iGuard: A Real-Time Anti-Theft System for Smartphones

Meng Jin[1], Yuan He[2], Dingyi Fang[1], Xiaojiang Chen[1], Xin Meng[1], Tianzhang Xing[1]
[1]Northwest University, China
[2]School of Software and TNLIST, Tsinghua University, China.
{mengj, mengxin}@stumail.nwu.edu.cn, he@greenorbs.com, {dyf, xjchen, xtz}@nwu.edu.cn

***Abstract*—Smartphone theft is a non-negligible problem that causes serious concerns on personal property, privacy, and public security. The existing solutions to this problem either provide only functions like retrieving a phone, or require dedicated hardware to detect thefts. How to protect smartphones from being stolen at all times is still an open problem. In this paper, we propose iGuard, a real-time anti-theft system for smartphones. iGuard utilizes only the inertial sensing data from the smartphone. The basic idea behind iGuard is to distinguish different people holding a smartphone, by identifying the *order* of the motions during the 'take-out' behavior and *how* each motion is performed. For this purpose, we design a motion segmentation algorithm to detect the transition between two motions from the noisy sensing data. We then leverage the distinct feature contained in each sub-segment of a motion, instead of the entire motion, to estimate the probability that the motion is performed by the smartphone owner himself/herself. Based on such pre-processed data, we propose a Markov Chain based model to track the behavior of a smartphone user. According to this model, iGuard instantly alarms once the tracked data deviate from the smartphone owner's usual habit. We implement iGuard on Android and evaluate its performance in real environments. The experimental results show that iGuard is accurate and robust in various scenarios.**

I. Introduction

Smartphone theft is a non-negligible problem that causes serious concerns on personal property, privacy, and public security. According to Consumer Reports [1], in 2013, more than 3 million smartphones were stolen in the U.S., up from 1.6 million in 2012. Unfortunately, by far there is not any effective and affordable anti-theft solution for smartphones. Many users choose to install a software on their smartphones so that the phone might be located, after it is stolen. This solution cannot protect the phone from being stolen. It may be helpful to retrieve the phone, suppose the pickpocket does not disable any functions of the protective software. A recent alternative solution is to use a Bluetooth based anti-theft device which can inform the owner if his/her smartphones is out of a secure proximity. This solution however requires dedicated hardware, which imposes additional cost and difficulties for users like the elderly or the children.

There is an urgent need for an anti-theft solution for smartphones, which is desired to be highly *sensitive* and *accurate*, and easy to use under all kinds of scenarios. Utilizing the built-in sensing capability and intelligence on the smartphones appears to be a promising direction. Nevertheless, existing activity recognition approaches fall short on one or more of the following aspects. The real-time recognition approaches [2–4] largely depend on the short-term features of user behavior, which can only identify distinctive or well-defined behavior. Note that the behavior of stealing a phone is relatively complex (including a sequence of motions). Without careful discrimination, that behavior is likely to be confused with the "take-out" motion performed by the smartphone owner. Although many efforts have been made to identify different users [5, 6], they have to depend on long-term features, e.g. life-style and biometric characters. Collecting such information usually consumes excessive time, which is clearly unacceptable for anti-theft scenarios.

In this paper, we for the first time propose a theft detection system for smartphones, named iGuard. iGuard is able to instantly identify whether the owner himself/herself is taking the smartphone, by exploiting only the built-in inertial sensors on the smartphone. Different from the existing user identification approaches which depend on long-term features of different users, iGuard can distinguish different users in real time, utilizing only one key behavior, namely taking out the smartphone. The idea behind iGuard comes from the following intuitive observations: *i) a person has distinctive habits (or order of motions), when he/she takes out a smartphone; ii) a person has distinctive features when he/she performs each independent motion in the behavior of taking out a smartphone.*

To obtain the above-mentioned fine-grained information in real time, however, is a very challenging task. It is also hard to find a simple mapping between the above suspected behavior (i.e., an unusual motion order or an unusual pattern of an independent motion) and the stealing activity. In iGuard, we provide a Markov Chain based model (named SAM), which continuously track motions of the smartphone user, and instantly estimate the probability (denoted as *PSelf*) that the motions are performed by the user himself/herself based on a joint consideration of the user's motion patterns and the order of motions. Once an abnormal *PSelf*, which indicates a suspected behavior, is detected, SAM identifies it as stealing.

For the application of SAM, we further design a motion segmentation algorithm to detect the transition between two motions from the noisy sensing data. We then leverage the distinct feature contained in each sub-segment of a motion, instead of the entire motion, to estimate the probability that the motion is performed by the smartphone owner himself/herself. The contributions of this paper are summarized as follows:

- Based on extensive experiments and observation, we propose a Markov Chain based model to continuously track motions of the smartphone user, which can detect

978-1-5090-5336-0/17/$31.00 ©2017 IEEE

theft with high sensitivity and accuracy.
- Based on this model, we propose iGuard, a theft detection system which can instantly identify whether the owner himself/herself is taking the smartphone, by exploiting only the built-in inertial sensors on the smartphone.
- We implement iGuard and evaluate its performance across various scenarios. The experiment results demonstrate the effectiveness of iGuard.

In the rest of this paper, we will present the related work in Section II. Then we describe our preliminary findings in Section III. We elaborate the design details of iGuard in Section IV and evaluate its performance in Section VI. Section VII concludes the whole paper.

II. Related Work

The design of iGuard shares some similar techniques with the existing works on human behavior recognition with sensor data. In this section, we briefly discuss those existing proposals, by classifying them into two main categories.

Feature based: Feature based behavior recognition approaches primarily leverage the distinctive features of different behaviors [3, 7–13]. They usually use a classifier that takes the extracted features of a behavior as the key inputs. For example, PBN [8] provides a behavior recognition approach based on data from sensor motes attached to a person's body and an AdaBoost classifier, which can identify 9 different activities with an accuracy of nearly 90%. RisQ [3] presents a smoking detection system that leverages the common features of smoking and a random forest classifier to recognize a series of motions during smoking. MoodScope [9] proposes a smartphone based approach which can infer the user's mood, by using a broad set of features extracted from user-smartphone interactions. AccelWord [7] proposes an acceleration data based solution for hotword detection, which extracts features of a predefined hotword and then identifies it using a pre-trained classifier.

Feature based behavior recognition approaches usually assume distinguishable features among different behaviors, which is however not available in the theft detection scenario, where the 'take-out' motions performed by different persons (i.e., the owner or others) probably have very similar patterns. Our experimental results in Section VI show that theft detection using a feature based method achieves an accuracy of less than 75%.

Similarity based: Similarity based behavior recognition approaches usually maintain a set of pre-constructed behavior profiles [5, 14–17]. They identify behaviors by calculating the similarity between the sampled signals and the behavior profiles. Typical similarity metrics include DTW (Dynamic Time Warping) distance [18], EMD (Earth Mover Distance), Euclidean distance, etc. For example, WiHear [15] uses DTW to recognize people's talk based on Wi-Fi signals, which can identify the pronunciation of 9 different alphabets. E-eyes [16] proposes a WiFi signature based daily activity identification system, which uses DTW and EMD to identify in-place activity and walking activity, achieving an accuracy of 96%. WiKey [17] proposes a keystroke recognition approach, which uses DTW to identify a keystroke sample based on its CSI-waveform shape.

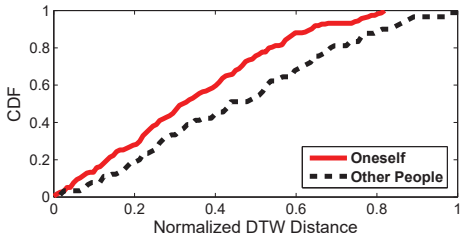


Fig. 1: DTW distance between two "take-out" motions performed by the same/different persons.

Similarity based behavior recognition approaches usually assume a fixed pattern of a certain behavior, which may not hold for the theft detection scenario. In practice, due to the mobility of the user and the smartphone gesture, even the "take-out" motions performed by the same person rarely have a constant pattern. Figure 1 shows the CDF of DTW distance between two "take-out" motions performed by the same/different persons. We can see that the overlap of the distributions of different motions is large, which implies that DTW distance is not an effective metric to distinguish them.

Different from the above two categories of approaches, iGuard leverages both a user's motion patterns and the order of motions for joint recognition of the theft behavior.

III. Preliminary

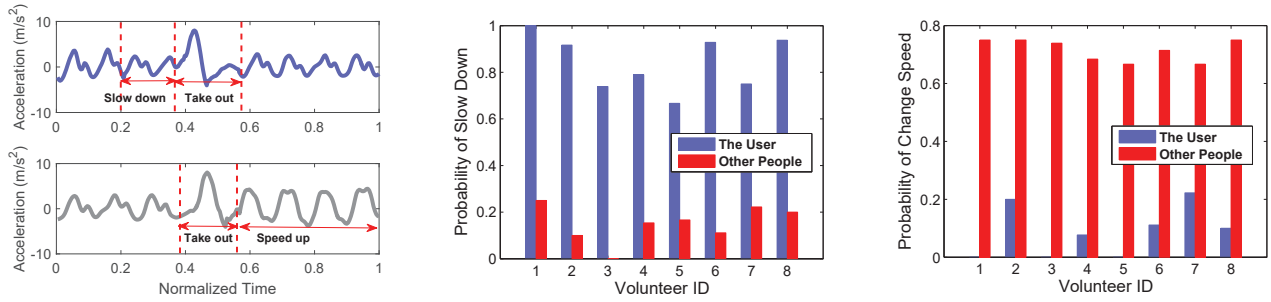
In this section, we first conduct a set of experiments to investigate how the behavior of taking out a smartphone affects the inertial sensor data. Specifically, we show that: i) *the user and the thief have consistent and distinguishable habits (or order of motions) when they taking out a smartphone;* and ii) *they also have consistent and distinguishable features when performing each motion in the behavior of taking out a smartphone.*

A. Data Collection

We develop a motion collection program based on Android, which can record the inertial sensor data when the user performing motions. We found 8 volunteers with ages ranging from 19 to 27 to perform a set of experiments. In the first experiment, we install our program on their smartphones and ask them to start the program during walking. The experiment lasts for 5 days and we collect 589 samples for the behavior of taking out a smartphone. In the second experiment, we ask the volunteers to pretend to be thieves to 'steal' the smartphones from the other volunteers. To imitate the motions of a thief, the volunteers try to take out the smartphone without being noticed. We collect 618 samples in this experiment.

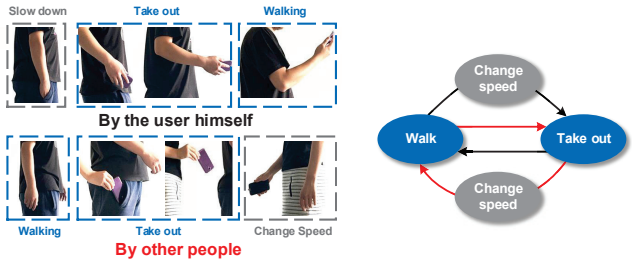
B. Data Analysis

1) Motion sequence of the behavior of taking out a smartphone performed by different people: In Figure 2(a), we plot the time series of acceleration data sampled during the take-out motion. Specifically, the figure at the top of Figure 2(a) plots the take-out motion performed by the user himself/herself, and



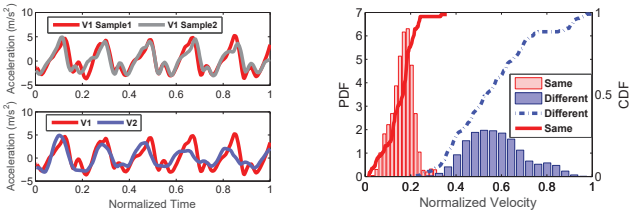
(a) (b) (c)

Fig. 2: Motion sequence of taking out a smartphone performed by the user and other people: (a) time series of acceleration sampled during the take-out motion; (b) probability of slow down before taking out the smartphone; (c) probability of speed change after taking out a smartphone.



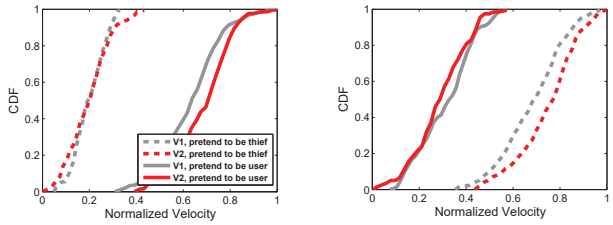
(a) (b)

Fig. 3: Difference between the take-out behavior performed by different people: (a) an example; (b) transitions between each motions.



(a) (b)

Fig. 4: The walking motion performed by different people: (a) time series of acceleration sampled during the walking motion; (b) distribution of DTW distance between two walking motion performed by the same and different volunteers.



(a) (b)

Fig. 5: The distribution of velocity for (a) picking and (b) putting up sub-segment that performed by different volunteers.

the figure at the bottom plots the take-out motion performed by other people. We can see that, if the smartphone is taken out by the user himself/herself, the sampled motion sequence is $\{$*walking, slow down, take out the smartphone, walking*$\}$. However, if the smartphone is taken out by others, the sampled motion sequence is $\{$*walking, take out the smartphone, speed up, walking*$\}$. The main causes of such a difference are: i) if the smartphone is taken out by the user himself/herself, he/she is conscious of this motion and thus will slow down habitually; ii) if the smartphone is taken out by others, he/she will speed up to flee soon after stealing the smartphone.

We further plot the statistical results in Figures 2(b) and 2(c), which verify the unbiasedness of our observation in Figure 2(a). The above observation implies that *people have distinguishable habits (or order of motions), when he/she takes out a smartphone.* Figures 3(a) gives an example to show the difference between the user himself/herself and others when they take out a smartphone. In addition, Figure 3(b) shows the transitions between each two motions, where the black arrow indicates a legitimate motion sequence while the red arrow indicates an illegitimate motion sequence.

2) Walking and Take-Out Motion Performed by Different People: We first discuss the **walking** motion performed by different people. In Figure 4(a), we plot the time series of acceleration data sampled during the walking motion. Specifically, the figure at the top of Figure 4(a) plots the acceleration data of two walking motions performed by the same volunteer, and the figure at the bottom plots the acceleration data of the walking motions performed by two different volunteers (V_1 and V_2). We observe that the walking motions performed by the same person are similar and at the same time different from that performed by another person. We further plot the statistical result in Figure 4(b) which shows the distribution of DTW distance between two walking motions performed by the same (the solid curve) and different (the dotted curve) volunteers, respectively. The figure shows that the overlap between the two curves is small.

Next we discuss the **take-out** motion performed by different people. As discussed in Section II, the take-out motions performed by the user and the thief have very similar features. Thus the existing feature based and DTW based behavior identification methods are not applicable in the theft detection scenario. Fortunately, we find that compared with the features extracted from the entire motion, *features extracted from sub-*

segments of a motion are clearly more distinguishable. Indeed, the user and the thief behave differently in different periods of a take-out motion. For example, a thief would pick the phone slowly and put up the phone fast to avoid raising awareness. However, the user acts differently. Such distinctive information will be averaged out if we measure the feature values of the entire motion.

As an example, Figures 5(a) and 5(b) plot the distribution of velocity of picking and putting up periods that performed by different volunteers (although not shown here, data of other features measured from other periods are also generated). The figures tell that: i) different volunteers (V_1 and V_2) have consistent behavior if they pretend to be the same role (the user or the thief); ii) different roles have distinct features during different periods of a motion.

The above observations imply that *a person has distinguishable features when he/she performs each independent motion (i.e., walking and take-out) in the behavior of taking out a smartphone.*

IV. System Design

In this section, we first describe the theft detection model and then present the overview of this work. After that we introduce the design details of iGuard.

A. SAM: A Model for Theft Detection

According to the observation in Section III, we propose to take both a user's performing pattern of an independent motion and the order of the motions into account for real-time theft detection. The problem of theft detection is stated as: given i) a sequence of motions in the behavior of taking out a smartphone; ii) the transition probability between each two motions; iii) the probability that an individual motion is performed by the user; what is the probability that the entire behavior is performed by the user?

We propose a Markov-based model (named SAM) to tackle this problem. Specifically, assuming the currently detected motion sequence is $\mathbf{S_n} = \{S_{n-1}, ..., S_{n-w}\}$, where w is the window for activity analysis. Based on the user's habit, we can get the transition probability between each two motions as $P(S_i|S_{i-1})$. Then the probability that the user performs a sequence of motion in the order of $\{S_{n-1}, ..., S_{n-w}\}$ can be estimated as $P_{seq} = \prod_{\text{i=n-w}}^{n} P(S_i|S_{i-1})$. Then we can give a normalized metric (denoted as *PoO*) to evaluate how likely the user performs the detected motion sequence $\mathbf{S_n}$ as follow:

$$PoO(\mathbf{S_n}) = max\left\{\frac{P_{seq} - \min\limits_{s\in S}(P_s)}{\max\limits_{s\in S}(P_s) - \min\limits_{s\in S}(P_s)}, 1 - \frac{rank(P_{seq}) - 1}{|S|}\right\} \quad (1)$$

where $\mathbf{S}$ is the set of all the possible motion sequences that include n motions starting with S_{n-w} and ending with S_n. $rank(P_{seq})$ is the ranking of P_{seq} among all the P_s ($s \in \mathbf{S}$).

In addition, we denote the probability that the motion S_i is performed by the user himself/herself as PoS_i. Thus based on the *PoS* sequence $\{PoS_{n-1}, ..., PoS_{n-w}\}$ and the estimated $PoO(\mathbf{S_n})$ for the motion sequence $\mathbf{S_n}$, we can give a normalized metric $PSelf(\mathbf{S_n})$ to describe how likely the motion sequence $\mathbf{S_n}$ is performed by the user as follows:

$$PSelf(\mathbf{S_n}) = PoO(\mathbf{S_n^{(p)}}) \cdot min\{PoS(\text{s·n}), ..., PoS(\text{s·n-w})\} \quad (2)$$

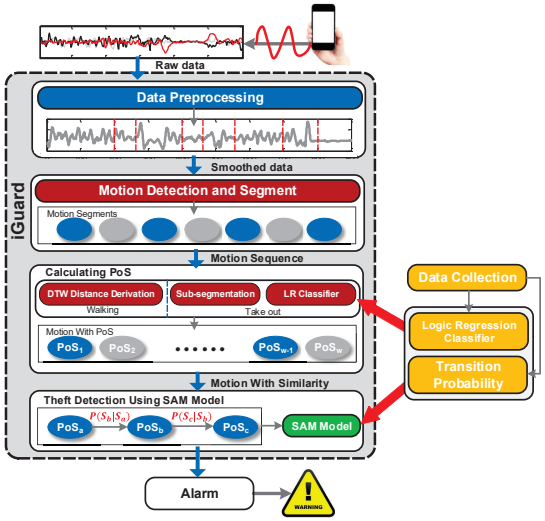


Fig. 6: Overview of iGuard.

Equation (2) implies that a low $PSelf(\mathbf{S_n})$ value, namely a high probability that the smartphone is stolen, is achievable under two conditions: i) the detected motion sequence $\mathbf{S_n}$ deviates from the smartphone owner's usual habit. ii) some motions in the sequence $\mathbf{S_n}$ exhibit low similarity with those performed by the user, leading to a low $min\{PoS(\text{s·n}), ..., PoS(\text{s·n-w})\}$. We verify the feasibility of using the SAM model for theft detection in Section IV-D.

While the focus of our work is on theft detection, the proposed SAM model can be tailored to many other application instances, such as smartphone customization, user authentication and even personal health monitoring, etc. We leave these exploitations to our future works.

iGuard overview. In this work, we leverage SAM and propose iGuard, a theft detection system that is able to instantly and accurately recognize who is taking out the smartphone using only the inertial sensors. Figure 6 shows the overview of iGuard.

First of all, the sampled sensor data are processed via a low-pass filter to remove high frequency noise. Then the *motion detection and segmentation* component extracts segments containing different motions from the smoothed sensor data. The *PoS estimation* component identifies the motion in each segment and outputs the motion sequence as $\{S_1, ..., S_w\}$, where w is the window for behavior analysis and we set $w = 3$ in iGuard. The extracted motion segments act as the input of the *PoS estimation* component which calculates the PoS_i for each motion. Specifically, based on the characteristics of walking and take-out motions, we develop two separate *PoS* calculating methods for them. For the walking motion, we use Dynamic Time Warping (DTW) technique. For the take-out motion, we calculate the *PoS* based on their sub-segment features and a logic regression classifier. Based on the SAM model, the *theft detection* component incorporates the *PoS* sequence $\{PoS_1, ..., PoS_w\}$ and the transition probability $P(S_i|S_{i-1})$ between each two motions, to identify whether the smartphone is stolen.

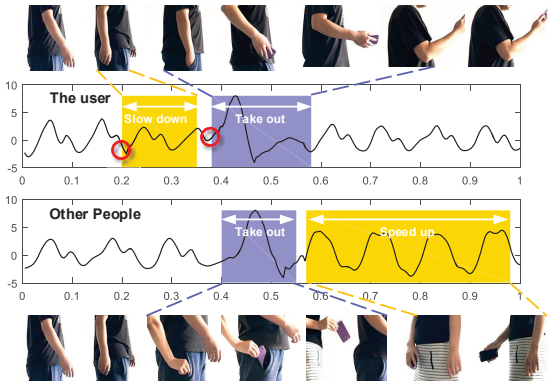


Fig. 7: Illustration of the segmentation procedure.

B. Motion Detection and Segmentation

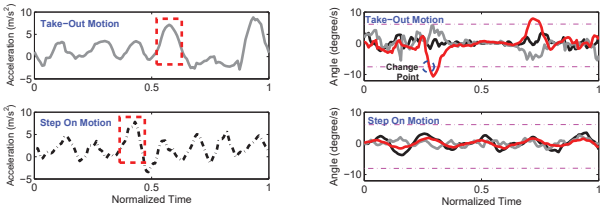
The goal of the motion detection and segmentation component is to extract the walking, take-out, and speed change motions from the smoothed inertial data, and output the motion sequence $\{S_1, ..., S_w\}$, as shown in Figure 7.

1) Walking detection: Since the goal of iGuard is to detect thefts when the user is walking, thus we should first identify the start of the walking activity before further data analysis. It is widely believed that, walking activity will cause a larger variance of inertial data than in-place activity (such as typing in text) [16]. Thus in iGuard, we use an empirical threshold on variance (as the authors of Travi-Navi [19] did in their work) to detect the start of a walking activity, and then trigger iGuard if a walking activity is detected.

2) Take-out detection: A naive take-out detection solution is to capture the sudden change in acceleration data with a predefined threshold. However, some other motions, such as step on a stair, might have a similar impact on the acceleration data, as shown in Figure 8(a). This will inevitably harm the detection quality. Fortunately, we find that such confusing motions only cause small change in the attitude of the smartphone. Thus although they cause a sudden change in acceleration data, they have marginal impact on the gyroscope data. The take-out motion which keeps changing the attitude of the smartphone will have a more obvious impact on the gyroscope data, as shown in Figure 8(b).

Based on the above observations, we propose to incorporate data from both the acceleration sensor and the gyroscope sensor to detect the take-out motions. Specifically, when the new data (acceleration data a_i and gyroscope data g_i) come, the algorithm will trace back to find the maximal difference between g_i (a_i) and samples collected within the last T seconds. If both the gaps $|g_i - g_m|$ and $|a_i - a_m|$ are greater than the thresholds g_{tr} and a_{tr}, we identify i as the start point of the take-out motion and then conduct a forward search afterwards to determine the entire take-out period.

3) Speed change detection: The goal of speed change detection is to determine whether the user changes his/her walking speed before and after the take-out motion. This is challenging because the slow-down and speed-up motions only result in a very small change in the sensor data, which is difficult to be detected from a dynamic data series. To address



(a) (b)

Fig. 8: Different between the take-out motion and the step on motion: (a) acceleration data; (b) gyroscope data.

this problem, we need a mechanism that is able to reliably detect small changes and insensitive to data dynamic.

We use the Cumulative Sum (CUSUM) test to detect the change in the walking speed. CUSUM uses sequential observations of the process and detects if the process mean has shifted by more than a specified threshold. The sequential nature of CUSUM is particularly attractive for our application where the sensor data arrive periodically. As the name suggests, the algorithm sequentially computes cumulative sums using the following two equations:

$$S_{hi}(i) = \max(0, S_{hi}(i-1) + x_i - T - k) \quad (3a)$$

$$S_{lo}(i) = \max(0, S_{lo}(i-1) + T - k - x_i) \quad (3b)$$

where $S_{hi}(0) = S_{lo}(0) = 0$, x_i is the process mean usually calculated from a sample of q observations, T is the target value of mean for which the process is considered to be under control. h is an adjustable parameter. When the value of $S_{hi}(i)$ or $S_{lo}(i)$ exceeds h, the process is considered to be changed. The value of k is calculated as $k = \delta\bar{\sigma}/2$, where $\bar{\sigma} = \sigma/\sqrt{q}$, σ is the process standard deviation and δ is the amount of shift in the process mean.

We apply the CUSUM on top of the walking segments before and after the take-out motion. When the cumulative sum of the sensor data computed over successive windows exceeds a certain threshold value, CUSUM identifies a change in the walking speed.

After identifying each motion in the behavior of taking out a smartphone, iGuard gets a **motion sequence** which might be labeled as $\{$*walking, (change speed), take out, walking*$\}$. This motion sequence is then fed to the *PoS estimation* component for further analysis.

C. PoS Estimation

The *PoS* estimation component aims to calculate the PoS_i for each motion in the motion sequence detected in the previous component, and outputs a *PoS* sequence $\{PoS_1, ..., PoS_w\}$. Based on the characteristics of walking and take-out motions, we develop two separate *PoS* estimation methods for them.

1) PoS for walking motion: Since different people have *stable* and *unique* biologic characters and walking habits, inertial data sampled during the walking motions performed by the same person are similar and at the same time different from that performed by another person (as shown in Section III). Thus the *PoS* of a walking motion can be estimated by calculating the similarity between the sampled data series and

Feature	Description	Category
Duration	Time duration for each sub-segment	**Duration Features**
Speed	Mean, max and variance of the movement speed of the smartphone	**Velocity Features**
distZ	Vertical displacement of the smartphone	**Displacement Features**
distXY	Horizontal displacement of the smartphone	
dist	Net displacement between the rest position and the peak position	
Roll velocity	Median and maximum angular velocity around the Y axis	**Angle Features**
Roll	Net angular change around the Y axis	
Pitch velocity	Median and maximum angular velocity around the X axis	
Pitch	Net angular change around the X axis	

TABLE I: Feature set extracted for each sub-segment.

the data series collected during the walking motion performed by the user himself/herself. Assuming that the walking motion that performed before the take-out motion is certainly performed by the user (i.e., with *PoS=1*), we can use it as a reference to calculate the *PoS* of the walking motion that performed after the take-out motion.

However, even two walking data series from the same people will misaligned at different segments (although they have a similar pattern). We propose to use Dynamic Time Warping (DTW) to align and quantify the similarity between two walking data series. Specifically, DTW matches each sample in one data series to one or more samples in another data series using dynamic programming. The objective of DTW can be stated as:

Given two time series $\mathbf{C_a}$ *and* $\mathbf{C_b}$:

$$\mathbf{C_a} = C_a[i], i = 1, ..., L_a; \mathbf{C_b} = C_b[i], i = 1, ..., L_b.$$

DTW aims to find a monotonic mapping function $f : I[1, L_a] \rightarrow I[1, L_b]$ *between* $\mathbf{C_a}$ *and* $\mathbf{C_b}$*, so as to minimize the distance between them*:

$$\sum_{i=1}^{L_a} (C_a[i] - C_a[f(i)])^2$$

where $I[1, L_a]$ *is the integers from 1 to* L_a.

In the context of iGuard, we segment a W_{walk} long walking data series before and after the take-out motion, and calculate their DTW distance. W_{walk} is the window for walking motion analysis.

Some practical challenges arise when we try to directly apply the above DTW algorithm to compare the walking motions: i) the high time-complexity (a quadratic function of the number of samples) of DTW is unacceptable for iGuard which has to detect the dissimilarity instantly; ii) the similarity calculated by DTW is in the form of an absolute value which is not bounded. In the design of iGuard, however, we need a probability value ranging from 0 to 1.

To address the above problems, we make some simple modifications to DTW. Instead of performing DTW directly on the two walking data segments $\mathbf{C_a}$ and $\mathbf{C_b}$, iGuard first normalizes them as $\hat{C}_a[i] = \frac{C_a[i] - min(\mathbf{C_a})}{max(\mathbf{C_a}) - min(\mathbf{C_a})}$ and $\hat{C}_b[i] = \frac{C_b[i] - min(\mathbf{C_b})}{max(\mathbf{C_b}) - min(\mathbf{C_b})}$. This brings all the sampled values into the range of $[0, 1]$. In addition, consider that the difference between walking speed before and after the take-out motion should be bounded within a few seconds, we can set a global path constraint when searching for the mapping function $f(\cdot)$ (as the authors of FOLLOWME [20] did in their work). In this way, the algorithm runs in linearithmic time.

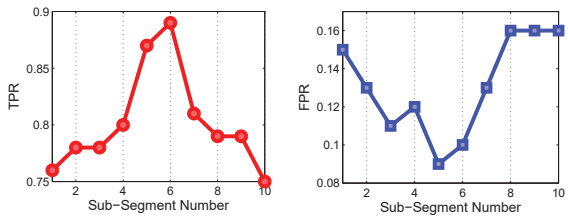


Fig. 9: the TPR and FPR of take-out motion recognition under different segment number.

After obtaining the mapping function, we calculate the normalized DTW distance between $\mathbf{C_a}$ and $\mathbf{C_b}$ as follow:

$$d(\mathbf{C_a}, \mathbf{C_b}) = \frac{\sum_{i=1}^{L_a} |C_a[i] - C_a[f(i)]|}{L_a} \quad (4)$$

The *PoS* of the walking motion can be calculated as $1 - d(\mathbf{C_a}, \mathbf{C_b})$.

2) PoS for take-out motion: As shown in Section II, the take-out motions performed by different roles have neither consistent nor distinguishable patterns. Thus the existing feature based and DTW based activity identification methods are not applicable in our scenario. Fortunately, we find in Section III that, compared with the features extracted from the entire motion *the features extracted from sub-segments of a take-out motion exhibit higher differentiation degree*. Thus we propose to first segment each take-out motion into sub-segments, then leverage the feature values measured from each sub-segment and the logistic regression classifier to calculate the *PoS* of the entire take-out motion.

However, it is challenging to determine the number of sub-segments that a motion should be segmented into. Specifically, if the time duration of a sub-segment is too short, the user may not have consistent behavior for that sub-segment. If the time duration of a sub-segment is too long, the distinctive information from the features is too much averaged out to be useful for distinguishing different roles. Figure 9 shows the TPR and FPR of recognizing whether the motion is performed by the others. The figure tells that: i) if we consider only the

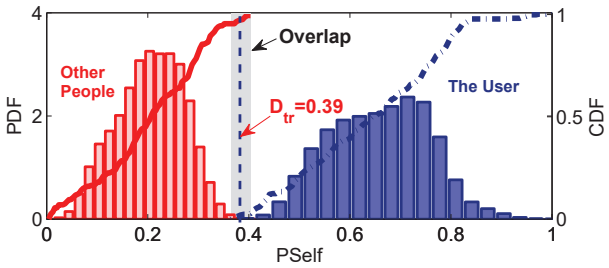


Fig. 10: Distribution of *PSelf* for the take-out motion sequences performed by the user and other people.

take-out motion independently for theft detection, the accuracy will topped off at $TPR = 86\%$. ii) TRP has a significant sharp when $N_{sg} = 6$ and FRP has a significant dip when $N_{sg} = 5$. Then we choose the segment number that brings the lowest FRP (i.e., $N_{sg} = 5$), since a high FRP may annoy the user by frequently triggering the alert when he/she takes out his/her own smartphone.

After segmenting the take-out motion, we compute for each sub-segment the features that can distinguish different roles. Table I shows a set of features computed for each segment. Methods for calculating these features is proposed in [7] and [3], and its design detail is therefore omitted. After extracting the features, iGuard obtains the *PoS* using the logistic regression classifier (training method of the classifier is given in Section V). We select logistic regression due to its exclusive advantages for our purpose: i) its output ranges from 0 to 1, which can be directly interpreted as *PoS*; ii) computational simplicity makes logistic regression feasible even under strict resource constraints, such as those on a smartphone.

After calculating the *PoS* for each motion, iGuard gets a motion sequence with *PoS* for each motion, such as $\{$*walking* $(PoS_1 = 1)$*, (change speed), take-out* $(PoS_2 = 0.9)$*, walking* $(PoS_3 = 0.9)\}$. This sequence is then fed to the next component for theft detection.

D. Theft detection using SAM model

After getting i) the motion sequence $\{S_1, ..., S_w\}$; ii) the *PoS* sequence $\{PoS_1, ..., PoS_w\}$; and iii) the transition probability between each two motions (obtained by the training study process shown in Section V), iGuard estimates the probability that the motion sequence $\{S_1, ..., S_w\}$ is performed by the user himself/herself (i.e., *PSelf*) by using our SAM model (i.e., Equation (2)).

Figure 10 shows the distribution of *PSelf* for the take-out motion sequences performed by the user and the others. We find that the distributions of the two kinds of behaviors are centered at different means and the overlap is small. This observation implies the feasibility of our SAM model. We set the detection threshold as $D_{tr} = 0.39$ for theft detection according to the experiment result. A motion sequence with *PSelf*< 0.39 will be treated as stealing.

V. Training

In this section, we explain how iGuard trains its logic regression classifier and the transition probability for the SAM model.

A. Training the Logic Regression Classifier

As shown in Section III, the features of the take-out motion are consistent among different people, but distinct between different roles (i.e., the user himself/herself or other people). Thus we do not need to train a classifier for every user, but only for each role. Briefly, two of the volunteers pretend to be the thief and the user. We ask the "user" to take out the smartphone by himself/herself for 100 runs and the "thief" to take out the smartphone from the "user" for 100 runs. To imitate motions of a real thief, the "thief" tries to take out the smartphone without attracting attentions of the "user". The collected inertial data are then used as training data to create the logic regression classifier. The created classifier is tested in Section VI.

B. Training the Transition Probability for SAM Model

Transition probabilities of the SAM model are obtained from the real world deployment of iGuard. Specifically, in the first 7 days, iGuard will not perform theft detection, but only detects the walking, take-out and speed change motions of the user and records the timestamps of these motions. To note that, we assume that the smartphone will not be stolen during this period and thus the walking and take-out motions are all labeled as "self-performed". Then iGuard can infer the transition probabilities between each two motions based on the collected motion sequences.

VI. Performance Evaluation

In this section, we present the evaluation of iGuard under various environments to show its accuracy and robustness.

A. Experiment Settings

Performance of iGuard is evaluated with three different smartphone models including Sumsung Galaxy S4/S5/Note3 (the results of Galaxy S5/Note3 are omitted due to space limitation), all of which are equipped with MEMS gyroscopes and accelerometers. The sampling rate is set as $50Hz$. The experiments are conducted in two scenarios: i) a corridor in a university building; and ii) a campus with a testing area of $365 \times 535m^2$ (as shown in Figure 12). 8 volunteers are participated in our evaluation.

We use the following metrics to evaluate the performance of iGuard: i) True Positive Rate (TPR): the fraction of the cases where iGuard correctly recognizes a theft event. ii) False Positive Rate (FPR): the fraction of the cases where iGuard mistakenly generates a false alarm when there is actually no theft event.

Comparison. We conduct comparative experiments to investigate the performance of the following approaches:

- *iGuard:* the proposed system in this paper.
- *PoS based method:* theft-detection using only the *PoS* of the take-out motion.
- *PoO based method:* theft-detection using only the *PoO* of the motion sequence.
- *Feature-based method:* theft-detection based on the features extracted from the entire take-out motion.

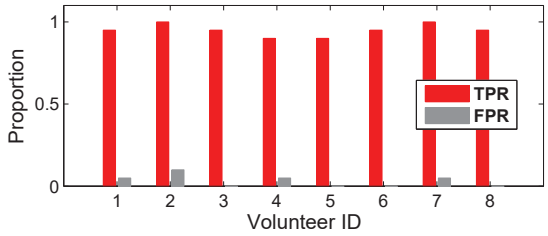


Fig. 11: Overall Accuracy of iGuard.

	iGuard	PoS	PoO	Feature-based
TPR	0.95	0.87	0.72	0.76
FPR	0.04	0.09	0.22	0.15

TABLE II: Overall Accuracy Comparison.

B. Accuracy of iGuard

To evaluate the performance of iGuard, we conduct two experiments in a corridor of our office building. In the first experiment, we ask the 8 volunteers to perform the behavior of taking out a smartphone for 20 runs. The only instruction provided is to walk as usual along the corridor, and take out the smartphone whenever they want. In the second experiment, one of the authors pretends to be the thief to steal the smartphone from the volunteers for 20 runs. The results are shown in Figure 11 and Table II.

We can see that in the relatively ideal condition (straight road without stair and curve) in the corridor, the TPR of iGuard can be as high as 0.95 (19% improvement compared with the Feature-based method) and the average FPR is as low as 0.04. TPR of the method using only the *PoS* of the take-out motion is only 0.87, which implies that classifying the take-out motion independently does not seem to work well. Meanwhile, considering only the order of the motion (i.e., *PoO*) also exhibits poor performance (with $TPR = 0.72$). We can also observe that *PoS* based method performs better than *PoO* based method. This indicates that we should increase the weighing coefficient of *PoS* in the SAM model.

C. Performance in Different Conditions

In this section, we evaluate iGuard under various conditions to show its robustness. Specifically, we select an experimental trace in our campus which includes three road conditions: straight road, roundabout and stairs as shown in Figure 12. We ask 4 of the volunteers to walk along the trace for 5 times, during which one of the authors will call them when they pass the testing points (i.e., $a \sim k$ as shown in Figure 12). The volunteers are asked to take out the smartphone when it rings. Then we ask another 4 of the volunteers to walk along the trace for 5 times, during which one of the authors pretends to be the thief to steal the smartphone from them. The experiment result is given in Figure 13.

According to the results in Figure 13, iGuard outperforms all other methods during the entire trace except for point g, where performance of *PoS* based method is comparable with iGuard. Nevertheless, no apparent performance degrade of iGuard is observed during the entire trace. We also find that

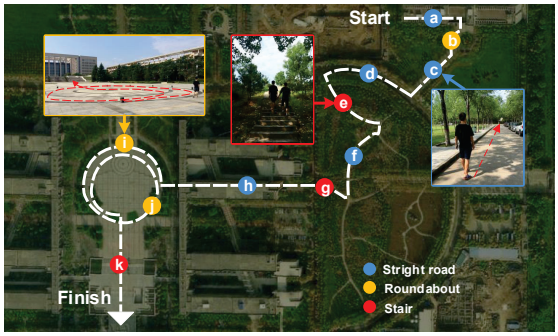


Fig. 12: Experimental trace in our campus.

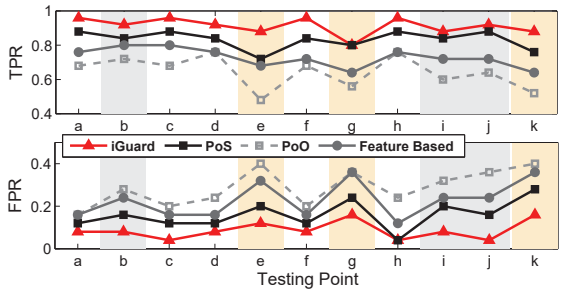


Fig. 13: TPR and FPR of different methods along the testing trace.

all the methods perform better for straight roads and worse for stairs and roundabouts. This is because the phone motion when walking on a straight road is much smoother and of lower motion frequency than that in the other two scenarios, which brings less noise for the motion detection and feature extraction. The *PoS* based method performs the closest to iGuard, outperforming other approaches at most of the time. The performance gap between the *PoS* based and the feature based method tells the gain of motion sub-segmentation which helps to capture the distinct features of the take-out motion.

For statistical comparison, we perform experiments in different scenarios, namely different road conditions, different walking speed of the user, and different smartphones. We perform about 80 runs for each scenario. The statistical results are displayed in Figure 14. Figure 14(a) shows the TPR and FPR of different approaches under different road conditions. The figure tells that all the four methods perform worst for stair road, with 9%, 10%, 29% and 12% performance degradation for iGuard, *PoS*, *PoO* and feature based methods respectively. This is because that the inertial data exhibits larger variance when the user going up or down stairs, which incurs high noise for motion detection and segmentation, especially for the detection of the change speed motion. Compared with going up and down stairs, the turning roads have less impact on the detection accuracy, leading to only $2\% \sim 6\%$ performance degradation.

Figure 14(b) shows the performance of different approachs under different walking speed ($1.3m/s$, $1.6m/s$ and $2.0m/s$) of the user. The figure tells that i) the walking speed does not have noticeable effects on TPR. ii) FPR of iGuard increases significantly when the user walking slowly. This is because the user will not 'slow down' the walking speed in this scenario,

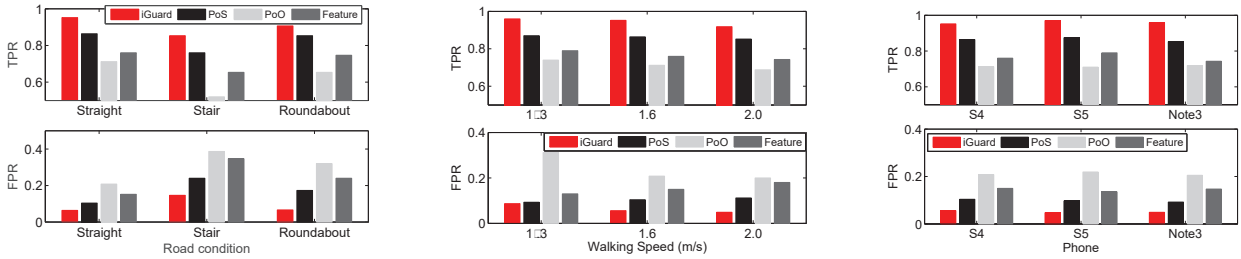


Fig. 14: Performance of different method under different condition: (a) different road condition; (b) different walking speed of the user; (c) different smartphone

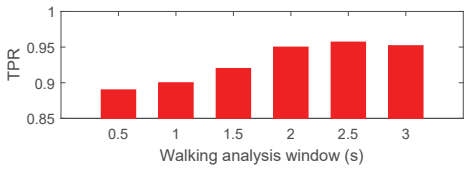


Fig. 15: Detection delay V.S. accuracy.

and thus iGuard will mistakenly treat the take-out motion performed by the user as stealing. Figure 14(c) shows the performance of iGuard on different smartphones (sumsung S4, S5 and note3), telling that iGuard is applicable on different platforms.

D. Detection Delay

The detection delay of iGuard is highly related to the length of the walking analysis window W_{walk}. Indeed, a short window will reduce the delay of iGuard, but incurs low detection accuracy. We conduct an experiment to evaluate the performance of iGuard under different W_{walk}, and the result is shown in Figure 15. The figure tells that when $W_{walk} \geq 2s$ the TPR exceeds 95% and stops growing, which means that iGuard can achieve high accuracy with only a 2s delay.

VII. Conclusion and Future Work

This paper presents iGuard, a real-time anti-theft system for smartphones. By smartly utilizing the distinct information from both the order of the motions during the 'take-out' behavior and the performing pattern of each individual motion, iGuard makes it possible to sensitively and accurately detect theft exploiting only the built-in inertial sensors on the smartphones. We implement iGuard on Android and evaluate its performance in real environments. The experimental results show that iGuard is accurate and robust in various scenarios. In the future work, we plan to expand iGuard to other applications such as smartphone customization, user authentication and even personal health monitoring, etc.

Acknowledgment

This work was supported by National Science Fund for Excellent Young Scientist No.61422207, The NSFC (61672428, 61572402, 61272461, 61672427, 61602381).

References

[1] Consumer Reports. Smart phone thefts rose to 3.1 million in 2013. http://www.consumerreports.org.
[2] S. Nirjon, J. Gummeson, D. Gelb, and K. Kim. TypingRing: A Wearable Ring Platform for Text Input. In *Proceedings of ACM MobiSys*, 2015.
[3] A. Parate, M. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis. RisQ: Recognizing Smoking Gestures with Inertial Sensors on a Wristband. In *Proceedings of ACM MobiSys*, 2014.
[4] S. Jain, C. Borgiattino, Y. Ren, M. Gruteser, Y. Chen, and C. Chiasserini. LookUp: Enabling Pedestrian Safety Services via Shoe Sensing. In *Proceedings of ACM MobiSys*, 2015.
[5] J. Ranjan and K. Whitehouse. Object Hallmarks: Identifying Object Users Using Wearable Wrist Sensors. In *Proceedings of ACM UbiComp*, 2015.
[6] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz. A Wearable System That Knows Who Wears It. In *Proceedings of ACM MobiSys*, 2014.
[7] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra. AccelWord: Energy Efficient Hotword Detection through Accelerometer. In *Proceedings of ACM MobiSys*, 2015.
[8] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles. PBN: Towards Practical Activity Recognition Using Smartphone-Based Body Sensor Networks. In *Proceedings of ACM SenSys*, 2011.
[9] R. Likamwa, Y. Liu, N. D. Lane, and L. Zhong. MoodScope: Building a Mood Sensor from Smartphone Usage Patterns. In *Proceedings of ACM MobiSys*, 2013.
[10] X. Zheng, J. Wang, L. Shangguan, Z. Zhou, and Y. Liu. Smokey: Ubiquitous Smoking Detection with Commercial WiFi Infrastructures. In *Proceedings of IEEE INFOCOM*, 2016.
[11] L. Liu, C. Karatas, H. Li, S. Tan, M. Gruteser, J. Yang, Y. Chen, and R. P. Martin. Toward Detection of Unsafe Driving with Wearables. In *Proceedings of ACM WearSys*, 2015.
[12] C. Xu, P. H. Pathak, and P. Mohapatra. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition using Smartwatch. In *Proceedings of ACM HotMobile*, 2015.
[13] H. Cheng, F. Sun, M. Griss, P. Davis, J. Li, and D. You. NuActiv: Recognizing Unseen New Activities Using Semantic Attribute-Based Learning. In *Proceedings of ACM MobiSys*, 2013.
[14] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-Home Gesture Recognition Using Wireless Signals. In *Proceedings of ACM MobiCom*, 2013.
[15] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni. We Can Hear You with Wi-Fi! In *Proceedings of ACM MobiCom*, 2014.
[16] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu. E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures. In *Proceedings of ACM MobiCom*, 2014.
[17] K. Ali, A. A. Liu, W. Wang, and M. Shahzad. Keystroke Recognition Using WiFi Signals. In *Proceedings of ACM MobiCom*, 2015.
[18] S. Salvador and P. Chan. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis*, 11(5):561 – 580, 2007.
[19] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. Travi-Navi: Self-deployable Indoor Navigation System. In *Proceedings of ACM MobiCom*, 2014.
[20] Y. Shu, K. G. Shin, T. He, and J. Chen. Last-Mile Navigation Using Smartphones. In *Proceedings of ACM MobiCom*, 2015.