Predicting User Traits From a Snapshot of Apps Installed on a Smartphone

Suranga Seneviratne[a,b]
suranga.seneviratne@nicta.com.au
Aruna Seneviratne[a,b]
aruna.seneviratne@nicta.com.au
Prasant Mohapatra[c]
prasant@cs.ucdavis.edu
Anirban Mahanti[b]
anirban.mahanti@nicta.com.au

[a]School of EET, University of New South Wales, Australia
[b]NICTA, Australia
[c]Department of Computer Science, University of California, Davis

Third party apps are an integral component of the smartphone ecosystem. In this paper, we investigate how user traits can be inferred by observing only a single snapshot of installed apps. Using supervised learning methods and minimal external information we show that user traits such as religion, relationship status, spoken languages, countries of interest, and whether or not the user is a parent of small children, can be easily predicted. Using data collected from over 200 smartphone users, specifically the list of installed apps and the corresponding ground truth traits of the users, we show that for most traits we can achieve over 90% precision. Our inference method can be used to provide services such as personalized content delivery or recommender systems for users. We also highlight privacy loss that can occur from unrestricted access to the app lists in popular smartphone operating systems.

I. Introduction

Smartphone usage is increasing and it is predicted to reach 50% of the global mobile device market by 2017 [11]. Smartphones allow third parties to develop *apps* to provide different services to users. Third party app developers publish these apps in app markets relevant to the mobile operating system, and the users can download and install them on their smartphones. Currently, Android and iOS operating systems, the two dominant smartphone platforms, together have about 91% share of the global smartphone market [9]; the official markets for Android and iOS are reported to have more than 800,000 apps each, and approximately 51 billion app downloads were predicted for the year 2013 [20].

Users decide to install apps depending on their requirements. For example, someone using local transit services is likely to install apps that provide transit schedules. Thus intuitively, the apps that a user has installed are potentially good indicators of their life style and interests.

In the two most popular smartphone operating systems, Android and iOS, apps need explicit permission to access personal data such as location, call logs, SMS, and social network profiles. The permission to access this information is requested from the user at the time of installation in Android and when an app attempts access the information for the first time in iOS. If the user does not wish to grant the permission, he/she can decide not to install that app or deny the permissions when prompted. Further, at any point in time users can check the permissions given to an app and decide to keep or uninstall the app.

In contrast, the list of apps installed on a user's smartphone can be obtained without user permission through any installed app in Android. Some advertisment libraries have reportedly embedded this feature to collect information about installed apps [13]. Although not as straight forward as in Android, iOS also allows apps to obtain the list of installed apps [2].

In this paper, we show that a single snapshot of installed apps can be used to predict various user traits with high accuracy. On one hand, this is a viable user profiling method that can be used to deliver personalized services without continuously tracking users' online activities or smartphone usage and maintaining historical records. On the other hand, giving access to the users' app lists to any app developer as it is done today in popular smartphone operating systems, poses privacy risks to the users.

We make the following contributions in this paper.

- We present the basic characteristics of user installed smartphone apps using three large datasets comprising over 9000 smartphone users.

- We show that various user traits can be predicted using only a snapshot of installed apps and minimal external information.
- We evaluate our methodology by using real-life data of over 200 smartphone users and show that presence of certain apps are strong indicators of some user traits.

Our paper is organized as follows. Section II discusses related work. Datasets we use are described in Section III. Section IV presents the methodology we used and Section V describes the evaluation of the predictions. Section VI discusses the implications of the findings and Section VII concludes the paper.

II. Related Work

Mining of temporal smartphone usage data has received much attention in recent years, with various end goals such as predicting users' interests, moods, future application installations, and future application use [8, 18, 21, 28, 22, 27]. Chittaranjan et al. [8] showed that smartphone users' personality can be predicted by applying data mining and machine learning techniques to their app, call, and SMS usage logs. LiKamWa et al. [18] used logs of websites visited, apps used, SMS logs, voice call logs, and email logs to infer users' mood. Shepard et al. [26] and Böhmer et al. [7] studied how app usage is dependent on contextual variables such as location, time of day, and day of week; researchers have used these contextual variables to predict users' future app usage [28, 22, 27]. Pan et al. [21] demonstrated that smartphone usage logs (e.g., app installation logs, call logs, Bluetooth logs) together with externally collected information such as friendship and affiliation can be used to predict future app installation behavior of users.

Privacy leakage due to third party advertisement libraries collecting user information through *over-permission* (i.e., asking for permissions that are not required for the app to function) has also received some attention [17, 13, 25]. Leontiadis et al. [17] studied more than 250,000 apps from Google Play Store and found that 73% of the free apps request at least one permission that poses a potential privacy risk to the user. However, this information leakage is under the control of the users as they can avoid installing apps which ask additional permissions.

User demographic prediction based on monitoring the web browsing history from server side [15] or from client side [12] has been explored previously. Hu et al. [15] predicted demographic attributes, age and gender by observing click-through logs of a large scale web site for a period of one week. Goel et al. [12] showed that it is possible to infer demographic attributes such as gender, age, ethnicity by observing client side browsing logs of 250,000 users for a period of one year.

In this paper, we focus on inferring user traits based on a single snapshot of the apps installed by a smartphone user, in contrast to some of the aforementioned works that require tracking and/or collecting user activity logs over a period of time. Our approach is complementary to that of Kosinski et al. [16] wherein user attributes are predicted using a single snapshot of Facebook "likes".

III. Datasets

Our objective is to identify user traits by taking a single snapshot of apps installed by smartphone users. In this paper, we focus on the following five user traits: religion, whether the user is single ('is single'), whether the user is a parent of small children under the age 10 ('is a parent'), mother tongue, and countries of interest to highlight the viability of the proposed method.

Evaluation of our user trait classifiers requires data from smartphone users, in particular the list of apps, installed by a user and the ground truth of the users' traits of interest. To this end, we collected a dataset of lists of apps and the ground truth user traits of the corresponding users using an Android application *Apptronomy* [24] as discussed in Section III.A. Additionally, we collect complementary datasets to evaluate the representativeness of our Apptronomy dataset, which are described in Section III.B. We conclude this section with a high-level characterization of our datasets.

III.A. Apptronomy dataset

Apptronomy is an Android application, which upon installation, lists and uploads the *installed apps* to a server and generates a random ID for that installation instance. We distributed *Apptronomy* among a group of volunteers and users recruited through Amazon Mechanical Turk [1]. From these participants, we collected their basic traits through a brief questionnaire. The questionnaire had 19 questions such as gender, age group, relationship status, language and users responded with the random ID generated by *Apptronomy*. Out of the 369 users who installed *Apptronomy*, 231 users answered the questionnaire. Some of the users did not answer some of the questions. For each

Table 1: Apptronomy dataset

User trait	Number of users
Country (Origin and Residence)	194
Language (Non-English mother tongue)	48
Is single?	195
Religion (Christianity, Islam, Hinduism, Buddhism)	79
Is a parent? (Of a child aged under 10)	229

Table 2: Summary of the datasets

	Appbrain	Appaware	Apptronomy
# of users	8653	841	369
# of apps	85770	24254	6341
# of installations	705004	94024	15710
Average # of apps/user	81	112	43
Median # of apps/user	51	75	34

trait we predict, we only considered those users for whom we have the corresponding ground truth traits from the survey (Table 1).

III.B. Crawled datasets

We crawled two popular *Social App Discovery* sites *Appbrain* [6] and *Appaware* [5], where users publicly share installed app lists. These web sites act as alternative app market places for Android and allow different methods of managing apps on a user's smartphone. One such option is to publicly share the installed apps so that new apps can be discovered through friends.

Table 2 provides the summary of the three datasets and we do a basic characterization of the datasets in the proceeding sub section.

III.C. Basic characteristics

Figure 1 shows the cumulative distribution function (CDF) of the number of apps per user for each dataset. More than 50% of the users had more than 20 apps installed. There is a significant difference in the number of apps between the two crawled datasets and the *Apptronomy* dataset. We attribute this to the difference of the users in the datasets. Users who use alternative app markets and share their app lists online are possibly more active smartphone users, who potentially can have higher number of apps compared to ordinary users. The average number of apps in the Apptronomy dataset is 43 and it corroborates what was reported in a Nielsen report [4], in particular that in 2012 an average US smartphone user had around 41 apps.

A noticeable difference is observed between Appbrain and Appaware in terms of number of apps. One possible reason for the difference between Appbrain and Appaware data is the length of time they have been in operation: Appbrain site was launched in 2010, and the Appaware site was launched in 2012. The older the site, the higher the possibility of having app lists which have not been synched with the website for a long period of time.

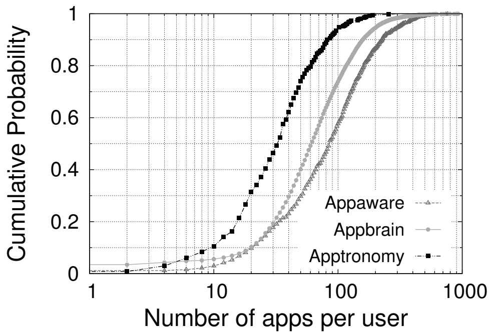


Figure 1: Number of apps per user

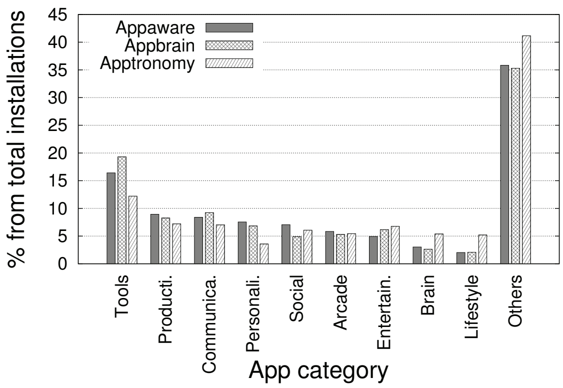


Figure 2: Percentage of installations by category

For each app in the three datasets, we queried Google Play Store and found the assigned app category. For the apps not found in Google Play Store we queried alternative app markets to obtain the category. Google Play Store and other app markets which get synched with it, categorize applications into 30 categories. For each dataset, we calculated the percentage of total installations for each category. Figure 2 shows the categories which had more than 5% share in at least one dataset.

Tools, *Productivity* and *Communications* are the top three categories across all three datasets and these three categories accounted for approximately 25-35% of the total installations. Across all three datasets, categories *Medical* and *Comics* had the lowest percentage of installations.

We also noticed that a number of users had apps that are not present in Google Play Store (29.8% in Appbrain, 3.7% in Appaware and 14.4% in Apptronomy). There can be multiple reasons for this, such as developer discontinuing the application, users using alternative app markets than Google Play Store, or Google removing the app from the market.

We classified the apps in the three datasets, into whether they are free or paid by querying Google Play Store. We found out that 89%, 85% and 90% apps in Appbrain, Appaware and Apptronomy datasets respectively, were free. Figure 3 shows the cumulative distribution function of the price of the paid apps found in our datasets. 50% of the paid apps cost less than AU$2.50.

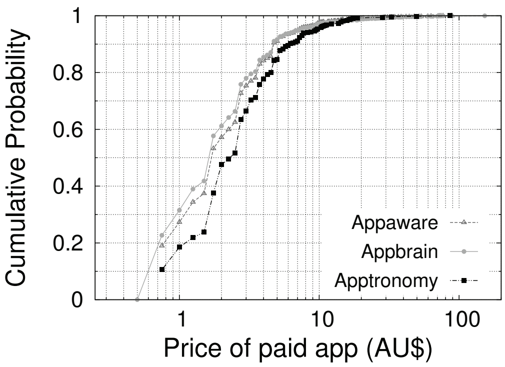
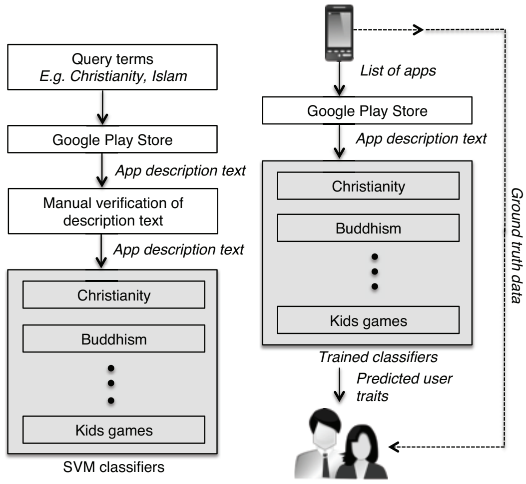


Figure 3: Price of paid apps in AU$



(a) Building SVM classifiers (b) Performance evaluation

Figure 4: Schematic diagram for training and evaluating the SVM classifiers

IV. User Trait Classifiers

For the user traits, 'is single' and 'is a parent' we trained binary SVM (*Support Vector Machine*) [10] classifiers which take the app description as the input and predict whether or not the given app is relevant to that particular trait. For the trait religion, we trained four binary SVM classifiers, one for each sub category: *Christianity, Buddhism, Hinduism and Islam*. This methodology is described in detail in Section IV.A and illustrated in Figure 4.

For the language prediction we used an external natural language processing API, and for country of interest prediction we used a publicly available list of country-wise popularity of apps. Language and country predictions are detailed in Sections IV.B and IV.C, respectively.

IV.A. SVM Classifiers

IV.A.1. User traits & pre-classified apps

To train and evaluate classifiers, it is necessary to first build a labeled dataset containing *positive examples* (i.e., apps that belong to a particular trait) and *negative examples* (i.e., apps that are not related to that particular trait). In this section, we describe the method used to identify positive and negative examples for the six binary SVM classifiers mentioned earlier.

For user traits religion, 'is single', and 'is a parent', we first identified positive examples by manually searching Google Play Store. For the religion trait, we focused on the following religions: *Christianity, Buddhism, Islam and Hinduism*. For each religion, we searched apps in Google Play Store by religion name, and from the results we selected the top-50 apps with an English app description as positive examples, following a manual inspection of the app description text for relevance. We applied a similar method for other traits. For the trait 'is single' we searched for the term *dating* and for 'is a parent', we searched for the term *kids games* and selected the top-50 apps that had English app descriptions and passed our manual verification process. After this process, we had 300 apps and the corresponding app descriptions manually classified as positive examples for traits religion (*Sub categories: Christianity, Buddhism, Islam and Hinduism*), 'is single', and 'is a parent', with 50 apps for each trait.

While the apps related to traits/sub categories other than the intended trait/sub category can be used as

Table 3: User trait classifiers performance
(* no of terms selected)

	Precision			Recall		
	10*	50*	100*	10*	50*	100*
Christianity	91%	100%	100%	77%	69%	69%
Buddhism	100%	92%	100%	85%	85%	92%
Hinduism	100%	100%	100%	100%	100%	100%
Islam	100%	92%	100%	92%	92%	85%
Dating	100%	100%	87%	77%	92%	100%
Kids games	92%	86%	92%	85%	92%	92%

negative examples (e.g., Buddhism, Islam, Hinduism, dating, and kids games apps as negative examples for the sub category Christianity), it may not be sufficient due to the diversity of the app space. Thus we selected the top-50 pre-installed apps, which are described in Section V, the top-25 paid apps and the top-25 free apps, which do not correspond to any of the intended traits as an additional source of negative examples.

For each trait/sub category, we selected the description text of the 50 positive examples and 350 negative examples and pre-processed the text using the standard text mining techniques of removing stops words, stemming the words and represented each app as a *tf-idf* weighted term vector. Then we used 70% of data for training and 30% for testing the SVM classifiers using machine learning tool Weka [14]. The SVM implementation in Weka uses sequential minimal optimization [23] and we selected a linear kernel in our classifiers, as linear kernels are reported to perform better for text mining tasks [19].

IV.A.2. SVM classifier performance

We evaluated the performance of the classifiers when top-10, top-50 and top-100 terms are selected based on the value of *information gain* [29] using the *precision* and *recall* metrics.

Let N_i be the number of predicted apps to be associated with trait i and let M_i be the number of apps actually associated to the trait i from the predicted apps. Let P_i be the number of apps associated with trait i in the testing dataset. Then the metrics *precision* and *recall* for user trait i are defined as:

$$Precision_i = \frac{M_i}{N_i} \qquad Recall_i = \frac{M_i}{P_i}$$

Table 3 summarizes these results. The classifiers have high precision in all the cases and a minimum recall of 69%. For the remainder of this paper, we report results of the classifiers with top-10 terms that show over 90% precision and over 75% recall for all the traits.

IV.B. Language Classifier

From the app description text, it is possible to identify the language of the app. There are various web services and software libraries that perform the language detection when text is given as an input. We used the Detect Language API [3] to identify the language of the app description text together with an associated confidence value.

IV.C. Country Classifier

For country classification, we used a list of the most popular apps in 23-countries, published by the Appbrain website [6]. Typically, these country-specific apps are from banks, ISPs, mobile operators, TV channels, and supermarkets of the country. We checked whether there were overlaps in the apps among the countries considered, and found no overlap when top-50 apps of each country were selected, and found overlap of less than 5 apps when the top-75 apps were selected. We ignored these overlapping apps from our analysis.

We checked for the presence of these apps in user app lists and if there was a match we report that the country is one of the countries of interest of that user. We checked the effect of the knowledge of top 25, 50, and 75 apps in this prediction.

While similar SVM classifiers can be trained for each country, it needs caution as in some countries English is not the most popular language. Thus the text pre-processing needs to be done for the language of the chosen country.

V. Performance Evaluation

V.A. Prediction & performance metrics

We used the classifiers explained in Sections IV.A, IV.B and IV.C to predict the traits of the users in *Apptronomy* dataset, for which we know ground truth information. For each user, we took the lists of installed apps, and for each app, the description text from Google Play Store was extracted. We excluded the pre-installed apps such as *Facebook, Twitter, vendor/operator specific apps* throughout this evaluation, by generating a list of pre-installed apps by examining 5 popular phone models from 5 different manufacturers covering 3 major Android versions.

The app description text was provided as the input to all the classifiers. If any of the classifiers provided a positive output for an app, we tagged the user with that particular user trait. For example, if a user is found to have an app identified as *Christianity* we predict the user to be a *Christian* and validate it with the ground truth information.

For each user trait, we made this decision at three threshold levels: presence of a single app is enough to make the decision (≥ 1), two or more apps must be present to make the decision (≥ 2) and three of more apps are required to make the decision (≥ 3). This process is summarized in Figure 4b. We used *precision* and *recall* metrics on a user trait basis to evaluate the performance of the predictions.

V.B. Results

Table 4 summarizes the performance of predictions. For user traits language, country, and religion we could achieve over 85% precision and over 20% recall for at least one of the threshold conditions. We could identify 10% of the users who revealed their relationship status as single at a 100% precision when we set the threshold criterion to two or more. User trait 'is a parent' had a low performance with 100% precision but with a 7% recall.

In general, as the number of apps required for evaluation increased, precision increases and recall decreases. Intuitively, as the number of apps matching to a user trait increases, the probability of a user actually having that trait also increases. However, the amount of change in precision and recall according to the threshold number of apps differs significantly. For example, for the language trait, when the threshold is moved from 1 to 2, precision increases by 24% while recall decreases by 8%. This is because if someone speaks a language other than English, he or she is more likely to have more than one app of that language. Thus recall decreases less. On the other hand, a user can have some non-English apps depending on the manufacturer of the phone or from where it was bought. These apps may falsely lead to the user being identified as a speaker of that language. As mentioned before, we removed such apps by compiling a list of pre-installed apps. However, this list is not exhaustive as it is not practical to cover all manufactures and operators. Further, the language detection API may not be perfect and it may misclassify given text to wrong languages. As a result, for the user trait language, a decision threshold of greater than 2 performs better as it eliminates these false positives.

In contrast, for the religion trait, recall drops by 19% when threshold criterion is changed from 1 to 2 with corresponding improvement of 10% in precision. This is because many users have only one app related to their religion and if that is ignored there is no way of figuring out the religion.

Table 4: Performance evaluation

	Precision			**Recall**		
	≥ 1	≥ 2	≥ 3	≥ 1	≥ 2	≥ 3
Language	62%	86%	82%	33%	25%	19%
Country						
Top-25	97%	100%	100%	17%	8%	5%
Top-50	98%	96%	94%	29%	12%	7%
Top-75	40%	63%	68%	37%	15%	9%
Religion	90%	100%	100%	24%	5%	3%
Is single?	70%	100%	100%	26%	10%	2%
Is a parent?	53%	78%	100%	26%	10%	7%

The performance of the trait 'is a parent' is hampered by the presence of apps that are tagged as kids games but are also popular among the adults. Such games are difficult to identify by the text description.

VI. Discussion

In this paper, we demonstrated how five basic user traits can be inferred by observing only a single snapshot of the installed apps of a user. There are a number of implications, both positive and possibly negative, of our findings.

The predictions can be used in applications where a user profile is needed such as micro-targeted advertising, user interface personalization, and recommendations of various kinds. The knowledge of the apps installed on a user's smartphone can be seen as means of instantly building a user profile. This approach to building a user profile compares favorably to user tracking techniques which can be expensive, time consuming, and perceived as being intrusive since these approaches monitor user activities across web sites or apps through the use of cookies or unique device identifiers. Our method could potentially be used in conjunction with user tracking approaches to address the *cold-start* problem in the tracking-based systems wherein an accurate user profile cannot be built until sufficient amount of data is collected over a period of time. Furthermore, our methodology can be readily extended to a range of other user traits by manually identifying a limited number of related apps from the smartphone app market. Some examples of such user traits are listed in Table 5.

Our classification method is effective when users have a diverse set of apps installed. For users with only the pre-installed apps and a very limited number of popular apps, our technique is less effective. This is the reason behind the lower recall values of the predictions. We believe that at best this is a temporary limitation; as users' familiarity with app markets in-

Table 5: Examples of other user traits

User trait	Example apps
Health conditions	Diabetes Diet, Stress Check by Azumio
Sexual Orientation	NearOx, DISTINC.TT
Education	ACCA Student Planner, CSAT UPSC Prep, Engineering Dictionary
Gender of children	Baby Doll House, Little Girl Salon
Utility profile	Bank, Insurance, ISP, Electricity etc. company apps
Interests	**Sports:** The Official ESPNcricinfo App, ESPN FC **Music:** Instrumental Hip Hop Rap Beats, Jazz Radio

creases, majority of the users are likely have a diverse range of apps. Further, our method is flexible to predict a range of user traits and therefore even if one trait fails for a user, there is the possibility that some other traits can be inferred correctly. For example we could predict at least one of the five traits correctly, for 42% of the users in the *Apptronomy* dataset when threshold 2 was selected for language, 1 was selected for traits religion, 'is single', 'is a parent' and Top-50 apps were selected for country with threshold 1.

In this work, we considered user traits that have direct associations to individual apps. It might be possible to predict another set of user attributes such as age, gender, and ethnicity by observing installation patterns at a larger scale. For example, if users in certain age groups tend to install more apps of certain categories than others, this information could be used to build additional classifiers. However, this approach requires obtaining the ground truth and the list of installed apps for a large number of users, whereas the method proposed requires only the information about a limited number of pre-classified apps to predict traits of an individual user.

The ability to successfully decipher user traits from a single snapshot of the list of installed apps poses privacy questions in today's smartphone ecosystem. For instance, explicit permissions are not required to access the list of installed apps on smartphones running Android, which raises the question of misuse by third-party apps and advertisement libraries. In the interest of protecting user privacy, smartphone environments such as Android and iOS should look to introduce more user-level control over the release of information pertaining to the list of installed apps.

VII. Conclusion

The paper presented, to the best of our knowledge, the first study of the use of installed apps on smartphones to infer user traits. By using the ground truth of over 200 smartphone users, we showed that certain user traits can be predicted with precision as high as 100% by only observing a single snapshot of the apps installed in the smartphone and a limited number of pre-classified apps.

The SVM classifier based methodology proposed in the paper is flexible and can be extended to infer various other user traits with minimal effort. We believe that lists of installed apps can be effectively used to infer user traits quickly and accurately, and these inferences can be used to drive applications such as user interface personalization, recommendations, and advertising, provided privacy threats are adequately managed.

References

[1] Amazon Mechanical Turk. `https://www.mturk.com/mturk`.

[2] iOS App detection. `http://www.ihasapp.com`.

[3] Language detection API. `http://detectlanguage.com`.

[4] Appnation: What a difference a year makes. `http://www.nielsen.com`, 2012.

[5] Appaware. `http://www.appaware.com`.

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

[7] 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 the 13th international conference on Human computer interaction with mobile devices and services*, pages 47–56, 2011.

[8] G. Chittaranjan, B. Jan, and D. Gatica-Perez. Who's who with big-five: Analyzing and classifying personality traits with smartphones. In *Proc. International Symposium on Wearable Computers*, pages 29–36, 2011.

[9] A. Cocotas. Android grabs a record share of the global smartphone market. `http://au.businessinsider.com`, 2013.

[10] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[11] eMarketer. Smartphone adoption tips past 50% in major markets worldwide. `http://www.emarketer.com`, 2013.

[12] S. Goel, J. M. Hofman, and M. I. Sirer. Who does what on the web: A large-scale study of browsing behavior. In *Proc. of the 6th International Conference on Weblogs and Social Media*, pages 130–137, 2012.

[13] M. C. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi. Unsafe exposure analysis of mobile in-app advertisements. In *Proc. of the 5th conference on Security and Privacy in Wireless and Mobile Networks*, pages 101–112, 2012.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[15] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen. Demographic prediction based on user's browsing behavior. In *Proc. of the 16th international conference on World Wide Web*, pages 151–160, 2007.

[16] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *National Academy of Sciences of the United States of America*, 110(15):5802–5805, 2013.

[17] I. Leontiadis, C. Efstratiou, M. Picone, and C. Mascolo. Don't kill my ads! balancing privacy in an ad-supported mobile application market. In *Proc. of HotMobile - 13th Workshop on Mobile Computing Systems and Applications*, 2012.

[18] R. Likamwa, Y. Liu, N. D. Lane, and L. Zhong. Moodscope: Building a mood sensor from smartphone usage patterns. In *Proc. of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, pages 389–401, 2013.

[19] L. M. Manevitz and M. Yousef. One-class svms for document classification. *The Journal of Machine Learning Research*, 2:139–154, 2002.

[20] mobiThinking. Global mobile statistics Section E: Mobile apps, app stores, pricing and failure rates. `http://mobithinking.com`, 2013.

[21] W. Pan, N. Aharony, and A. Pentland. Composite social network for predicting mobile apps installation. In *Proc. of the National Conference on Artificial Intelligence*, pages 821–827, 2011.

[22] A. Parate, M. Böhmer, D. Chu, D. Ganesan, and B. M. Marlin. Practical prediction and prefetch for faster access to applications on mobile phones. In *Proc. of the International Joint Conference on Pervasive and Ubiquitous Computing*, pages 275–284, 2013.

[23] J. C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. 1999.

[24] S. Seneviratne. Apptronomy. `https://play.google.com/store/apps/details?id=com.test.apptronomy`, 2013.

[25] S. Shekhar, M. Dietz, and D. S. Wallach. Adsplit: Separating smartphone advertising from applications. In *Proc. of the 21st USENIX Conference on Security Symposium*, pages 28–28, 2012.

[26] C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum. Livelab: Measuring wireless networks and smartphone users in the field. *SIGMETRICS Perform. Eval. Rev.*, 38(3):15–20, Jan. 2011.

[27] Y. Xu, M. Lin, H. Lu, G. Cardone, N. Lane, Z. Chen, A. Campbell, and T. Choudhury. Preference, context and communities: A multi-faceted approach to predicting smartphone app usage patterns. In *Proc. of the International Symposium on Wearable Computers*, pages 69–76, 2013.

[28] T. Yan, D. Chu, D. Ganesan, A. Kansal, and J. Liu. Fast app launching for mobile devices using predictive user context. In *Proc. of the 10th International Conference on Mobile Systems, Applications, and Services*, pages 113–126, 2012.

[29] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of the 14th International Conference on Machine Learning*, pages 412–420, 1997.