liu, H. Li, T. Xie, and Q. Mei, "PRADO: Predicting app adoption by learning the correlation between developer-controllable properties and user behaviors," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, p. 79, 2017.
[65] J. Lee, K. Lee, E. Jeong, J. Jo, and N. B. Shro, "Context-aware application scheduling in mobile systems: What will users do and not do next?" in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 1235–1246.
[66] C. Tan, Q. Liu, E. Chen, and H. Xiong, "Prediction for mobile application usage patterns," in *Proc. Nokia MDC Workshop*, vol. 12, 2012, pp. 1–4.
[67] T. Yan, D. Chu, D. Ganesan, A. Kansal, and J. Liu, "Fast app launching for mobile devices using predictive user context," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 113–126.
[68] Z.-X. Liao, Y.-C. Pan, W.-C. Peng, and P.-R. Lei, "On mining mobile apps usage behavior for predicting apps usage in smartphones," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 609–618.
[69] Z.-X. Liao, S.-C. Li, W.-C. Peng, S. Y. Philip, and T.-C. Liu, "On the feature discovery for app usage prediction in smartphones," in *Proc. IEEE 13th Int. Conf. Data Mining (ICDM)*, Dec. 2013, pp. 1127–1132.
[70] A. Parate, M. Bohmer, D. Chu, D. Ganesan, and B. M. Marlin, "Practical prediction and prefetch for faster access to applications on mobile phones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 275–284.
[71] X. Zou, W. Zhang, S. Li, and G. Pan, "Prophet: What app you wish to use next," in *Proc. ACM Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 167–170.
[72] C. Shin, J.-H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 173–182.
[73] R. Baeza-Yates, D. Jiang, F. Silvestri, and B. Harrison, "Predicting the next app that you are going to use," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 285–294.
[74] Y. Wang, N. J. Yuan, Y. Sun, F. Zhang, X. Xie, Q. Liu, and E. Chen, "A contextual collaborative approach for app usage forecasting," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 1247–1258.
[75] K. Huang, C. Zhang, X. Ma, and G. Chen, "Predicting mobile application usage using contextual information," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 1059–1065.
[76] Y. Xu, M. Lin, H. Lu, G. Cardone, N. Lane, Z. Chen, A. Campbell, and T. Choudhury, "Preference, context and communities: A multi-faceted approach to predicting smartphone app usage patterns," in *Proc. Int. Symp. Wearable Comput.*, 2013, pp. 69–76.
[77] J. P. Achara, G. Acs, and C. Castelluccia, "On the unicity of smartphone applications," in *Proc. 14th ACM Workshop Privacy Electron. Soc.*, 2015, pp. 27–36.
[78] S. Zhao, G. Pan, Y. Zhao, J. Tao, J. Chen, S. Li, and Z. Wu, "Mining user attributes using large-scale app lists of smartphones," *IEEE Syst. J.*, vol. 11, no. 1, pp. 315–323, Mar. 2017.
[79] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti, "Predicting user traits from a snapshot of apps installed on a smartphone," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 18, no. 2, pp. 1–8, 2014.
[80] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti, "Your installed apps reveal your gender and more!" *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 18, no. 3, pp. 55–61, 2015.
[81] R. Xu, R. M. Frey, E. Fleisch, and A. Ilic, "Understanding the impact of personality traits on mobile app adoption—Insights from a large-scale field study," *Comput. Hum. Behav.*, vol. 62, pp. 244–256, Sep. 2016.
[82] E. Malmi and I. Weber, "You are what apps you use: Demographic prediction based on user's apps," in *Proc. ICWSM*, 2016, pp. 635–638.
[83] Z. Qin, Y. Wang, H. Cheng, Y. Zhou, Z. Sheng, and V. Leung, "Demographic information prediction: A portrait of smartphone application users," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 3, pp. 432–444, Jul./Sep. 2016.
[84] D. H. Park, Y. Fang, M. Liu, and C. Zhai, "Mobile app retrieval for social media users via inference of implicit intent in social media text," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 959–968.
[85] E. L. Murnane, S. Abdullah, M. Matthews, M. Kay, J. A. Kientz, T. Choudhury, G. Gay, and D. Cosley, "Mobile manifestations of alertness: Connecting biological rhythms with patterns of smartphone app use," in *Proc. 18th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services*, 2016, pp. 465–477.
[86] P. Welke, I. Andone, K. Blaszkiewicz, and A. Markowetz, "Differentiating smartphone users by app usage," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 519–523.
[87] H. Cao and M. Lin, "Mining smartphone data for app usage prediction and recommendations: A survey," *Pervasive Mobile Comput.*, vol. 37, pp. 1–22, Jun. 2017.
[88] A. Karatzoglou, L. Baltrunas, K. Church, and M. Bohmer, "Climbing the app wall: Enabling mobile app discovery through context-aware recommendations," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 2527–2530.
[89] T. Liang, L. He, C.-T. Lu, L. Chen, P. S. Yu, and J. Wu, "A broad learning approach for context-aware mobile application recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 955–960.
[90] K. Zhu, L. Zhang, and A. Pattavina, "Learning geographical and mobility factors for mobile application recommendation," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 36–44, May/Jun. 2017.

[91] B. Liu, D. Kong, L. Cen, N. Z. Gong, H. Jin, and H. Xiong, ''Personalized mobile app recommendation: Reconciling app functionality and user privacy preference,'' in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 315–324.

[92] H. Zhu, H. Xiong, Y. Ge, and E. Chen, ''Mobile app recommendations with security and privacy awareness,'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 951–960.

[93] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, ''Addressing cold-start in app recommendation: Latent user models constructed from twitter followers,'' in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2013, pp. 283–292.

[94] B. Liu, Y. Wu, N. Z. Gong, J. Wu, H. Xiong, and M. Ester, ''Structural analysis of user choices for mobile app recommendation,'' *ACM Trans. Knowl. Discovery from Data*, vol. 11, no. 2, p. 17, 2016.

[95] B. Guo, J. Li, V. W. Zheng, Z. Wang, and Z. Yu, ''CityTransfer: Transferring inter-and intra-city knowledge for chain store site recommendation based on multi-source urban data,'' *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, p. 135, 2018.

[96] X. Liu, H. H. Song, M. Baldi, and P.-N. Tan, ''Macro-scale mobile app market analysis using customized hierarchical categorization,'' in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[97] F. Sarro, A. A. Al-Subaihin, M. Harman, Y. Jia, W. Martin, and Y. Zhang, ''Feature lifecycles as they spread, migrate, remain, and die in app stores,'' in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 76–85.

[98] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, ''Mining mobile user preferences for personalized context-aware recommendation,'' *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, p. 58, 2015.

[99] O. Yurur, C. H. Liu, Z. Sheng, V. C. M. Leung, W. Moreno, and K. K. Leung, ''Context-awareness for mobile sensing: A survey and future directions,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 68–93, 1st Quart., 2016.

[100] B. Guo, Z. Yu, D. Zhang, H. He, J. Tian, and X. Zhou, ''Toward a group-aware smartphone sensing system,'' *IEEE Pervasive Comput.*, vol. 13, no. 4, pp. 80–88, Oct. 2014.

[101] B. Yan and G. Chen, ''AppJoy: Personalized mobile application discovery,'' in *Proc. 9th Int. Conf. Mobile Syst., Appl. Services*, 2011, pp. 113–126.

[102] K. Shi and K. Ali, ''Getjar mobile application recommendations with very sparse datasets,'' in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 204–212.

[103] S. Shen, X. Lu, Z. Hu, and X. Liu, ''Towards release strategy optimization for apps in Google play,'' in *Proc. 9th Asia–Pacific Symp. Internetware*, 2017, p. 1.

[104] W. Martin, F. Sarro, and M. Harman, ''Causal impact analysis for app releases in Google play,'' in *Proc. 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, 2016, pp. 435–446.

[105] R. Garg and R. Telang, ''Inferring app demand from publicly available data,'' *Manage. Inf. Syst. Quart.*, vol. 37, no. 4, pp. 1253–1264, 2013.

[106] A. Ghose and S. P. Han, ''Estimating demand for mobile applications in the new economy,'' *Manage. Sci.*, vol. 60, no. 6, pp. 1470–1488, 2014.

[107] S. Comino, F. M. Manenti, and F. Mariuzzo, ''To upgrade or not to upgrade? The release of new versions to survive in the hypercompetitive app market,'' in *Proc. Int. Workshop App Market Anal.*, 2016, pp. 37–42.

[108] M. Nayebi, H. Cho, H. Farrahi, and G. Ruhe, ''App store mining is not enough,'' in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 152–154.

[109] A. Dimitrios and S. G. Hall, *Applied Econometrics*. New York, NY, USA: Macmillan, 2015.

[110] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, ''Collaborative denoising auto-encoders for top-n recommender systems,'' in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, 2016, pp. 153–162.

[111] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, ''Neural collaborative filtering,'' in *Proc. 26th Int. Conf. World Wide Web. Int. World Wide Web Conf. Steering Committee*, 2017, pp. 173–182.

[112] X. Wu, B. Shi, Y. Dong, C. Huang, and N. Chawla, ''Neural tensor factorization,'' 2018, *arXiv:1802.04416*. [Online]. Available: https://arxiv.org/abs/1802.04416

[113] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial nets,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[114] D. Cao, X. He, L. Nie, X. Wei, X. Hu, S. Wu, and T.-S. Chua, ''Cross-platform app recommendation by jointly modeling ratings and texts,'' *ACM Trans. Inf. Syst.*, vol. 35, no. 4, p. 37, 2017.

[115] W. Pan, N. Aharony, and A. Pentland, ''Composite social network for 1301 predicting mobile apps installation,'' in *Proc. AAAI*, 2011, pp. 821–827.

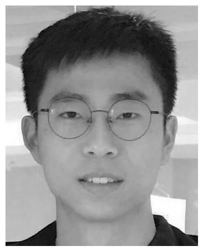
[116] X. Sun and N. Ansari, ''EdgeIoT: Mobile edge computing for the Internet of Things,'' *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[117] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu, and E. M. Tapia, ''MobileMiner: Mining your frequent patterns on your phone,'' in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 389–400.

[118] K.-J. Stol, T. D. LaToza, and C. Bird, ''Crowdsourcing for software engineering,'' *IEEE Softw.*, vol. 34, no. 2, pp. 30–36, Mar./Apr. 2017.

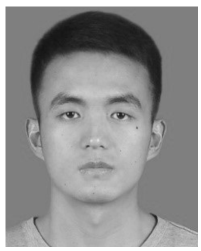
BIN GUO (SM'14) received the Ph.D. degree in computer science from Keio University, Japan, in 2009.

He is currently a Professor with Northwestern Polytechnical University, Xi'an, China. He was a Postdoctoral Researcher with Institut TELECOM SudParis, France. His research interests include ubiquitous computing, mobile crowd sensing, and human–computer interaction.



YI OUYANG received the B.E. degree in computer science and technology from the Xi'an University of Technology, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree in computer science with Northwestern Polytechnical University, Xi'an.

His research interests include data mining and mobile app analytics.



TONG GUO received the B.E. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2016, where he is currently pursuing the master's degree.

His research interests include data analysis and machine learning.

LONGBING CAO (SM'06) received the Ph.D. degree in pattern recognition and intelligent systems from the Chinese Academy of Science, and the Ph.D. degree in computing sciences from the University of Technology Sydney.

He is a Professor and the Founding Director of the UTS Advanced Analytics Institute. His current research interests include data science, artificial intelligence, behavior informatics, and their enterprise applications.

ZHIWEN YU (SM'13) received the Ph.D. degree of engineering in computer science and technology from Northwestern Polytechnical University, in 2005.

He was a Research Fellow with Kyoto University, Kyoto, Japan, from 2007 to 2009, and an Alexander Von Humboldt Fellow with Mannheim University, Mannheim, Germany, from 2009 to 2010. He is currently a Professor with Northwestern Polytechnical University. His current research interests include pervasive computing and human–computer interaction.