Guardian Angel: A Smartphone Based Personal Security System for Emergency Alerting

Hansong Guo*, Liusheng Huang*, He Huang†, Zehao Sun*
Jiade Peng‡, Zhuolong Yu*, Zhenyu Zhu*, Hongli Xu*, Hengchang Liu*
*School of Computer Science and Technology, University of Science and Technology of China
†School of Computer Science and Technology, Soochow University
‡School of Software Engineering, University of Science and Technology of China
{guohanso, luciola, sa614201, yzl123, zzy7758}@mail.ustc.edu.cn
{lshuang, xuhongli, hcliu}@ustc.edu.cn, huangh@suda.edu.cn

***Abstract*—In recent years, the number of reports about people assaulted in China and all over the world is significantly increasing, and most of the victims in these assault cases are females, who often lack enough strength to revolt when facing dangerous situations. At the same time smartphones are becoming more and more ubiquitous and being utilized in many fields of our daily lives nowadays, so we argue that smartphones should also play a more important role in the field of personal security. In this paper, we propose a novel system called Guardian Angel for personal security against assaults synthetically leveraging accelerometers, GPS, microphones and vibration motors, which have already been equipped on most off-the-shelf smartphones. As opposed to existing solutions, Guardian Angel follows a more practical, more energy-efficient and more intelligent manner. When confronted with danger, users can ask for assistance by knocking on their smartphones. Guardian Angel recognizes these knock actions with the help of a supervised classifier constructed from thirteen different features. To reduce energy consumption and system overheads ulteriorly, Guardian Angel mines safe locations automatically from GPS positions of users' daily lives utilizing a DBSCAN(Density-Based Spatial Clustering of Applications with Noise)-based algorithm and enters a dormant state when in safe locations. We conduct a series of experiments on Google Nexus 5 smartphones, and evaluation results demonstrate that Guardian Angel achieves 92.8% true positive rate and 0.0% false positive rate about recognizing users' requests and making responses correspondingly.**

***Keywords*—Personal Security; Smartphones; Pattern Recognition; Ubiquitous Computing**

I. INTRODUCTION

Smartphones(and other devices) with abundant advanced built-in sensors are becoming more and more pervasive in our daily lives, which paves the avenues towards a smart world in which an increasing number of ubiquitous computing technologies are explored, proposed and implemented. These significant researches focus on many fields, such as healthcare [1∼4], localization [5] and human computer interaction [6∼9] and make our lives more efficient [10∼12], more intelligent and more enjoyable, but little attention has been paid to the field of personal security. However, violent crimes are seriously threatening the safeties of victims' lives and properties. We are shocked at the statistics of crimes released by the FBI that 1,120,614 incidents of crimes against people with 1,289,799 victims occurred in United States, 2013. In other words, a violent crime occurred in just every 28.1 seconds [13].

†H. Huang is the corresponding author.

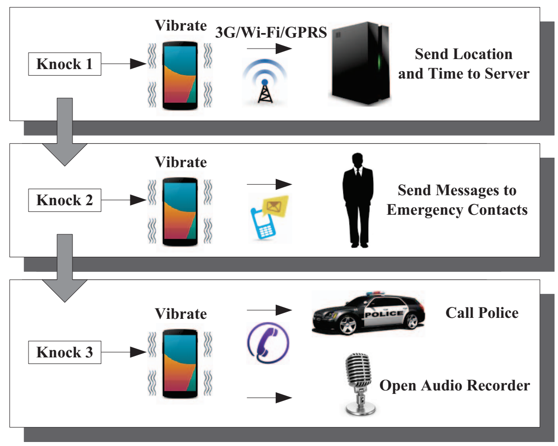


Fig. 1: Overview of Guardian Angel.

Motivated by the aforesaid statistic data and the increasing number of reports about people(especially females) assaulted in China and all over the world, we explore the possibility of applying ubiquitous computing technology to the field of personal security.

In this paper, we present Guardian Angel, a system that exploits the combination of accelerometers, GPS, microphones and vibration motors already equipped on most commodity smartphones and transforms these smartphones into reliable bodyguards. Users can knock on their smartphones asking for help directly through Guardian Angel when any threatening or potentially dangerous situations are encountered. As shown in Fig. 1, we design three kinds of smartphone response behaviors according to the urgency degrees of various circumstances. Every time after recognizing a knock action, the smartphones will vibrate using built-in motors as confirmations to avoid misrecognitions. Users can cancel false alerts(scarcely any) through particular user interfaces of the smartphones.

The main contributions of our work are as follow:

- We proposed Guardian Angel, which is, to the best of our knowledge, the first personal security system leveraging multiple types of sensors embedded in smartphones to interact with users intelligently and effectively. Then we implemented this system on Android-based smartphones and evaluated it in real world.

978-1-4673-7211-4/15 $31.00 © 2015 IEEE
DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.54

- We determined a set of discriminative features that can be utilized to recognize knock actions, some of which have been widely exploited in acoustics-related researches previously. To our surprise, these acoustics-related features also worked pretty well on recognizing knock actions in Guardian Angel.
- We constructed a series of classifiers based on different classification algorithms to recognize knock actions, including Decision Tree, Random Forest, Naive Bayes, Multi Layer Perceptron, k-Nearest Neighbors and Support Vector Data Description. After comprehensive considerations of performance about recognizing knock actions from activities of daily lives and total time consumed, we ultimately chose the Random Forest in Guardian Angel.
- We designed a DBSCAN-based algorithm to mine safe locations from GPS positions of users' daily lives. So when in safe locations, Guardian Angel can enter a dormant state to reduce energy consumption and system overheads ulteriorly.

II. Related Work

Existing Smartphone-Based Works. Personal security-related works received increasing attention especially after the 2012 Delhi gang rape case. EmergenSee [14], for instance, can transmit live videos, audios and GPS positions to pre-selected contacts by tapping a button. There are some other similar off-the-shelf works, including bSafe [15], LifeLine Response [16], Guardian Safety Net [17], Citizen COP [18], Circle of 6 [19] and so on. However, in real urgent situations, all works above share a common drawback: it is still time-consuming for users to take out and unlock smartphones, open applications and tap some specific buttons. iGoSafely [20] is another personal security work whose alert function can be triggered by either unplugging the headsets or shaking the smartphones for a while, which is novel but still not perfect. As for unplugging the headsets, on the one hand, users have to take the headsets with them all the time. On the other hand, false alerts may be caused when users really want to unplug the headsets rather than calling for help, whose number will be numerous when iGoSafely is utilized in practice. Activating the alert function by shaking the smartphones, which is not concealed enough, may push users into more dangerous situations. For example, if the users take out their smartphones and shake them when being pointed by guns, the criminals will certainly be irritated and the smartphones may be robbed. EmergenSee, iGoSafely and other works mentioned above are the best of all off-the-shelf works today, but still not practical and intelligent enough.

Previous Location-Related Researches. Massive significant location-related researches have been conducted. Lian et al. [21] put forward a novel location naming approach, which can provide concrete and meaningful names of locations for GPS coordinates. A ranked list of the most possible semantic names will be presented to users based on their current locations, time and check-in histories. In [22], Yuan et al. proposed a framework focusing on discovering functional zones in cities using human trajectories. Their work provided powerful tools for urban science-related studies. Zheng et al. [23] discovered collaborative locations and activities for recommendation using locations-related GPS positions and users' comments. Yoon et al. [24] designed an efficient itinerary recommendation system based on GPS trajectories collected from multiple users. Locations are not only useful but also private. Suzuki et al. [25] proposed a method to anonymize the users' locations in realistic environments.

III. Recognize Knock Actions

In this section, we firstly describe the computational process of recognizing knock actions. Then we illustrate several activities of daily lives that may be confused with knock actions and present our segmentation algorithm. Finally, we introduce the features employed to recognize knock actions.

A. Computational Process

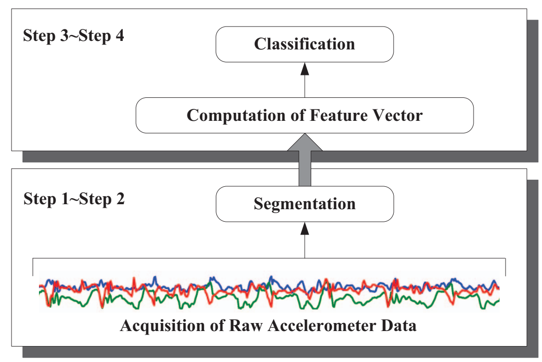


Fig. 2: Computational process of recognizing knock actions.

In our knock actions recognition process, some dedicated processing steps are performed. The first step is the acquisition of raw accelerometer data from smartphones, which are organized in the form of triples(x, y, z) with corresponding timestamps. Then an End-Points Detection based segmentation algorithm is exploited to extract segments we are interested in from accelerometer data time series. In the third step, Guardian Angel computes a feature vector consisting of time-domain and frequency-domain features which contain important cues for distinguishing segments generated by various actions. Finally, Guardian Angel constructs a classifier, which outputs the recognition results, namely, knock actions or other actions using features extracted in the former step.

B. Activities of Daily Lives

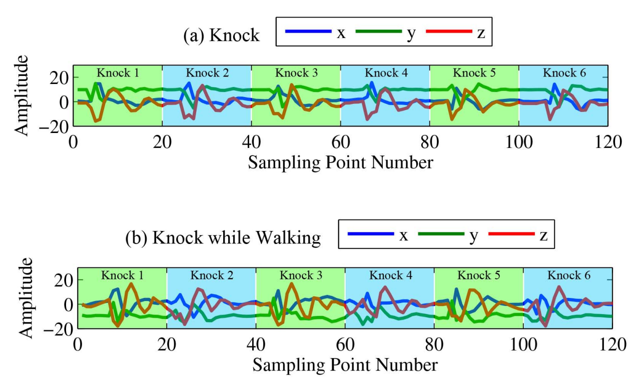


Fig. 3: Accelerometer data generated by knock actions.

In this part, we illustrate several segments generated by knock actions and activities of daily lives. Fig. 3(a) and (b) plot the waveform of accelerometer data generated by knock actions while being stationary and walking respectively. Fig. 4(a), (b) and (c) plot the waveform of accelerometer data generated by three typical activities of daily lives, which are going upstairs and downstairs, walking and running.

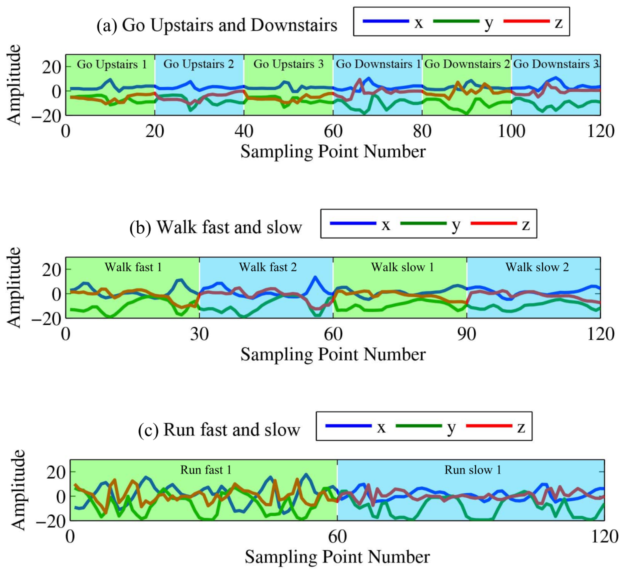


Fig. 4: Accelerometer data generated by activities of daily lives.

C. Segmentation Algorithm

Accelerometer data time series are divided into segments by a segmentation algorithm. Sliding Windows [26], [27] and End-Points Detection [28], [29] are two of the most popular segmentation methodologies. Sliding Windows-based segmentation algorithm produces massive segments, the processing, storage and classification of which will consume enormous energy, computing capabilities and memories, so the segmentation algorithm we utilized in Guardian Angel is based on End-Points Detection, which focuses on particular segments we are interested in and filters out extraneous segments, as shown in Algorithm 1.

Algorithm 1 End-Points Detection based Segmentation

Input:
z_{i1}, z_{i2}, z_{i3}: the most recent three local extrema on z-axis.
t_{i1}, t_{i3}: corresponding timestamps of z_{i1}, z_{i3}.
δ, η: threshold parameters.
Output:
segments we are interested in.
if abs($z_{i2}-z_{i1}$)+abs($z_{i3}-z_{i2}$)$>\delta$ **then**
  if abs($t_{i3}-t_{i1}$)$<\eta$ **then**
    create a new segment S_i;
    calculate t_i ,which is the mean of t_{i1} and t_{i3};
    add the closest ten sampling points before t_i to S_i;
    add the closest ten sampling points after t_i to S_i;
    output S_i;
  end if
end if

D. Computation of Feature Vector

Features can be regarded as abstractions of raw accelerometer data and proper features can represent the main characteristics of data segments accurately. In this paper, we extract a set of features, which are organized in the form of a feature vector, from massive raw accelerometer data. Then the feature vector can be utilized as inputs to classification algorithms to recognize knock actions among activities of daily lives. All features are divided into three categories, namely time domain features, frequency domain features, and statistics features, as shown in TABLE. I.

Features		
Time	3	X-Min., X-Max., AbsX-Ave.
	3	Y-Min., Y-Max., AbsY-Ave.
	3	Z-Min., Z-Max., AbsZ-Ave.
	1	All-Ave.
Statistics	1	Z-Kurtosis
Frequency	1	Z-Spectral Slope
	1	Z-Peak Frequency

TABLE I: Feature set for recognizing knock actions.

- **Time Domain Features.** Time domain features focus on waveform characteristics and they can be obtained from data segments directly with very small computational complexity and storage memory.
Min. & Max. & Ave. Amplitude. These features describe the basic shape of accelerometer data and have been extensively exploited in various works, especially threshold-based algorithms.

- **Statistics Features.** Statistics features capture the distribution characteristics of consecutive accelerometer sampling points.
Kurtosis. This feature weighs how the amplitude decays near the extreme points, namely the peakedness and flatness. Larger kurtosis values indicate more peaked distribution. The Kurtosis of segment S_i is calculated as:
$$Kurtosis_i = \frac{m_i \sum_{j=1}^{m_i} (z_{ij} - \bar{z}_i)^4}{(\sum_{j=1}^{m_i} (z_{ij} - \bar{z}_i)^2)^2}$$
where z_{ij} indicates the z-axis value of the j-th sampling point in segment S_i and $\bar{z}_i$ indicates the mean z-axis value of all sampling points in segment S_i. m_i is the total number of sampling points in segment S_i.

- **Frequency Domain Features.** Frequency domain features pay attention to periodic natures. We transform the time series of accelerometer data into spectrum employing Fast Fourier Transform(FFT) in this paper.
Spectral Slope. This feature denotes the energy distribution at various frequencies. In this paper, we apply linear regression to accelerometer data after Fast Fourier Transformation, which produces a numerical result indicating the slope of the best-fitting line through the spectrum on frequency domain. The Spectral Slope of segment S_i is

calculated as:

$$SpectralSlope_i = \frac{1}{\sum_{j=1}^{n_i} a_{ij}} \frac{n_i \sum_{j=1}^{n_i} a_{ij} f_{ij} - \sum_{j=1}^{n_i} a_{ij} \sum_{j=1}^{n_i} f_{ij}}{n_i \sum_{j=1}^{n_i} f_{ij}^2 - (\sum_{j=1}^{n_i} f_{ij})^2}$$

where f_{ij} indicates the value of the j-th frequency component of segment S_i and a_{ij} indicates the corresponding amplitude of frequency component f_{ij}. n_i is the total number of frequency components of segment S_i.

Peak Frequency. This feature measures the frequency at which the maximum spectral amplitude occurs. The Peak Frequency of segment S_i is calculated as:

$$PeakFrequency_i = f_{ij'}$$
$$j' = \underset{j \in [0, n_i]}{\operatorname{argmax}} |F_j(z_i)|$$

where $|F_j(z_i)|$ indicates the amplitude of the j-th frequency component of segment S_i after Fast Fourier Transformation and z_i indicates the z-axis values of all sampling points in segment S_i.

IV. Mine Safe Locations

Algorithm 2 Mine Safe Locations

Input:
D_r: data set of n_r GPS positions.
ε: radius parameter.
$MinPts$: density threshold parameter.
Output:
D_s: data set of n_s GPS positions, which are safe locations.
Phase 1:
for each GPS position d_i in D_r **do**
  mark d_i as **unvisited**;
  calculate den_i, which is the number of GPS positions in ε-neighborhood of d_i;
end for
Phase 2:
while at least an **unvisited** GPS position **do**
  randomly select an **unvisited** GPS position d_i;
  mark d_i as **visited**;
  if $den_i > MinPts$ **then**
    create a new cluster C_i;
    add d_i to C_i;
    for each GPS position d_j in ε-neighborhood of d_i **do**
      if d_j is **unvisited then**
        mark d_j as **visited**;
        if $den_j > MinPts$ **then**
          Depth First Growth(d_j, C_i);
        end if
        add d_j to C_i;
      end if
    end for
  end if
end while
Phase 3:
for each cluster C_s **do**
  calculate c_s, which is the center of C_s;
  add c_s to D_s;
end for

In this section, we introduce the algorithm utilized to mine safe locations, which is based on DBSCAN(Density-Based Spatial Clustering of Applications with Noise) [30].

If the ε-neighborhood of a GPS position contains at least $MinPts$ other GPS positions, then this GPS position is a core GPS position. Primarily, our algorithm finds core GPS positions and their neighborhoods within a radius ε using the algorithm described in Algorithm 2 Phase 1. Then our algorithm connects core GPS positions and their neighborhoods recursively to form dense regions as clusters. Related algorithms are provided in Algorithm 2 Phase 2 and Algorithm 3. Finally as shown in Algorithm 2 Phase 3, our algorithm calculates the centers of these clusters as safe locations.

Algorithm 3 Depth First Growth

Input:
d_j: a GPS position in D_r.
C_i: a cluster of GPS positions.
for each GPS position d_k in ε-neighborhood of d_j **do**
  if d_k is **unvisited then**
    mark d_k as **visited**;
    if $den_k > MinPts$ **then**
      Depth First Growth(d_k, C_i);
    end if
    add d_k to C_i;
  end if
end for

V. System Implementation

We implemented Guardian Angel, our proposed personal security system, on Google Nexus 5 smartphones running Android 5.0.1 operating system. Fig. 5 presents five screen shots to illustrate the key functions of Guardian Angel.

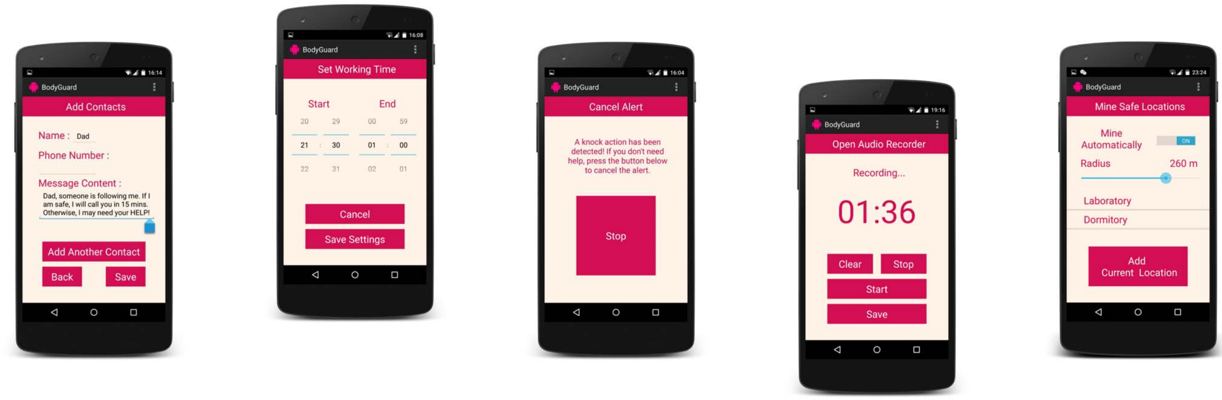
Add Emergency Contacts. When incidents occur, Guardian Angel will send messages with real-time GPS positions to users' pre-selected emergency contacts immediately. Users can add new contacts through an interface, as shown in Fig. 5(a).

Set Working Time. In order to save energy and system resources, users are allowed to set working time through the interface presented in Fig. 5(b), only during which Guardian Angel will be in working status.

Cancel Alerts. If any false alert occurs(scarcely any), users can cancel it through the big button illustrated in Fig. 5(c).

Open Audio Recorders. Audio recordings of crime scenes are significant evidences, which will be extremely helpful for police to solve cases. So when users are in danger, Guardian Angel will activate the recorders automatically, as shown in Fig. 5(d).

Mine Safe Locations. Another novel function which reflects the intelligence of Guardian Angel is that it can mine safe locations automatically. The most important benefit of this function is energy saving. Guardian Angel mines the most frequent n locations from GPS positions of users' daily lives in the last seven days and regards them as safe locations. When in the areas within r meters around the safe locations, Guardian Angel stops sampling accelerometer data and enters a dormant state, which can save a tremendous amount of energy. r can be pre-defined by users through the interface presented in Fig. 5(e).



(a) Add Emergency Contacts (b) Set Working Time (c) Cancel Alerts (d) Open Audio Recorders (e) Mine Safe Locations

Fig. 5: Key functions of Guardian Angel.

VI. Evaluation

In this section, we present the results of our experiments. This section consists of three parts. In the first part, we demonstrate the experimental results of recognizing knock actions. In the following part we study the experimental results of mining safe locations. Finally, we report the real world evaluation results of our system in the third part.

A. *Recognizing Knock Actions*

1) Time of Calculating Every Feature Vector

TABLE. II shows the average time of calculating every feature vector from a dataset containing 100 segments. We observe that the whole feature vector computation process can be accomplished within about 0.55ms.

Features		Time
Time	AbsX-Ave.	(0.38±0.05)ms
	X-Min., X-Max.	
	AbsY-Ave.	
	Y-Min., Y-Max.	
	AbsZ-Ave.	
	Z-Min., Z-Max.	
	All-Ave.	(0.18±0.06)ms
Statistics	Z-Kurtosis	(0.20±0.08)ms
Frequency	Z-Spectral Slope	(0.09±0.03)ms
	Z-Peak Frequency	(0.14±0.05)ms
Total		(0.55±0.05)ms

TABLE II: Time(Avg.±Std.Dev.) of calculating every feature vector.

2) Experimental Results of Recognizing Knock Actions

In this part, we examine how well Guardian Angel can recognize knock actions from activities of daily lives utilizing six different classification algorithms. As shown in TABLE. III, our experimental dataset contains 100 knock actions performed when users are under still status, 80 knock actions performed when users are walking and 146672 segments generated by all kinds of activities in 4 days of daily lives.

Activities	Total
Knock Actions(still)	100 times
Knock Actions(while walking)	80 times
Daily Lives (Walk, Run, Go Upstairs and so on)	4 days

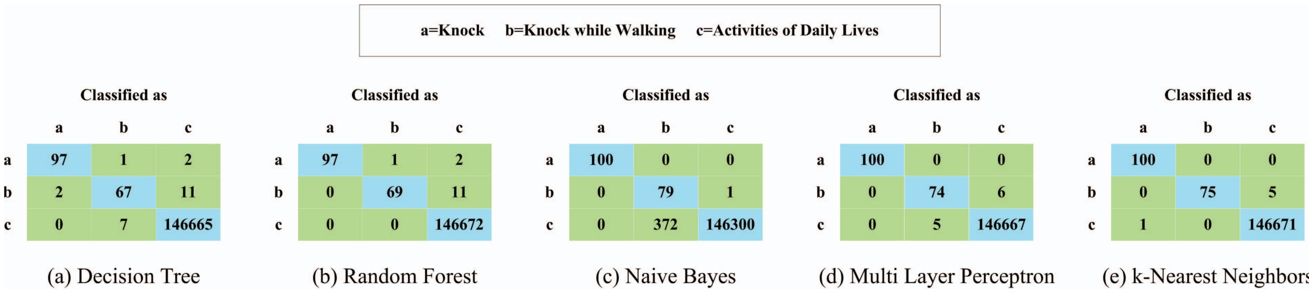
TABLE III: Experimental dataset for recognizing knock actions.

Multi-Class Classification Algorithms. We construct five multi-class classifiers based on different classification algorithms [31], [32], which are Decision Tree, Random Forest, Naive Bayes, Multi Layer Perceptron and k-Nearest Neighbors respectively. Then we conduct a series of 10-fold cross-validation experiments: Firstly, we randomly partition our experimental dataset into 10 mutually exclusive folds and all folds each have an approximately equal size. Secondly, training and testing processes are performed 10 times, that is to say, in the i-th iteration, fold i is retained for testing and the remaining 9 folds are used for training. Fig. 6 presents the confusion matrixes, Fig. 7 illustrates the true positive rate(left), false positive rate(right) and Fig. 8 reports the total time consumed in 10-fold cross-validation classification experiments of these five classifiers. The true positive rate and false positive rate are calculated as:

$$TP\ Rate = \frac{\#\ of\ True\ Positives}{\#\ of\ (True\ Positives + False\ Negatives)}$$

$$FP\ Rate = \frac{\#\ of\ False\ Positives}{\#\ of\ (True\ Negatives + False\ Positives)}$$

We can make three main observations here. Firstly, as presented in Fig. 6 and Fig. 7, Naive Bayes outputs the largest number(372 times) of false positive recognitions of knock actions, which is much larger than the numbers output by other classification algorithms. But at the same time, Naive Bayes also achieves the highest true positive rate of recognizing knock actions, which can recognize 99.4% knock actions correctly. Secondly, as illustrated in Fig. 6 and Fig. 7, Random



a=Knock b=Knock while Walking c=Activities of Daily Lives

(a) Decision Tree

	Classified as a	b	c
a	97	1	2
b	2	67	11
c	0	7	146665

(b) Random Forest

	Classified as a	b	c
a	97	1	2
b	0	69	11
c	0	0	146672

(c) Naive Bayes

	Classified as a	b	c
a	100	0	0
b	0	79	1
c	0	372	146300

(d) Multi Layer Perceptron

	Classified as a	b	c
a	100	0	0
b	0	74	6
c	0	5	146667

(e) k-Nearest Neighbors

	Classified as a	b	c
a	100	0	0
b	0	75	5
c	1	0	146671

Fig. 6: Confusion Matrixes of different classifiers.

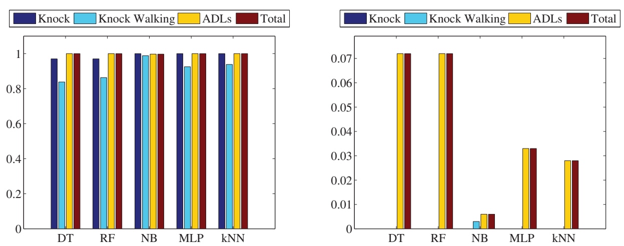


Fig. 7: TP Rate(left) and FP Rate(right) of different classifiers.

Forest is the only classification algorithm which outputs 0.0% false positive rate of recognizing knock actions, and the true positive rate is as high as 92.8%. Thirdly, as reported in Fig. 8, the total time consumed in 10-fold cross-validation classification experiments of k-Nearest Neighbors and Multi Layer Perceptron are 1835.48s and 302.09s respectively but the total time of Random Forest is only 6.82s. The former two are 269 and 44 times as long as the latter one respectively.

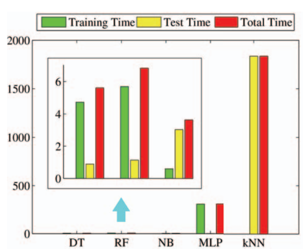


Fig. 8: Total time consumed of different classifiers.

One-Class Classification Algorithm. We construct a one-class classifier based on SVDD(Support Vector Data Description) [33]. The numbers of knock actions in training set and testing set are 162 and 18 respectively and the number of segments of daily lives in testing set is 146672. The true positive rate, false positive rate, false negative rate and precision of recognizing knock actions of SVDD are shown in TABLE. IV. The false negative rate and precision are calculated as:

$$FN\ Rate = \frac{\#\ of\ False\ Negatives}{\#\ of\ (True\ Positives + False\ Negatives)}$$

$$Precision = \frac{\#\ of\ True\ Positives}{\#\ of\ (True\ Positives + False\ Positives)}$$

TP Rate	FP Rate	FN Rate	Precision	Time
1.000	0.087	0	0.014	122.47s

TABLE IV: Recognition results of SVDD.

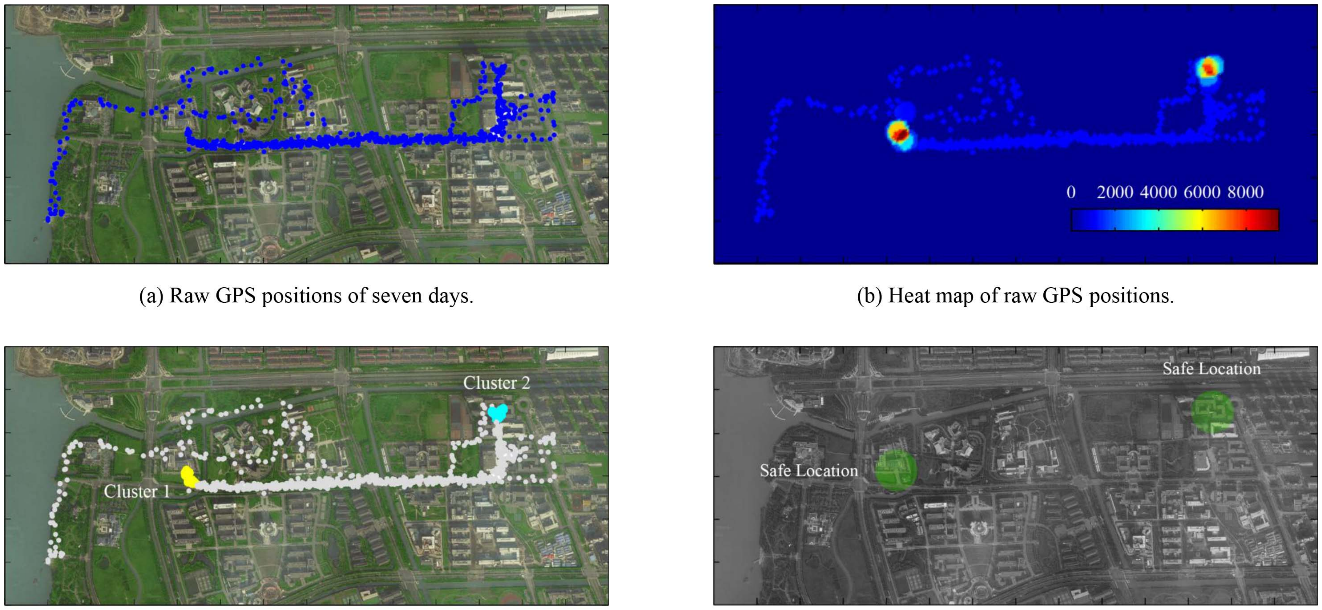
We can make three main observations from TABLE. IV. Firstly, SVDD can recognize 100.0% knock actions correctly. Secondly, meanwhile, SVDD outputs numerous false positive recognitions of knock actions. Thirdly, SVDD is also time-consuming. The total time consumed for classification of SVDD is 122.47s. As mentioned before, Random Forest can complete the classification process(both training and testing) 10 times on same dataset within only 6.82s. Even so, the former is 18 times as long as the latter.

B. *Mining Safe Locations*

1) Experimental Results of Mining Safe Locations

In this part, we present the experimental results of mining safe locations. Guardian Angel logs GPS positions once within every 30 seconds. What is worth mentioning here is that smartphones can't receive the GPS signal when in buildings. So Guardian Angel will log the last GPS position as current one when the GPS signal is not available. To save energy, if the GPS signal can't be detected for too long, GPS will be closed for a while, during which Guardian Angel will still keep logging the last GPS position in every sampling period. The acquisition of GPS positions will not consume too much energy because users often stay in buildings for most of the time.

Guardian Angel mines safe locations from GPS positions of the last seven days utilizing our DBSCAN-based algorithm. The experimental results are presented in Fig. 9. Fig. 9(a) plots all these raw GPS positions, among which many are **overlapped** because GPS positions logged by Guardian Angel are the same when smartphones are in buildings. Fig. 9(b) draws the heat map of these raw GPS positions, which reflects the density of GPS positions. We can observe two obvious hotspots, which are the laboratory and dormitory respectively. We regard a place as a safe location, if people stay there for more than five hours a day, so we assign 4200 to density threshold parameter $MinPts$ in Algorithm 2 according to the sampling period(30s) of Guardian Angel. We utilize 50(m) as the value of the radius parameter ε in Algorithm 2. Experimental results output by our DBSCAN-based algorithm are illustrated in Fig. 9(c) and (d). Excluding noise data, which are colored in gray, all GPS positions are divided into two



(a) Raw GPS positions of seven days.

(b) Heat map of raw GPS positions.

(c) Clusters output by our DBSCAN-based algorithm.

(d) Results of mining safe locations.

Fig. 9: Experimental results of Guardian Angel about mining safe locations.

clusters as shown in Fig. 9(c), and we mark these two clusters using yellow and cyan respectively. In Fig. 9(d), there are two green circles representing safe locations, whose centers are calculated from the two clusters mentioned in Fig. 9(c).

C. *Real World Evaluation*

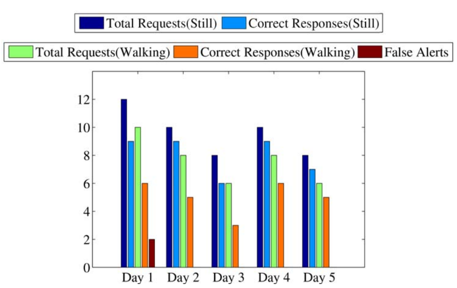


Fig. 10: Results of responding to users' requests in real world.

Finally, we evaluate how well Guardian Angel can respond to users' requests in real world. Fig. 10 illustrates the number of requests performed by users under the condition of being still and walking respectively, correct responses under corresponding conditions and false alerts. We can observe that the number of requests varied between 14 and 22 in these five days. Of all the 86 requests, Guardian Angel responded to 65 requests correctly with only 2 false alerts in all five days, which we believe are extremely few. In our interviews with the participants in our experiments, they said that they did't feel uncomfortable from these few false alerts that were raised by Guardian Angel. In summary, the real world evaluation results provide strong evidence of the effectiveness of Guardian Angel in responding to users' requests in real world scenarios.

VII. Discussion

Crime Rate. In future, we will take historical crime rate data into consideration to make Guardian Angel more intelligent and more efficient. In other words, Guardian Angel will compare the historical crime rate data which corresponds to current time and location with certain thresholds in real time and design diverse strategies accordingly. For example, if users are in low-crime areas, Guardian Angel will turn off the accelerometers, which can reduce energy consumption and system overheads ulteriorly.

Acoustic Clips. Depending on analyzing numerous supervisory video recordings, we find that there are a variety of acoustic clues (e.g., "help") in potentially dangerous and criminal situations. So in future, we will exploit these acoustic hints using speech recognition technology in Guardian Angel to ulteriorly improve the user experience, such as usability, creditability and so on. Sphinx [34] is an open source speech recognition engine provided by Carnegie Mellon University, which can be integrated in Guardian Angel. In future, the Guardian Angel will combine continuous Hidden Markov Models with Conditional Random Fields and utilize Mel-Frequency Cepstral Coefficients as features to implement its speech recognition function.

Interviews with Participants. In our exit interviews with the participants, they frankly love this personal security system. Many of them, especially some girls, want Guardian Angel on their personal smartphones immediately. A participant, who is a girl, said, "Guardian Angel is really amazing. I feel at ease when it is with me though I walk late and alone at night. When can I get it?"

VIII. Conclusion

In this paper, we present Guardian Angel, which is, to the best of our knowledge, the first personal security system synthetically utilizing smartphones' built-in sensors, including accelerometers, GPS, microphones and vibration motors. Guardian Angel recognizes knock actions with the help of a supervised multi-class classifier constructed from thirteen features. To save energy consumption and system overheads, Guardian Angel mines safe locations automatically leveraging a DBSCAN-based algorithm and enters a dormant state when in safe locations. We implemented our proposed system on Android based smartphones and conducted a series of experiments. Real world evaluation results demonstrate that Guardian Angel can respond to users' requests correctly with few false alerts, which is quite promising for emergency alerting on personal security.

IX. Acknowledgement

This paper was supported by National Science and Technology Major Project under Grant No. 2012ZX03005009, National Science Foundation of China under Grant No. U1301256, 61170058, 61272133, BJ2260080039 and G-G2260080042, Special Project on IoT of China NDRC(2012-2766), Research Fund for the Doctoral Program of Higher Education of China No. 20103402110041, and USTC Innovation funding DG2260010011.

References

[1] Y. Tang, S. Wang, Y. Chen, and Z. Chen, "Ppcare: A personal and pervasive health care system for the elderly," in *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*. IEEE, 2012, pp. 935–939.

[2] Z. Zhu, X. Su, F. Jiang, J. Li, and J. Ye, "A user-centric pervasive computing services model for medical and health-care," in *Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on*. IEEE, 2007, pp. 841–846.

[3] K.-C. Wang, M.-C. Huang, Y.-H. Hsieh, S.-Y. Lau, C.-H. Yen, H.-L. C. Kao, C.-W. You, H.-H. Chu, and Y.-C. Chen, "Soberdiary: a phone-based support system for assisting recovery from alcohol dependence," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 2014, pp. 311–314.

[4] L. Chen, I. Hussain, R. Chen, W. Huang, and G. Chen, "Blueview: a perception assistant system for the visually impaired," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 143–146.

[5] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[6] F. Tian, L. Xu, H. Wang, X. Zhang, Y. Liu, V. Setlur, and G. Dai, "Tilt menu: using the 3d orientation information of pen devices to extend the selection capability of pen-based user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 1371–1380.

[7] C. Shin, J.-H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 173–182.

[8] S. T. Perrault, E. Lecolinet, J. Eagan, and Y. Guiard, "Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 1451–1460.

[9] H. Watanabe, T. Terada, and M. Tsukamoto, "Ultrasound-based movement sensing, gesture-, and context-recognition," in *Proceedings of the 2013 International Symposium on Wearable Computers*. ACM, 2013, pp. 57–64.

[10] M. He, A. Chin, E. Chen, and J. Tian, "Efficient personalized recommendation of mobile web content using an em-based clustering method."

[11] M. Lv, D. Zhang, L. Chen, and G. Chen, "Detecting traffic congestions using cell phone accelerometers," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 2014, pp. 107–110.

[12] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 316–324.

[13] "FBI Statistics of Crimes," http://www.fbi.gov/stats-services/crimestats.

[14] "EmergenSee," http://emergensee.com/.

[15] "bSafe," http://getbsafe.com/.

[16] "LifeLine Response," http://www.llresponse.com/.

[17] "Guardian Safety Net," http://www.guardiansafetynet.com/.

[18] "Citizen COP," http://www.citizencop.org/.

[19] "Circle of 6," http://www.circleof6app.com/.

[20] "GoSafely," http://www.igosafely.com/.

[21] D. Lian and X. Xie, "Learning location naming from user check-in histories," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 112–121.

[22] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," 2014.

[23] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with gps history data," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1029–1038.

[24] H. Yoon, Y. Zheng, X. Xie, and W. Woo, "Smart itinerary recommendation based on user-generated gps trajectories," in *Ubiquitous Intelligence and Computing*. Springer, 2010, pp. 19–34.

[25] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio, "A user location anonymization method for location based services in a real environment," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 398–401.

[26] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, and G. J. Norman, "Classification accuracies of physical activities using smartphone motion sensors," *Journal of medical Internet research*, vol. 14, no. 5, p. e130, 2012.

[27] N. Győrbíró, Á. Fábián, and G. Hományi, "An activity recognition system for mobile phones," *Mobile Networks and Applications*, vol. 14, no. 1, pp. 82–91, 2009.

[28] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 234–239.

[29] T. Gu, S. Chen, X. Tao, and J. Lu, "An unsupervised approach to activity recognition and segmentation based on object-use fingerprints," *Data & Knowledge Engineering*, vol. 69, no. 6, pp. 533–544, 2010.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[31] Y. Zheng, B. Jeon, D. Xu, Q. J. Wu, and H. Zhang, "Image segmentation by generalized hierarchical fuzzy c-means algorithm," *Journal of Intelligent and Fuzzy Systems*.

[32] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for ν-support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.

[33] D. Tax, "DDtools, the Data Description Toolbox for Matlab," July 2014, version 2.1.1.

[34] "Carnegie Mellon University Sphinx Speech Recognition Engine," http://cmusphinx.sourceforge.net/.