Is This You? Identifying a Mobile User Using Only Diagnostic Features

Anthony Quattrone, Tanusri Bhattacharya, Lars Kulik, Egemen Tanin, James Bailey
The University of Melbourne
quattronea,tbhattachary,lkulik,etanin,baileyj@unimelb.edu.au

ABSTRACT

Mobile smart phones capture a great amount of information about a user across a variety of different data domains. This information can be sensitive and allow for identifying a user profile, thus causing potential threats to a user's privacy. Our work shows that diagnostic information that is not considered sensitive, could be used to identify a user after just three consecutive days of monitoring. We have used the *Device Analyzer* dataset to determine what features of a mobile device are important in identifying a user.

Many mobile games and applications collect diagnostic data as a means of identifying or resolving issues. Diagnostic data is commonly accepted as less sensitive information. Our experimental results demonstrate that using only diagnostic features like hardware statistics and system settings, a user's device can be identified at an accuracy of 94% with a Naive Bayes classifier.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Models; H.2.8 [**Database Applications**]: Data Mining

Keywords

Mobile Privacy; Predictive Modeling; Inference Attacks; Mobile Analytics

1. INTRODUCTION

Modern smart phones capture a great amount of personal information about a user across a variety of different data domains. Extractable diagnostic features could be collected on a regular basis by a large number of mobile apps by the fact that many smart phones have Internet access.

Mobile app marketplaces such as Google Play and Apple App Store are convenient for both the application developers and the mobile users providing centralized services for downloading third party applications. This has led to an explosion of mobile application development and their usage [11]. Many of these applications capture raw data from a user's device and upload it to a remote database in order to deliver certain services. While these applications can provide significant benefit to the users, they can also impose potential risk to disclose sensitive user information.

In smart phone applications, location data collected via GPS, Wi-Fi, RFID or Bluetooth sensors is considered as the most sensitive information causing the most severe privacy risks [8, 5, 14, 16]. Sensitive personal data can also be captured through camera, microphone, accelerometer sensors installed in smart phones. There is also other seemingly less-sensitive information such as hardware statistics or system settings that could be easily accessed. These features can be easily extracted by a mobile application like Device Analyzer [1].

In this paper, we have analyzed the Device Analyzer dataset [15] to see what features are important in order to identify a user's device other than the obvious sensitive information. To make the data more accessible for our analysis, we first transformed the raw dataset to a aggregated dataset to provide context about each user at a daily level. A web application has been developed as a part of this aggregation process to describe the daily level context of a Device Analyzer user.

We have modeled a Naive Bayes classifier to learn a user's device using less sensitive features such as hardware statistics. Our experiment shows that using only information like manufacturer name, internal and external memory usage and system settings, a user profile can be predicted at an accuracy of 94%. Only three consecutive days of monitoring diagnostic features are necessary to identify a user profile, a time period that is short for normal app usage. A mobile app has access to direct features that can uniquely identify a device such as a WIFI mac address, however diagnostic information used in our experiments is less suspected in posing as a private threat and more widely distributed to remote servers. For example, a mobile hardware manufacturer company may have access to the usage of harware statistics of its customers for analysing the performance.

This finding is a threat to user privacy as an adversary could learn the identity of a user profile given they have access to an additional dataset that contains the user's name or if a user moves to pay for a service and reveals their name to complete a transaction. Once the identity of a user is

MUM '14, November 25 - 28 2014, Melbourne, VIC, Australia
Copyright 2014 ACM 978-1-4503-3304-7/14/11 ...$15.00.
http://dx.doi.org/10.1145/2677972.2677999

known, this could lead to further intrusions. For example, a user may be targeted with unsolicited marketing.

Our main contributions are as follows:

- We provide an approach to aggregate the dataset at a daily level.
- Our experiment on the aggregated data demonstrates that diagnostic features of a mobile device such as hardware statistics and system settings can be sufficient to predict the user.
- Using a Naive Bayes classifier, we obtain a accuracy of 94% to predict a user's device from less sensitive mobile information.

2. RELATED WORK

Smart phones are becoming the number one convergence device that stores most of a user's personal data. Information privacy in pervasive computing is a established research discipline with roots dating back to the early 1980s when PCs were gaining mainstream adoption. There is now growing concern of the privacy implications associated with smart phones.

Traditionally smart phones have been used for location based services and social networking applications. As a result considerable research has been performed to protect user privacy in location sharing applications [8, 5, 14, 16]. For example, Consolvo et al. have discovered the three important factors, why, what and when for sharing location information with service providers [8]. In [14], a privacy-aware location sharing application, called *Locaccino*, has been developed. Anthony et al. have studied whether users' preferences to share location information vary with respect to place or social context [5].

Research into the privacy implications of third-party app usage is a relatively new topic. Automated systems have been proposed for both Android and iOS to detect privacy leaks at an API method call level. While these systems are very useful in detecting where sensitive information is leaked, they do not indicate if it is justified given the functionality of the app.

With the advancement of smart phone applications, people are becoming more concerned about the privacy of other sensitive data in their phones, for example photos, contacts, etc. According to the survey performed by Ben-Asher et al., the sensitivity of mobile data depends on the data type and the context of use [6]. Chin et al. have performed a survey on 60 smart phone users to determine their attitudes towards security and privacy. They found that people are more concerned about privacy on the smart phones than their laptops [7].

A real-time privacy monitoring application, called *TaintDroid* was recently developed for smart phones. According to their result, *TaintDroid* could identify misusage of users location and device identification information for 20 applications out of 30 popular Android applications [11].

The iOS platform relies on a strict auditing process to protect their users as opposed to a permissions system. A system called PiOS [10] gained much attention demonstrating the ability to deconstruct an iOS application and demonstrate where privacy leaks occur. PiOS found that most of the applications that were analyzed on iOS do not leak much personal information, however more than half leaked the device ID.

Another system called Stowaway was demonstrated in [12] which compares method calls made by an app relative to the permissions the app developers request in the Android Manifest. Interestingly it was found that one third of applications are not following the least privilege path with their permission requests. For example, a developer may request fine grained location access but only actually use coarse grained access. This is likely a result of developers not understanding how to request permissions correctly due to unclear documentation.

Similar work has been performed in the online privacy area by using browser configuration features to determine if a browser can be uniquely identified in [9]. It was demonstrated that given common plugins like Java and Flash are installed, 94.2% of browsers in the sample are unique.

In this work, we argue that less-sensitive mobile information such as hardware statistics and system settings can cause potential threats to a user's privacy.

3. METHODOLOGY AND EXPERIMENTS

We have used the Device Analyzer dataset [15] to see what features are important in order to identify a user's device other than the obvious sensitive information. The complete Device Analyzer dataset contains smart phone usage from over 17,000 devices. Given the nature and scale of the data, we followed a traditional data-mining approach performing preprocessing on the data, feature selection and modeling. A web application was developed to view Device Analyzer user data. Our experimentation focused on finding correlations between features and establishing their predictive power.

3.1 Preprocessing

The dataset generated from the Device Analyzer app is stored in a low-level detailed format. To make the data more manageable, we parsed the data extracting all key and value pairs and aggregated it to a daily level per handset. User profiles that did not capture memory usage were filtered out of our model.

The following process was performed:

- Iterate over every data file for each Device Analyzer user;
- Extract handset ID and date keys and create a data structure to store daily level data;
- Use the handset ID and date as a hash to store each daily data structure in a hash table;
- When a numerical feature is found, derive SUM, COUNT, AVG, MIN, MAX and update the daily data structure;
- When a categorical feature is found e.g. system settings lock mode, take the value at end of the day and update the daily data structure;

Feature	% Off	% On	% Null
System Settings Lock	52	24	24
System Settings Sound Effects	50	36	14
System Settings Device Stay On	85	9	7

Table 1: System Settings Features Distribution

- Produce an output file for each device after all data is iterated;
- Combine each output files into a single file which can be uploaded into a relational database.

3.2 Feature Selection

Our aim was to test the predictive power of diagnostic features, the following features were used for our model:

- Device Manufacturer
- Device Model
- Device Locale (Language Setting)
- Internal Memory Size
- Free Internal Memory
- External Memory Size
- Free External Memory
- System Settings Lock
- System Settings Sound Effects Toggle
- System Settings Screen Stay On Duration
- System Settings Device On While Charging

The dataset indicates that a wide variety of Android devices are being used. After preprocessing, our dataset contains 18 mobile manufacturers and 56 mobile device models.

To estimate and visualize the distribution of the continuous featues across user profiles, Kernel Density Estimation (KDE) was preformed. In Figure 1a, it can be seen that total internal device memory is more evenly distribued than internal free memory in Figure 1b. Conversely, in Figure 1c and Figure 1d, both external memory available and external memory free is well distributed. Most user profiles appear to have a similar setting for how long the screen stays on when there is no user input as described in Figure 1e. There is also a bias in the dataset to specific locales (Figure 1f).

Percentage distributions of discrete featues is presented in Table 1. Half the population of users do not specify a phone lock setting on a given day while a third of users have a lock. Sound effects are turned off be at least half the users in the population and most users allow the device to turn off while charging.

3.3 Data Modeling and Classifier

To uniquely identify a user, the model needed to classify the handset ID as it identifies the user's device. Naive Bayes was a good candidate given it has proven to be efficient when there are many classes [13].

3.4 Implementation

The parser was developed in C using libraries libcsv and uthash. After preprocessing the data, the output file was imported into a MySQL [2] relational database to allow for analysis and feature extraction.

Train/Test Split (%/%)	Accuracy (%)	Macro Avg Precision	Macro Avg Recall
70/30	93.75	0.921	0.949

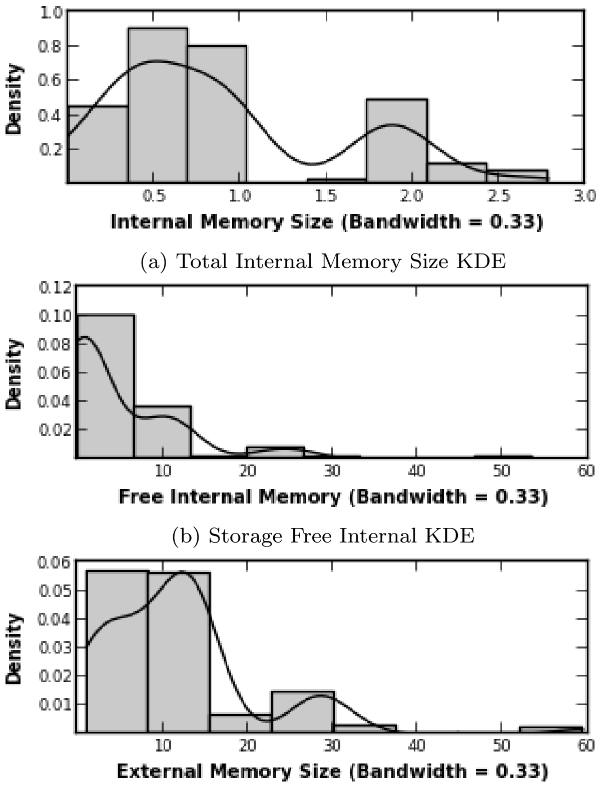
Table 2: Experimental Results Using a Naive Bayes Classifier

3.5 Experimental Results

Using only the diagnostic features described in *Feature Selection*, the model produced using Naive Bayes was accurate.

Our sample for analysis contained 223 days of data in which 66 user profiles could be uniquely identified. In the preliminary analysis we trained and tested our model on devices that contained at least two days of captured data, however we found that this was not sufficient to uniquely determine a user. Thus, only devices in the dataset with at least three days of captured data were considered for analysis. In practice three days is still a small amount of time because a mobile user will use apps for significantly longer periods which are likely to increase the accuracy of our approach further. Days in which the Device Analyzer app did not capture the external memory free feature for a device were also filtered. Numerical memory features were scaled based on the maximum for the respective feature.

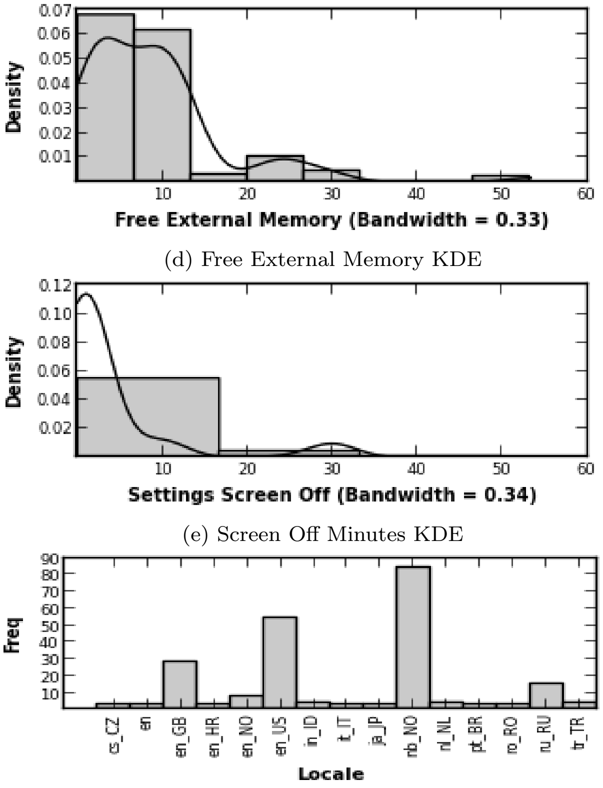
The dataset was distributed ensuring that each device has 70% of the data points as training data and evaluated on the remaining 30%. Table 2 describes the accuracy, precision and recall values of the classifier. It can be seen that a user's device can be identified at an accuracy of 93.75% with our model. The macro average of precision and recall values across all the classses are 92.1% and 94.9%, respectively.



(a) Total Internal Memory Size KDE

(b) Storage Free Internal KDE

(c) External Memory Total KDE



(d) Free External Memory KDE

(e) Screen Off Minutes KDE

(f) Locale Distribution

Figure 1: Kernel Density Estimation (KDE) plots of continuous featues

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a methodology to predicting a device ID based on diagnostic features. We also presented a web application that can be used to easily navigate through the Device Analyzer dataset and present the information in a more comprehensive visual format.

Our results indicate that when using features that would not commonly be considered sensitive are captured over a number of days, a simple Naive Bayes classifier can produce an accurate model to identify a user.

In the future we aim to determine if certain features can be used to predict other features in the dataset.

5. REFERENCES

[1] Device Analyzer Application. `https://play.google.com/store/apps/details?id=uk.ac.cam.deviceanalyzer`.

[2] MySQL Database. `http://www.mysql.com`.

[3] R-Project. `http://www.r-project.org`.

[4] Weka 3: Data Mining Software in Java. `http://www.cs.waikato.ac.nz/ml/weka`.

[5] Denise Anthony, Tristan Henderson, and David Kotz. Privacy in location-aware computing environments. *IEEE Pervasive Computing*, 6(4):64–72, 2007.

[6] Noam Ben-Asher, Niklas Kirschnick, Hanul Sieger, Joachim Meyer, Asaf Ben-Oved, and Sebastian Möller. On the need for different security methods on mobile phones. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 465–473. ACM, 2011.

[7] Erika Chin, Adrienne Porter Felt, Vyas Sekar, and David Wagner. Measuring user confidence in smartphone security and privacy. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 1. ACM, 2012.

[8] Sunny Consolvo, Ian E Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location disclosure to social relations: why, when, & what people want to share. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 81–90. ACM, 2005.

[9] Peter Eckersley. How unique is your web browser? In MikhailJ. Atallah and NicholasJ. Hopper, editors, *Privacy Enhancing Technologies*, volume 6205 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2010.

[10] Manuel Egele, Christopher Kruegel, Engin Kirda, and Giovanni Vigna. Pios: Detecting privacy leaks in ios applications. In *NDSS*, 2011.

[11] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. Taintdroid: an information flow tracking system for real-time privacy monitoring on smartphones. *Communications of the ACM*, 57(3):99–106, 2014.

[12] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 627–638, New York, NY, USA, 2011. ACM.

[13] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[14] Eran Toch, Justin Cranshaw, Paul Hankes-Drielsma, Jay Springfield, Patrick Gage Kelley, Lorrie Cranor, Jason Hong, and Norman Sadeh. Locaccino: a privacy-centric location sharing application. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*, pages 381–382. ACM, 2010.

[15] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. Device analyzer: Understanding smartphone usage.

[16] Jason Wiese, Patrick Gage Kelley, Lorrie Faith Cranor, Laura Dabbish, Jason I Hong, and John Zimmerman. Are you close with me? are you nearby?: investigating social groups, closeness, and willingness to share. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 197–206. ACM, 2011.