Comprehension of Ads-supported and Paid Android Applications: Are They Different?

Rubén Saborido, Foutse Khomh, Yann-Gaël Guéhéneuc, Giuliano Antoniol
Polytechnique Montréal, Canada

***Abstract*—The Android market is a place where developers offer paid and–or free apps to users. Free apps can follow the freemium or the ads-business model. While the former offers less features and the user is charged for unlocking additional features, the latter includes ads to allow developers to get a revenue. Free apps are interesting to users because they can try them immediately without incurring a monetary cost. However, free apps often have limited features and–or contain ads when compared to their paid counterparts. Thus, users may eventually need to pay to get additional features and–or remove ads. While paid apps have clear market values, their ads-supported versions are not entirely free because ads have an impact on performance. The hidden costs of ads, and the recent possibility to form family groups in Google Play to share purchased apps, make it difficult for developers and users to balance between visible and hidden costs of paid and ads-supported apps.**

In this paper, first, we perform an exploratory study about ads-supported and paid apps to understand their differences in terms of implementation and development process. We analyze 40 Android apps and we observe that (i) ads-supported apps are preferred by users although paid apps have a better rating, (ii) developers do not usually offer a paid app without a corresponding free version, (iii) ads-supported apps usually have more releases and are released more often than their corresponding paid versions, (iv) there is no a clear strategy about the way developers set prices of paid apps, (v) paid apps do not usually include more functionalities than their corresponding ads-supported versions, (vi) developers do not always remove ad networks in paid versions of their ads-supported apps, and (vii) paid apps require less permissions than ads-supported apps. Second, we carry out an experimental study to compare the performance of ads-supported and paid apps and we propose four equations to estimate the cost of ads-supported apps. We obtain that (i) ads-supported apps use more resources than their corresponding paid versions with statistically significant differences and (ii) paid apps could be considered a most cost-effective choice for users because their cost can be amortized in a short period of time, depending on their usage.

***Index Terms*—Android; Performance metrics; Advertisements**

I. Introduction

The Android market Google Play is the place where Android developers and users meet to offer and to find apps. In this market, developers may wish to maximize the number of downloads and ratings of their apps while users may be more interested in finding the best and cheapest apps that fulfill their needs. In general, developers want to maximize their profits while users want to reduce their costs.

There exist different business models for Android apps[1]. One model is the pay to download in which users pay the app before downloading. The opposite model is the free app-only model, in which developers do not seek any monetary reward but, most likely, coverage for their apps. Between these two extreme models stand the freemium model, in which developers offer apps for free but ask money for unlocking more features, and the ads-supported model, in which developers offer apps for free but use ads to generate revenues.

On the one hand, both developers and users are interested by free apps: developers to showcase their apps and users to test out these apps for free. On the other hand, developers may offer, in addition to their free versions, paid apps and include in the free versions ads. These ads-supported apps offer less or similar features than their corresponding paid versions and they use ad networks to display ads that provide revenue to developers. While paid apps have clear market values (their prices), ads-supported versions are not entirely free because ads in apps have an impact on app ratings and users' privacy [1] but also on performance metrics as CPU, memory, network usage, and power consumption [2], [3].

Yet, users are sometimes reluctant to pay for apps when ads-supported versions of the same or similar apps exist for free[2]. For this reason, and to increase the numbers of purchases of paid apps, Google launched in August 2016 the concept of "Family Group"[3]. When users set up a family group on Google Play, the family manager can invite up to five people to the group and they can share purchased apps. Thus, the prices of the purchased apps is divided by the numbers of family members.

To the best of our knowledge, there are currently no recommendations for developers regarding the business model to choose for their apps and the impact of this choice. In this paper we want to understand the usage of the ads-supported model by developers and users and, consequently, we ask the general question: *"Are ads-supported and paid Android apps different?"*. We answer this question in two steps, by asking: *"How is the ads-supported business model used?"* and *"How is the ads-supported business model impacting performance and price?"*

We answer the first question through an exploratory study about the ads-business model comparing ads-supported and paid apps to understand their differences and development process. We analyze 40 Android apps, 20 ads-supported apps downloaded from Google Play and their corresponding paid

[1] https://developer.android.com/distribute/monetize/index.html

[2] https://goo.gl/31IXuZ

[3] https://support.google.com/googleplay/answer/6286986

versions bought in the same market. We analyze the frequency of releasing of ads-supported and paid apps and the evolution of the prices of paid apps across releases. In addition, we collect and process information about all the developed Android apps offered in Google Play by developers of the selected apps to know the proportion of paid apps with respect to free ones in the marketplace. We also compare both ads-supported and paid apps in terms of required permissions, used ad networks, and offered features.

We answer the second research question by carrying out an experimental study to compare the performance of ads-supported and paid apps and we propose four equations to estimate the cost of ads-supported apps. We want to make explicit the hidden costs of ads when considering the possibility to form family groups. Thus, we want to provide some advices to developers, to seize and act on the balance between visible and hidden costs of paid and ads-supported apps. We collect performance metrics running each app in a real phone (repeating 30 times each measure to allow for statistical tests and effect size). Using these data, we determine the cost of ads-supported apps due to ads depending on the network usage, the battery drained, and the time in which a data plan is over, to estimate the time in which an ads-supported app overtakes its paid version.

From our exploratory study, we observe that (i) ads-supported apps are preferred by users although paid apps have a better rating, (ii) developers do not usually offer a paid app without a corresponding free version, (iii) ads-supported apps usually have more releases and are released more often than their corresponding paid versions, (iv) there is no a clear strategy about the way developers set prices of paid apps, (v) paid apps do not usually include more features than their corresponding ads-supported versions, (vi) developers do not always remove ad networks in paid versions of their ads-supported apps, and (vii) usually paid apps require less permissions than ads-supported apps.

From our experimental study, we confirm that (i) ads-supported apps use more resources than their corresponding paid versions and we conclude that differences are statistically significant and (ii) we show that, depending on the usage, paid apps could be considered as a better choice for users because its cost could be amortized in a short period of time.

The remainder of the paper is organized as follows. Section II presents the data collection process used in our exploratory research and experimental study. Section III describes the experiments carried out and the observations found by our exploratory research. Section IV presents and discusses the experimental study about performance and cost of ads-supported apps. Section V discusses threats to the validity of our study. Finally, Section VI reports related works and Section VII concludes, summarizing and discussing our findings and providing some advices to developers.

II. Study Data Collection

We want to analyze ads-supported and paid versions of Android apps to understand the usage of the ads-business model by developers and users. In addition, we want to study the balance between the costs of ads in free apps and the costs of paid apps, while considering their features and sharing among family groups. Finally, we want to provide developers with some advices about the impact of the business model on both users and developers.

A. Selection of Subject Apps

The context of our study is the official Android marketplace, Google Play, and the subjects are ads-supported and paid Android apps available in this marketplace. For each category in Google Play, we randomly selected eight paid apps with an ad-supported version available. Over the resulting 128 apps we selected 65 which worked on our phone and contained visible ads. Among these 65, we randomly selected 20 from different developers. Our selection process is akin to a stratified random sampling of paid apps with two strata: ad-supported version and working on our phone. We bought the paid apps and we downloaded their corresponding ads-supported versions.

Selected apps belong to 11 different categories and the number of downloads for ads-supported and paid apps was in the range $[500, 5,000,000]$ and $[50, 1,000,000]$, respectively. For all of this we consider that we have a reduced but representative sample of Android apps to carry out an exploratory research and an experimental study.

Table I shows the apps selected for our study. The second column shows the identifier associated with each app. The third and fourth columns contain information related to the category to which the app belongs and the developer who develops the app, respectively. Next column shows information about the versions and the number of downloads in the marketplace for the selected apps. All of this information was obtained from Google Play in May 2016, when apps were downloaded (ads-supported apps) and bought (paid apps).

B. Analysis and Monitoring of Subject Apps

We collect information about releases from Google Play and AppBrain. The latter is a public source for information about Android apps where we can find data on all apps on Google Play, resources for developers, and statistics about the Android ecosystem. We obtain information about permissions using the Android tools `aapt` and `dumpsys`. To analyze ad networks in Android apps we use the Android app AppBrain Ad Detector, which is available for free in Google Play.

To compare ads-supported and paid apps in terms of performance metrics, for each app, we create a Robotium scenario to run an exercise the apps. Robotium[4] is an open-source test automation framework for Android development that makes it easy to write powerful and robust automatic black-box UI tests for Android apps. We use the commercial easy-to-use tool Robotium Recorder[5] to create Robotium test cases based on component identifiers and not on absolute or relative coordinates.

[4] https://github.com/RobotiumTech/robotium

[5] http://robotium.com/products/robotium-recorder

TABLE I
APPS SELECTED FOR THE STUDY.

				Ads-supported version			Paid version		
App	Id	Category	Developer	Version	Rating	Downloads	Version	Rating	Downloads
AccuWeather Platinum	A01	Weather	Accuweather.com	3.4.2.12	4.3	50.000.000 - 100.000.000	4.0.1	4.2	100.000 - 500.000
Pocket Salsa	A02	lifestyle	Modernistik	2.14	4.3	100.000 - 500.000	2.14	4.8	10.000 - 50.000
DroidEdit Pro (code editor)	A03	Productivity	André Restivo	1.23.1	4.1	1.000.000 - 5.000.000	1.23.1	4.6	50.000 - 100.000
How to Tie a Tie Pro	A04	Books & Reference	Artfonica	2.6.2	4.6	5.000.000 - 10.000.000	2.7	4.6	10.000 - 50.000
Sleep Time+ Smart Alarm Clock	A05	Health & fitness	Azumio Inc.	1.36.1026	4.1	1.000.000 - 5.000.000	1.36.1026	4.2	1.000 - 5.000
Perfect Body Building Plan Pro	A06	Health & fitness	Bodybuilding-Apps.com	1.2.0	4.2	100.000 - 500.000	1.1.6	4.6	1.000 - 5.000
CountDownr	A07	Tools	SpeedyMarks	1.1.0	2.6	5.000 - 10.000	1.1.0	3.5	100 - 500
Dictionary.com Premium	A08	Books & Reference	Dictionary.com, LLC	5.2.2	4.6	10.000.000 - 50.000.000	5.2.2	4.7	100.000 - 500.000
Calculator Plus	A09	Productivity	Digitalchemy, LLC	4.9.2	4.5	10.000.000 - 50.000.000	4.9.4	4.8	50.000 - 100.000
Moon+ Reader Pro	A10	Books & Reference	Moon+	3.5.3	4.4	10.000.000 - 50.000.000	3.5.3	4.7	500.000 - 1.000.000
PRO PDF Reader	A11	Books & Reference	Ivan Ivanenko	4.4.1	4.0	10.000.000 - 50.000.000	4.4.4	4.4	10.000 - 50.000
Popster	A12	Finance	ITIOX	2.1	3.7	10.000 - 50.000	2.1	4.3	500 - 1.000
Video Player HD Pro	A13	Media & Video	wowmusic	1.0.8	4.4	1.000.000 - 5.000.000	1.0.2	4.5	1.000 - 5.000
Time Trial Stopwatch	A14	Sports	Liuto Apps	1.2.2	3.3	500 - 1.000	1.2.2	3.5	50 - 100
Mighty Grocery Shopping List	A15	Shopping	Mighty Pocket	4.0.15	4.2	100.000 - 500.000	4.0.15	4.5	50.000 - 100.000
Hair Color Studio Premium	A16	lifestyle	ModiFace	1.4	3.3	5.000.000 - 10.000.000	1.4	3.9	10.000 - 50.000
Date / Calendar Converter Full	A17	Tools	NoM	1.7	4.5	5.000 - 10.000	1.4	5.0	100 - 500
OBDII Trouble Codes	A18	Transportation	NoM	1.14	3.9	100.000 - 500.000	1.13	4.5	10.000 - 50.000
Maths Formulas	A19	Books & Reference	NSC Co.	9.1.1	4.4	1.000.000 - 5.000.000	9.1	4.6	1.000 - 5.000
Relax Melodies P: Sleep & Yoga	A20	Health & Fitness	Ipnos Software	3.2	4.4	5.000.000 - 10.000.000	3.3	4.8	100.000 - 500.000

The scenarios typically start the apps, skip the initial tutorials if they exist, and wait for 100 seconds in the main activity which contains visible ads in the free apps. The same activity is used for the paid app to make a fair comparison. Because the default refresh rate value of ad networks use to be 60 seconds, setting the length of the scenario to 100 seconds guarantees that ads are loaded at least once. In our experiments, the scenarios are run 30 times while performance metrics are collected.

We use a LG Nexus 4 Android phone, equipped with the Android Lollipop operating system (version 5.1.1, build number LMY47V), which is connected to an external power supplier to get power consumption measurements using a hardware based approach. CPU, memory, and network usages are collected using well known methods. All of this is commented in detail in Section IV.

III. Exploratory Study

We refine our first high-level research question, *How is the ads-supported business model used?*, into seven detailed research questions (*RQs*). In this section, we introduce and discuss the details of the experiments that we carried out to address each of the *RQs*. For each research question we describe our motivations, the data-collection approach, and discussions of our observations.

RQ 1: Do developers prefer ads-supported or paid apps?

Motivation: Developers can offer in Google Play independent free apps (including or not ads), independent paid apps, or both free and paid versions of the same app. We analyze the number and type of apps offered in the marketplace by the developers of the apps shown in Table I to understand what type of apps developers usually prefer.

Approach: We extract from Google Play information about the existing apps in the marketplace offered by developers. Then, we count the numbers of independent free apps, of independent paid apps, and of free apps with a corresponding paid version.

Results: In total, developers offer 331 apps in Google Play. Out of these 331 apps, 116 free apps have the corresponding paid versions, 88 are without paid apps, and 11 are without free version. Thus, we conclude that (i) 70.09% of the free apps have corresponding paid versions, (ii) 26.59% of apps do not have corresponding paid versions, and (iii) only 3.32% of paid apps do not have corresponding free versions.

Developers prefer offering both free and paid version of their apps in Google Play.

RQ 2: Do users prefer ads-supported or paid apps?

Motivation: When considering ads-supported and paid apps, users can decide to install the free app that contains ads or buy the paid app without ads. We compare ads-supported and paid apps in terms of numbers of downloads and users' ratings, which can be considered as measures of success [4], [5]. This information is useful for developers to know the point of view of users about ads-supported apps and their paid versions.

Approach: In Google Play, the numbers of downloads and users' ratings are shown for each app in the marketplace. A rating is a number between one and five calculated as the weighted average of all users' ratings. The number of downloads is given as a range specifying how many times an app was downloaded and installed. For each ads-supported and paid app we compare these two metrics.

Results: Table I shows that ads-supported apps are always downloaded more than their paid counterparts. Considering the central values of the ranges defining the numbers of downloads we conclude that ads-supported apps are downloaded, in average, 113 times more than their corresponding paid versions. Regarding ratings, for 19 apps (90%), paid versions have better ratings than their corresponding ads-supported versions. For only two apps (10%), free versions have ratings greater than or equal to their paid versions. In average, ratings for paid apps is 4.43 while for ads-supported apps is 4.09.

Users prefer ads-supported apps although they rate paid apps better.

RQ 3: How often are ads-supported apps released?

Motivation: In Google Play, developers can release their apps as often as they consider it necessary. We analyze the frequency of releasing of ads-supported and paid apps to understand how developers release free and paid apps.

Approach: We crawl the AppBrain website[6] to extract information about the previous releases of the studied apps. The collected information is stored in a CSV file that we use to compare the frequency of releasing of both ads-supported and their corresponding paid apps.

Results: Fig. 1 shows the distribution of the frequencies of releasing of ads-supported and paid versions of the subject apps, where the symbol ◊ shows the average value. In average, ads-supported apps have 21.10 releases while paid apps have 17.90 releases. We also study the number of days between consecutive releases for ads-supported and paid apps. Ads-supported apps are released, in average, every 84 days (median 43 days) while paid apps are released less frequently: every 101 days in average (median 51 days). In addition, we observe that developers usually release ads-supported apps before their corresponding paid versions.

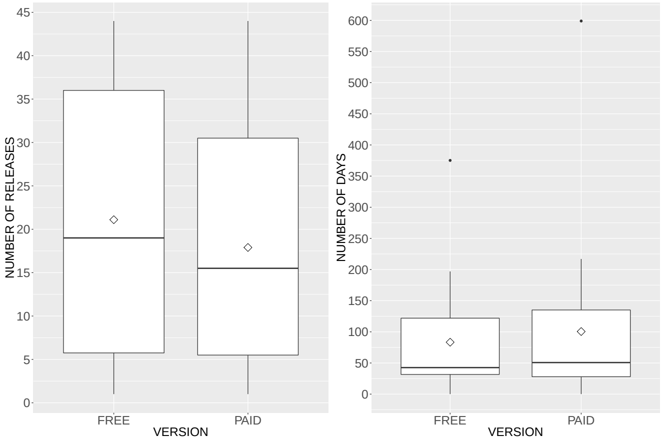

Fig. 1. Release frequencies for ads-supported and paid apps.

Fig. 2 shows the comparison of releasing frequencies for three apps, for space constraints, chosen among the apps selected for this study. We chose these three apps because they illustrate different ways of releasing apps. For each app is shown the date of each release and the release number, for both ads-supported and paid apps. For A05, the ads-supported version was released before the paid one. Then, when the free app was mature enough, the paid app was released and, from this point in time on, both ads-supported and paid apps were released concurrently. When considering the A16, the first version for both ads-supported and paid versions were released at the same time. Then, ads-supported and paid apps were released alternatively with a difference of few days between them. For A18, the paid app was released and, one month later, the free version was released, probably due to a small number of downloads for the paid version.

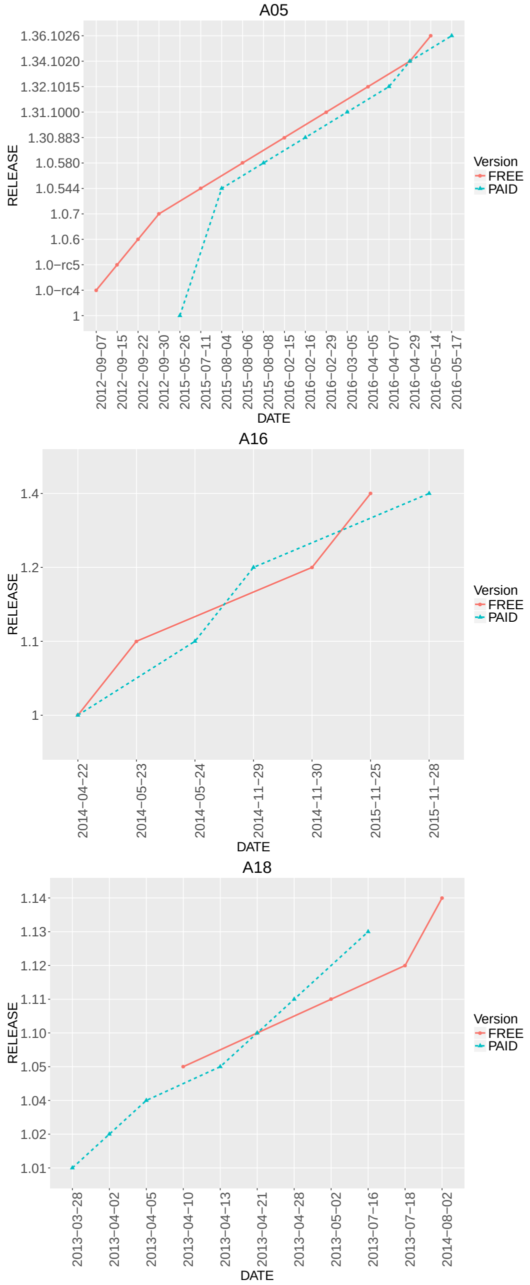

Fig. 2. Comparison of releasing frequencies for three apps.

> *Ads-supported apps usually have more releases and are released more often than their corresponding paid versions.*

RQ 4: How does the price of paid apps evolve?

Motivation: Paid apps are not free by definition and have a price that can be changed by developers at any time. We study the evolution of the prices of different releases of paid apps to understand how developers set and update prices.

Approach: The information about the releases of apps as explained above also contained the price of paid apps for each

[6] http://www.appbrain.com

of their releases. Using this information, we can study the evolution of prices of paid apps.

Results: Fig. 3 shows the evolution of the cost for each paid app and release. It does not show a clear trend. A05, A07, A10, and A16 maintained their prices constant over different releases while prices were increased for A09, A15, A19, and A20. For the other apps the price fluctuated over different releases.

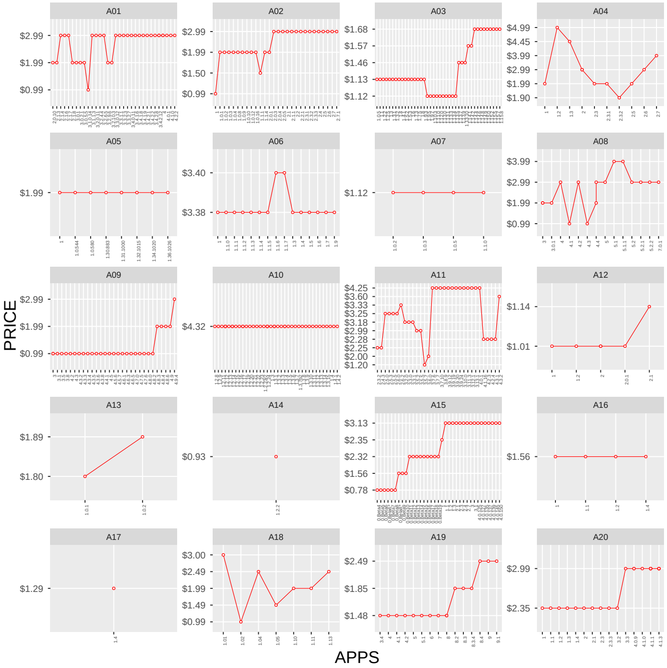

Fig. 3. Price evolution of paid apps.

> *There is no a clear trend in the evolution of prices of paid apps over different releases.*

RQ 5: Do ads-supported apps have reduced features?

Motivation: Developers introduce ads in their free apps to get a revenue. They release paid versions without ads and, sometimes, with more features. We compare the features of ads-supported and paid apps to know if ads-supported apps have as much features as the paid ones.

Approach: Information about the existence of ads or feature limitations in ads-supported apps is available in the descriptions of the apps in Google Play. For each app, we manually check the app description to assess the features offered by both the ads-supported and paid versions of the apps.

Results: In six cases (A03, A10, A14, A15, A18, and A20), paid apps offer more features than their ads-supported versions. It means that only 30% of the studied apps offer more features in their paid versions. In the majority of cases (70%), the ads-supported and paid versions of an app are identical in terms of features. Therefore paid apps never offer less features than their corresponding ads-supported versions, which is expected.

> *Paid apps do not usually include more features than their corresponding ads-supported versions.*

RQ 6: What are the ad networks used?

Motivation: Ads-supported apps use ad networks, which allow developers to include ads in their apps by providing API and content. Some ad networks, such as AdMod, offer ads mediation[7], which lets developers serve ads to their apps from multiple networks increasing the monetization of their apps by sending ad requests to the ad network with the highest bid. When using ad networks, developers set a refresh rate defining how often ads are reloaded. The lower the value the more often are ads reloaded. We investigate the numbers of ad networks and the values of the refresh rates used by developers in their apps.

Approach: We analyze the numbers of ad networks used by apps using Appbrain Ad Detector, which reports over 70 different aspects of apps, including which ad networks are embedded in apps. For each app, for both their ads-supported and paid versions, we retrieve the ad networks that they use. Also, we run each ads-supported app for three minutes and we measure the time between each ad reload to determine their refresh rates.

Results: The number of ad networks used by ads-supported apps is in the range $[1, 6]$ (average 1.80). The ad network AdMob, which is powered by Google, is used in all the ads-supported apps with 11 of them (55%) using this ad network exclusively.

> *The ad network provided by Google is the most used in ads-supported apps, although developers sometimes include additional ad networks.*

Considering paid apps, among the 20 apps used in our study, six paid apps (30%) contain ad networks and these networks are identical to those in their corresponding free versions. Specifically, A01, A05, A08, A15, A16, and A19, include ad networks in their paid versions, although the official Android documentation[8] states that the usage of ad networks increases the sizes of apps and users often avoid downloading large apps[9].

> *Developers do not always remove ad networks in paid versions of their ads-supported apps, which increases the apps sizes.*

Regarding the refresh rate of ads, ad networks recommend to set it to a value in the range of $[30, 120]$ seconds, although the default value use to be 60 seconds. A zero value means that ads would only be loaded once. We observe that only three apps (15%) have null refresh rates. For nine apps (53%), developers set the refresh rate to 60 seconds while they set

[7] https://firebase.google.com/docs/admob/android/mediation
[8] https://firebase.google.com/docs/admob/android/lite-sdk
[9] https://developer.android.com/topic/performance/reduce-apk-size.html

30 seconds for seven apps (41%). Only one app (6%) has a different refresh rate, in this case of 40 seconds. Thus, most developers keep the default refresh rate proposed by the ad network[10] while few developers use the minimum refresh rate proposed by ad networks.

We estimate the average refresh rate of ads as $\lceil \frac{100}{1.88} \rceil = 56$ seconds. Where 1.88 is the average number of ads reloaded in 100 seconds by ads-supported apps. This average refresh rate value has been estimated omitting the three apps with a null value.

Most developers use the default refresh rate for ads-supported apps.

RQ 7: What is the impact of ads on permissions?

Motivation: Permissions is a mechanism that enforces restrictions on the operations that apps can perform. They are requested in the manifest file of the app. Book *et al.* [1] showed that ad networks are increasingly taking advantage of the permissions requested by the app, possibly to compromise the users' privacy. As it is said in the Android documentation[11], it is strongly recommended to minimize the number of permissions that apps request. We compare ads-supported and paid apps in terms of numbers and types of granted permissions.

Approach: Requested and granted permissions can be obtained using the Android tools `aapt` and `dumpsys`, respectively. We developed a Python script that obtains, from the `apk` file of each app, the list of granted permissions for both ads-supported and paid apps. Then, we apply the `diff` command to analyze the differences between ads-supported and paid versions of each app. Finally, we compute, for the paid apps, the numbers of granted permissions removed and the numbers of granted permissions added in comparison to their corresponding ads-supported versions.

Results: Table II shows that, for three apps (15%), the paid versions include new permissions that do not exist in the ads-supported versions. CHECK_LICENSE is used to apply license controls to apps published through Google Play. GET_ACCOUNT, MANAGE_ACCOUNTS, and USE_CREDENTIALS are permissions related to the Android `AccountManager` class for logging in with Google and validate the user. The READ_PHONE_STATE permission allows read only access to phone state and it is used in combination with CHECK_LICENSE permission for licensing validation.

For seven apps (35%), both ads-supported and paid versions have exactly the same numbers and types of permissions. Concerning paid versions, for 10 apps (50%), permissions are removed in comparison to ads-supported versions (mostly permissions related to network access and location, which are required to load ads). Specifically, for A04, A09, A11, A14, A17, and A18, the paid versions require less permissions than the ads-supported ones because the former do not need to access to the Internet, but it is required when ads are included in the free version. Table III shows the list of paid apps that grant less permissions than their corresponding ads-supported versions.

TABLE II
APPS THAT GRANT MORE PERMISSIONS FOR THE PAID VERSION.

App	Permissions added to the paid version
A02	com.android.vending.CHECK_LICENSE
A03	android.permission.GET_ACCOUNTS android.permission.MANAGE_ACCOUNTS android.permission.USE_CREDENTIALS
A10	android.permission.READ_PHONE_STATE com.android.vending.CHECK_LICENSE

TABLE III
APPS THAT GRANT LESS PERMISSIONS FOR THE PAID VERSION.

App	Permissions removed in the paid version
A01	com.google.android.providers.gsf.permission.READ_GSERVICES android.permission.READ_PHONE_STATE
A04	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
A06	com.google.android.providers.gsf.permission.READ_GSERVICES com.android.vending.BILLING android.permission.READ_PHONE_STATE com.google.android.gms.permission.ACTIVITY_RECOGNITION android.permission.ACCESS_WIFI_STATE
A09	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
A11	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
A14	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
A16	android.permission.ACCESS_COARSE_LOCATION
A17	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
A18	android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE
A20	com.android.vending.BILLING android.permission.READ_EXTERNAL_STORAGE android.permission.ACCESS_COARSE_LOCATION android.permission.WRITE_EXTERNAL_STORAGE

The READ_PHONE_STATE and the READ_GSERVICES permissions are required by some ads providers to get some information about the phone and read Google service configuration data. The BILLING permission is needed for sending in-app billing requests and managing in-app billing transactions using Google Play. Ads-supported versions of A06 and A20 need this permission because they offer the possibility of unlocking additional content. The ACTIVITY_RECOGNITION permission is required by app A06 to integrate additional ads SDK into the AdMob advertising platform. Permissions to read from and write to external storage are granted by some ads providers (as MoPub or MillennialMedia). Even if they are not mandatory, ads providers recommend them but developers can choose to use them or not.

[10] https://support.google.com/admob/answer/3245199?hl=en

[11] https://goo.gl/jckp86

In general, paid apps require less permissions than their corresponding ads-supported versions. Yet, the validation of their licensing may require extra permissions which are not needed by their corresponding ads-supported versions.

IV. Experimental Study

We answer our second high-level research question, *How is the ads-supported business model impacting performance and price?*, through an experimental study over the ads-supported and paid apps selected and shown in Table I. Past studies [2], [3] discussed the impact and hidden costs of ads on performance metrics. However, they did not support their reports using statistical tests and, therefore, their conclusions could be statistically invalid. We complement these previous studies with measures that we can analyze statistically to check if ads-supported apps are more costly in terms of performance metrics than paid apps because of the presence of ads. In addition, we propose different equations to determine the cost of free apps due to ads.

For each app, we create a simple scenario to start the app, skip the initial tutorial (if present), and wait for 100 seconds in the main activity that contains ads in the free version. These scenarios are run 30 times automatically while we collect performance metrics. Before the runs, the screen brightness is set to the minimum value and the phone is set to keep the screen on. To avoid any kind of interference during the measures, only the essential Android services are run on the phone.

We collect the following performance metrics (CPU, memory, network usages, and power consumption) as follows.

CPU usage: It is collected using the approach proposed by Gui *et al.* [3]. The `top` command is run on the phone in the background, obtaining the percentage of CPU usage associated to an Android app. Every second, this information is added to a file stored on the phone.

Memory usage: It is measured using the `dumpsys meminfo` command on the phone. Every second, this information is obtained for the process associated with the Android app and added to a file stored on the phone. We measure memory using the Proportional Set Size (PSS), which is proposed in the Android documentation[12] and which is a measure of the app's RAM use, including shared pages across processes. This metric is different from the one used by Gui *et al.* [3], where the Resident Set Size (RSS) was used, which indicates how many physical pages are associated with a process and which is less precise.

Network usage: It is collected using the `tcpdump`[13] command on the phone as in recent works [3], [6], which captures packets from the WiFi and cellular connections. We use this tool via `adb` to capture the numbers of bytes transmitted over the network connection.

Power consumption: It is measured using a digital oscilloscope, a TiePie Handyscope HS5, which offers the LibTiePie SDK, a cross-platform library for using TiePie engineering USB oscilloscopes in third party applications. We use this device because it allows higher frequencies than the Monsoon Power Monitor[14]. We set the resolution of the oscilloscope to 16 bits and the frequency to $125kHz$. Therefore, the device takes a measure each eight microseconds. The mobile phone is powered by a power supply and, between both we connect, in series, a uCurrent[15] device, which is a precision current adapter for multimeters, converting the input current I proportionally to the output voltage V_{out}. Knowing I and the voltage supplied by the power supply V_{sup}, we use Ohm's law to calculate the power usage P as $P = V_{sup} \cdot I$. The diagram of the connection is shown in Fig. 4.

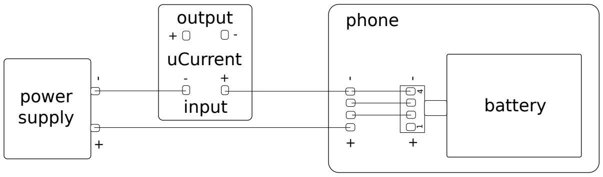

Fig. 4. Connection between power supply and the Nexus 4.

Because the phone is connected via USB to the PC to send and receive data, we disabled the USB charging on the phone. We developed a simple Android app to enable or disable USB charging in Nexus 4 phones. This app is free and available for download in Google Play[16].

Fig. 5 shows the distribution of power consumption for each app individually. We observe that the ads-supported versions are always more power-consuming than the paid ones for the considered measures. A similar trend is obtained for CPU, memory, and network usages (we thus omit the figures).

We perform a Wilcoxon rank sum test to check if the differences observed between the values of the measures of ads-supported and paid versions are significant. Our null hypothesis is that the distributions of the measures of paid apps and that of their corresponding ads-supported versions differ by a location shift of μ (the average value), expecting that paid apps have a better performance. We consider the difference to be significant if the obtained p-value is lower than 0.05. In addition, we compute the effect size using the Cliff's Delta function from the `R` software [17] when the comparison is significant.

We do not get a significant difference and the null hypothesis cannot be rejected for only one app (A01) and only one metric (memory usage). For all the other apps and measures, there are statistically significant differences with a large effect and we can reject the null hypotheses. We conclude that paid apps are more efficient in terms of CPU, memory, network, and power consumption because of the absence of ads.

[12] https://developer.android.com/studio/profile/investigate-ram.html

[13] http://www.androidtcpdump.com/

[14] https://www.msoon.com/LabEquipment/PowerMonitor/

[15] http://www.eevblog.com/projects/ucurrent/

[16] https://play.google.com/store/apps/details?id=ruben.nexus4usbcharging

[17] https://cran.r-project.org/web/packages/effsize/

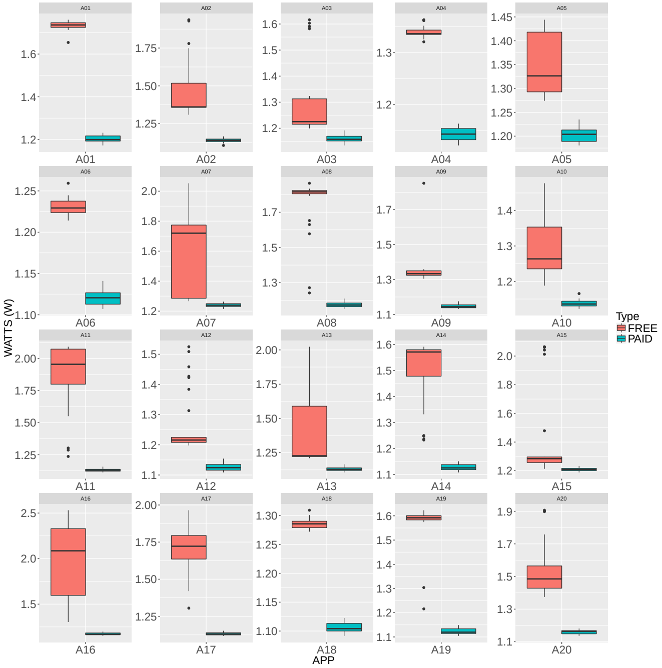

Fig. 5. Power usage by app.

Fig. 6 shows the differences in percentages for each performance metric between ads-supported and paid apps, where the symbol ◇ shows the average value. Ads-supported apps consume, in average, 5.88%, 42.15%, 93.19%, and 21.27% more than paid apps for CPU, memory, network, and power consumption, respectively.

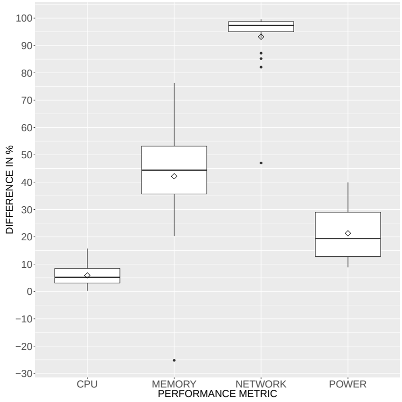

Fig. 6. Performance metric differences for ads-supported and paid apps.

Ads-supported apps use more resources than their corresponding paid versions and differences are statistically significant and the effect size large.

From the collected data, we obtain that, in average, ads-supported apps increase network usage by 1.28MB (median 0.56MB). The average data network usage is so high because A01 consumes a lot of data in its free version (average 14.43MB in comparison to the 0.06MB consumed by the paid app). Using the median value, we estimate the average network usage of each ad. Taking into account that in 100 seconds, in average, 2.6 ads are loaded, we conclude that 0.21MB $\left(\frac{0.56MB}{2.6}\right)$ is the average network usage of each independent ad. We thus estimate the network cost of ads using the price of $15 per gigabyte as provided by the AT&T company, a popular American Internet services provider. We thus determine that each load of an ad potentially costs end users $0.00315 in network charges. Concerning power consumption, in average, ads-supported apps increases it by 0.34W (median of 0.27W). Instead of power, we estimate the cost of ads in terms of energy consumption. The average difference in terms of energy consumption between ads-supported and paid apps is 36.35J (median 20.29J). Considering this fact we conclude that 13.98J $\left(\frac{36.35J}{2.6}\right)$ is the average energy consumption of each independent ad.

The average network usage and energy consumption of independent ads is 0.21MB and 13.98J, respectively.

Taking into account the average energy consumption of ads we use Eq. 1 to calculate the percentage of battery charge that is consumed by ads. E is the average energy consumption of independent ads in joules (J), and V and C are the voltage and electric charge (in mAh), respectively, of the phone battery. Replacing constants by the Nexus 4 values ($V = 3.8$ and $C = 2100mAh$), we conclude that, in average, each independent ad consumes 0.0486% of the total battery.

$$\%_{charge} = \frac{E}{V} \times \frac{1000}{C \times 3600} \times 100 \tag{1}$$

We analyze the impact of the refresh rate of ads on the real cost of ads-supported apps calculating the total network (in MB) consumed by ads in ads-supported apps using Eq. 2, where D is the running time of the app (in seconds), R_{rate} is the refresh rate of ads (in seconds), and $network_{ads}$ is the average network usage of ads in MB.

$$network = \left(\frac{D}{R_{rate}}\right) \times network_{ads} \tag{2}$$

We define a similar equation to analyze the impact of the refresh rate of ads on battery life for ads-supported apps using Eq. 3, where $battery_{ads}$ is the average percentage of battery consumed by ads, which can be calculated using Eq. 1.

$$battery_{drained} = \left(\frac{D}{R_{rate}}\right) \times battery_{ads} \tag{3}$$

In addition, given a data plan size, we calculate the time (in seconds) in which the data limit would be reached because of ads using Eq. 4, where $datasize$ is the size of the data plan in MB, R_{rate} is the refresh rate of ads (in seconds), and $network_{ads}$ is the average network usage of independent ads in MB.

$$D_{dataplan} = \frac{datasize \times R_{rate}}{network_{ads}} \quad (4)$$

There is a point in time in which the hidden costs of ads-supported apps overtake the clear costs of paid apps and, thus, ads-supported apps could be more expansive. To estimate the time in which an ads-supported app overtakes its paid version, we define Eq. 5, where $price_{app}$ is the cost of the paid app, R_{rate} is the refresh rate of ads (in seconds) in the free app, $network_{ads}$ is the average network usage of each independent ad in MB, $price_{MB}$ is the MB price of overage data, and $members$ is the number of people in the Google Family Group sharing the app.

$$amortization = \frac{price_{app} \times R_{rate}}{network_{ads} \times price_{MB} \times members} \quad (5)$$

Although the cost of network usage and battery life could be considered small, it depends on the use of the app as follows. Let us suppose that we use the ads-supported version of A11, which is a popular PDF book reader. A11 uses a refresh rate of 60 seconds. If a user uses A11 100 minutes every day for a month, Eq. 2 shows that about 630MB of data would be spent only in ads. If the data plan is limited, for instance, to 500MB, Eq. 4 shows that it would be over in 24 days and the user would be billed for 130MB that supposes \$1.95. Regarding battery life and considering Eq. 3, because the app is used every day for 100 minutes, the battery percentage would be decreased an additional 4.86% each day, for only loading ads. Using Eq. 5, and considering that A11 costs \$3.49, we obtain that if the data plan is over and the free app is used only by one member of the family group for 19 hours, the cost of data overage due to ads would be higher than the price of the paid app. If we consider that up to five users can belong to a family group, the cost of data overage would be higher than the price of the paid version in only four hours.

Depending on use and service provider, paid apps are sometimes less expensive because their costs are amortized in a short period of time.

V. Threats to Validity

Threats to internal validity concern factors, internal to our study, that could have influenced our results. We collected information about releases from Google Play and AppBrain. The latter is a public source for information about Android apps and we cannot guarantee that it does not introduce any bias. However, we compared information existing in AppBrain and Google Play for the last version of the apps used in our study and confirmed that the information offered by AppBrain was consistent. We only considered the last versions because they are the only version available in Google Play. To obtain information about the ad networks used in the apps we used AppBrain Ad Detector, which is available for free in Google Play. We verified the information offered by this app using the free app Addons Detector, also available in Google Play. Considering performance metrics, we computed the power and energy consumption using well-known theory and scenarios. In addition, we replicated several times our measures to ensure statistical validity. Concerning the cost of ads-supported apps, given that we cannot accurately simulate all users contexts (user's location and timezone) and because we do not know for how long ads are cached before they are reloaded, we chose to uninstall apps between different runs. Although this fact reduced the memory overload in the real phone during the experiments, it probably artificially increased the network usage because ads were not always cached.

Threats to construct validity concern relationship between theory and observation and the extent to which the measures represent real values. We used a Nexus 4 phone, the same model used in previous studies [6]–[10]. Our measurement apparatus had a higher or the same number of sampling bits as previous studies and our sampling frequency was one order of magnitude higher than past studies. Overall, our measures were more precise or at least as precise as those in previous studies. Network and CPU usages were collected using the tool `tcpdump` and the command `top` on the phone, respectively, which may have introduced extra energy consumption [3]. We confirmed that running `tcpdump` and `top` did not have noticeable impact on energy consumption and we extended this observation to `dumpsys meminfo`, the command used to collect information related to memory usage.

Threats to conclusion validity concern the relationship between experimentation and outcome. While the analyses related to performance metrics was supported by appropriate statistical procedures, the exploratory study was qualitative in nature and, hence, we did not use any statistical analysis. Our estimations of the hidden cost of ads and, therefore, of the cost of ads-supported apps were based on the data collected in the experimental study, which was limited to 20 apps and to the Nexus 4 phone. Further validations on a larger set of apps and–or different phone is desirable to make our findings more generic.

Threats to external validity concern the generalization of our findings. The study was limited to a reduced number of apps. We consequently must accept the App Sampling Problem [11], which exists when only a subset of apps are studied, resulting in potential sampling bias. We selected apps belonging to different categories and developed by different developers. More experiments over larger set of apps must be carried out to generalize our findings to the entire Android ecosystem.

We ensure the replicability of our study by making all collected data and scripts available in a replication package[18]. In addition, all the figures and tables omitted in the paper, because of the space limitation, are also available in our replication package.

VI. Related Work

Power consumption and data traffic of mobile apps are nowadays a hot topic given the popular use of mobile devices

[18] It will be available in June 2017.

and these issues have been addressed in many papers [6], [12]–[14], just to name a few contributions. In addition to power consumption and network usage, other metrics have been analyzed in Android apps. In [2] was presented ProfileDroid, a monitoring and profiling system for characterizing Android app behaviors at multiple layers: static, user, OS and network. In addition, the impact of ads on performance has been studied in some recent works. In [2] is observed that ads-supported versions of apps could end up costing more than their paid versions especially on limited data plans, due to increased advertising/analytics traffic. In [3] is studied 21 Android apps and they shown that the use of ads leads to mobile apps that consume significantly more network data and have increased energy consumption. They also found that these hidden costs can impact the ratings given to an app.

Permissions and its relation with ad networks has been also studied. In [1] is shown that the use of most permissions has increased over the last several years, and that more libraries are able to use permissions that pose particular risks to user privacy and security.

There are several approaches focusing narrowly on how to improve mobile ads. In [15] a prototype was implemented with prefetching and caching techniques to improve the energy consumption and network usage of ads. In [16] an ad delivery framework was proposed, which predicts user context to identify relevant ads. More recently, in [17], was proposed AD-APT, a system to avoid the adverse implications of mobile advertising on device energy and network usage. AD-APT strikes a balance between these two performance metrics refactoring ad-supported apps automatically to adjust the frequency of mobile ad occurrences at runtime.

To the best of our knowledge, the ads-business model and differences between ads-supported and paid Android apps have not been explored before. Although it is obvious the impact of ads on performance metrics it was not studied if this impact is significant or not. We do that and, in addition, we propose few equations to get an estimation about the cost of ads-supported apps.

VII. Conclusion

The Android market is a place where developers offer paid and–or free apps to users. Free apps can follow the freemium or the ads-business model . While the latter includes ads to allow developers to get a revenue, the former offers less features and the user is charged for unlocking additional features.

In this paper, we focused on the ads-business model to understand the differences in the implementation and development process of ads-supported and paid apps. We asked “Are ads-supported and paid Android apps different?”. To answer our question, we asked “How is the ads-supported business model used?” and “How is the ads-supported business model impacting performance and price?”.

For the first question, we conducted an exploratory study on a set of 40 apps from Google Play. We bought 20 paid apps and we downloaded their corresponding ads-supported versions. We observed that (i) although paid apps have a better rating ads-supported apps are preferred by users, (ii) developers do not usually offer a paid app without a corresponding free version, (iii) paid apps usually have less releases and are released less often than their corresponding free versions, (iv) there is no a clear strategy about prices set by developers to paid apps, (v) ads-supported apps do not usually include less features than their corresponding paid versions, (vi) developers do not always remove ad networks in paid versions of their ads-supported apps, and (vii) ads-supported apps usually require more permissions than paid apps.

To answer the second question, we carried out an experimental study showing that ads-supported apps use more resources than their corresponding paid versions with statistically significant differences. We estimated that the average network usage and energy consumption of ads is 0.21MB and 13.98J, respectively. Finally, based on that, we offered different equations to estimate (i) the network usage of ads-supported apps, (ii) the percentage of battery drained due to ads in free apps, (iii) the time in which a data plan is over due to the presence of ads in free apps, and (iv) the time in which an ads-supported app overtakes its paid version.

We conclude that ads-supported and paid apps are not so different because developers usually start releasing free apps and later modify these apps to release them into the marketplace as paid versions. From our observations we advise developers to: (1) remove ad networks in paid apps because they increase the app size and the number and types of Android permissions, (2) take into account the number and type of required permissions when a license validation approach is used, (3) consider the impact on ads-supported apps performance due to the refresh rate of ads.

We recommend developers to use approaches as AD-APT, which offers a choice between basic policies to adjust the frequency of mobile ad occurrences. A faster way of reducing the cost of ads-supported apps is setting the refresh rate of ad networks to higher values. Although this action makes ads-supported apps more efficient, it could have an impact on the chance of getting revenues. For this reason more studies about the trade-off between performance of ads-supported apps and revenues due to ads should be carried out as future works. In addition to this, we want to research most popular ad networks and their impact on permissions, performance, and revenues. We also plan to study and compare Android licensing validation approaches, usually used by paid apps, in terms of permissions and performance.

References

[1] T. Book, A. Pridgen, and D. S. Wallach, “Longitudinal Analysis of Android Ad Library Permissions,” *CoRR*, vol. abs/1303.0857, 2013. [Online]. Available: http://arxiv.org/abs/1303.0857

[2] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, “ProfileDroid: Multi-layer Profiling of Android Applications,” in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom ’12. New York, NY, USA: ACM, 2012, pp. 137–148. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348563

[3] J. Gui, S. Mcilroy, M. Nagappan, and W. G. J. Halfond, “Truth in Advertising: The Hidden Cost of Mobile Ads for Software Developers,” in *Proceedings of the 37th International Conference on Software Engineering (ICSE)*, May 2015.

[4] M. Harman, Y. Jia, and Y. Zhang, “App store mining and analysis: MSR for app stores.” in *MSR*, M. Lanza, M. D. Penta, and T. Xie, Eds. IEEE Computer Society, 2012, pp. 108–111. [Online]. Available: http://dblp.uni-trier.de/db/conf/msr/msr2012.html#HarmanJZ12

[5] L. L. A. Gomes, A. L. Fontão, A. J. S. Bezerra, and A. C. Dias-Neto, “An Empirical Analysis of Mobile Apps’ Popularity Metrics in Mobile Software Ecosystems,” Maceió (Brazil), Oct. 2016.

[6] R. Saborido, G. Beltrame, F. Khomh, E. Alba, and G. Antoniol, “Optimizing User Experience in Choosing Android Applications,” in *Proceedings of the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Mar. 2016.

[7] M. Linares-Vásquez, G. Bavota, C. Bernal-CÃąrdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, “Mining Energy-greedy API Usage Patterns in Android Apps: An Empirical Study,” in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: ACM, 2014, pp. 2–11. [Online]. Available: http://doi.acm.org/10.1145/2597073.2597085

[8] C. Sahin, P. Tornquist, R. Mckenna, Z. Pearson, and J. Clause, “How Does Code Obfuscation Impact Energy Usage?” in *ICSME’14*, 2014, pp. 131–140.

[9] P. Huang, T. Xu, X. Jin, and Y. Zhou, “DefDroid: Towards a More Defensive Mobile OS Against Disruptive App Behavior,” in *Proceedings of the The 14th ACM International Conference on Mobile Systems, Applications, and Services*, Singapore, Singapore, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2906388.2906419

[10] C. Sahin, L. Pollock, and J. Clause, “From Benchmarks to Real Apps: Exploring the Energy Impacts of Performance-directed Changes,” *Journal of Systems and Software*, pp. –, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121216000893

[11] W. Martin, M. Harman, Y. Jia, F. Sarro, and Y. Zhang, “The App Sampling Problem for App Store Mining,” in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR ’15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 123–133. [Online]. Available: http://dl.acm.org/citation.cfm?id=2820518.2820535

[12] M. Linares-Vásquez, G. Bavota, C. E. B. CÃąrdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, “Optimizing Energy Consumption of GUIs in Android Apps: A Multi-objective Approach,” in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2015. New York, NY, USA: ACM, 2015, pp. 143–154. [Online]. Available: http://doi.acm.org/10.1145/2786805.2786847

[13] A. Hindle, A. Wilson, K. Rasmussen, E. J. Barlow, J. C. Campbell, and S. Romansky, “GreenMiner: A Hardware Based Mining Software Repositories Software Energy Consumption Framework,” in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: ACM, 2014, pp. 12–21. [Online]. Available: http://doi.acm.org/10.1145/2597073.2597097

[14] A. Pathak, Y. C. Hu, and M. Zhang, “Where is the Energy Spent Inside My App?: Fine Grained Energy Accounting on Smartphones with Eprof,” in *Proceedings of the 7th ACM European Conference on Computer Systems*, ser. EuroSys ’12. New York, NY, USA: ACM, 2012, pp. 29–42. [Online]. Available: http://doi.acm.org/10.1145/2168836.2168841

[15] A. J. Khan, V. Subbaraju, A. Misra, and S. Seshan, “Mitigating the true cost of advertisement-supported "free" mobile applications.” in *HotMobile*, G. Borriello and R. K. Balan, Eds. ACM, 2012, p. 1. [Online]. Available: http://dblp.uni-trier.de/db/conf/wmcsa/hotmobile2012.html#KhanSMS12

[16] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft, “Breaking for Commercials: Characterizing Mobile Advertising,” in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC ’12. New York, NY, USA: ACM, 2012, pp. 343–356. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398812

[17] A. Pamboris, G. Antoniou, C. Makris, P. Andreou, and G. Samaras, “AD-APT: Blurring the Boundary Between Mobile Advertising and User Satisfaction,” in *IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft’16)*, ser. Proceedings of the 3rd IEEE/ACM International Conference on Mobile Software Engineering and Systems, vol. 1. Austin, Texas, United States: IEEE, May 2016.