

# Characterizing Smartphone Usage Patterns from Millions of Android Users

Huoran Li, Xuan Lu, Xuanzhe Liu <sup>‡</sup>  
Peking University  
[{lihuoran, luxuan, xzl}@pku.edu.cn](mailto:{lihuoran, luxuan, xzl}@pku.edu.cn)

Tao Xie  
University of Illinois  
Urbana-Champaign  
[taoxie@illinois.edu](mailto:taoxie@illinois.edu)

Kaigui Bian  
Peking University  
[bkg@pku.edu.cn](mailto:bkg@pku.edu.cn)

Felix Xiaozhu Lin  
Purdue University  
[xzl@purdue.edu](mailto:xzl@purdue.edu)

Qiaozhu Mei  
University of Michigan  
[qmei@umich.edu](mailto:qmei@umich.edu)

Feng Feng  
Wandoujia Lab  
[jackfeng@wandoujia.com](mailto:jackfeng@wandoujia.com)

## ABSTRACT

The prevalence of smart devices has promoted the popularity of mobile applications (a.k.a. apps) in recent years. A number of interesting and important questions remain unanswered, such as why a user likes/dislikes an app, how an app becomes popular or eventually perishes, how a user selects apps to install and interacts with them, how frequently an app is used and how much traffic it generates, etc. This paper presents an empirical analysis of app usage behaviors collected from millions of users of Wandoujia, a leading Android app marketplace in China. The dataset covers two types of user behaviors of using over 0.2 million Android apps, including (1) app management activities (i.e., installation, updating, and uninstallation) of over 0.8 million unique users and (2) app network traffic from over 2 million unique users. We explore multiple aspects of such behavior data and present interesting patterns of app usage. The results provide many useful implications to the developers, users, and disseminators of mobile apps.

## Categories and Subject Descriptors

D.4.8 [Operating Systems]: Performance—*Measurement, Modeling and Prediction*; D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

## General Terms

Performance, Measurement

## Keywords

Android apps; app management; app popularity; app performance; app stores.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

IMC'15, October 28–30, 2015, Tokyo, Japan.

© 2015 ACM. ISBN 978-1-4503-3848-6/15/10 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2815675.2815686>.

## 1. INTRODUCTION

The release of iPhone in 2007 has opened a new era of mobile computing. Eight years later, smart devices such as iPhones and Android devices have played an indispensable role in our daily lives. The increasing popularity of mobile devices and apps has induced an evolution of software industry. The emergence of online marketplaces and platform APIs has been a game changer to developers, market intermediaries, and consumers, making them more and more willing to develop, disseminate, and use mobile apps rather than Web-based services [24].

The prevalence of mobile apps has also generated a huge volume of app usage data. Understanding user behaviors of using mobile apps is meaningful to all the stakeholders mentioned above. For example, marketplace operators can identify popular or problematic apps and provide effective app recommender systems; developers can understand why their apps are liked or disliked by the users and therefore improve the app design; end-users can have a better knowledge of how particular apps are consuming their time and bandwidth, and therefore optimize their decisions of selecting apps. Due to the lack of a sizable repository of such behavioral data, most of these important questions remain unanswered.

Previous studies have shown evidence about different users having different patterns of using mobile apps [19, 23, 7, 1]. Most of these studies were conducted using rather small behavior datasets, typically collected from a small number of student volunteers. The analyses also suffer from various selection biases introduced by college students.

Moreover, most behavioral signals were collected through a monitoring app voluntarily installed on the subjects' devices. Such a design is difficult to be applied by the variety of crowd due to the security and privacy concerns. As a result, very few studies could scale up to millions of users.

We present a comprehensive analysis of app usage behaviors collected from millions of Android users<sup>1</sup>. Our data comes from a leading Android app marketplace in China,

<sup>1</sup>All users involved in our dataset are anonymized by removing the identifiers. A sample of dataset has been released on DatCat. Researchers can search over DatCat by the title of this paper or directly access our server <http://45.56.95.4/description.html>. Contact the corresponding author: [xzl@pku.edu.cn](mailto:xzl@pku.edu.cn) for technical issues.

called **Wandoujia**<sup>2</sup>. Similar to other app marketplace operators in China, such as Baidu, 360Safe, and Tencent, Wandoujia provides its own management app that facilitates users to search, browse, download, install, update, and uninstall apps. Once being launched, the Wandoujia management app runs as a background system-level service without requiring the “root” privilege. It also provides the user option to collect network activity information of each app on device, e.g., data traffic and access time under different networks.

The dataset that we study covers over 0.2 million (260,172) Android apps. The data mainly contains two types of user behaviors: (1) app management activities (i.e., installation, updating, and uninstallation) of over 0.8 million (8,112,145) unique users; (2) app network usage of 2 million (2,031,007) unique users. Based on this dataset, we conduct a *systematic descriptive analysis* that includes the distributions of app popularity, possible reasons of app selection, life cycles of abandoned apps, fine-grained network analysis of apps, and the impact of economic factors on app usage.

To the best of our knowledge, our work is the first descriptive analysis of millions of users of Android marketplaces. Although most users are Chinese, the usage patterns learned from the millions could generalize to other populations. Based on such a new dataset, we not only validate some conclusions from previous efforts, but also explore some new knowledge of user behaviors on app usage. The paper aims to answer the following questions:

- **How can we characterize the app popularity among millions of users?** Compared to previous studies that mainly investigated app popularity by downloads [17], we explore the app popularity from various aspects, including the monthly downloads of an app, the number of unique devices an app was installed on, the network traffic of an app, and the network access time of an app. Distributions of all these aspects comply with the Pareto principle [14], i.e., a small portion of apps account for substantial popularity. These findings can help marketplace operators and network providers improve their workloads for serving the most popular apps, and organize efficient caching or prefetching mechanisms to enable fast downloads and delivery of these popular apps.
- **How do mobile users choose and manage apps?** Compared to previous studies that usually examined the co-installed apps [24, 17], we further explore why the apps are installed and how soon they are likely to be uninstalled. We try to understand the possible reasons for co-installing apps, i.e., apps provided by the same **vendor/developer** or with similar **functionalities** are more likely to be installed together. We first study the uninstallation patterns of apps by devising a new metric called the *temporal installation/uninstallation ratio* to indicate how much an app is likely to be abandoned. These findings can help app marketplace operators improve their recommendation systems.
- **How do different apps perform in network activity?** Compared to previous studies [25, 24, 7], our network analysis is conducted at fine granularity. We

study the network activities in terms of the data traffic and the access time under Wi-Fi and cellular, respectively. We further investigate where the network activity occurs (foreground or background). Besides identifying some “network intensive” apps, we find apps that may “suspiciously” keep continuous network connections or consume traffic at the background. For example, it is surprising that some tools, such as flashlight apps and namecard apps, generate unexpected background traffic against their main purposes. We are also surprised to find that a large number of apps keep long-lived TCP connections at the background after they are launched. These findings can help marketplace operators to locate some “problematic” apps, and help users choose alternative apps to reduce potential threats caused by these apps.

- **How do economic factors affect app selection and network usage?** Compared to previous studies that relied on free/paid apps to measure economic factors [23], our study is performed in a quite different way. Based on the hypothesis that the price of a device model can reflect the user’s financial background, we classify users into different groups according to their device models. We make the first report on how device models impact the apps selection and data traffic, e.g., users with lower-end device models tend to use local or less network-intensive apps. Based on our findings, app marketplace operators can make more accurate app recommendations, and app developers can target specific user groups to optimize user experiences.

The remainder of this paper is organized as follows. Section 2 describes the dataset. Sections 3-6 describe the inferred app usage patterns in four aspects, app popularity patterns, management patterns, network patterns, and price-sensitive patterns, respectively. Section 7 summarizes the findings and implications to different stakeholders. Section 8 discusses the threats to validity. Section 9 makes comparisons with related work, and Section 10 concludes the paper.

## 2. DATASET

In this section, we briefly introduce the Wandoujia marketplace and describe the features of our dataset collected by the Wandoujia management app. To protect the user privacy and assure the ethnics of our research, we also discuss how the data is processed.

### 2.1 Wandoujia

Our data is from Wandoujia<sup>3</sup>, a free Android app marketplace in China. Wandoujia was founded in 2009 and has grown to be a leading Android app marketplace. Like other marketplaces, third-party app developers can upload their apps to Wandoujia and get them published after authenticated. Compared to other marketplaces such as Google Play, apps on Wandoujia are all free, although some apps can still support “in-app purchase”.

Users have two channels to access the Wandoujia marketplace, either from the Web portal, or from the Wandoujia management app. The Wandoujia management app is a native Android app, by which people can manage their apps,

<sup>2</sup><http://www.wandoujia.com>

<sup>3</sup>Visit its official site via <http://www.wandoujia.com>.

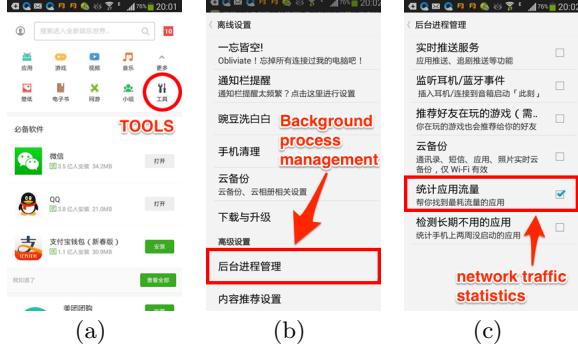


Figure 1: Screenshots of advanced settings in the Chinese version of the Wandoujia management app (the advanced settings is not supported in the current English version). (a) is the homepage of the Wandoujia management app, where users can navigate to “settings” by clicking the text circled by red; (b) refers to the background management service setting, which is highlighted by the red rectangle; (c) is to toggle whether to allow Wandoujia to collect the data of network activities.

e.g., downloading, searching, updating, and uninstalling apps. The logs of these management activities are all automatically recorded.

Beyond these basic features, the Wandoujia management app is developed with some advanced but optional features that can monitor and optimize a device. These features include network activity statistics, permission monitoring, content recommendation, etc. All features are developed upon Android system APIs and do not require “root” privilege. Users can decide whether to enable these features, as shown in Figure 1. However, these features are supported only in the Chinese version.

The Wandoujia management app is automatically launched and it works as a system-wide service after the device that installs the app starts up. The data collected on each device is uploaded to the Wandoujia server every day.

## 2.2 Data Collection

As of 2015, Wandoujia has over 250 million users. Each user is actually associated with a unique Android device, which could be either a smartphone or tablet computer. In the study of this paper, we collected one-month usage data from August 4, 2013 to September 2, 2013. The volume of our data set is 1.2 TB.

Our one-month dataset covers more than 0.2 million (260,172) Android apps. The data of user behaviors consists of two types: (1) the *data of app management activities* (i.e., installation, update, and uninstallation) (2) the *data of app network usage* (i.e., the traffic and access time per app).

### 2.2.1 App Management Activities

App management activities consist of downloading<sup>4</sup>, updating, and uninstalling apps. When an app is installed

<sup>4</sup>In the Wandoujia management app, a pop-up of installation wizard is presented to users when an app is downloaded. So we treat “download” and “installation” equally.

or updated via the Wandoujia management app, its installation counter is automatically increased by one and a log entry of this activity is created. The logs of uninstallation via the Wandoujia management app are processed in a similar way. In the Wandoujia management app, collecting app management activities is always enabled.

In our one-month data, we collected the management activity logs from **8,112,145 unique users** (devices in fact). We denote the dataset as “Universal User Set”. The logs of management activities are used to explore an app’s popularity, and can implicitly reflect the app’s quality.

### 2.2.2 Network Activities

When the advanced features are enabled, the Wandoujia management app collects daily network statistics of each app, if the app generates network connections either from Wi-Fi or cellular (2G/3G/LTE). However, if an app is never launched or generates no network connections, the app is not recorded in the network statistic logs.

To reduce the overhead of runtime monitoring, the Wandoujia management app does not record network details of each session of an app. Instead, it summarizes the total daily traffic and access time of an app by examining flows at the TCP level. The traffic and access time are captured for Wi-Fi and cellular, respectively.

In particular, the traffic and access time generated from **foreground** and **background** are treated separately. The Wandoujia management app determines whether an app is running at “foreground” by probing the Android system stack for every 2 seconds. In this way, the “foreground” access time can imply how long the user interacts with an app. The Wandoujia management app checks whether an app running at “background” every 2 minutes. If any network activity is detected during this interval, this app is regarded to be “online” and its “background” access time is increased by 2 minutes. Such time interval is reasonable to initiate and release a TCP connection.

In summary, the statistic of network activity provides 8 combinations of information, i.e., 2 *metrics* (access time and traffic) \* 2 *modes* (Wi-Fi and Cellular) \* 2 *states* (foreground and background).

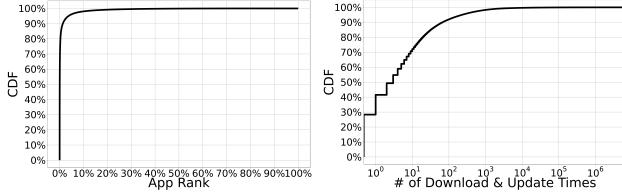
As the statistic of network activity is an optional feature for end-users, the covered users are a subset of the “Universal User Set”. We take into account only the users who successively contributed the statistics for more than 3 weeks. In our one-month dataset, the network activity covers **2,031,007 unique users**. We denote such a dataset as “Networked User Set”.

### 2.2.3 Device Models and Price

The Wandoujia management app also records the model information of each device, e.g., Samsung Galaxy Note 2, Google Nexus. We leverage the device models to classify users. Finally there are 12,091 different device models in total. Such a result implies the heavy fragmentation of Android devices. To better organize these models, we collected their listing price information when they were firstly put onto market.

## 2.3 User Privacy

We took a series of steps to preserve the privacy of involved users in our dataset. First, all raw data collected for this study was kept within the Wandoujia data warehouse



(a) Percentage of Downloads against App Rank    (b) Downloads of an App against # of Download & Update Times

Figure 2: App Popularity by Downloads

servers (which live behind a company firewall). Second, our data collection logic and analysis pipelines were completely governed by three Wandoujia employees<sup>5</sup> to ensure compliance with the commitments of Wandoujia privacy stated in the *Term-of-Use* statements. Finally, the Wandoujia employees anonymized the user identifiers. The dataset includes only the aggregated statistics for the users covered by our study period.

### 3. APP POPULARITY

In this section, we make a **macro-level analysis** on app distribution. In most previous studies [17, 19], apps are usually ranked by their number of downloads on the marketplaces. To make a comprehensive analysis, we use four metrics: **(1) the number of downloads of the app; (2) the number of unique devices that download the app; (3) the aggregated data traffic generated by the app; (4) the aggregated access time that users interact with the app.** The former two metrics can indicate how many users an app has, and the latter two can indicate how much an app is really used.

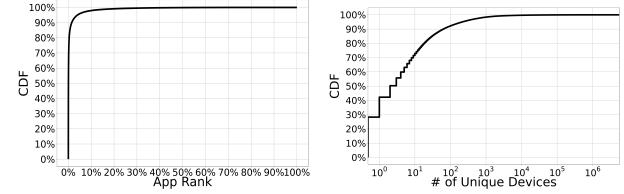
#### 3.1 Popularity by Downloads

First, we investigate the most intuitive metric of app popularity, i.e., the number of downloads of an app. Most app stores take the number of downloads (i.e., total, monthly, or weekly) as an indicator to rank app popularity. We then investigate the app downloads from the management activities of the **Universal User Set**. It is a common observation that people often update their installed apps. Hence, we compute total downloads of an app by aggregating its number of downloads and updates in our dataset.

Various evidence reports that the Pareto principle exists in networked application domains such as web content, audio, and video downloads [14], i.e., *20% of the objects account for 80% downloads*. Figure 2(a) shows that the cumulative distribution function (CDF) of the percentage of app downloads against the app ranking by downloads. The results show that the app downloads follow the Pareto principle. For example, 10% apps account for 96.2% downloads of all apps. Such finding validates a similar distribution of app downloads reported previously [17].

We also explore the distributions of downloads of each app. Figure 2(b) indicates that about 80% apps are downloaded or updated less than 10 times in our one-month dataset.

<sup>5</sup>One co-author, Feng Feng, is a co-founder and current CTO of Wandoujia. He supervised the process of data collection and de-identification.



(a) Percentage of Unique Subscribers against App Rank    (b) Unique Subscribers of an App against # of Unique Devices

Figure 3: App Popularity by Unique Devices

A substantially large number of apps are downloaded, updated, or uninstalled only once. Such finding indicates a “long-tail” of apps that are rarely active.

#### 3.2 Popularity by Unique Subscribers

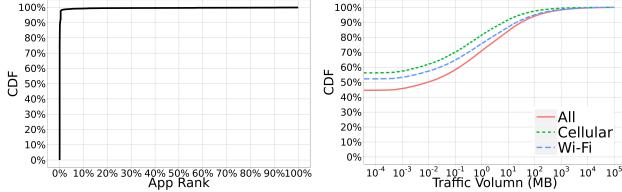
Although the number of downloads of an app is a natural indicator for popularity, it is also possible that some apps are frequently downloaded or updated due to “fraud” behavior. For example, apps can be downloaded and updated by automated programs to increase their ranks on marketplaces. To validate whether the “mostly downloaded apps” also have more users, we aggregate the **unique subscribers (devices)** that ever downloaded, updated, or uninstalled an app. Compared to the number of downloads of an app, each device is counted only once as a subscriber. Figure 3(a) shows that the CDF of the percentage of unique subscribers against app ranking by downloads. We can find that the distribution of subscribers of an app still complies with the Pareto principle, i.e., about 10% apps account for 95.1% users. In this way, we could conclude that **the more an app is downloaded, the more users it usually owns**.

Meanwhile, we explore how many users an app owns in Figure 3(b). It shows that about 80% apps have been installed, updated, or uninstalled by only less than 10 unique devices.

However, it is observed that some apps may have high downloads but rather limited users/devices. We find that one app used by only 4 unique subscribers has 973 downloads in one month. We further explore its download logs, and find that its downloading actions were all performed in a quite short interval (within two days). There are two possible reasons. One is that all of the 4 devices (subscribers) are testing devices used by the app developers. The other is that the app developers may purposely increase the downloads. **Although such kind of apps is rare, the number of their downloads may mislead end-users or even bring threats.**

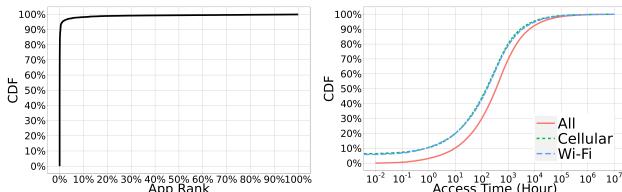
#### 3.3 Popularity by Network Traffic

Either downloads or unique devices of an app can just indicate that an app is downloaded and installed. However, we cannot judge whether the app is really used by users. Not surprisingly, **mobile users may never use an app after first launching it, but do not uninstall it either**. From the logs of the **Networked User Set**, an app cannot generate network logs if it is never launched by users. Hence, we employ the network activities to examine whether the app is really used. Although we may miss some apps that



(a) Percentage of Traffic against App Rank      (b) Traffic of an App

Figure 4: App Popularity by Data Traffic



(a) Percentage of Access Time against App Rank      (b) Access Time of an App

Figure 5: App Popularity by Access Time

are usually used offline, e.g., PDF readers or dictionaries, most of currently popular apps heavily rely on network and can be covered.

For better illustration, we distinguish *Wi-Fi*, *Cellular*, and *All* (*Wi-Fi* + *Cellular*) of each app. The data traffic comes from both foreground and background. In contrast, we take into account only the access time from foreground, because such time indicates how long users really interact with the app.

Figure 4(a) and Figure 5(a) illustrate the CDF of the percentage of aggregated traffic/access time of apps against app ranking by downloads, respectively. We can find that the Pareto principle still holds for the network activities of apps.

We also show the aggregated data traffic/access time of an app in Figure 4(b) and Figure 5(b), respectively. We can observe that about 97% apps consume less than 100 MB traffic volume in one month, and about 95% apps are used less than 100 hours.

### 3.4 Summary

From the preceding macro-level analysis of popularity, we can conclude that the app popularity generally complies with the Pareto principle in terms of downloads, unique users, data traffic, and access time. This finding not only validates results reported in previous studies [24, 17], but also helps marketplace operators and network providers identify the “real” popular apps that are frequently downloaded, updated, and used. To make the apps fast delivered to end-users, the marketplace operators can leverage this distribution to allocate resources for serving these apps, design efficient cache mechanisms, or quickly redirect user requests to the app providers. Knowing which apps are more adopted by users, network providers can provide better bandwidth, cache/prefetch contents to users, or explore value-added services such as advertisements to increase revenue.

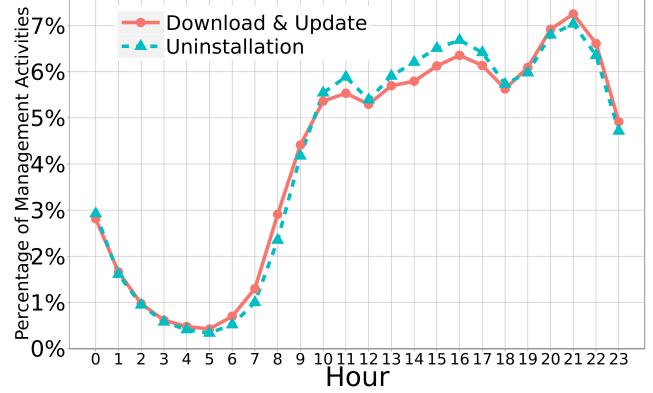


Figure 6: **Distribution of Diurnal App Management Activities.** Each point on the curve represents the percentage of activities performed during the one-hour time interval against the total activities, during the whole day. *For example, activities during 10:00 am-11:00 am account for about 6% of all activities.*

## 4. APP MANAGEMENT PATTERNS

In this section, we investigate how users manage their apps, e.g., which apps they often install and when they install these apps, which apps are more likely to be uninstalled, and why these apps are uninstalled by users.

### 4.1 Diurnal Management Patterns

App management activities include downloading, updating, and uninstalling apps. Exploring app management activities is motivated for various reasons. As app management activities are usually made manually, they can implicitly reflect the density and frequency of user interactions with the devices. For marketplace operators, this information can help them understand when a large number of concurrent user requests would arrive, so that the marketplace operators can optimize their servers for faster and more reliable network bandwidth. For app developers, they can leverage such information to publish the latest versions of apps at a proper time, to attract more downloads and updates.

We investigate the diurnal downloading, updating, and uninstallation distribution. We aggregate activities of downloading and updating an app, because they both reflect the users’ interest towards this app. As shown in Figure 6, the app management activities are “periodically” performed during a day. The extent of app downloading and updating activities keeps growing from 6:00 am and reaches the first peak around 11:00 am. The downloading and updating activities decline slightly between 11:00 am to 12:00 pm. It is not very surprising because users may take lunch at this time. The same observation could be found between 4:00 pm to 6:00 pm, i.e., the time on the way back home or at dinner. We can also find that about 32% downloading and updating activities are performed during 7:00 pm to 11:00 pm, where they reach the maximum around 9:00 pm. Such distribution is quite consistent with human regularity. After 9:00 pm, the downloading and updating activities decline quite sharply, and reach the minimum around 5:00 am. However, at the midnight, downloading and updating activ-

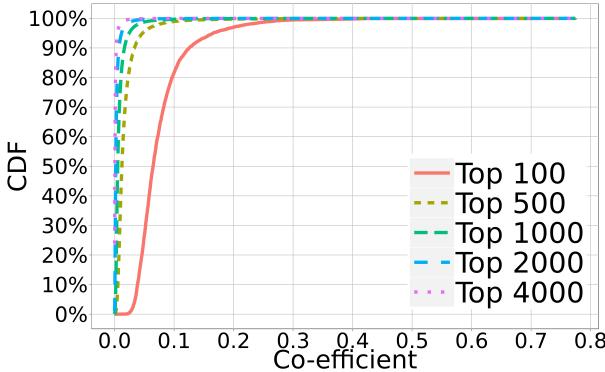


Figure 7: Coefficiency of installed apps.

ties occupy about 7% in total, implying that there are still a large number of active users.

Activities of uninstalling apps present a similar distribution to the ones of downloading/updating apps. However, knowing when users uninstall apps may be less meaningful, because the uninstallation activities do not have interactions with the marketplace operators or app providers.

## 4.2 App Selection Patterns

Knowing user interests and needs towards apps is useful. For marketplace operators, such information can help improve their recommendation systems. For app developers, such information can help improve their apps to work better with other co-installed apps, or explore potential collaboration opportunities with other developers. For network providers, such information can help them provide a more personalized data plan to bundle correlated apps and download them.

In fact, a lot of previous studies investigated how users select apps [24, 11, 19, 17], and some interesting findings were reported. From these studies, a common metric is to check the “cluster effect”: which apps are more likely to be selected together. We adopt the same metric, but investigate it at two levels: the **micro-level** of co-installed apps, and the **meso-level** of correlated app categories.

From our previous analysis, substantial apps are rarely downloaded (less than 10 times or 10 devices), so we choose only the top 4,000 apps by their downloads. Each app is downloaded by at least 20 devices. To better explore the underlying reasons why apps are selected, we organize the apps into categories according to the classification system of Wandoujia. Table 1 summarizes the categories ordered by the number of apps. We also present the number of users and network summary of each category, which are used for analysis in Section 5.

### 4.2.1 Co-Installed Apps

We first study co-installed apps. Given two apps  $app_m$  and  $app_n$ , we employ the classical **Jaccard Similarity Coefficient** (denoted as  $\lambda$ ) to measure the possibility how they are likely to be installed together. We denote the number of unique devices that install either  $app_m$  or  $app_n$  as  $\mathbb{D}$  ( $app_m \cup app_n$ ), and the number of unique devices that install both  $app_m$  and  $app_n$  as  $\mathbb{D}$  ( $app_m \cap app_n$ ). We then compute  $\frac{\mathbb{D}(app_m \cap app_n)}{\mathbb{D}(app_m \cup app_n)}$ .

Figure 7 shows the **Jaccard Similarity Coefficient** of the top- $N$  apps, where  $N$  varies from 100 to 4,000. With the increasing the number of  $N$ , the CDF of  $\lambda$  indicates that there are a very small fraction of apps being co-installed together. The  $\lambda$  of 90% app pairs is lower than 0.1, if both  $app_m$  and  $app_n$  have 2,000 unique devices. Given the  $\lambda$  value of 0.1,  $app_m$  and  $app_n$  share only 200 unique devices. In contrast, we focus on the app pairs that take a higher value of  $\lambda$ , and explore **why these apps are more likely to be installed together**. We choose the pairs whose  $\lambda$  is more than 0.1. Rather than exploring the detailed textual descriptions of the apps, we focus on two properties *vendor* and *category*.

The *vendor* information of an app can reflect the developer, provider, or owner of the app. Usually, the naming rules of an Android app can reflect its vendor information. For example, the package `com.tencent.mm` can be processed by removing the general term “com” and the usage term “mm”, and the vendor information “tencent” is extracted. By clustering the vendors, we find that **a number of co-installed apps with high  $\lambda$  value come from the same vendor**. For example, the pair of < 360Safe (a mobile anti-virus app used by 719,258 devices), 360 AppStore (a marketplace app used by 560,600 devices) > has the  $\lambda$  value of 0.358, and these two apps are both provided by Qihoo<sup>6</sup>. Furthermore, the  **$\lambda$  value could be much higher, if two apps developed by the same vendor belong to the same category**. For example, the pair of < MI Video (a media player used by 54,217 devices), MI Voice Assistant (a voice assistant used by 75,221 devices) > holds the  $\lambda$  value of 0.604, and the two apps are both provided by XIAOMI<sup>7</sup>; the pair of < WeChat (used by 1,567,563 devices), QQ (used by 1,606,222 devices) > holds the  $\lambda$  value of 0.7, and the two apps are provided by Tencent.

There are many possible reasons why apps from the same vendor are often co-installed. A vendor may focus on a specific application domain, e.g., Tencent is the largest messaging service provider in China. Tencent QQ is the most popular instant messaging app in China; WeChat not only supports instant messaging, but also provides social communication features such as content sharing. Another reason is that there might be “**in-app bundled installation**” in some apps. For example, when users install an app, the app’s vendor may implicitly or explicitly recommend users to install other apps. For simple validation, we make field studies by selecting 50 apps from well-known app developers such as Qihoo, Baidu, and Tencent, and install them manually. 14 apps out of the 50 apps recommend installing other apps in their installation process, and 8 apps of these 14 “bundled” installations suggest apps provided by the same provider.

### 4.2.2 Correlation of App Categories

The *category* information of an app indicates the functionality and application domains of the app. We can infer the user needs and interests according to their selected app’s category. Furthermore, knowing the correlations of app categories can suggest marketplace operators to organize apps

<sup>6</sup>Qihoo is one of the largest anti-virus software vendors in China.

<sup>7</sup>XIAOMI is one of the largest mobile phone manufacturers in China.

Table 1: Chosen Top Apps by Category.

App Category	Apps	Users ( $10^6$ devices)	Downloads ( $10^6$ times)	Traffic (GB)	Access-Time ( $10^7$ hours)	C-Traffic	C-Time	W-Traffic	W-Time
GAME	1,227	3.87	15.15	13,669.71	0.38	2.98%	5.19%	0.76%	6.39%
NEWS_AND_READING	274	1.17	1.97	13,143.17	0.23	3.11%	2.91%	0.72%	3.95%
VIDEO	238	2.86	6.52	1,196,978.79	0.38	28.41%	1.42%	81.08%	10.54%
TOOL	227	3.84	9.43	77,329.87	0.68	15.63%	10.79%	4.40%	9.46%
SYSTEM_TOOL	217	3.37	7.54	34,012.16	0.25	3.05%	3.37%	2.17%	4.24%
SOCIAL	188	2.18	4.01	35,926.76	0.35	8.96%	4.77%	1.94%	5.66%
EDUCATION	172	1.68	2.98	13,893.55	0.34	1.46%	5.35%	0.87%	4.71%
LIFESTYLE	156	1.68	2.85	2,388.59	0.07	0.72%	1.00%	0.12%	1.06%
TRAVEL	111	1.62	2.75	8,182.24	0.03	0.78%	0.53%	0.52%	0.25%
PERSONALIZATION	104	1.49	3.68	7,426.38	0.86	0.85%	12.03%	0.46%	13.67%
FINANCE	99	0.32	0.50	382.60	0.02	0.13%	0.24%	0.02%	0.26%
COMMUNICATION	85	4.09	8.45	54,394.71	2.85	24.74%	49.01%	2.26%	35.26%
SHOPPING	78	1.57	3.00	21,808.51	0.07	3.16%	0.65%	1.32%	1.60%
PRODUCTIVITY	75	0.76	1.17	2,712.50	0.01	0.18%	0.17%	0.18%	0.26%
MOTHER_AND_BABY	48	0.10	0.15	525.72	0.01	0.07%	0.04%	0.03%	0.12%
MUSIC	43	2.33	3.39	49,540.12	0.17	5.66%	2.47%	3.08%	2.49%
SPORTS	27	0.31	0.36	61.40	0.00	0.02%	0.05%	0.00%	0.04%
IMAGE	23	0.14	0.17	801.64	0.00	0.06%	0.01%	0.05%	0.03%
TRAFFIC	14	0.10	0.12	78.10	0.00	0.02%	0.03%	0.00%	0.01%

The users, downloads, traffic, and access time are all computed by aggregating the data of each app in the category.

The percentile of *W-Traffic* (*C-Traffic*) and *W-Time* (*C-Time*) refer to the data traffic and foreground access time over Wi-Fi (*W*) and cellular (*C*) network, respectively.

from different categories to enable the fast delivery of potentially co-installed apps to end-users.

Given two app categories  $M$  and  $N$ , we denote the number of unique devices that install an app either in  $M$  or  $N$  as  $\mathbb{D}(M \cup N)$ , and the number of unique devices that install apps from both in  $M$  and  $N$  as  $\mathbb{D}(M \cap N)$ . We then compute  $\frac{\mathbb{D}(M \cup N)}{\mathbb{D}(M \cap N)}$  to indicate how likely that the apps in  $M$  and  $N$  are installed together. We also take into account the special case, where  $M=N$ , indicating how many users install more than one app in this category.

Figure 8 shows the probability distribution that apps from different categories are selected together. The categories are sorted in the descending order of the number of apps. An immediate observation is that categories having more apps are more closely correlated. In addition, apps providing related functionalities are more likely to be selected together. For example, users may want to share a video from a video-player app to friends by employing a communication app (e.g., correlation between *COMMUNICATION* and *VIDEO* is 0.77), or use a viewer app to open a document that is received by an instant messenger app (e.g., correlation between *TOOL* and *COMMUNICATION* is 0.88).

It is not surprising that users may install more than one app in some categories. For example, in *GAME*, *TOOL*, and *COMMUNICATION*, the correlations are all more than 0.8. The result suggests that users have more interests and needs in these categories.

### 4.3 Uninstallation Patterns

We next to explore a question: **how an app is likely to be disliked or abandoned by users**. Such question is quite crucial to app developers, marketplace operators, and end-users. App developers can know how much their apps are not appreciated by users, so that they can find and fix problems in time to avoid user loss. Marketplace operators can improve their recommendation systems to filter unpopular, low-quality, or even malware apps. End-users can avoid downloads of frequently abandoned apps and potential threats.

For the question, the absolute number of unistallations of an app may not be a good indicator. For example, apps with high uninstalls may also have high downloads. So we compute the metric of *installation/uninstall ratio*, denoted as  $\Omega(app_i)$ , to indicate how likely an app would be abandoned.

Given an app  $app_i$ , we first compute  $\frac{\sum \mathbb{I}_{device_i}}{\sum \mathbb{U}_{device_i}}$ , where  $\sum \mathbb{I}_{device_i}$  and  $\sum \mathbb{U}_{device_i}$  represent the number of devices that installed and uninstalled  $app_i$ , respectively. We extract all devices that appear in both installation and uninstallation logs of  $app_i$  from the **Universal User Set**. For each device, we order the  $app_i$ 's installation and uninstallation log entries by their timestamps, in the form of  $< I_{t_1}, I_{t_2}, \dots >$  and  $< U_{t_1}, U_{t_2}, \dots >$ . Here,  $t_i$  refers to the timestamp when the installation or uninstallation action was performed. In particular, an uninstallation action could not be used unless an installation action was made previously, i.e.,  $I_{t_i} \prec U_{t_i}$ . Doing so assures that  $app_i$  is exactly uninstalled.

The lower value of  $\Omega(app_i)$  an app holds, the higher likelihood it is abandoned. For better illustration, Figure 9(a) shows the scattered distribution of  $\Omega(app_i)$  (denoted as “I/U ratio”) of top 12,000 apps ranked by their downloads. The median value of  $\Omega$  is about 7.46. The  $\Omega$  of 80% apps’ is less than 11. **In this way,  $\Omega$  can exactly tell how much an app is actually abandoned by users.** However,  $\Omega(app_i)$  is not a good signal to comprehensively reflect how much an app is disliked by users, because users may not always uninstall an app even if they do not need the app any longer.

To better understand the users' attitude towards apps, we evaluate the lifecycle of abandoned apps by combining the temporal information to  $\Omega$ . Such evaluation is motivated by an intuition that an app is likely to be a disliked one if it is uninstalled shortly after installed. To this end, we compute the app's lifecycle by the timestamps of installation and uninstallation. We have two immediate observations. First, from Figure 9(b), if an app is exactly uninstalled, its lifecycle can be identified. **About 40% abandoned apps can “survive” for only less than one day, and about 93% abandoned apps can “survive” for less than a week.**

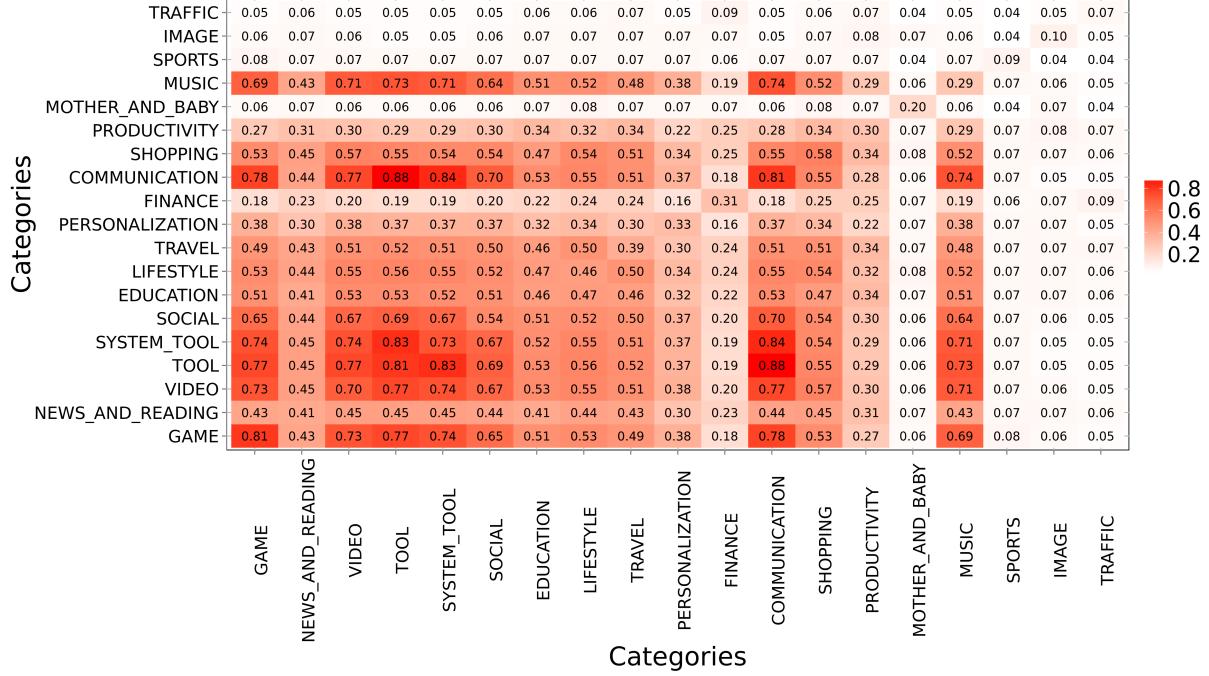


Figure 8: Heatmap of illustrating the relationship of co-installed app categories.

Second, from Figure 9(c), we can find a positive correlation among  $\Omega$  and lifecycle of abandoned apps. In other words, apps with a lower  $\Omega$  are more likely to be uninstalled within a shorter interval.

## 5. NETWORK ACTIVITY PATTERNS

Understanding network activities of apps is a highly interesting topic. Previous studies have already revealed some findings on network usage of apps, e.g., TCP flows on tier-1 network [24], or usage logs by field studies [20, 23]. In contrast, our study is performed at a much finer granularity. First, we distinguish daily data traffic and access time from Wi-Fi and cellular network, respectively. Second, we distinguish daily data traffic and access time from foreground and background, respectively.

Based on the granularity of network activities, our study aims to explore some findings not covered by previous efforts. End-users can know which apps are network-intensive, and result in more data traffic and battery life. In this way, end-users can identify “undesirable” apps, optimize improper network privilege, or even remove these apps. Marketplace operators can also detect some potentially problematic apps. App developers can fix possible bugs, and OS-vendors can patch to avoid threats. Network providers can leverage network behavior and suggest proper data plans for end-users.

### 5.1 Data Traffic Patterns

First, we identify “network-intensive” apps, i.e., apps that consume more data traffic than others. We aggregate apps by their categories and summarize the total traffic consumption (in GB) from Wi-Fi and cellular, respectively. As shown in Table 1, *VIDEO* apps are the most “traffic-intensive”. Apps from the *VIDEO* category consume 81.08% of Wi-Fi

traffic and 28.41% cellular traffic against all apps. Interestingly, apps from *TOOL* and *SYSTEM\_TOOL* consume a lot of data traffic. The apps in these two categories include apps of input method, anti-virus, app management, etc. Users heavily rely on these apps to manage, optimize, and personalize their devices.

We then classify data traffic into two dimensions: (1) Wi-Fi and cellular; (2) foreground and background. Such classification can provide us more insights of traffic consumption. We show the detailed summary in Table 2.

#### 5.1.1 Traffic of Wi-Fi and Cellular

In most categories, it is not surprising that Wi-Fi accounts for more than 60% in total traffic. In the categories of *VIDEO*, *TOOL*, *MUSIC*, *SYSTEM\_TOOL*, *SHOPPING*, and *EDUCATION*, more than 80% data traffic is from Wi-Fi. A possible reason is that most of these apps are usually used in places with stable Wi-Fi connection, e.g., at home or cafe. The situation is quite different in *COMMUNICATION*. The traffic from cellular network accounts for about 39.9%. It is not surprising because users may use *COMMUNICATION* apps whenever a network connection is available.

#### 5.1.2 Traffic of Foreground and Background

We then identify the foreground and background traffic of an app, respectively. Often, the foreground traffic is generated when users interact with the app. In an Android OS, a foreground app can be determined if the app is currently at the top of the activity stack. In contrast, the background traffic implies that the app is still connecting to network even users do not interact with it. From Table 2, the foreground traffic accounts for more than 60% in most categories. Foreground traffic accounts for about 50%

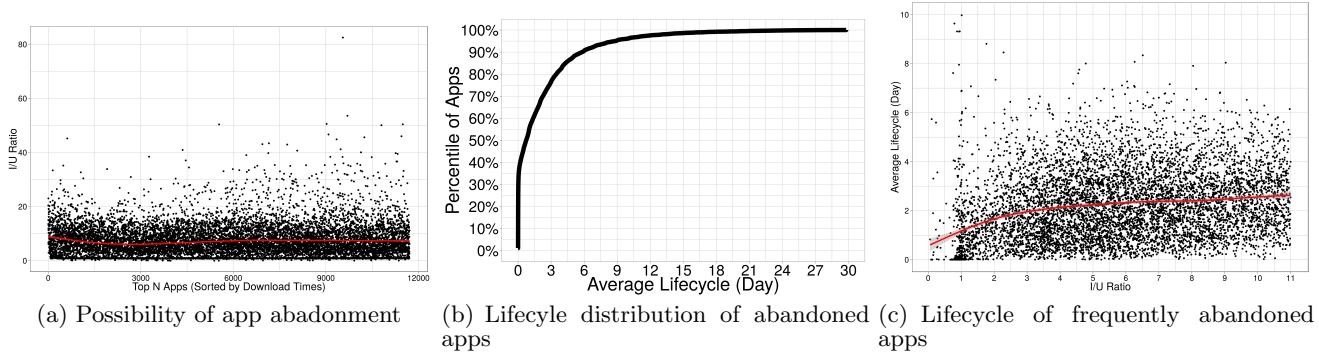


Figure 9: Lifecycle of abandoned app.

Table 2: Network Summary by App Category

App Category	<i>C</i> -Traffic (B)	<i>W</i> -Traffic (B)	<i>C</i> -Traffic (F)	<i>W</i> -Traffic (F)	<i>C</i> -Time (B)	<i>W</i> -Time (B)	<i>C</i> -Time (F)	<i>W</i> -Time (F)
<b>VIDEO</b>	0.81%	45.13%	1.28%	52.78%	42.62%	56.66%	0.10%	0.63%
<b>TOOL</b>	8.16%	39.13%	9.56%	43.14%	48.57%	50.42%	0.57%	0.43%
<b>COMMUNICATION</b>	12.42%	15.90%	27.48%	44.20%	48.01%	46.85%	3.15%	1.99%
<b>MUSIC</b>	4.35%	35.19%	5.67%	54.80%	49.23%	50.09%	0.36%	0.32%
<b>SOCIAL</b>	7.26%	20.65%	14.63%	57.47%	48.43%	50.41%	0.57%	0.59%
<b>SYSTEM_TOOL</b>	5.07%	51.57%	2.80%	40.55%	50.02%	49.48%	0.23%	0.26%
<b>SHOPPING</b>	3.29%	17.09%	9.42%	70.21%	43.34%	56.42%	0.08%	0.17%
<b>EDUCATION</b>	3.76%	39.38%	5.46%	51.40%	45.57%	52.83%	0.90%	0.69%
<b>GAME</b>	10.34%	43.11%	8.80%	37.74%	48.13%	51.34%	0.26%	0.28%
<b>NEWS_AND_READING</b>	5.91%	24.64%	14.83%	54.62%	43.43%	55.25%	0.60%	0.71%

*W* and *C* refer to Wi-Fi and Cellular, respectively.  
*B* refers to background and *F* refers to foreground.

in four categories, i.e., *SYSTEM\_TOOL* (43.35%), *GAME* (46.54%), *TOOL* (52.7%), and *VIDEO* (54.1%). We infer that some apps in these categories keep consuming a large amount of traffic, when users switch to use other apps, or the screen-off traffic occurs with device sleeping [9]. This observation draws our attention. Usually, *GAME* apps may embed some third-party advertisement libraries (adlibs). Besides adlibs, *VIDEO* apps may often prefetch content. Hence, the background traffic of these apps should be necessary. If we can find reasonable explanations for background traffic, we denote such behavior as “reasonable”. For example, some management apps such as Wandoujia and anti-virus apps often need downloading or updating actions at background. Otherwise, we annotate an app with “unknown” if we can not judge whether the background traffic is really necessary. We manually check some top apps with high traffic consumptions to see whether these consumptions are reasonable. We find 14 apps that we are not quite clear why they need background network privilege<sup>8</sup>. For example, on average, an alarm clock app (*at.samsung.powersleep*) daily consumes about 13 MB cellular traffic and 156 MB Wi-Fi traffic at background for a user among about 300 users; a LED flashlight app (*com.chenlei.flashlightfree*) daily consumes about 7 MB cellular traffic and 5 MB Wi-Fi traffic at background for a user among about 20 users. Users may suffer from lots of unnecessary loss of background cellular data if they use these apps.

<sup>8</sup>We list these apps and their network activities on <http://www.sei.pku.edu.cn/~liuxzh/IMC2015/>

There are some reasons for producing background network activities. One reason is that ad libraries are widely used in a lot of apps, and may download and update advertisements according to user contexts. We simply validate such reason from two aspects. First, we crawl the user reviews of the 14 “suspicious” apps on Wandoujia, and check whether advertisement issues are reported. From user reviews, 3 out of 14 apps are found with complaints about advertisement. Second, by disassembling the .apk files and conducting program analysis, we find that 3 out of 14 apps include at least one of popular advertisement libraries published on AppBrain<sup>9</sup>. Although we are still not sure whether and how much these advertisement libraries consume background network, we believe such finding already brings awareness to end-users about the potential traffic loss.

Another reason is that the misuses or even malicious granting of network permissions. Since our current case analysis is manual, we plan to leverage our previously developed WHYPER tool [15] to further dig out “suspicious” apps by combining their permissions and descriptions.

In summary, the results show that users may be unaware that the traffic is “silently but maybe unexpectedly” consumed at background. It reminds users to alert or kill background network activities after launching these apps. Background network activities may lead to unnecessary data loss in a data plan, or imply some threats.

<sup>9</sup><http://www.appbrain.com/stats/libraries/ad>. AppBrain is a well-known platform for Android users and developers.

## 5.2 Access Time Patterns

We then investigate the access time of network activities. Intuitively, access time may reflect two important insights. First, the foreground access time of an app indicates how long a user interacts with it. Therefore, such metric can imply how much the user likes or needs the app. Second, similar to background data traffic, background access time indicates how long an app connects to network when users do not interact with it. Therefore, the background access time can imply the “liveness” of the app after it is launched.

Similar to the study of data traffic, we first illustrate the access time distribution among app categories, as shown in Table 1. When exploring the foreground access time, it is not surprising that the *COMMUNICATION* apps account for 49% cellular time and 35.26% against all apps. It is also interesting to find that users spend a lot of time on *TOOL* (10.79% under cellular and 9.46% under Wi-Fi).

We investigate how much foreground and background account for network time, respectively. From Table 2, we are surprised to observe that foreground time accounts for only less than 2% (by aggregating *W-Time(F)* and *C-Time(F)*) in total network access time, whereas the background time occupies more than 98% (by aggregating *W-Time(B)* and *C-Time(B)*). In other words, most apps still keep “long-and-live” TCP connection at background after being launched, although they are not used by users. The background network time may be reasonable for apps that heavily rely on network, e.g., *COMMUNICATION*, and *SOCIAL*. For example, most of these apps require auto synchronization or notification. However, it hardly makes sense that many apps from other categories have continuous network connection at background. Although background network activities do not always reflect more data traffic, they still occupy network connections and may lead to unnecessary energy cost.

## 6. PRICE-SENSITIVE PATTERNS

The preceding measurements provide us information on some patterns of app selection and network usage. In this section, we further explore these patterns by classifying users, i.e., **how these patterns are influenced by the prices of devices**. Our goal is motivated by two concerns: (1) **do users with devices of different prices have different needs and interests in using apps?** (2) **if yes to (1), how much do the prices of devices matter?** In other words, we would like to investigate whether the user’s usage patterns are price sensitive.

### 6.1 Device Model Clustering

We cluster users by their device models from the **Networked App Set**, that covers about 2 million users. We finally have 12,091 device models. We present the number of users of each device model in Figure 10. It is observed that 96% device models have less than 500 users. This number of users accounts for only less than 0.1% in our dataset. It demonstrates the heavy fragmentation of Android devices. Therefore, we take a set of “*popular*” device models, each of which has at least 500 users. We further manually check these device models and merge some duplicated device models. For example, GT 7100/7102/7108 all refer to Galaxy Note 2 with the same hardware specification, but customized by different network carriers such as China Mobile and China Unicom. In this way, we finally label 327

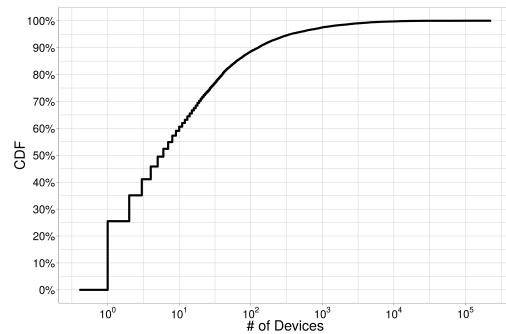


Figure 10: CDF for Number of Users of Device Models

“*popular*” device models. Then, we categorize the 327 device models according to their on-market sale prices. Although the price of a device is cut down gradually after the device is sold, such categorization could still roughly reflect economic factors. We put each device model into one of five groups, which are numbered as 1-5 in Table 3.

### 6.2 Apps Used Among Groups

We investigate the number of used apps among different groups, and present the distribution in Figure 11. Note that we examine the apps from the **Networked User Set**, indicating that they are exactly used by users. An immediate observation is that **the higher price a device model has, the more apps are used on this device model**. Such finding might come from two reasons. First, the more expensive device models usually provide more powerful hardware specifications, e.g., faster CPU, larger RAM, and higher definition of screen. The computation power should motivate the users to install and use more apps. Second, more expensive device models usually come from more famous manufacturers, such as Samsung, Motorola, LG. Such observation is attributed to the fact that Android systems customized by these manufacturers may pre-load more apps than cheaper devices, and may lead to the problem of “*bloatware*”. We download the factory ROM files of some representative device models from ROMJD<sup>10</sup>. For example, the Samsung Galaxy Note 2 (Group 5) has 18 pre-installed apps that generate network activities, and the number on Galaxy S4 (Group 4) is 15. In contrast, Meizu M040 (Group 1) has only 9 pre-installed apps that generate network activities. Although we sample only some device models, the finding can support our hypothesis to some degree.

### 6.3 Network Activity Among Groups

Similar to the preceding analysis of network activities, we show the CDF of data traffic and access time from different groups in Figure 12. **We can find that the usage of Wi-Fi also tends to take a positive correlation with device model price**. In other words, the more expensive a device model is, the more Wi-Fi traffic its users tend to consume. However, **the situation is quite different for cellular traffic**. The differences of cellular traffic among Group 1-3 are quite marginal, where 80% users consume less than 150 MB cellular traffic in one month. From

<sup>10</sup>ROMJD is a website sharing Android ROM files. <http://www.romjd.com/>

Table 3: Device Model by On-Market Price

Group (price in Chinese Yuan)	# of Device Models	Typical Device Model	# of Covered Users
Group_1: 1,000-	57	HUAWEI-T8951, HUAWEI-C8812, Samsung SCH-I739	216,249
Group_2: 1,000-1,999	98	Motorola XT681, Meizu M040, Samsung GT-I9082	385,811
Group_3: 2,000-2,999	76	Xiaomi MI-1S, Xiaomi MI-2, Samsung GT-S7568, LG-P970	329,361
Group_4: 3,000-4,999	69	Samsung Galaxy S3, Nexus 4, HTC One	517,262
Group_5: 5,000+	27	Samsung Galaxy Note 2, Galaxy S4, SCH-N719	378,504

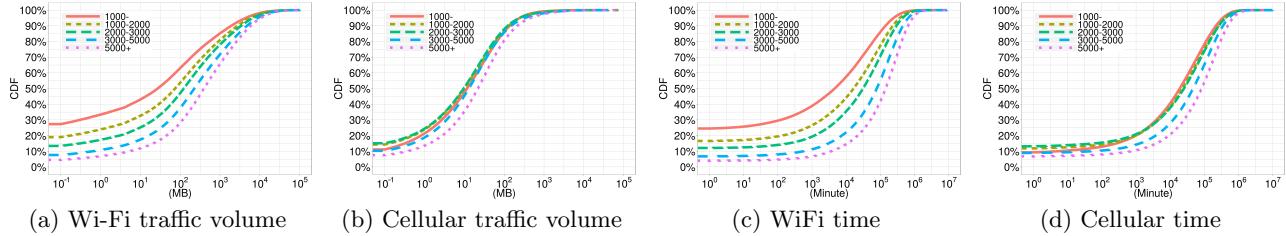


Figure 12: Network Activity Distribution among User Groups

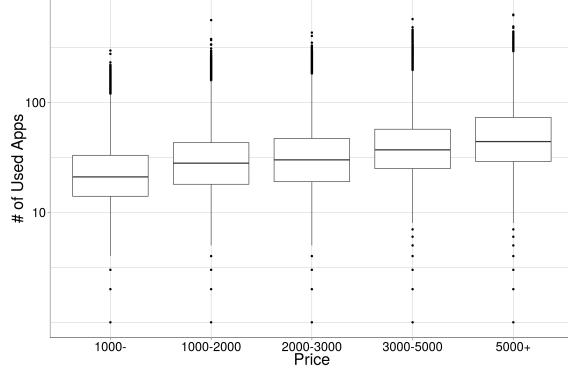


Figure 11: Number of App Usage Across Different Groups

Group 4, the cellular traffic and the access time begin to increase. Users from Group 5 consume the most cellular traffic and the most access time. About 5% users from this group use about 1GB cellular traffic. There may be two possible reasons. First, as device models from Groups 4 and 5 are more powerful, their users tend to connect to the network and download more contents. Second, users from these two groups could be supposed to have better economic background, so they are likely to afford a higher-cost data plan to use.

#### 6.4 User Interest on Similar Apps

To explore whether users' interests are affected by device prices, we study how users from different groups perform in selecting apps with the same or similar functionalities. We choose two typical types of networked apps: (1) **News Reader**: we choose 5 popular apps: Netease, Sohu, Phoenix, Tencent, and Sina; (2) **Browser**: we choose 8 popular browsers: Chrome, FireFox, Opera, Maxthon, UCWeb, 360Safe, Baidu, and Sogou. We show the foreground time against device models. Some preference tendencies could be found. For news reader apps, users from Groups 4 and 5 tend to use the Netease app, but users from Groups 1-3 tend to use the Sohu app. A similar finding could be found

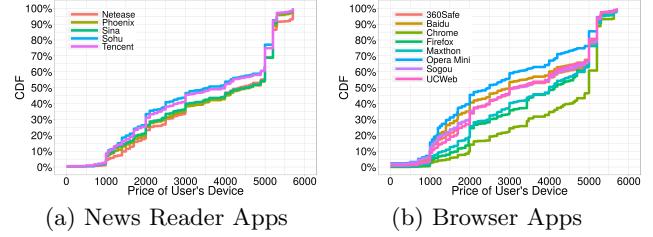


Figure 13: Similar App Preferences among User Groups

in **Browser**. Users from Groups 4 and 5 tend to use **Chrome**, but users from Group 1-3 tend to use **UCWeb** and **360safe**.

For better understanding the user preferences, we assume that the features and performance may impact users from different groups. First, users in Groups 1-3 may prefer local apps over international apps. For example, **UCWeb** and **360safe** are both developed by local app providers in China, and preferred over **Chrome**. Second, user interface designs may influence target user groups. For example, the **Tencent** news reader often embeds a video that fits the textual content browsed by users. Considering 42% of its users are from Group 3, we can suppose that users in this group are interested in watching videos instead of textual content. In contrast, users from Groups 4-5 seem to prefer to go through the textual content, so they prefer the **Netease** news reader. Third, some specific features might motivate different user preferences. For example, **Opera Mini** is said to save traffic by offloading computation onto cloud, and its 60% users are from Groups 1, 2, and 3. In summary, such finding implies that device prices has impacts on app selection and usage, and reflect different user interests and needs.

## 7. IMPLICATIONS

We have investigated the user behaviors and infer some patterns. In this section, we summarize our findings. We not only validate some findings that were reported in previous studies based on a smaller scale of users, but also identify

Table 4: Summary of Findings and Implications

Findings in Diverse Usage Patterns	Implications for Relevant Stakeholders
The popularity distribution of apps typically follows the “Pareto” effect. A large number of smartphone apps are with only quite few download count and subscriber devices, and contribute marginally to network usage. Such a finding has been mentioned in various previous studies [2, 5, 7, 17].	The <b>marketplace operators</b> can identify which apps are really popular and more frequently downloaded and used. The marketplace operators should put the .apk files of these popular apps “as close as possible” to end-users, e.g., by deploying Content Delivery Networks (CDN) servers. They can further design efficient mechanisms such as cache, prefetching, and large bandwidth to make these apps fast delivered. The apps that are downloaded by a limited number of devices in a short interval could alert marketplace operators and end-users for further examination.
<b>Diurnal distribution of apps management</b> is quite periodical during a day. Böhmer <i>et al.</i> [2] also reported some similar results.	The <b>marketplace operators</b> can better allocate resources such as server and bandwidth to adapt the downloading/updating requests. The <b>app developers</b> can publish their up-to-date app versions at proper time to attract more downloads.
It is not surprising that some apps are frequently installed together such as <i>COMMUNICATION</i> and <i>SOCIAL</i> . Apps that come from the same vendor or have similar functionalities are more likely to be installed together.	The <b>marketplace operators</b> can improve recommendation systems, e.g., cluster the frequently co-installed apps at a close location to allow end-users to fast download them. The <b>app developers</b> can leverage this knowledge to find “mutual-composition” opportunities and design some APIs, e.g., navigating to a <i>social networking</i> app from a <i>news reader</i> app to share content. Some co-installed apps come from the same vendors. The <b>end-users</b> can be cautious to avoid undesirable app downloads. The <b>advertisers</b> can deliver relevant ads in some apps. For example, earphone advertisers can focus on users who use <i>MUSIC</i> and <i>VIDEO</i> apps as they are correlated with <i>SHOPPING</i> apps.
An app’s installation/uninstallation ratio performs a positive correlation to its lifecycle. Most “disliked” apps are likely to be uninstalled within 2 days.	The <b>marketplace operators</b> can know which apps are disliked and further explore why they are abandoned. The <b>end-users</b> can judge an app’s popularity even without user reviews according to its installation/uninstallation ratio.
Some apps are more network intensive than others with similar functionalities. Similar findings were reported previously [16].	The <b>marketplace operators</b> can recommend end-users proper apps that have similar functionalities but better fit the users’ data plan. The <b>network providers</b> can also leverage this knowledge and provide special data plan. For example, some network providers in China make special data-plan contract with <i>music</i> (such as Baidu Music ( <code>com.ting.mp3.oemc.android</code> ) and Kuwo Music ( <code>cn.kuwo.player</code> )) and <i>video</i> apps (such as Youku ( <code>com.cibn.tv</code> )), and users can pay for this data plan independently and enjoy unlimited cellular traffic to download video/audio files or enjoy them online.
Some apps may consume a large amount of traffic volume at background, but the behaviors might not be reasonable or necessary, e.g., the <i>flashlight</i> and <i>namecard</i> apps. Some apps are found using advertisement libraries.	Such findings can remind the <b>marketplace operators</b> to pay attention to these apps and warn the <b>end-users</b> and the <b>app developers</b> . The <b>OS-vendors</b> can benefit from our findings by preventing these apps from potential threats.
A lot of apps keep long-lived TCP connection when they are not “used” by users.	Background network connections would consume resources and energy, or even imply some malicious behaviors. Such a finding reminds that the <b>end-users</b> should periodically “clean up” their devices or terminate the threads of unused apps running at background.
Although fragmentation is quite severe in Android devices, most device models have only few users.	Such findings can suggest the <b>app developers</b> focus on mainstream devices and make their apps adaptive to them.
The app selection is diverse among users who take different device models. For example, lower-end device users prefer the <i>Opera</i> browser as it is said to save traffic.	The <b>marketplace operators</b> can recommend functionality-similar apps to users with different device models.
The network usage is diverse among different users who have different price levels of device models. Lower-end device users heavily rely on Wi-Fi. In contrast, higher-end devices are likely to use more cellular than lower-end device users.	Such findings can predict how much data plan a device will probably use based on the same or similar device models. In this way, the <b>marketplace operators</b> can recommend <i>end-users</i> different apps according to their own data plan. The <b>app developers</b> can also provide some features to adapt different users, e.g., making their apps prefetch contents under Wi-Fi network for the lower-end users.

In the left column, we summarize the findings from our study. In the right column, we suggest some implications for various stakeholders.

some new insights and suggest some implications. We list the findings and implications in Table 4.

## 8. THREATS TO VALIDITY

One potential limitation of our work is that the dataset is collected from a single app marketplace in China. This limitation may have introduced some biases caused by app marketplace specific policies, and some of our findings may not always hold in other app marketplaces. Care should be given to generalize our findings to another marketplace.

Another limitation is that the users under study are mainly Chinese, and the region differences should be considered. The same limitation also exists in most of previous studies that were conducted over users from a specific region, e.g., some states in USA [24]. To alleviate this issue, we conduct our study over millions of users. We believe that such scale of users could reduce the threats. In fact, our findings such as the Pareto principle of app popularity and user interests of co-installed apps have validated the findings from previous studies.

Because the analysis is done on one month of usage data, some of our findings may not generalize to a longer period of time. Mobile apps are updated very frequently, and some potential security threats of apps, e.g., overprivilege of network permissions, might be already fixed in their up-to-date versions. Meanwhile, we realize that one-month data is not sufficient to predict app quality in the latest Wandoujia marketplace. We plan to evaluate our findings based on a new half-year dataset and investigate how users and app usage evolve.

## 9. RELATED WORK

Understanding user behavior of mobile apps establishes a foundation for different stakeholders in the research community of mobile computing, e.g., app developers, network providers, marketplace operators, and OS vendors. A plethora of studies have been made from different perspectives.

**Understanding User Behavior by Field Study.** Given that collecting large-scale user data is hardly feasible for most studies, learning user behavior by field study is always a straightforward way. A lot of studies were performed over specific user groups. Rahmati *et al.* [19, 18] performed a four-month field study of the adoption and usage of smartphone-based services by 14 novice teenage users. Tossell *et al.* [23] applied a naturalistic and longitudinal logs-based approach to collect real usage data from 24 iPhone users in the wild. Sanzi *et al.* [20] collected data from 387 Android users in India, where users pay for cellular data consumed, with little prevalence of unlimited data plans. Falaki *et al.* [6] found that web browsing contributed over half of the traffic. Using detailed traces from 255 volunteer users, Falaki *et al.* [7] conducted a comprehensive study of smartphone use and found immense diversity of users, by characterizing intentional user activities. Lim *et al.* [12] made a questionnaire-based study to discover the diverse usages from about 4,800 users across 15 top GDP countries. Yan *et al.* [25] developed and deployed an app to collect usage logs from over 4,600 users to find their similar interests and explore mobile apps recommendation systems for smartphone apps. For a study closely to ours, Xu *et al.* [24] presented usage patterns by analyzing IP-level traces of thousands of users from a tier-1 cellular carrier in U.S. They identified

traffic from distinct marketplace apps based on HTTP signatures and present aggregate results on their spatial and temporal prevalence, locality, and correlation.

Some field studies were made on specific apps. Böhmer *et al.* [2, 3] made a field study over three popular apps such as Angry Bird, Facebook, and Kindle. Patro *et al.* [16] deployed a multiplayer RPG app game and an education app, respectively, and collected diverse information to understand various factors affecting application revenues.

**Mining App Marketplace Data.** Some types of app related data like user reviews, star ratings, and like/dislike voting are publicly accessible. Chen *et al.* [4] presented AR-Miner to extract informative user reviews and group them using topic modeling. Fu *et al.* [8] presented WisCom, a system that can analyze tens of millions user ratings and comments in mobile app markets. Pestas *et al.* [17] monitored and mined four popular third-party Android app marketplaces and showed that the app popularity distribution deviates from commonly observed Zipf-like model.

**Predicting Apps to Use.** Some studies target predicting the “*to-be-used*” apps by collecting user logs. Shin *et al.* [22, 21] collected a wide range of smartphone information from 23 users, extracted and analyzed features related to app prediction. Liao *et al.* [11, 10] proposed a temporal-based app predictor to dynamically predict the apps which are most likely to be used. Montoliu *et al.* [13] presented a framework to discover places-of-interest from multimodal mobile phone sensory data. Do *et al.* [5] presented a framework for predicting where users will go and which app they are to use in the next ten minutes from the contextual information collected by smartphone sensors.

Compared to these studies, the major differences of our study include the unique dataset covering millions of users, some unique information such as app installation, uninstallation, and diverse network usage. Although Chinese users take up majority of all users, we believe that behavior patterns inferred from millions of users under study should be more generalized and comprehensive than those from volunteers. With our dataset, we also validate some results that were reported over smaller scale of users. For example, a small set of apps account for substantially a large portion of downloads [17] and unique users [24], some apps are more likely to be installed together [24, 6, 7], and some functionality-similar apps may vary in terms of performance [20]. However, besides using a different dataset collected from millions of users, our study explores uniquely new findings that were not covered previously:

1. First, we make comprehensive measurement of the app popularity from various aspects including downloads, users, and diverse network activities.
2. Second, we explore which apps are likely to be uninstalled and the lifecycle of the abandoned apps.
3. Third, beyond reporting the co-installation of apps, we further explore the possible reasons why these apps are selected together.
4. Fourth, we make a fine-granularity analysis of network activities to identify the “network-intensive” apps and “problematic” apps that consume traffic at background.

- Finally, we study the economic factor by the price of device model, and explore how it impacts on user behaviors on apps usage.

## 10. CONCLUSION

We conducted a systematic descriptive analysis of a large collection of mobile app usage behaviors from millions of Android users. Interesting usage patterns are with respect to app popularity, app management, app selection, app abandonment, network usage, and so on. Our findings provide valuable implications for different stakeholders in the mobile app industry and in the research community of mobile computing.

This paper mainly focuses on the descriptive analysis of the data. Many findings of the analysis lead to interesting research questions that can be immediately explored. Such research tasks include predicting app quality and popularity, optimizing app performance, and improving app recommendations, etc. Development of these directions can directly improve the Wandoujia management app and benefit millions of users.

## Acknowledgment

This work was supported by the National Basic Research Program (973) of China under Grant No. 2014CB347701, the Natural Science Foundation of China (Grant No. 61370020, 61421091, 61222203, 61572051, 61528201). Tao Xie's work was supported in part by National Science Foundation under grants no. CCF-1349666, CCF-1409423, CNS-1434582, CCF-1434590, CCF-1434596, CNS-1439481, and CNS-1513939. Qiaozhu Mei's work was supported in part by the National Science Foundation under grant no. IIS-1054199.

## 11. REFERENCES

- A. Apaolaza, S. Harper, and C. Jay. Understanding users in the wild. In *Proc. of W4A*, page 13, 2013.
- M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with angry birds, Facebook and Kindle: a large scale study on mobile application usage. In *Proc. of MobileHCI*, pages 47–56, 2011.
- M. Böhmer and A. Krüger. A study on icon arrangement by smartphone users. In *Proc. of CHI*, pages 2137–2146, 2013.
- N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang. AR-miner: mining informative reviews for developers from mobile app marketplace. In *Proc. of ICSE*, pages 767–778, 2014.
- T. M. T. Do and D. Gatica-Perez. Where and what: Using smartphones to predict next locations and applications in daily life. *Pervasive and Mobile Computing*, 12:79–91, 2014.
- H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In *Proc. of IMC*, pages 281–287, 2010.
- H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *Proc. of MobiSys*, pages 179–194, 2010.
- B. Fu, J. Lin, L. Li, C. Faloutsos, J. I. Hong, and N. M. Sadeh. Why people hate your app: making sense of user feedback in a mobile app store. In *Proc. of KDD*, pages 1276–1284, 2013.
- J. Huang, F. Qian, Z. M. Mao, S. Sen, and O. Spatscheck. Screen-off traffic characterization and optimization in 3g/4g networks. In *Proc. of IMC*, pages 357–364, 2012.
- Z. Liao, S. Li, W. Peng, P. S. Yu, and T. Liu. On the feature discovery for app usage prediction in smartphones. In *Proc. of ICDM*, pages 1127–1132, 2013.
- Z. Liao, Y. Pan, W. Peng, and P. Lei. On mining mobile apps usage behavior for predicting apps usage in smartphones. In *Proc. of CIKM*, pages 609–618, 2013.
- S. L. Lim, P. J. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden. Investigating country differences in mobile app user study behavior and challenges for software engineering. *IEEE Transactions on Software Engineering*, 40(5):40–64, 2014.
- R. Montoliu, J. Blom, and D. Gatica-Perez. Discovering places of interest in everyday life from smartphone data. *Multimedia Tools Appl.*, 62(1):179–207, 2013.
- M. E. J. Newman. Power Laws, Pareto Distributions and Zipf's Law. *Contemporary Physics*, 46:323, 2005.
- R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications. In *USENIX Security*, pages 527–542, 2013.
- A. Patro, S. K. Rayanchu, M. Grieppentrog, Y. Ma, and S. Banerjee. Capturing mobile experience in the wild: a tale of two apps. In *Proc. of CoNEXT*, pages 199–210, 2013.
- 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. of IMC*, pages 277–290, 2013.
- A. Rahmati, C. Tossell, C. Shepard, P. T. Kortum, and L. Zhong. Exploring iphone usage: the influence of socioeconomic differences on smartphone adoption, usage and usability. In *Proc. of MobileHCI*, pages 11–20, 2012.
- A. Rahmati and L. Zhong. Studying smartphone usage: Lessons from a four-month field study. *IEEE Trans. Mob. Comput.*, 12(7):1417–1427, 2013.
- A. A. Sani, Z. Tan, P. Washington, M. Chen, S. Agarwal, L. Zhong, and M. Zhang. The wireless data drain of users, apps, & platforms. *Mobile Computing and Communications Review*, 17(4):15–28, 2013.
- C. Shin and A. K. Dey. Automatically detecting problematic use of smartphones. In *Proc. of Ubicomp*, pages 335–344, 2013.
- C. Shin, J. Hong, and A. K. Dey. Understanding and prediction of mobile application usage for smart phones. In *Proc. of Ubicomp*, pages 173–182, 2012.
- C. Tossell, P. T. Kortum, A. Rahmati, C. Shepard, and L. Zhong. Characterizing web use on smartphones. In *Proc. of CHI*, pages 2769–2778, 2012.
- Q. Xu, J. Erman, A. Gerber, Z. M. Mao, J. Pang, and S. Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *Proc. of IMC*, pages 329–344, 2011.
- B. Yan and G. Chen. Appjoy: personalized mobile application discovery. In *Proc. of MobiSys*, pages 113–126, 2011.