CHANGE OF POSSESSION/NON-POSSESSION DETECTION FOR SMARTPHONES

By

Savitri Gadagi

A thesis submitted in partial fulfillment

of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Kiran Balagani Dr. Paolo Gasti

Thesis Advisor Thesis Co-Advisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Anand Santhanakrishnan Dr. Batu Chalise

Committee Member Committee Member

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Nada Anid

Dean of the School of Engineering & Computing Sciences

NEW YORK INSTITUTE OF TECHNOLOGY

New York, NY

2017

Copyright © Savitri Gadagi 2017

All Rights Reserved

### ABSTRACT

Change of possession/non-possession detection for smartphones

By

Savitri Gadagi, Master of Science

New York Institute of Technology, 2017

In this thesis, we investigated whether change of possession and non-possession of smartphone can be detected using movement signals collected from the smartphones. We aim to propose a mechanism that augments continuous authentication mechanisms by minimizing number of authentication points. Instead of performing periodic authentication, we look for the occurrence of events of interest that indicate change of possession of the smartphone. We evaluated detection accuracies on data collected from 29 smartphone users over multiple sessions. Data was collected while users performed typing activity in walking condition. During each session, users were involved in four activities: (1) the user hands the smartphone to the proctor (give); (2) the proctor forcefully grabs the smartphone from the user (grab); (3) the user places the smartphone on the table to perform a task (Jenga); and (4) the user turns back while walking (turn). We used 37 sessions to train a classifier (random forest) on these events, and 15 sessions to evaluate the classifier’s performance. Our experiments show that the proposed technique can distinguish between one of these events and the lack of events with a true positive (TP) rate of 92.6% and 97.4% respectively. Further, our technique can distinguish between give, grab and lack of events with a TP rate of 81%, 43.8% and 97% respectively.

### ACKNOWLEDGMENTS

I express my sincere and heartfelt gratitude to Dr. Paolo Gasti and Dr.Kiran Balagani. It has been such an honor to be your student, and to work with you. Thank you for your mentorship, kindness, guidance, support and inspiration.

My sincere thanks to committee members, Dr. Anand Santhanakrishnan and Dr. Batu Chalise, for their valuable insights, guidance, and support.

I express my very profound gratitude to my parents and to my husband, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Savitri Gadagi

**TABLE OF CONTENTS**

ABSTRACT iii

ACKNOWLEDGMENTS iv

CHAPTER 1 : INTRODUCTION 1

1.1 Contributions and novelty of this work 3

1.2 Organization 3

CHAPTER 2 : RELATED RESEARCH 4

CHAPTER 3 : OUR DATASET 7

3.1 Data Collection 7

3.2 Data Preprocessing 9

CHAPTER 4 : TRAINING AND TESTING 11

CHAPTER 5 : FEATURES AND CLASSIFIERS 15

5.1 Time domain features 15

5.2 Frequency domain features 16

CHAPTER 6 : RESULTS AND DISCUSSIONS 18

6.1 Influence of event covered by the window 18

6.2 Influence of window size 20

CHAPTER 7 : CONCLUSION AND FUTURE WORK 24

REFERENCES 25

# 

#### INTRODUCTION

Currently, user authentication on smartphone is performed when the user unlocks the phone. This type of authentication is commonly referred to as one-time authentication. At present, majority of the phones use one-time authentication methods such as passcodes, graphical patterns, and fingerprint. An important drawback of one-time authentication is that the adversary can take over the smartphone post-authentication. For instance, if a user forgets to lock the smartphone and leaves it unattended, adversary can use it without needing authentication. Under this attack, the adversary has unmitigated access to the content of the smartphone. In addition, one-time authentication methods are vulnerable to guessing [8], spoofing [15] and side channel attacks [6].

With continuous authentication, theuser is authenticated at fixed intervals post login. Common mechanisms used for one-time authentication are not suitable for continuous authentication, because these techniques require explicit input, and are therefore disruptive of the user’s workflow. To address this problem, researchers have been looking at behavioral biometric signals, such as hand movement [1], gait [13], voice [14], touchscreen interaction [2-8], and keystroke dynamics [9-12], as a way to continuously authenticate users. These signals are generated during normal smartphone use, and can be collected in the background, without interrupting the user’s workflow. Though continuous authentication of smartphone users has a security advantage over login-time authentication, it also has several drawbacks:

1. Because of possibly high authentication latency (usually between 30 seconds and 2 minutes), the adversary can gain possession of the smartphone well before the end of the current authentication window. For this reason, it would be ideal to re-authenticate as soon as possible after the adversary has gained possession of the smartphone.
2. It is unnecessary to re-authenticate a user who has been recently authenticated, if the phone can determine that no other user has had access to the smartphone since authentication. By authenticating only after detecting an event of interest, the smartphone can reduce the number of authentication points within a session, and as a result minimize the authentication energy consumption, and reduce the occurrence of false rejections.

In this thesis, we present a protocol that identifies events of interests, such as change of possession and non-possession, with the goal of minimizing the number of authentication points, and to reduce authentication latencies when events of interest are detected. We aim to determine if the legitimate user is still in possession of the smartphone. Instead of periodic authentication, we look for events, which signify disruption of users possession of the smartphone. Events of interest for this study include:

1. Change of possession events: The user loses the possession of the phone to someone else. For example, the adversary might grab the smartphone or the user voluntarily hands over the smartphone to someone else.
2. Non-possession events: The user loses possession of the phone by putting the smart phone on a table or in a bag.

These events are considered in contrast to a Non-Event, which indicates that the user continuously possesses the smartphone.

## Contributions and novelty of this work

In this thesis, we propose a novel change of possession and non-possession detection framework for smartphones. Our evaluation of data collected from 29 subjects shows that movement signals captured during an event differ from comparison to a non-event in measurable ways. The proposed system is able to detect events and non-events with TP rate of 92.6% and 97.4%, and FP rate of 2.6% and 7.4% respectively and, is able to distinguish between a give, grab and non-event with TP rate of 81%, 43.8% and 97%, FP rate of 2.8%, 6% and 9.5% respectively.

## Organization

This thesis is organized as follows. Chapter 2 reviews the related research in this field. Chapter 3 provides details on our dataset and describes steps involved in cleaning the dataset. Chapter 4 presents the protocol involved in preparing training and testing sets. In Chapter 5, we discuss computation of features and the classifiers used. In Chapter 6, we present the results of our evaluation. In Chapter 7, we conclude the thesis and propose future work.

# 

#### RELATED WORK

Smartphones incorporate sophisticated sensors, such as inertial sensors, camera, touchscreen, and magnetic compass. Inertial sensors are explored in various papers for tasks including activity recognition [18], sleep monitoring [21], fitness monitoring [22], and smartphone authentication [1-7]. One of the very promising areas where sensors are being studied is to measure the touch data and micro movements of the smartphones, as users interact with their smartphones. These behavioral signals can be continuously measured to authenticate the smartphone users. Various behavior-based continuous authentication strategies have been proposed that employ users walking pattern [13], keystroke pattern [9-12], and touchscreen input behavior [2-6]. Though continuous authentication of smartphone users has a security advantage, it has drawbacks in terms of authentication latency and energy usage. Frank.et.al [3] employed touchscreen input as a data for continuous authentication. They proposed 30 features generated from user interactions with smartphone using swipe gestures. Frank et al. showed that the basic navigation activities like scrolling are distinct enough for continuous authentication of the smartphone users. To show this result, they used 11-12 strokes. They were able to achieve an EER of 2%-3%, and authentication latencies of 43 seconds. In practice, the authentication latency could be higher than 43 seconds based on the application context. For instance, a user watching a video on her smartphone might need several minutes to generate 11-12 strokes. This renders continuous authentication ineffective when the adversary gains possession of the phone before next authentication, because the adversary will be able to use the phone for 21 seconds on average. Our work addresses the challenges of continuous authentication by immediately authenticating if users activity is disrupted.

There are few strategies proposed to address energy expenditure of smartphone user authentication. First, Sedenka et al. [24] used sensor based optimization strategies, such as switching the sensors on and off depending on contextual attributes [25] and reducing sensor sampling rate [24]. Second, Riva et al. [23] proposed progressive authentication to reduce number of authentication points. The proposed system used face recognition, voice recognition, proximity, and motion sensors to identify legitimate user and reduced number of explicit authentications by 42%. This work shares common goal with our work in terms of reducing number of authentication points. However, our work differs from [23] as follows: (1) we used population data to identify change of possession of the smartphone, while [23] used face recognition, voice, and proximity data to establish user identity. Use of face, voice and proximity data raises security and privacy concerns, as the data carries personally identifiable data (who the user is?) and expose private information (what is user saying what is user’s location?). (2) We performed experiments on dataset containing 29 users, while [23] evaluated their protocol from dataset of 9 users. (3) We were able to detect change of possession, just by using inertial sensors embedded in the smartphone, while [23] used proximity and motion sensors to detect whether the phone is next to another device of same user. This networked communication between devices might be expensive in terms of smartphone’s energy consumption.

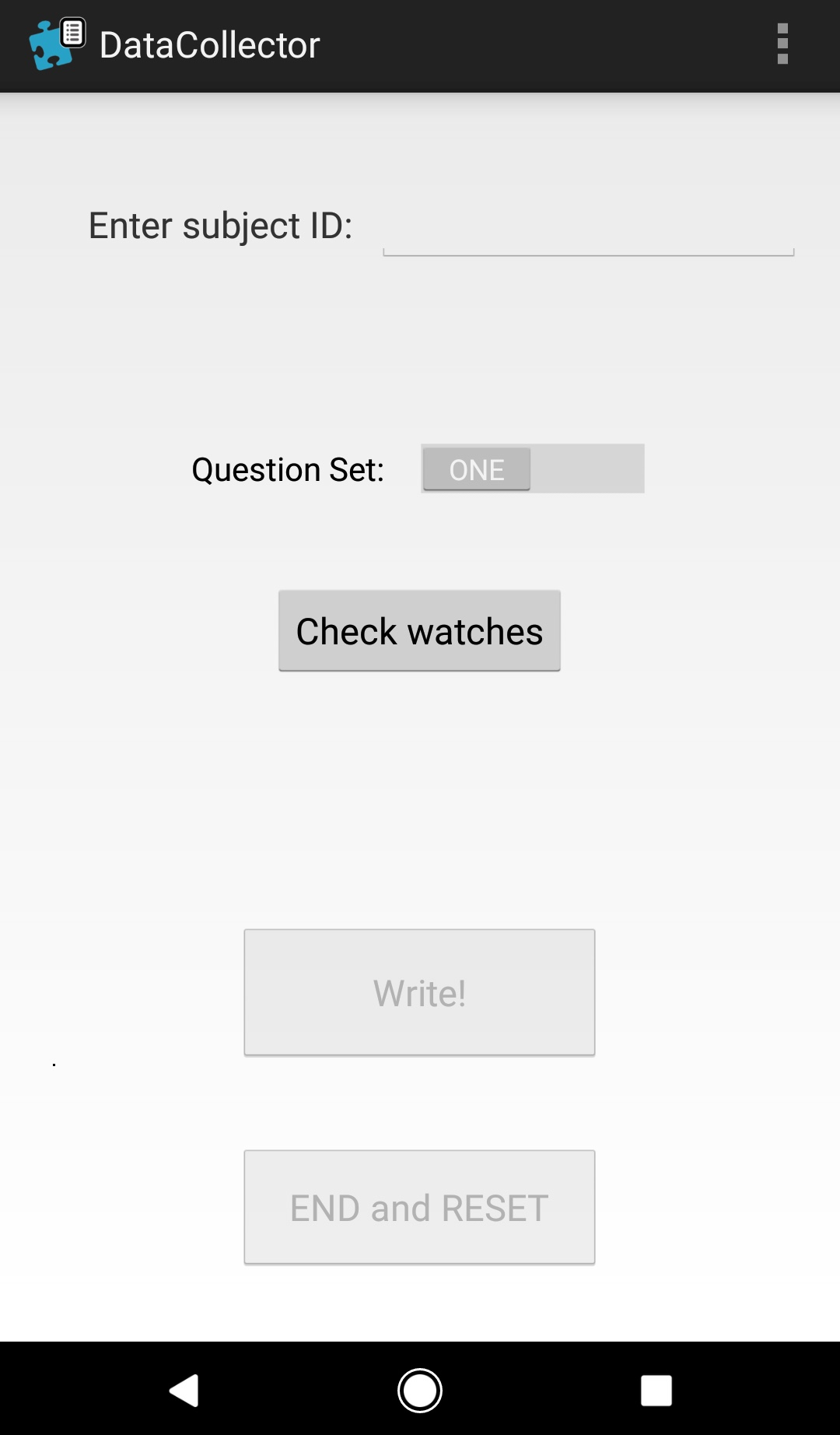
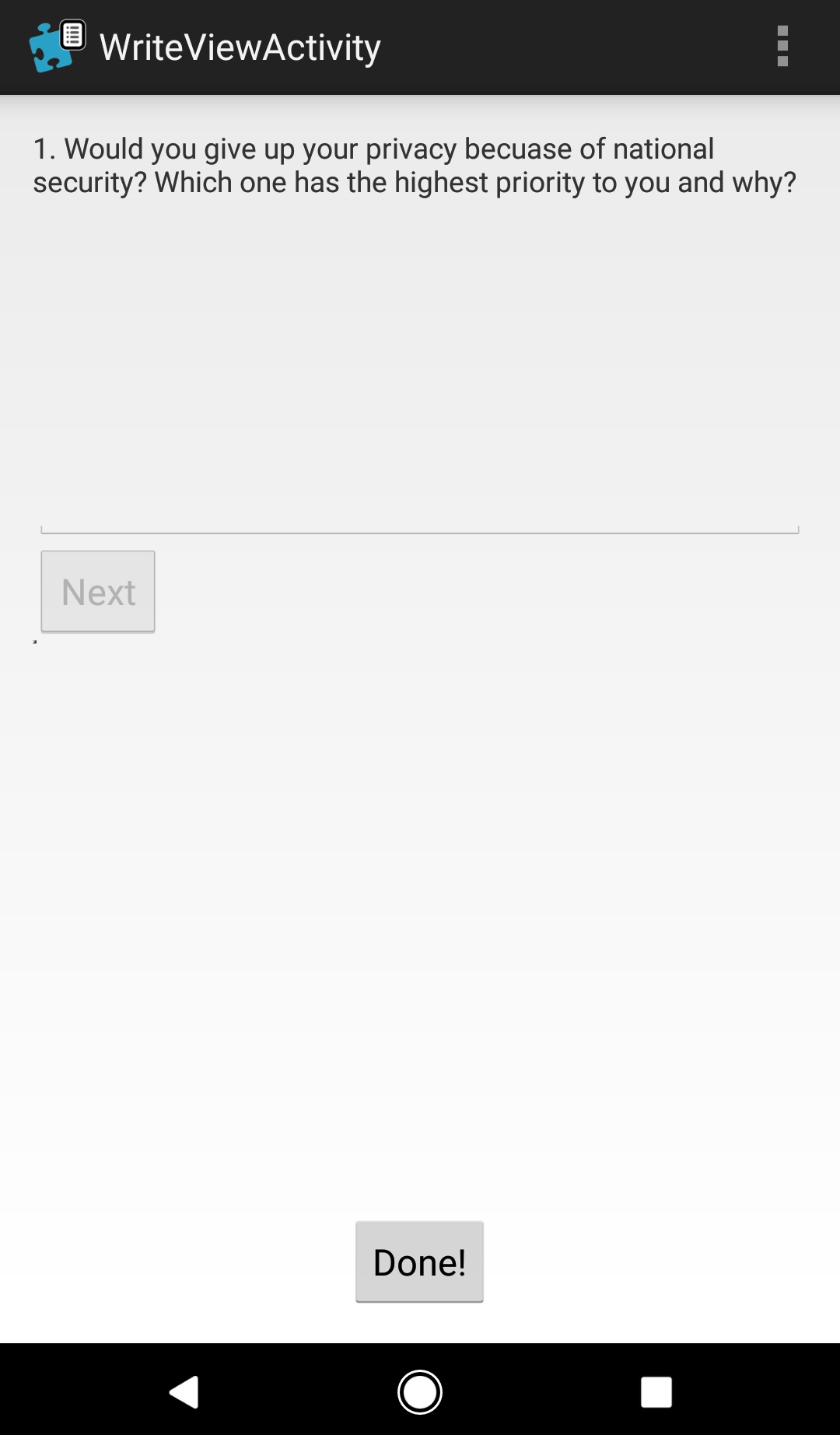
In this thesis, we build on the work of Chamani [26] on change of possession detection system for smartphone users. This work extends Chamani’s work as follows: (1) in Chamani’s work, only two event types (grab and give) were annotated as “events”. The signals in rest of the event space were considered as non-events. In particular, the movement signals during Jenga were considered non-events, and actions performed by the proctor when she handed the phone back to subjects were also considered non-events. Because Jenga generates movements similar to give and grab, an events and actions performed by proctor do not belong to the user, we annotate them as separate events. (2) Chamani extracted features exclusively from accelerometer signals. We extended the protocol to extract features from gyroscope signals, and established a pipeline to combine features from accelerometer and gyroscope. (3) Chamani’s work derived frequency features from accelerometer signal, while in this thesis, we introduced time domain features and report comprehensive performances using frequency domain features, time domain features and a combination of both.

# 

#### OUR DATASET

## Data Collection

We collected data from 29 smartphone users from a population of graduate and undergraduate students (9 male, and 20 female). 23 participants attended two sessions, and 6 participants attended one session. This resulted in total of 52 sessions. Each session took about 15 minutes to complete. During each session, users were asked to use an android smartphone loaded with our data collection software. The application displayed 10 questions in sequence. User answered each question by typing minimum of 250 characters. Figure3-1 (a) and Figure3-1 (b) show the interfaces of the data collection application.

1. (b)

Figure3-1 Interface of data collection application on smartphone Figure3-1 (a) Main page, Figure3-1 (b) Page where user typed answers to questions

While typing, users were asked to walk from the room where proctor was standing to a hallway. Approximately 100 feet down the hallway, a mark was made with blue tape to limit the walking distance for the user. Once user reached the blue tape, the user walked back to the room where the proctor was standing. Every time user walked back to the room, the proctor asked the user to perform an activity. These activities are: Give, Grab, Jenga, and Turn. Table 3-1 describes each activity. In each session, users were involved in a total of 10 activity instances: 3 give, 3 Jenga, 2 turn, and 2 grab.

Table 3‑1 Describes the Give, Grab, Jenga, Turn, and Walk activities.

|  |  |
| --- | --- |
| **Activity** | **Description** |
| Give | The proctor asks the user to give her the smartphone. This activity starts when user starts to extend her arm to give the smartphone, and ends when the smartphone is firmly in proctor’s hand and face of the proctor is fully visible in phone’s front-end camera. |
| Grab | The proctor forcefully grabs the smartphone from the user. This activity starts when phone is removed from user’s hands, and ends when the phone is firmly in proctor’s hand and face of the proctor is fully visible in phone’s front-end camera. |
| Jenga | The proctor asks the user to play Jenga. This activity starts when user moves his arm to place the phone on the table, and ends when phone is placed on the table (the phone stops moving). |
| Turn | The proctor asks the user to turn back and to continue to walk and type. |
| Walk | The user types and walks with no disruption. |

The Android application we developed collects following data:

1. Sensor data: Sensor data includes X, Y, Z signals from the accelerometer, magnetometer and gyroscope sensors. AccX, AccY, AccZ, measures acceleration force applied to the smart phone in X, Y, Z axes respectively [16]. GyroX, GyroY, GyroZ measures rate of rotation of the device around three physical axes [16].
2. Touch data: X, Y coordinates of the touchscreen and keystroke data.
3. Video using front-end camera.
4. Subjects answers to the set of questions displayed on the screen.
5. Smartwatch data: Accelerometer, gyroscope and heart rate signals were collected from two smart watches. At the end of every session, data collected from the watches was transferred to the phone. This data was not used in our experiments.

## Data Preprocessing

The X, Y, Z coordinates from the accelerometer and gyroscope were pre-processed in order to calculate overall movement of the device in 3-dimensional space. We calculated magnitude of the vector point for both accelerometer and gyroscope signals.

Magnitude of acceleration, accMag is calculated as:

= Equation 3‑1

Magnitude of gyroscope signal, gyroMag is calculated as:

= Equation 3‑2

We plot *accMag* (as shown in Figure3-2) to illustrate the entire event space and extract events and non-events.

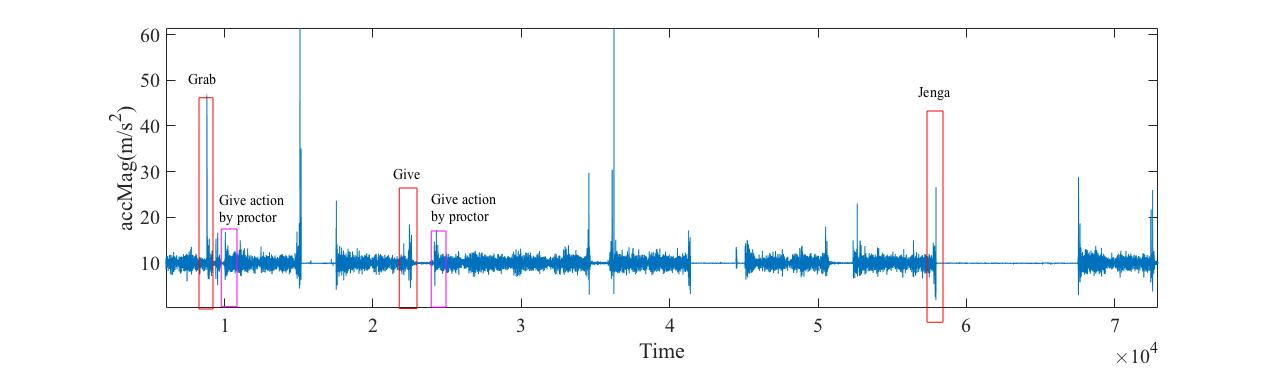


Figure 3‑2 Plot of accMag movement signals in time domain. Signals highlighted in red rectangular boxes are the Give, Grab, and Genga performed by the user. Signals highlighted in pink are the actions performed by the proctor.

As shown in the Figure3-2, the application also collected data when the proctor handed the smartphone back to the user, after a give or grab event. However, because this action is not performed by the user, we did not consider the corresponding data in our analysis.

# 

#### TRAINING AND TESTING

We used 37 sessions to train the classifier, and 15 sessions for testing. We split the entire event space in overlapping windows of length *t* as shown in the Figure4-1. We then measured the percentage of event information covered by each window. Based on this percentage, each window was labeled as Give, Grab or Non-event. For instance, in the figure4-1, windows composed of at least 60% Give signals were labeled as a Give, and windows composed of less than 60% Give signals were labeled as Non-event.

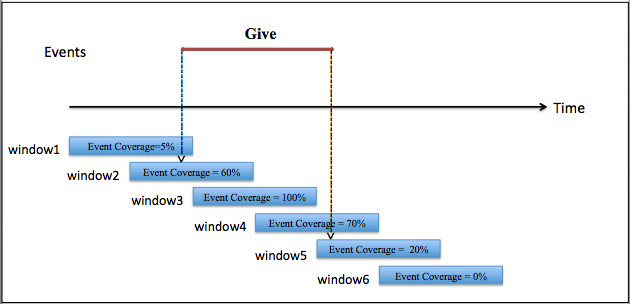


Figure 4‑1 Event space is split in overlapping windows.

We then fused the raw data points from three consecutive overlapping windows to form a fused-window. As shown Feature4-2, window1, window2, window3 were fused to form first fused-window. During fusion, raw data points from three windows were concatenated such that, data points in overlapped portion were considered only once.

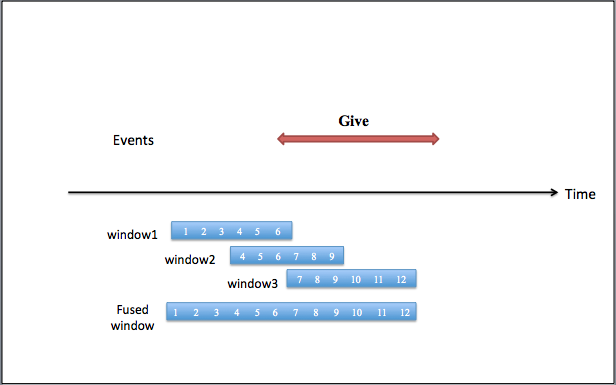


Figure 4‑2 Raw data points from three consecutive overlapping windows were concatenated to form one fused window. Data points in overlapped portion were considered only once.

As shown in Figure4-3, raw data points from window1, window2 and window3 were fused to form first fused-window and raw data points from window4, window5 and window6 were fused to form second fused-window. Features were computed from a fused-window to form a feature vector. The class label to a feature vector was obtained by fusing labels from the windows forming a fused-window. Each window label was assigned a priority based on type of event. Grab was considered the highest priority event, Give was considered second highest priority event, and Non-Event was the least priority event. The feature vector was labeled with highest priority event among the three windows forming the fused-window. Details of priority assignment and rules that were followed to label feature vector are discussed next.

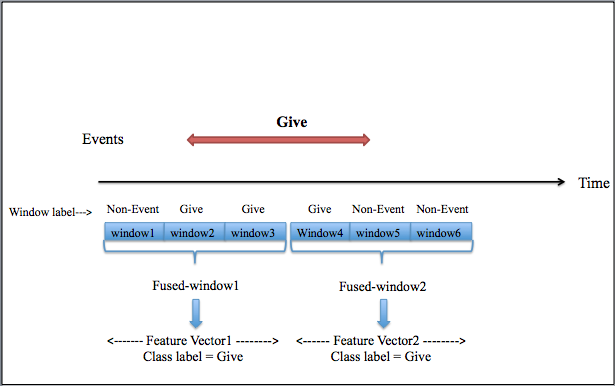


Figure 4‑3 Three overlapping windows were fused to form a fused-window. Features were extracted from fused-window to form a feature vector.

1. If at least one of the three windows was labeled as Grab, then the feature-vector was labeled as Grab. For example, a fused-window composed of windows labeled Grab, Give and Non-Event, resulted in a feature vector labeled as Grab. Similarly, a fused-window composed of windows labeled Grab, Give, Give or Grab, Non-Event, Non-Event resulted in the feature vectors labeled as Grab.
2. If none of the windows were labeled as Grab and at least one of the windows was labeled as Give, then the feature vector was labeled as Give. For Example, a fused-window composed of windows, Give, Give, and Non-Event resulted in a feature vector labeled as Give. Similarly, a fused-window composed of windows, Give, Non-Event, and Non-Event resulted in a feature vector labeled Give.
3. If none of the windows forming a fused-window were Grab or Give, then the feature vector was labeled as non-event. Meaning, the combination Non-Event, Non-Event and Non-Event resulted in Non-Event.

Because user walked for the majority of the time in a session, the event-to-non-event ratio in our dataset is roughly 1:45. This lead to class imbalance, with event class as the minority class and non-event as the majority class. To avoid class imbalance issues [17], we under-sampled the non-event class for the training set. We selected the same number of non-events and events for training set by choosing a random subset of non-event windows. However, we did not under-sample the testing set.

# 

#### FEATURES AND CLASSIFIERS

We derived two types of features: Frequency domain features, and time domain features. These features were computed from data using accelerometer and gyroscope sensors. Features extracted from the time domain were concatenated with features from frequency domain to form one feature vector. Computation of these features is discussed next.

## Time domain features

Motivation for using the time domain features was based on the observation that the amplitude of the signal varies between an event and non-event. As shown in Figure 3-2, the amplitude of the signal increases during an event, in comparison to a non-event. We extracted three temporal features from accelerometer magnitude (accMag) and three from gyroscope magnitude (gyroMag), resulting in 6 temporal features. Listed below are the temporal features and their description.

1. Maximum accMag reading in the fused-window. Measures maximum motion of the smartphone in a 3D space.
2. Standard deviation of accMag readings in the fused-window.
3. Mean of accMag readings in the fused-window. Measures average motion of the smartphone in a 3D space.
4. Maximum gyroMag reading in the fused-window. Measures maximum rotation of the smartphone in a 3D space.
5. Standard deviation of gyroMag readings in the fused-window.
6. Mean of gyroMag readings in the fused-window. Measures average rotation of the smartphone in a 3D space.

## Frequency domain features

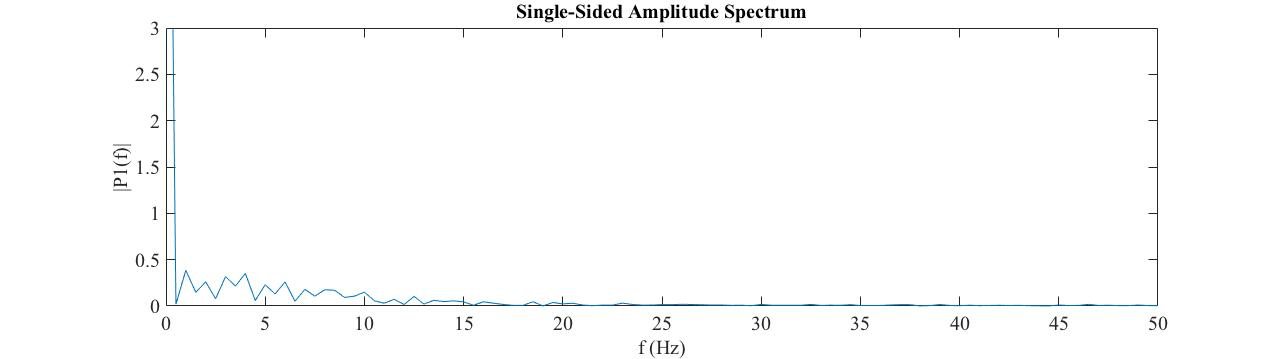
We used frequency domain features based on the assumption that the frequency components might have different signatures during an event compared to during non-event. In particular, frequency features might better distinguish a Give from a Grab, as Give is a smooth action and Grab is more of a jerky action. We used Fast Fourier Transform to find frequency components in the signal and converted the signal into frequency domain. As we collected accelerometer and gyroscope data at a sample rate of 100Hz, Fourier transform resulted in coefficients between 0 and 50. We then split the spectrum of FFT domain into 10 bins. The average amplitude of each bin was used as one feature, leading to 10 features in frequency domain. Figure5-1 and Figure5-2 demonstrate different signatures generated during a Non-event and Grab respectively.

Figure 5‑1 Frequency domain plot of non-event signal.

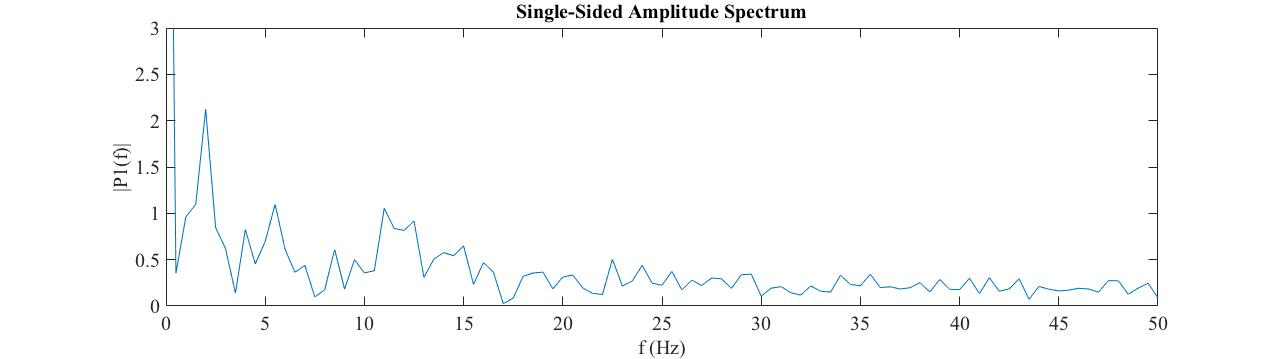


Figure 5‑2 Frequency domain plot of Grab signal

Figure5-3 summarizes the different steps involved in our protocol.

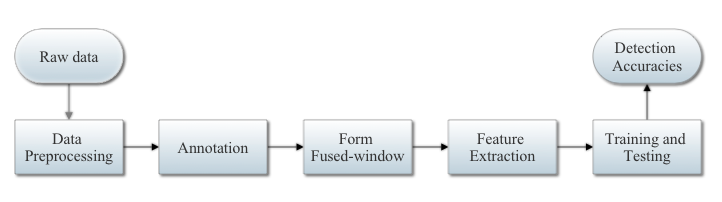


Figure 5‑3 Flow diagram depicting our protocol

# 

#### RESULTS AND DISCUSSIONS

Two main factors influenced detection performance in our protocol: (1) event coverage by the windows, and (2) window size. Influence of these factors is discussed next.

## Event Coverage by the Windows

As we split the event space in overlapping windows, the event information covered by the windows ranged from 1% to 100%. In order to get better insights about the influence of event coverage on the performance, we generated the performance reports for range of event coverage thresholds, from 10% to 100%. For instance, a coverage threshold of 50% means, the windows covering less than 50% of the event information were labeled non-events and the windows covering 50% to 100% were labeled events. As shown in the Table 6-1, 10% coverage threshold resulted in poor performance. Results improved as we increased the event coverage threshold from 10% to 60%. This is because the windows covering 10% of the event information carry 90% of nonevent information, therefore resulting in poor classification results. When we increased the coverage threshold above 60%, the performance started to degrade. This is because labeling windows with 100% event coverage as events also means that windows with, say 80% to 99% event coverage were labeled as non-events. This affects the performance in many ways, such as:

1. The majority of windows in the range 1% to 99% fall in the non-event section, and very few instances with 100% coverage fall in the event section. This reduces number of event instances available for training set. In practice, difference between a window with 90% event coverage and 100% event coverage is minimal. This leads to increased misclassifications.
2. With high coverage threshold, experiment might miss the entire event. For example, with 100% coverage threshold, if an event is split between three overlapping windows, each one covering 80% of the event information, then experiment will miss the entire event. An event coverage threshold of 60% provided better performance results by balancing the aforementioned factors.

Table 6‑1 Classifier result for window size t=2 seconds and event coverage of 10% to 100%. Results improved as the event coverage threshold was increased from 10% to 60%. When the event coverage threshold was above 60%, the performance started to degrade.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Event (Give, Grab)** | | **Non-event** | |
| **Event Coverage** | **TP Rate** | **FP Rate** | **TP Rate** | **FP Rate** |
| 10% | 84% | 3.1% | 96.90% | 16% |
| 20% | 87.10% | 3.6% | 96.40% | 12.9% |
| 30% | 89.30% | 3.7% | 96.30% | 10.7% |
| 40% | 87% | 2.9% | 97.10% | 13.0% |
| 50% | 91.50% | 3.8% | 96.20% | 8.5% |
| 60% | 92.50% | 2.9% | 97.20% | 8.4% |
| 70% | 90.00% | 2.7% | 97.30% | 10% |
| 80% | 89.20% | 3.3% | 96.70% | 10.8% |
| 90% | 89.10% | 4.2% | 95.80% | 10.9% |
| 100% | 85.50% | 4.0% | 96% | 14.5% |

## Influence of Window Size

On average, Grabs in our experiments were shorter than Gives. Choosing the correct window size for both types of events was critical for better classification results. In what follows, we discuss circumstances that might lead to low classification accuracies:

1. If a grab event is one second long and the window is longer than one second, even if the window covers 100% of the event, we have a feature vector that is only partially composed of the event at hand.
2. If a give event is 3 seconds long, and the window is 1 second, the feature vector might have too little event information.
3. A grab event of length 1 second might be split between two consecutive windows of length 3 seconds. The experiment will miss the entire event, as none of the windows meet threshold of 3 second and 60% event coverage. In this case, performance figures reported may be misleading.

In Table6-2, we present the results for window size t = 1, 1.5, 2, and 2.5 seconds to demonstrate above 3 cases.

Table 6‑2 Classification results for window size t=1, 1.5, 2, and 2.5 seconds and event covered by window=60%.

|  |  |  |
| --- | --- | --- |
|  | **Event** | **Non-event** |
| **Window size** | **TP Rate** | **TP Rate** | **Number of events missed by protocol** |
| 1 second | 86.0% | 96.8% | 0 |
| 1.5 second | 85.0% | 97.0% | 0 |
| 2 second | 92.5% | 97.2% | 1 |
| 2.5 second | 93.8% | 96.2% | 5 |

Based on the analysis of window size and event coverage by the window, we choose 2 second as window duration, and 60% for event coverage. Table 6-3 and 6-4 summarize the performance results for combination of accelerometer and gyroscope signals in time and frequency domain.

As shown in Table6-3, irrespective of accelerometer or gyroscope signal, time domain features out performed frequency domain features in distinguishing event and non-event. When we combined features from time domain and frequency domain for accelerometer and gyroscope signals, the classifier achieved better results with TP rate of 92.6% and 97.4 % for event and non-event respectively.

Table 6‑3 Detection accuracy for event, non-event.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Signal** | **Domain** | **Event (Give, Grab)** | | **Non-Event** | |
| **TP rate** | **FP rate** | **TP rate** | **FP rate** |
| accMag | Time | 92.60% | 4% | 96.00% | 7.4% |
| Frequency | 91.60% | 2.7% | 97.30% | 8.4% |
| Time + Frequency | 91.60% | 2.5% | 97.50% | 8.4% |
| gyroMag | Time | 90.50% | 3.4% | 96.60% | 9.5% |
| Frequency | 89.50% | 3% | 97.00% | 10.5% |
| Time + Frequency | 91.60% | 3.6% | 96.40% | 8.4% |
| accMag  +  gyroMag | Time | 90.50% | 3% | 97.00% | 9.5% |
| Frequency | 89.50% | 2.6% | 97.40% | 10.5% |
| Time + Frequency | 92.60% | 2.6% | 97.40% | 7.4% |

As shown in Table6-4, frequency domain features outperformed time domain features in distinguishing Give from Grab. When we combined frequency features with time domain features for both accelerometer and gyroscope, we attained better results. Classifier distinguished between a give, grab and non-event with TP rate of 81%, 43.8% and 97% respectively.

Table 6‑4 Detection accuracy for Give, Grab and Non-event

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Signal** | **Domain** | **Give** | | **Grab** | | **Non-event** | |
| **TP rate** | **FP rate** | **TP rate** | **FP rate** | **TP rate** | **FP rate** |
| accMag | Time | 73.00% | 4.0% | 43.80% | 0.8% | 96.30% | 7.4% |
| Frequency | 77.80% | 2.8% | 31.00% | 0.8% | 97.40% | 8.4% |
| Time +  Frequency | 77.80% | 3.3% | 34.40% | 0.7% | 97.00% | 8.4% |
| gyroMag | Time | 60.30% | 3.3% | 34.40% | 1.2% | 96.90% | 10.5% |
| Frequency | 71.40% | 3.2% | 34.40% | 0.9% | 97.00% | 10.5% |
| Time +  Frequency | 71.40% | 3.3% | 31.30% | 0.8% | 97.00% | 10.5% |
| accMag +  gyroMag | Time | 73.00% | 3.3% | 46.90% | 0.7% | 97%% | 9.5% |
| Frequency | 76.20% | 2.5% | 40.60% | 0.8% | 97.60% | 10.5% |
| Time +  Frequency | 81.00% | 2.8% | 43.8% | 0.6% | 97.00% | 9.5% |

As shown in Table6-5 and Table6-6, it was easy to distinguish between events and non-events. However, distinguishing between Give and Grab was not that easy. In particular, Grabs were confused with Gives, 43.8% of the Grabs were correctly classified as Grab and rest of the Grab were classified as Give.

Table 6‑5 Confusion matrix for event vs. non-event, when time and frequency domain features were extracted from accMag and gyroMag

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Predicted class** | |
| **Event** | **Non-event** |
| **Actual Class** | **Event** | 88 | 7 |
| **Non-event** | 68 | 2394 |

Table 6‑6 Confusion matrix for Give, Grab, and Non-event classes, when time and frequency domain features were extracted from accMag and gyroMag

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Predicted class** | | |
| **Give** | **Grab** | **Non-event** |
| **Actual Class** | **Give** | 51 | 6 | 6 |
| **Grab** | 15 | 14 | 3 |
| **Non-event** | 55 | 9 | 2398 |

# 

#### CONCLUSION AND FUTURE WORK

In this thesis, we investigated whether change of possession and non-possession of the smartphone can be detected using movement signals collected from inertial sensors. We carefully designed data collection protocol to capture events of interest during natural smartphone usage from 29 subjects. We designed a model to extract 30 features from raw data generated during an event and non-event. The framework trained random forest classifier using features extracted during Give, Grab and Non-events. The classifier was able to detect events and non-events with TP Rate of 92.6% and 97.4% respectively, while these results are promising, there are still challenges in distinguishing a Give from Grab. Majority of the Grabs were mixed with Give. We assume that, this issue will be fixed if larger dataset is used to train the classifier. Hence we plan to collect more data and test the protocol with larger dataset. Our results suggest that, change of possession and non-possession can be extended to different activities (e.g., reading and map navigation), body motion context (e.g., sitting), and application level features (e.g., pause duration between strokes/gestures and pause frequencies).

Our future work will involve extracting features from individual axes of each sensor in addition to magnitude, and extracting different types of features such as wavelet features in addition to frequency domain and time domain. We plan to compare the classification performance of different classifiers on our dataset.

### REFERENCES

[1] Zdenka Sitova, Jaroslav Sedenka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, Kiran S. Balagani,“HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users”, IEEE Transactions On Information Forensics And Security, Vol. 11, No. 5, May 2016

[2] Cheng Bo, Lan Zhangy, Taeho Jung, Junze Han, Xiang-Yang Li, Yu Wang “Continuous User Identification via Touch and Movement Behavioral Biometrics”. Performance Computing and Communications Conference (IPCCC), 2014 IEEE International

[3] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, “Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication,” IEEE Transactions on Information Forensics Security, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[4] T. Feng, Z. Liu, Kyeong-An Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. “Continuous mobile authentication using touchscreen gestures”. In Homeland Security (HST), 2012 IEEE Conference on Technologies for, pages 451–456, 2012.

[5] Xi Zhao,Tao Feng,Weidong Shi, ”Continuous Mobile Authentication Using A Novel Graphic Touch Gesture Feature ”. Biometrics : Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on 29Sept.-2 Oct.2013

[6] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, “Beware, your hands reveal your secrets!” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur, Scottsdale, AZ, USA, Nov. 2014, pp. 904–917.

[7] Lingjun Li, Xinxin Zhao, and Guoliang Xue “Unobservable Re-authentication for Smartphones”, in Proc. NDSS, 2013, pp. 1–16.

[8] Data genetics: Pin analysis. <http://www.datagenetics.com/blog/september32012/.Accessed>: 2015-11-08

[9] N. L. Clarke and S. M. Furnell, “Authenticating mobile phone users using keystroke analysis,” Int. J. Inf. Secur., vol. 6, no. 1, pp. 1–14, 2007.

[10] A. Buchoux and N. L. Clarke, “Deployment of keystroke analysis on a smartphone,” in Proc. Austral. Inf. Secur. Manage. Conf., 2008, p. 48.

[11] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri, “Keystroke dynamics authentication for mobile phones,” in Proc. ACM Symp. Appl.Comput., 2011, pp. 21–26.

[12] P. Campisi, E. Maiorana, M. Lo Bosco, and A. Neri, “User authentication using keystroke dynamics for cellular phones,” IET Signal Process., vol. 3, no. 4, pp. 333–341, Jul. 2009.

[13] M. Derawi, C. Nickel, P. Bours, and C. Busch. ”Unobtrusive user authentication on mobile phones using biometric gait recognition”, in International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010.

[15] Srlabs: Spoofing fingerprints. <https://srlabs.de/spoofingfingerprints/.Accessed:2105-11-08>.

[16] Android API Guide. <https://developer.android.com/guide/topics/sensors/sensors_overview>

[17] He, H. and E.A. Garcia, “Learning from imbalanced data”*.* IEEE Transactions on Knowledge and Data Engineering.2009. 21(9): p. 1263-1284.

[18] Jennifer R. , Kwapisz, Gary M, Weiss, Samuel A. Moore “Activity Recognition using cellphone accelerometers.”

[19] A. De Luca, A. Hang, F. Brudy, C. Linder, and H. Hussmann, ”Touch me once and I know it’s you!:implicit authentication based on touch screen patterns”, in ACM CHI, 2012,pp 987-996

[20] H. Seo, E. Kim, et al., ”A novel biometric identification based on a users input pattern analysis for intelligent mobile devices,” IJARS 2012.

[21] K. Huang, Xiang Ding, Jing Xu. ”Monitoring Sleep and Detecting Irregular Nights through Unconstrained Smartphone Sensing “ 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom) [2015 IEEE 12th Intl Conf on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7508813) 10-14 aug.2015

[22] [Gary M. Weiss](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Gary%20M.%20Weiss.QT.&newsearch=true), [Jeffrey W. Lockhart](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Jeffrey%20W.%20Lockhart.QT.&newsearch=true), [Tony T. Pulickal](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Tony%20T.%20Pulickal.QT.&newsearch=true), [Paul T. McHugh](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Paul%20T.%20McHugh.QT.&newsearch=true), [Isaac H. Ronan](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Isaac%20H.%20Ronan.QT.&newsearch=true), [Jessica L. Timko](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Jessica%20L.%20Timko.QT.&newsearch=true)“Actitracker: A Smartphone-Based Activity Recognition System for Improving Health and Well-Being”, [Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7795280) 17-19 Oct. 2016

[23] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In Proceedings of 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10,202,pages 301-316,2012

[24] J. Sedenka, S. Govindarajan, P. Gasti and K. Balagani. “Secure outsourced biometric authentication with performance evaluation on smartphones”. TIFS, (8), 2014.

[25] C. Bo, L. Zhang, X-Y. Li, Q. Huang, and Y. Wang. “Silentsense: Silent user identification via touch and movement behavioral biometrics”. In MobiCom, 2013.

[26] Keyvan Chamani: “Change of possession detection for smartphones”. New York Institute of Technology Master Thesis, May 2017