| **Title**          | Detecting Phantom Data Usage on Smartphones with Analysis of Contextual Information |
|-------------------|-----------------------------------------------------------------------------------|
| **Author(s)**     | Jiang, Shiqi; Zhou, Pengfei; Li, Mo                                               |
| **Citation**      | Jiang, S., Zhou, P., & Li, M. (2015). Detecting Phantom Data Usage on Smartphones with Analysis of Contextual Information. International Journal of Distributed Sensor Networks, 2015, 135150-1. |
| **Date**          | 2015                                                                              |
| **URL**           | http://hdl.handle.net/10220/39075                                                  |
| **Rights**        | © 2015 Shiqi Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. |
Research Article

Detecting Phantom Data Usage on Smartphones with Analysis of Contextual Information

Shiqi Jiang, Pengfei Zhou, and Mo Li

School of Computer Engineering, Nanyang Technological University, Singapore 639798

Correspondence should be addressed to Shiqi Jiang; sjiang004@ntu.edu.sg

Received 9 February 2015; Accepted 9 April 2015

Copyright © 2015 Shiqi Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the wide development of smartphones, mobile data usage has enjoyed rapid growth in recent years. Unfortunately many users are plagued with Phantom Data Usage (PDU), which refers to the unexpected mobile data usage that does not accord with user perception. We investigate about 400 real PDU issues and find the causes of PDU are not only the exceptions of applications, for example, software bugs or malware, but also the user’s personalized misuse. Based on the observations that each user preserves specific data usage patterns under particular environmental context, we present PDS, a PDU detection system, which automatically detects whether the current data usage is consumed as expected. Results from our evaluation experiments show that 72% of PDU cases detected by PDS are confirmed by users.

1. Introduction

Nowadays smartphones serve as not only phones but also the primary social communication and entertainment tools involving the email services, SNS, mobile games, and so forth. The number of third-party smartphone applications enjoyed years of explosive growth and many apps incur data traffic through the cellular or Wi-Fi network. Global mobile data traffic has doubled in 2012 and grows by 72% up to 23 EB in 2013. Due to the rapid technology evolution and wide coverage of cellular networks, for example, 3G and LTE, most users buy the cellular network data plan for mobile data usage. Although the per unit data cost is becoming cheaper, users are paying more money as the total data consumption grows. While more money is spent, users feel more and more confused about their mobile data bills. According to a recent survey [2], most users believe only 50% of their data bills are indeed consumed by themselves and the rest are inexplicable. Similar complaints arise in USA [3] that a number of users find that mysterious charges occur which is not noticed in time. Many users are paying for such unclear and inexplicable data usage, which is named Phantom Data Usage (PDU) in this paper.

In the users’ point of view, PDU refers to the exceptional data usage which does not accord with their normal perception. PDU is a real and emerging problem. Rather than being unusual and isolated cases, many PDU issues affect a significant number of users. When we randomly sample 100 mobile data usage issues, for 82 issues users fail to determine the causes, regarding them as PDU. Only 18 issues are due to the certain causes, for example, software bugs, malicious features of applications, or the normal heavy usage of users. In this paper, by analyzing about four hundred real PDU issues, we find that the users’ personalized misuse is a major cause of PDU. The misuse of applications is a collateral consequence of the development of smartphone industry. On one hand, the smartphone gains its popularity and smartphone users widely spread across different age groups and professions. On the other hand, smartphone applications are mostly developed by third-party or individual developers with different level of programming skills and design considerations, leading to a wide variety of applications. As a result, users might be confused across different applications, which causes the misuse problems which were not noticed by most users. Therefore it would be beneficial if smartphones can automatically tell if some applications are being misused by the certain user, in other words, detecting PDU.

Existing data usage monitors on smartphones can derive data usage information. Some third-party data monitoring applications like My Data Manager [4] can record the detailed data usage history for each app. However such tools cannot differentiate normal data usage and PDU. Large data usage
consumption does not always indicate PDU. The data usage information needs to be further analyzed to validate whether it accords with users’ usage habits. There have been some approaches [5, 6] proposed in the literature for abnormal network traffic detection and they can be potentially adapted for PDU detection. However most of them are developed to detect the network intrusion on the Internet. Directly applying those methods for PDU detection on smartphones is inappropriate due to the high uncertainty and the diversity of PDU issues. Firstly the root causes of PDU are highly different from that of the Internet. Traditional methods are mostly learning-based while the PDU characteristics are unknown. Secondly the situations on smartphones are more complicated and the usage behaviors are highly personalized, which makes it difficult to extract patterns in generic.

To the best of our knowledge, PDU issues on smartphones have been rarely studied. Inferring the occurrence of PDU from a large number of applications is challenging. First, a high data consuming app might not necessarily cause PDU. To differentiate normal data usage and PDU, we need to study how the data usage is supposed to be consumed, which is different across different applications. The basic observation is that the environmental context affects the way users use mobile applications. For instance, a Facebook user might be used to watching video at home but only browsing texts and images in office. By analyzing the traces from 7 volunteers for three weeks, we reveal that the user context (i.e., time, locations, and movements) affects not only the application usage (which application is used) but also the data usage patterns (how the application is used). Second, the context related data usage patterns are highly personalized depending on the user’s usage patterns. In this paper, we learn each user’s usage habits from the historical data usage records, extract their data usage profiles, and build a data usage model for each user.

We present PDS, a system tool to detect PDU on smartphones. PDS records the detailed data usage information of each application, the user context information (e.g., movements, time, and locations), and relevant system events. By analyzing such information, PDS detects whether the current data usage accords with users’ usage habits. Specifically, PDS profiles the data usage in different circumstances. It then compares the current traffic with that in the most similar circumstance to dig out PDU. To make it practical, PDS meets several criteria including low monitoring overhead, high detection accuracy, and flexibility to different applications and users.

We evaluate PDS on real users’ devices. We solicit 7 Android users with different smartphone experience and usage habits. In three weeks PDS detects 63 PDU cases in total and 45 of them (72%) are confirmed by users themselves. The system overhead is also carefully examined.

2. Phantom Data Usage Issues

To understand PDU issues in real world, we collect 412 data usage issues from the following forums: (1) popular Android forums including XDA-Developers and Android forums; (2) manufacture forums like Microsoft Community, Google Group, and Apple support discussions; and (3) service provider forums, for example, AT&T community. For the issues that are confirmed to be resolved, we study their causes from the replies and feedback. For the open issues whose causes are still unknown in the forums, we download and test the related applications mentioned in the complaints. We also interview 10 users from open issues. By discussion with them, we try to understand the details when PDU issues occur and study the reason why they regard the data usage as PDU. We make the following observations.

(i) Most of the data usage issues are for unknown reasons to users, regarded as PDU. Illustrated in Figure 1(a), software bugs including client bugs and server bugs contribute about 19% of sampled data usage issues. Indeed only 21% of issues are caused by malware which is generally considered as the main PDU cause by users. More than half of sampled issues are related to PDU. It indicates that PDU is a pervasive problem for smartphone users. For most data usage issues, it is difficult for users to recognize the causes and figure out the solutions.

(ii) More than 90% of PDU issues occur on popular applications. We download the 61 applications mentioned in the unknown issues. Figure 1(b) shows the majority of the applications are reputed and popular with the high download rank, including Facebook, Dropbox, and Google Map. These applications are supposed to be well-designed and ought to have no serious problems. The further test verifies our ideas. Among these apps, only 4 are problematic due to client or server bugs, which indicates the causes of most PDU are not the exceptions of applications but might be from users themselves.

(iii) The misuse of users is the major cause of PDU. We study the discussions of the unknown issues and find the potential cause is the misuse of applications. Misunderstanding and misconfiguration can result in misusing certain applications. For example, a user switches on the autoupdate feature in App Store and the data usage consumed by the autoupdate might be regarded as PDU. Note that although the features (like autoupdate) are normal and by-designed, the users might take the data consumption of them as PDU.

(iv) The occurrence of PDU issues does not follow generic patterns. We interview 10 users from 10 unknown issues. We have discussions with them and try to understand the details when PDU occurs. Different users have different application usage habits and preferences.

Now we give a clear definition of PDU. According to our study, users determine PDU when the data usage consumption does not accord with their expectations. In this paper, we borrow the concept of Quality of Experience (QoE), which is the difference between perceptions and expectations for a service, to infer PDU. The perceptions and expectations are normally developed based on many
factors, for example, word-of-mouth, past experiences, and the intensity of personal needs. We make use of QoE in a simplified way by leveraging the usage habits of individuals as a measure of user expectation. PDU is the data usage consumption that is inconsistent with the usage habits of individuals.

3. Context versus Data Usage

Before presenting the system design, we discuss our observation on context and data usage patterns. Prior works [7–9] provide the insights of the comparison of the usage time among different applications, that is, how long the user uses the application. The data usage, however, is different from the application usage time. It corresponds to the consumed network traffic which is highly related to the user’s operations on the application, that is, how the user uses the application. Different user operations result in distinctive data usage patterns. For instance, the operations on Dropbox can be divided into browsing and downloading, whose data usage profiles (Section 4.3) are highly different. As our observation reveals in this paper, the user’s operations on each application are related to the user context. To detect PDU, we have to not only study the data usage patterns of different applications but also investigate the usage patterns of each application in different context settings.

We collect the traces of 18 popular apps in 3 categories (news, games, and SNS) from 7 volunteers for three weeks using the information collector (Section 4.1). The time, cellular tower ID, and network traffic traces are recorded. The locations are labeled manually by volunteers as home, office, canteen or market, and so forth. Inspired by the prior work, we select the time, locations, and movement status as the factors. Specifically, we record the data usage traces of each app considering the user operations on the app, which are classified using the method introduced in Section 4.3. Based on the experimental study, we make the following observations.

Time. We begin with the analysis of temporal data usage patterns. Our study shows different time of day influences both the overall data usage of different apps and user’s operations on the apps. Figure 2(a) (left) plots the average amount of data usage of two apps (games and SNS) from two users. As depicted in Figure 2(a) (left), the data usage of each application is different across different time of the day and different users have different application usage habits. For user A, news apps are most likely used in the morning and SNS apps are preferred in the afternoon while user B spends less time on news app but more time on games. In Figure 2(a) (right), the two users’ operations of using the SNS app are plotted in different colors. We see users A and B have different operation combinations on the app and the distribution also changes with time.

Locations. Our study reveals that the user’s locations influence (1) which apps a user prefers to use and (2) the user operations as well. Figure 2(b) depicts the data usage of different apps at different locations for the same user. As shown in the figure, the user prefers to launch different apps in different locations. News apps are most likely launched at office and SNS apps are usually used in the classroom. All the three apps have different user operations and their distributions in different locations are also different in different locations.

Movements. We extract the user’s movement statuses from the mobile phone’s cellular tower ID variation. We use a binary value to present the movements; that is, we consider the user is static if the ambient cellular tower ID does not change and moving otherwise. In other words, we consider whether the user is at particular locations (like home and office) or moving in between them (like walking in streets or on bus). Figure 2(c) shows the data usage of three apps from 3 users. We see that all the users prefer different apps at different movement statuses. For example, all of them prefer to use news apps when static. But when moving, user A is used to
using SNS apps or playing games, but user C prefers playing games mostly.

Although we investigate the three factors separately, their influence on the user’s operations and application data usage is not independent of each other. In the following section, we jointly leverage the three factors to build a PDU detection system.

4. PDS Design and Implementation

In this section, we introduce Phantom Detection System (PDS) by describing its overall architecture and core components. PDS is designed to effectively collect and analyze data usage information. PDS leverages data usage patterns, user’s data usage habits in different context settings, which enables
PDS to evaluate whether the consuming data usage accords with users’ usage habits to detect PDU. Figure 3 shows the system overview of PDS. PDS comprises three components. The information collector samples the network traffic, the context information, and the related system events. The data usage model (DUM) is triggered by certain system events including the network status changes and application maintenance messages. Given the input context information, DUM tries to find the record with the most similar context settings from the usage history dataset which contains pairs of context settings and the corresponding data usage profile. Then the decision engine (DE) compares the corresponding data usage profile with the input network traces. DE determines the inconsistent traces as PDU and otherwise updates the usage history dataset. Next we will introduce the key techniques in each component.

4.1. Information Collector. The information collector records the time-stamped information including the network traffic, the users’ locations, and the system events. We implement the information collector on Android phones. We leverage `iptables` which is an inherent library in Android Open Source Project (AOSP). To execute it on custom ROMs, we build `iptables` for different CPU architectures including ARM v5, ARM v7, MIPS, and x86 as well. We add logging rules into input and output chains of `iptables` and capture the network traffic information via monitoring the kernel message (KMSG). As shown in Table 1, the detailed network traffic information contains (1) time stamp recorded in UNIX format. The time is further quantized into 6 intervals, that is, 8 am–12 pm, 12 pm–4 pm, 4 pm–8 pm, 8 pm–12 am, 12 am–4 am, and 4 am–8 am (2) UID. UID is the application identifier. Each application is assigned a unique ID when installed, and (3) IP address and port number. For each network operation, its source and destination IP address and port numbers are recorded. We collect the package length and its protocol type as well.

To get the locations, we leverage the cellular tower ID instead of GPS for two concerns: to achieve the low sensing overhead since GPS is energy-hungry, while at the same time avoiding using the precise locations for privacy protections. Most of commercial smartphones are able to sense the signals from several nearby cellular towers. In the implementation, we make use of top 3 cellular tower ID as the location identity. We evaluate the variance of cellular tower ID to extract the user’s movement status, static or moving.

We mainly collect two types of system events, the network status changes and application maintenance (installation and upgrade) messages. We make use of the Broadcast mechanism of Android to receive these event notifications timely and efficiently. Other mobile systems also implement similar mechanisms.

4.2. Data Usage Model. The goal of data usage model (DUM) is to find the record containing the most similar context settings with the input context information from the usage history dataset, which we regard as the optimization problem.

---

**Figure 3: Phantom Detection System (PDS) overview: once triggered, PDS processes the data collected periodically.**

**Table 1: The format of network traffic traces.**

| Time stamp | UID | In     | Out     | Src. addr. | Src. port | Dst. addr. | Dst. port | Len. | Proto. |
|------------|-----|--------|---------|------------|-----------|------------|-----------|------|--------|
| 1378709492618 | 10074 | wlan0  | 172.22.100.60 | 49132 | 111.13.12.89 | 80 | 52 | TCP   |
| 1378709492619 | 10074 | wlan0  | 111.13.12.89 | 80 | 172.22.100.60 | 49132 | 40 | TCP   |
| 1378709492619 | 10074 | wlan0  | 111.13.12.89 | 80 | 172.22.100.60 | 49132 | 40 | TCP   |
| 1378709492620 | 0     | wlan0  | 172.22.100.60 | 51220 | 155.69.24.6 | 53 | 59 | UDP   |
| 1378709492620 | 10032 | wlan0  | 172.22.100.60 | 51385 | 173.194.38.130 | 443 | 60 | TCP   |
| 1378709492621 | 10032 | wlan0  | 173.194.38.130 | 443 | 172.22.100.60 | 51385 | 52 | TCP   |
Given the context information $S^c_r$ when application $r$ is running, it contains three parts: $S^c_r = \{t^c_r, t^r_r, m^c_r\}$, where

(i) $t^c_r$ is the quantized time interval and $t^r_r \in \{T_1, T_2, \ldots, T_n\}$ which is mentioned in Section 4.1,

(ii) $L_r$ are the locations which are presented as the sequences of $(n^c_r, n^r_r, n^m_r)$ in the order of RSS where $C$ is the cellular tower ID,

(iii) $m^c_r$ is movements status which can be extracted from the variance of the cellular tower ID sensed by the mobile phones; specifically we get two types of movements, steady and riding: $m^c_r \in \{M_{steady}, M_{riding}\}$.

For each application $r$, the usage history dataset contains a list $L^h_r$ of $(S^h_r, U^h_r)$ where $S$ is the context settings and $U$ is the corresponding data usage profile. To retrieve the most similar context settings with $S^c_r$ from $L^h_r$, we minimize the object function quantifying the overall bias between two context settings:

\[
E = E_T + E_L + E_M
\]

\[
E_T = \sum_r \| t^h_r - t^c_r \|_2
\]

\[
E_L = \sum_r \sum_i idmatching(n^h_r, n^c_r, l^c_r, l^h_r)
\]

\[
E_M = \sum_r \| m^h_r - m^c_r \|_2
\]

In (la), the overall bias object function $E$ comprises three terms. $E_T$ captures the constraints that the times $t^h_r$ and $t^c_r$ must belong to the same time period. $E_L$ captures the fact the locations should be as closed as possible. In particular, we use the function idmatching [10] to compute the matching score of two cellular tower sequences. If $n^h_r$ and $n^c_r$ are the same and $l^c_r, l^h_r$ are considered matching and mismatching otherwise. Each match is assigned a positive match score according to its RSS rank (i.e., 1, 2, and 3). For each mismatch, a penalty cost ($-2$) is set. At last $E_M$ minimizes the difference between the movement statuses. The movement status should be the same. Given the input context information $S^c_r$, we can optimize (la) to get the record containing the most similar context setting $S^h_r$ and its corresponding data usage profile $U^h_r$.

4.3. Decision Engine. The goal of the decision engine is to evaluate the current data usage $U^c_r$. Based on the observations in Section 3, $U^c_r$ might be PDU if the similarity between $U^c_r$ and $U^{h,b}_r$ is low. To compute the similarity, we first profile the current data usage. In this paper, we leverage a set of statistical features of network traffic [5, 6] including (a) $L_{max}$, the maximum packet length among the records of $U$, (b) $L_{avg}$, the average packet length of the records in $U$, and (c) $F$, the frequency of records in $U$, $F = N/T$, where $T$ is the time length of $U$.

We use $(L_{max}, L_{avg}, F)$ as the profile of the given data usage. Then we define a measure of similarity between the profiles of $U^c_r$ and $U^{h,b}_r$.

\[
s_y(U^c, U^h) = 1 - \sum w_j \cdot \varphi_j(U^c, U^h), \quad j \in \{m, a, f\}.
\]

In (2), $w_j, j \in \{m, a, f\}$, is the weight which is predefined. $\varphi_j, j \in \{m, a, f\}$, is the similarity function for $L_{max}, L_{avg}$, and $F$. In our implementation, we make $\varphi_\theta = \|g_{\theta} - g_{\theta}\|$, where $\theta \in \{L_{max}, L_{avg}, F\}$.

To evaluate the similarity model, we collect the network traffic traces of 20 different operations on 8 apps. We conduct each operation twice and get 40 traces totally. As seen in Figure 4(a), the traces of same operations lead to the high similarity. However there are clearly 4 pairs of different operations apart from the diagonal gaining the high similarity as well, hence false positives. The further analysis reveals the combinations of network requests of different operations might result in the similar statistical network traffic features. For instance, operation #6 is downloading multiple documents and operation #13 is browsing the page with lots of images. They obtain the closed $L_{avg}$ and $F$.

To improve our approach, we introduce the service grouping which is based on an observation: different operations on an app trend to access different IP/Port servers. For example, operation #6 accesses the document server and the image server is visited by operation #13. The IP/port servers can be used to indicate the operations. To apply service grouping, we take the following steps. (1) Group the records of network traffic traces according to their IP address and port number. The record group with the same IP address and port number is called a service. (2) Profile the network traffic of each service. (3) Compute $s_y$ for each same service of $U^c$ and $U^h$. (4) Let the overall similarity $s_{ov}$ be the sum of the similarities of all the same services.

Seen in Figure 4(b), after introducing the service grouping, the false positives are canceled. We determine the default DE similarity threshold to be 0.8 by offline training.

If the current data usage is consistent with that in the usage history, DE updates the usage history dataset. In our implementation, we use the average of statistical features to update the corresponding data usage profiles in the usage history dataset.

5. Evaluation

In this section, we assess the effectiveness and overhead of PDS. We evaluate PDS on real user phones. We recruit 7 Android users from our university campus. The volunteers have 5 different phone models running Android system. We install PDS on the users’ phones and collect the traces and detection results for three weeks. With the permission of users, results are submitted to our server for the further analysis. PDS anonymizes all the data to ensure users’ privacy.

Practically, it is difficult to reproduce PDU issues or control the occurrences of PDU due to their high dependence on the personalized usage. Therefore we make use of the user feedback as the ground truth. The users are further separated into two groups according to their smartphone usage.
usage experience. Group #1 contains 5 experienced users. They either have rich experience in using Android devices or are Android developers. In contrast group #2 contains 2 users who are fresh to Android phones. All the volunteers are asked to label every PDU report as true or false according to the related statistical data usage information.

For group #1, we set a low DE threshold to ensure that all the suspicious data usage is reported. As depicted in Figure 5, the reported case shows all the suspicious data usage cases; the PDU case is determined by the default threshold of PDS; the confirmed case shows the cases which the user considers as PDU among the reported cases. In the three weeks, there are 200 suspicious data usage cases reported. Among them 88 cases are detected as PDU by PDS and 63 cases are confirmed as PDU by users.

Table 2 shows the further breakdown of the statistics on the results of group #1. TP stands for the case which is detected as PDU by PDS and confirmed by users. FN shows the PDU case which is not detected by PDS but is labeled as PDU by users among all the reported cases. Then the average true positive rate of PDS from these 5 users is 72.2%

Next we present the DE similarity distribution of the PDU cases confirmed by users in group #1 (true positives in Table 2). Illustrated in Figure 6, more than 80% of PDU cases obtain the DE similarity around 0.8. It is consistent with the default setting of DE similarity threshold. The result also guarantees the high quality of the labels from group #1 users.

For group #2, we record the size of the usage history dataset which not only leads to the storage overhead but also affects the execution time retrieving the context settings. As Figure 7 shows, after two weeks the size approximately reaches its maximum level and after three weeks the size approaches its plateau. For both users, the usage history dataset of an app averagely contains 67.5 records.

We measure the energy consumption on three Android smartphones: HTC G14, Nexus 4, and Samsung Galaxy S2. We operate the smartphones as usual and use a Monsoon Power Monitor to capture the battery consumption in real time. Figure 8 shows the instant battery consumption when the system is in idle state and running PDS. As shown in Table 3, PDS will consume extra 4.5% energy in the idle state with radio and WiFi open and screen closed. And for the
normal usage cases, battery life will reduce 10.2% due to the overhead of PDS.

We use Android Debug Bridge (ADB) to dump the memory usage in run time. In the case of normal use, PDS occupy 19 MB in average. When only logging, memory utilization is lower, about 12 MB. When the detection procedure is triggered, memory usage will reach around 25 MB. This level of memory occupied belongs to common usage among Android utilization tools.

6. Related Work

Application Usage Prediction. A number of studies focus on the application usage prediction. By analyzing the users' locations and their usage of applications, My Experience presents that users prefer to use Short Message Service (SMS) when moving [11]. The context information is also analyzed in [12] which finds the locations are highly related to the application usage. A large-scale study is conducted in [13]. It discusses the application usage and several major context factors, for example, time and location, and finds that the use of some apps is strongly related to previously used apps. There are also several applications based on the studies. By predicting the specific application usage, PCS [14] enables the crowdsourcing on smartphones with low energy consumption. Some approaches are also proposed to improve the smartphone performance, like developing a dynamic home screen [15] and enabling a fast launch of the predictive application [9]. While these studies successfully explore the application usage and how to leverage it, it is more important to understand the fine-grained data usage patterns for detecting PDU.

Anomaly Detection. Anomaly detection refers to the problem of finding patterns that do not conform to expected behavior. In [16], anomaly detection is applied to detect fraudulent credit card applications and fraudulent credit card usage. By monitoring calling activities on mobile phone, several anomaly detection techniques [17] are used to detect the

---

**Table 3**: Energy consumption: we collect the energy profile in four states: (1) cellular radio and Wi-Fi on, screen off, and phones in idle mode. (2) Cellular radio and Wi-Fi on, screen off, and PDS running in background. (3) Cellular radio and Wi-Fi on, screen on, and some other applications running normally. (4) Cellular radio and Wi-Fi on, screen on, and PDS and some other applications running simultaneously. The battery capacity is 3000 mAh.

| Number | Description     | Memory (MB) | Battery life |
|--------|-----------------|-------------|--------------|
| 1      | Idle + radio    | 12          | 21.95        |
| 2      | Detector + radio| 15          | 19.74        |
| 3      | Normal          | 21          | 7.74         |
| 4      | Detector + normal| 25      | 7.02         |

---

**Figure 6**: The DE similarity distribution of the PDU cases confirmed by users in group #1.

**Figure 7**: The growth of usage history data set in group #2.

**Figure 8**: Instant battery consumption of PDS.
mobile phone fraud. In [18], the authors focus on detecting disease outbreak in a specific area and unusual medical time sequences in the health caring domain. Detecting PDU is also a problem of anomaly detection but the complicated PDU characteristics and high user diversity make this problem highly different from previous anomaly detection issues.

7. Discussions and Future Work

Only on Android? Currently we develop PDS only on Android phones. Although several mechanisms of Android make it easier to be implemented, our approach is not limited to any specific platforms. The information we make use of is also available on other platforms including iOS and Windows Phone. For example, tcpdump [19] running on iOS can provide all the data usage information. The similar APIs also can be found on other platforms. Furthermore, PDS can not only serve as an application but also be integrated into existing data usage management software. Currently we design and implement PDS and show the basic performance. Next release and test in public would be one of the future works.

Startup Stage. PDS is a history-based detection system. With the growth of the usage history dataset, the detection accuracy is increasing as well. However PDS highly depends on both the quantity and the quality of the usage history dataset. In the startup stage, how to build a high-quality usage history dataset in a short time is a practical problem for PDS. Currently PDS needs to involve the users to ensure the correctness of the usage history. One of our future works is to reduce user interventions. We make a preliminary observation that the applications of the same categories might share the similar data usage patterns. Based on the observation, we can train models offline for some popular applications, which can be used to cross-refered when building the usage history dataset for similar applications.

8. Conclusion

This paper makes advances towards automatically detecting Phantom Data Usage (PDU) on smartphones. To achieve the goal, we present a characterization study of PDU based on about 400 data usage issues in real world. We also make the observation that the context affects how the users use applications and each user preserves specific data usage patterns under particular context. Based on it, we present PDS, a system tool to automatically detect PDU. The evaluation results show PDS achieves about 72% of PDU cases that can be successfully detected by PDS and confirmed by users.

Conflict of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the reviewers for their constructive insights and valuable comments for improving the quality of the paper. They acknowledge the support from Singapore MOE AcRF Tier 2 Grant MOE2012T21070 and NTU Nanyang Assistant Professorship (NAP) Grant M4080738.020.

References

[1] Cisco, “Cisco visual networking index: global mobile data traffic forecast update, 2013–2018,” 2014, http://goo.gl/Z6U9nI.
[2] More than thirty percent of users believe their mobile data is stolen, 2013, http://goo.gl/elnHMz.
[3] C. Starrett, “U.S. iPhone users reporting phantom data usage,” 2010, http://goo.gl/JO9XVu.
[4] My Data Manager, http://goo.gl/vZWmL0.
[5] A. Lakhina, M. Crovella, and C. Diot, “Mining anomalies using traffic feature distributions,” ACM SIGCOMM Computer Communication Review, vol. 35, no. 4, pp. 217–228, 2005.
[6] T. Qin, X. Guan, W. Li, P. Wang, and Q. Huang, “Monitoring abnormal network traffic based on blind source separation approach,” Journal of Network and Computer Applications, vol. 34, no. 5, pp. 1732–1742, 2011.
[7] Y. Xu, M. Lin, H. Lu et al., “Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns,” in Proceedings of the 17th ACM International Symposium on Wearable Computers (ISWC ’13), pp. 69–76, September 2013.
[8] T. M. T. Do, J. Blom, and D. Gatica-Perez, “Smartphone usage in the wild: a large-scale analysis of applications and context,” in Proceedings of the ACM International Conference on Multimodal Interaction (ICMI ’11), pp. 353–360, November 2011.
[9] T. Yan, D. Chu, D. Ganesan, A. Kansal, and J. Liu, “Fast app launching for mobile devices using predictive user context,” in Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys ’12), pp. 113–126, 2012.
[10] P. Zhou, Y. Zheng, and M. Li, “How long to wait? Predicting bus arrival time with mobile phone based participatory sensing,” in Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys ’12), June 2012.
[11] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay, “MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones,” in Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys ’07), pp. 57–70, June 2007.
[12] H. Verkasalo, “Contextual patterns in mobile service usage,” Personal and Ubiquitous Computing, vol. 13, no. 5, pp. 331–342, 2009.
[13] 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 Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI ’11), pp. 47–56, September 2011.
[14] N. D. Lane, Y. Chon, L. Zhou et al., “Piggyback CrowdSensing (PCS): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities,” in Proceedings of the...
C. Shin, J.-H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," in Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp ’12), pp. 173–182, 2012.

S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in Proceedings of the 27th Hawaii International Conference on System Sciences (HICSS ’94), pp. 621–630, January 1994.

C. Aggarwal, “On abnormality detection in spuriously populated data streams,” in Proceedings of the Fifth SIAM International Conference on Data Mining, chapter 8, pp. 80–91, SIAM, 2005.

J. Lin, E. Keogh, A. Fu, and H. Van Herle, “Approximations to magic: finding unusual medical time series,” in Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems, pp. 329–334, June 2005.

TcpDump, http://www.tcpdump.org/.
