

Master's Thesis

Modeling and Discovering Human Behavior from Smartphone Sensing Life-Log Data

Department of Electronics and Computer Engineering

Graduate School, Chonnam National University

MAFRUR, Rischana

June 2015

Modeling and Discovering Human Behavior from Smartphone Sensing Life-Log Data

Department of Electronics and Computer Engineering
Graduate School, Chonnam National University

MAFRUR, Rischan

Supervised by Professor CHOI, Deok Jai

A dissertation submitted in partial fulfillment of the requirements for the Master of Engineering in Computer Engineering.

Committee in Charge:

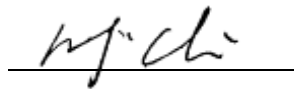
KIM, Soo Hyung



KIM, Kyung Baek



CHOI, Deok Jai



June 2015

Table of Contents

LIST OF FIGURES	iii
LIST OF TABLES	iv
(Abstract)	v
1. INTRODUCTION	1
1.1 Overview	1
1.2 Contribution	2
2. DATASET	2
2.1 Data Acquisition	2
2.1.1 Application Data Collector	3
2.1.2 Dataset Description	4
2.1.3 Dataset that used in this research	10
2.2 Data Pre-processing	10
2.2.1 Data Cleansing	11
2.2.2 Dataset Transformation	11
2.3 Feature Extraction	12
2.3.1 Define Human Activity and Behavior	12
2.3.2 Features Description and Extraction	13
2.3.3 Human and Machine Time	16
2.3.4 List of the Final Features	17
3. HUMAN BEHAVIORS MODELING	19
3.1 Background and Problem Statement	21
3.2 Proposed Methods	22
3.2.1 Overall architecture	22
3.2.2 Similarity Pattern Detection	22
4. EXPERIMENTAL RESULTS	24
4.1 Result and Discussion	24
4.1.1 Behavior Identification	24
4.2 Experimental Evaluation	24
4.2.1 Time Execution and Performance	25

4.2.2 Limitation	26
5. RELATED WORKS	27
5.1 Smartphone Personal Data.....	Error! Bookmark not defined.
5.2 Modeling and Discovering Human Behaviors	Error! Bookmark not defined.
6. CONCLUSIONS	30
Bibliography	32
References	33
(국문초록).....	35
ACKNOWLEDGEMENT	37

LIST OF FIGURES

Figure 2-1. Funf Open Sensing Framework.....	3
Figure 2-2. User personal database in user smartphone	3
Figure 2-3. Strings.xml file in project directory.....	5
Figure 2-4. Inside the string.xml file.....	5
Figure 2-5. Data preprocessing flows.....	11
Figure 2-6. Sample output of the features extraction in Pre-Processing II.	16
Figure 2-7. Sample output of the features extraction in Pre-Processing III (Final Features).19	
Figure 3-1. Example data visualization from two of students in the same day for four days.	20

LIST OF TABLES

Table 2-1. List of probes and time period of recording	4
Table 2-2. List of probes and types.....	7
Table 2-3. Data Summarization from 47 students.	9
Table 2-4. List of features and the values	15

Modeling and Discovering Human Behavior from Smartphone Sensing Life-Log Data

MAFRUR, Rischan

Department of Electronics and Computer Engineering

Graduate School, Chonnam National University

(Supervised by Professor CHOI, Deok Jai)

(Abstract)

Today, personal data is becoming a new economic asset. Personal data which generated from our smartphone can be used for many purposes such as identification, recommendation system, and etc. In this research, we have collected user personal data from many users, around 37 students during 2 months. We develop new approach that can be used to identify human behavior motifs based on user personal data from their smartphone. The data which generated by users smartphone are heterogeneous because those data produced by variety of sensors. Sometimes, the data from one or more sensors does not available. To handle that problem, we use many of sensors and tried to combine them rather than only use one of sensor. We have implemented our approach to demonstrate the feasibility and effectiveness of our approach to identify human behavior. Furthermore, we evaluate our approach and present the details in this thesis.

1. INTRODUCTION

1.1 Overview

Nowadays, smartphone capability has increased significantly. Smartphone has equipped with high processor, bigger memory, bigger storage and etc. With this equipment, smartphone has capability to running complex application. Many sensors also has embedded to the smartphone. With those sensors and log capability of smartphone, we can develop many useful system or application in different domain such as healthcare (elderly monitoring system [1] [2]) human fall detection [3] [4], transportation (monitoring road and traffic condition [5]), personal [6] [7] and social behavior [8] [9], environmental monitoring (pollution [10], weather) and etc. To develop such system, we have to collect the user personal data and then analyze it. In this research, we have collected user personal data to identify human behavior. Every person has unique behavior (behavior model). Example cases, in the context of daily behavior: Alice is research student in one of university in Korea. Every working day, he wakes up, takes a shower, breakfast, and goes to his campus at 8:40 AM. He living in dormitory, he walks from dormitory to his lab (campus) takes 10 minutes. Usually, he arrived in his lab at 9 AM and then sits on his chair and starts working. This example is one of the human daily routine in working day. Based on this story, we can used Alice's smartphone sensor data to define and build Alice's behavior model.

There are two ways to collect personal data from the users based on user involvement. First, participatory sensing and then the second, opportunistic sensing. Participatory sensing means the application still need user's intervention to complete their task. The examples for such application need user to taking text input for each time period, taking picture and etc. On

the other hand, opportunistic sensing means application does not need user's intervention to complete their task, users not involved in making decisions instead smart phone itself make decisions according to the sensed and stored data. In this thesis, to collect user personal data, we follow opportunistic method because we do not want to bothering user much. Based on those data, we identified human behavior and create their behavior model.

1.2 Contribution

Our contribution in this work are: (1) We have developed an application data collector which can collect user personal data and its following opportunistic method. This application does not bothering users, there is nothing to do after user install this application. (2) We have developed system that can identify human behavior based on their smartphone personal data. (3) Instead of identifying human behavior we also have developed system which can create human behavior model.

2. DATASET

2.1 Data Acquisition

This section explain about the data acquisition, we divide this section to three main parts are: application data collector, dataset description, and dataset that used in this research. Application data collector's section explain about our application which is used in this research to collect user personal data. Dataset description's section explain about our dataset itself, dataset that we have collected from user's smartphone. Dataset that used in this research's section explain about the lists of data that we used in this research. Not all data that we collected are used in this research, we only use several data which related with our research goals.



Figure 2-1. Funf Open Sensing Framework

The screenshot shows a smartphone screen with the 'My Files' app open. The status bar at the top shows the time as 9:15 PM. The app displays a list of files in the '/storage/emulsa' directory. Each file entry includes a file name, its size, and its creation date and time.

File Name	Size	Date/Time
3700877c-29...7_default.db	24KB	2015/01/15 2:39 PM
3700877c-29...7_default.db	6.8MB	2015/01/15 3:24 PM
3700877c-29...1_default.db	8.2MB	2015/01/15 4:25 PM
3700877c-29...6_default.db	3.3MB	2015/01/15 5:24 PM
3700877c-29...4_default.db	2.6MB	2015/01/15 6:26 PM
3700877c-29...1_default.db	2.8MB	2015/01/15 7:24 PM
51b5c75b-c8...9_default.db	300KB	2015/01/15 12:12 PM
51b5c75b-c8...8_default.db	7.1MB	2015/01/15 1:12 PM
51b5c75b-c8...8_default.db	3.2MB	2015/01/15 2:12 PM
633062f6-ef3...7_default.db		

Figure 2-2. User personal database in user smartphone

2.1.1 Application Data Collector

To develop application data collector, we do not create from scratch, we use Funf library. The Funf Open Sensing Framework is an Android-based extensible framework, originally developed at the MIT Media Lab, for doing phone-based mobile sensing. Funf provides a reusable set of functionalities enabling the collection and configuration for a broad range of data types. Funf is open sourced under the LGPL license. Figure 2-1 shows Funf framework can collect many of sensing from smartphone such location, movement, communication and usage, social proximity, and many more. In this thesis, we do not describe details about Funf

architecture, more details about Funf architecture can be seen in the main site of Funf¹ and also Funf developer site².

Table 2-1. List of probes and time period of recording

No.	Probes	Interval,duration (s)
1.	Location	300
2.	Wi-Fi	300
3.	Bluetooth	300
4.	Battery	300
5.	Call Log	86400
6.	SMS Log	86400
7.	Applications Installed	86400
8.	Hardware Info	86400
9.	Contacts	86400
10.	Browser Search Log	86400
11.	Browser Bookmark	86400
12.	Light Sensor	120,0.07
13.	Proximity	120,0.07
14.	Temperature	120,0.07
15.	Magnetic Field	120,0.07
16.	Pressure	120,0.07
17.	Activity Log	120,0.07
18.	Screen Status	120,0.07
19.	Running Application	120,0.07

2.1.2 Dataset Description

Our application follows opportunistic sensing method. It because we do not want to bothering user much. To do that, we have to define the time (interval and duration) first in our application. Interval means how many times in second system will send data request to the smartphone. An example, we set interval 300 seconds means 5 minutes so application will

¹ <http://www.funf.org/>

²² <https://code.google.com/p/funf-open-sensing-framework/>

request and store the data for every 5 minutes. Duration is the measure of continuance of any object or event in time. Duration is used in sensor's data because without duration is useless to collect the sensors data. An example of duration setting, when we set interval 120 seconds or two minutes and duration 0.07 s means the application will send data request to the smartphone for every 2 minutes and the system will record the data during 0.07 seconds. Table 2-1 shows the interval and duration of each probes. Those interval and duration have been tested and we thought those setting was optimum one. We can change those setting by change the value on the string.xml in android project. Figure 2-3 shows the string.xml file in the android project directory and Figure 2-4 shows inside the string.xml file, we can change the values of interval and duration in that file.

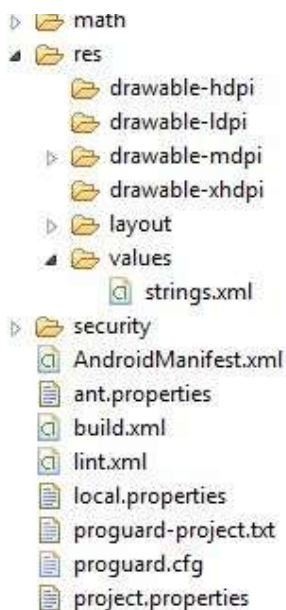


Figure 2-3. Strings.xml file in project directory

```
{
  "@type": "edu.mit.media.funf.probe.builtin.ContactProbe",
  "@schedule": {
    "interval": 86400,
    "opportunistic": true,
    "strict": true
  }
},

{
  "@type": "edu.mit.media.funf.probe.builtin.LightSensorProbe",
  "@schedule": {
    "interval": 120,
    "duration": 0.07,
    "opportunistic": true,
    "strict": true
  }
},

{
  "@type": "edu.mit.media.funf.probe.builtin.ProximitySensorProbe",
  "@schedule": {
    "interval": 120,
    "duration": 0.07,
    "opportunistic": true,
    "strict": true
  }
},
}
```

Figure 2-4. Inside the string.xml file

To make easy, we classify the data that we want to collect to three of categorization, are:

1. On Request Data (Current Data)
2. Historical Data (Saved in Android database system)
3. Continuous Data (Sensors data)

On request data means we ask the current values (information) from android system such as location, battery, nearby Bluetooth and etc. Historical data means the data that stored in android database system so we only need to access and copy those data from android database system to our application, the example of historical data are contact, call log, SMS log, and etc. Continuous data means we can get those data continuously such as sensor data (accelerometer, gyroscope, magnetic field, and etc).

We are living in time dimension space, every event has time variable. In our data, every value that returned from the user smartphone has timestamp value. Funf already has features to define timestamp, Funf using UNIX UTC (Coordinated Universal Time) which is (Unix time or POSIX time or Unix timestamp) is the number of seconds that have elapsed since January 1, 1970. To convert UNIX time to the human readable time, we can use POSIX function in R or another programming language. Data that we collected using our application will be stored in SQLite database format with (*.db) extension, the view of data can be seen in Figure 2-2. To open those database, we can use SQLite browser that can be download in SQLite browser main site³. The table in all of databases contain four columns, *_id* is automatically generated by database engine, *name* means the name of probes (sensors), *timestamp* column is time when system store the data to the phone's storage, and *value* is the value that returned from the sensors. Table 2-2 shows the list of probes (sensor

³ <http://sqlitebrowser.org/>

data) that provided by our application. The total of probes which provided by our application are 19 probes and we use 9 probes in this research.

Table 2-2. List of probes and types

No.	Name of Probes	Explanation	Used
On Request Data			
1.	SimpleLocationProbe	GPS data (user location)	X
2.	WifiProbe	Nearby Wi-Fi signals	X
3.	BluetoothProbe	Nearby Bluetooth signals	X
4.	BatteryProbe	Battery status	X
Historical Data			
1.	CallLogProbe	User call log	X
2.	SmsProbe	User SMS log	X
3.	ApplicationsProbe	List of application installed	
4.	HardwareInfoProbe	User's smartphone hardware info	
5.	BrowserBookmarksProbe	User Bookmarks	
6.	BrowserSearchesProbe	User Browser log	
7.	ContactProbe	User contact (phonebook)	
Continuous Data			
1.	LightSensorProbe	Measures the ambient light level (illumination) in lx	
2.	ProximitySensorProbe	Measures the proximity of an object in cm relative to the view screen of a device.	
3.	TemperatureSensorProbe	Measures the temperature of the device in degrees Celsius (°C).	
4.	MagneticFieldSensorProbe	Measures the ambient geomagnetic field (x, y, z) in μT	
5.	PressureSensorProbe	Measures the ambient air pressure in hPa or mbar.	
6.	ScreenProbe	Screen phone (on and off)	X
7.	RunningApplicationsProbe	List of running applications	X
8.	ActivityProbe	User activity log based on accelerometer sensor (none, low, and high activity)	X

To understand the value from each probes, we give the example value of location data, the name of probe is "*Simple Location Probe*". Location is one of the most important information

from the user. Location value that returned by our application is like in the box below:

```
{ "mAccuracy":1625.0, "mAltitude":0.0, "mBearing":0.0, "mElapsedRealtimeNanos":21989372000000, "mExtras":{ "networkLocationSource":"cached", "networkLocationType":"cell", "noGPSLocation":{ "mAccuracy":1625.0, "mAltitude":0.0, "mBearing":0.0, "mElapsedRealtimeNanos":21989372000000, "mHasAccuracy":true, "mHasAltitude":false, "mHasBearing":false, "mHasSpeed":false, "mIsFromMockProvider":false, "mLatitude":35.1837595, "mLongitude":126.9052379, "mProvider":"network", "mSpeed":0.0, "mTime":1403484137091}, "travelState":"stationary"}, "mHasAccuracy":true, "mHasAltitude":false, "mHasBearing":false, "mHasSpeed":false, "mIsFromMockProvider":false, "mLatitude":35.1837595, "mLongitude":126.9052379, "mProvider":"network", "mSpeed":0.0, "mTime":1403484137091, "timestamp":1403484137.255}
```

That data which from location probes is representing a geographic location. A location can consist of a latitude, longitude, timestamp, and other information such as bearing, altitude, velocity and etc. All locations generated by the *LocationManager* are guaranteed to have a valid latitude, longitude, and timestamp (both UTC time and elapsed real-time since boot) and all other parameters are optional. The details documentation about the data itself can be accessed in our projects site⁴, open “Data Documentation” directory. In this research, we use location data but we do not use all of those data, probably in this case, we only use longitude and latitude data to define user location. The reason why our application collect all of those data is probably another researchers want to use those data such as bearing, accuracy and etc for another purposes.

We store the data from all of students in archive file. The size of all of data after extracted is around 28.7 GB. Extracted data contain 47 directories in different name for each student’s

⁴ <https://github.com/rischanlab/Rfunf>

data. The result of data summarization which contain with name of directories, size, starting point, and ending point can be seen in Table 2.3. Starting point means when the student start the application, and ending point means when the student stop the application.

Table 2-3. Data Summarization from 47 students.

No.	Data ID	Size (MB)	Starting Point	Ending Point
1.	ENFP_0719	628	6/30/2014 8:26	8/20/2014 0:18
2.	ENFP_0773	664	6/26/2014 12:34	8/18/2014 4:58
3.	ENFP_2012	661	6/27/2014 6:11	9/2/2014 3:57
4.	ENTJ_5868	6890	6/27/2014 5:31	8/13/2014 0:00
5.	ENTJ_6454	121	6/26/2014 5:32	8/6/2014 18:53
6.	ENTJ_6966	272	7/2/2014 7:24	8/19/2014 11:22
7.	ENTP_5623	455	6/30/2014 4:49	8/19/2014 20:57
8.	ESFJ_2301	145	6/27/2014 5:31	8/20/2014 2:58
9.	ESFJ_9284	158	6/26/2014 12:34	8/18/2014 4:58
10.	ESFP_0912	278	6/26/2014 5:28	8/18/2014 8:53
11.	ESFP_3295	-		
12.	ESFP_4634	486	6/27/2014 5:25	8/20/2014 4:10
13.	ESFP_7467	607	6/26/2014 5:27	8/19/2014 7:18
14.	ESTJ_0371	2390	7/3/2014 16:21	8/16/2014 21:03
15.	ESTJ_3022	183	6/26/2014 5:28	8/18/2014 23:22
16.	ESTJ_5071	1920	7/2/2014 2:34	9/11/2014 1:49
17.	ESTJ_5190	258	7/30/2014 6:04	8/24/2014 1:43
18.	ESTJ_5824	173	6/26/2014 5:29	8/18/2014 3:51
19.	ESTJ_6510	756	6/27/2014 5:30	8/20/2014 8:09
20.	ESTP_4301	232	6/26/2014 5:29	8/20/2014 4:39
21.	ESTP_5154	990	6/27/2014 5:31	8/13/2014 0:00
22.	INFP_1993	432	6/26/2014 5:31	8/20/2014 0:31
23.	INTJ_5498	342	6/26/2014 5:28	8/20/2014 2:49
24.	INTJ_7906	312	6/14/2014 11:00	8/16/2014 23:01
25.	INTP_3739	1030	6/27/2014 5:28	8/18/2014 5:58
26.	INTP_6399	199	6/26/2014 5:29	8/12/2014 8:32
27.	INTP_9712	180	6/26/2014 5:37	8/16/2014 18:05
28.	ISFJ_2057	183	6/27/2014 5:32	8/14/2014 23:19
29.	ISFJ_2711	767	7/31/2014 0:51	8/20/2014 6:59
30.	ISFJ_7328	133	6/30/2014 7:09	8/19/2014 23:37
31.	ISFP_4030	2380	6/27/2014 6:11	9/2/2014 3:57

32.	ISFP_4282	613	6/27/2014 5:27	8/20/2014 2:46
33.	ISTJ_0178	158	6/26/2014 5:28	8/19/2014 5:05
34.	ISTJ_0386	284	6/26/2014 5:27	8/19/2014 7:18
35.	ISTJ_2068	339	6/26/2014 5:29	8/18/2014 5:30
36.	ISTJ_2837	186	6/27/2014 5:27	8/22/2014 5:41
37.	ISTJ_3052	131	6/27/2014 5:27	8/20/2014 3:41
38.	ISTJ_4659	325	7/2/2014 2:34	9/11/2014 1:49
39.	ISTJ_4667	156	6/26/2014 5:29	8/15/2014 10:44
40.	ISTJ_4700	170	7/3/2014 6:50	8/25/2014 13:08
41.	ISTJ_4753	363	6/26/2014 5:29	8/18/2014 23:48
42.	ISTJ_4968	95	7/3/2014 16:21	8/16/2014 21:03
43.	ISTJ_9139	473	7/3/2014 16:21	8/20/2014 5:57
44.	ISTJ_9576	198	7/4/2014 1:00	8/18/2014 7:12
45.	ISTP_3948	500	6/26/2014 5:29	8/20/2014 1:28
46.	ISTP_7676	365	6/27/2014 5:31	8/19/2014 22:11
47.	XXXX_XXXX	434	6/27/2014 5:31	8/21/2014 6:02

2.1.3 Dataset that used in this research

Table 2-2 List of probes and types shows that all of data that we collected form user's smartphone. Not all of those data are used in this research. We give symbol ("X") in the last column (*used column*) to the data which we used in this research. The data that we used are: On request data: GPS location, Nearby Wi-Fi, Nearby Bluetooth, Battery; Historical data: Call log and SMS log; Continuous data: Smartphone screen, Running applications, user activity log. The total dataset that we used are 9 probes.

The total of students who participated are 47 students. From those data not all data are full available. Some of students does not have SMS log, or another data, the reason they do not have SMS data probably he prefers to uses application messenger such as Kakao, Whatsapp, etc instead of SMS application. In this research, we use data from 37 students which all of data are available during 2 months.

2.2 Data Pre-processing

The data which collected from user's smartphone are not clean, means the data has a noise

and duplication. In this section, we explain about the data pre-processing which is contain with two subchapters are data cleansing and data transformation.

2.2.1 Data Cleansing

Funf library which we used as base of our application, has a problem in historical data collection. Historical data is the data which has been stored in android database system such as contact, SMS log, call log, and etc. We use 86400 second interval, means the application copy those data from android database system to our application database one time every day. It makes duplication in our database and we have to care about it. Another problem is system does not always work well, sometimes something wrong happened and the user's smartphone return value such as NA, error, or/and has no value. We use R programming language to create module which can remove this duplication and clean the noisy data.

2.2.2 Dataset Transformation

As we mentioned in data description's section that the size of all of the data is around 28 GB. To process those of data, when we load all of those data in the same time it will spend computer resource especially RAM. R environment system will load all of data that will be process in RAM. To handle that problem, we have to define what kind of data that we want to use and store those data to another file, in this case, we use csv files.

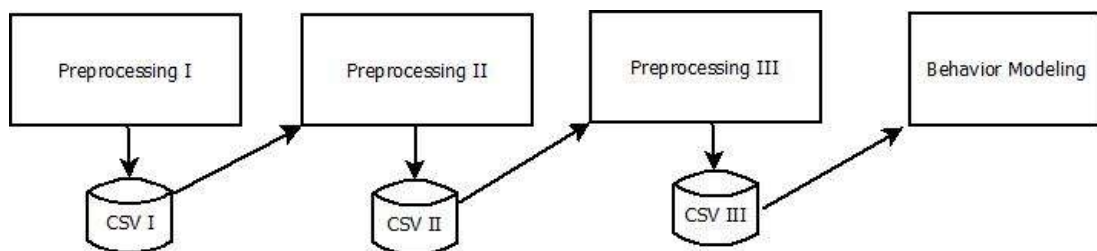


Figure 2-5. Data preprocessing flows

We have three kind of preprocessing modules and each module will store new data to csv

file. Figure 2-5 shows preprocessing process and dataset transformation from preprocessing I until behavior modeling module. Preprocessing I will load all of raw data, removing duplication data, cleansing data, and select the most important data that have been defined. Preprocessing I will store the result data to the CSV I database. Preprocessing II will load the CSV I data not the raw data, in this process features extraction applied. The result of Preprocessing II stored in CSV II. Preprocessing III load the CSV II data and transform the data to the fit format before creating behavior model applied. This ways will reduce time processing and computer resource's usage.

2.3 Feature Extraction

Features are functions of the original measurement variables that are useful for classification and/or pattern recognition. Feature extraction is the process of defining a set of features, which will most efficiently or meaningfully represent the information that is important for analysis and classification. In this stage, before we are extracting the features we have to define first what the features that we want to use.

2.3.1 Define Human Activity and Behavior

To extract the features, we have to know first what the human behavior is. In this thesis, we define that human behavior is human daily activities which carried out continuously. As we mentioned in introduction section, about the Alice' daily activities from he wakes up until arrives to his lab room in working day. We call that Alice's activities are Alice's behavior because that activities carried out continuously by Alice in his working day.

In terms of human daily activities, we have to consider about four important things are:

1. What kind of activity (e.g. meeting, studying, exercising, and etc)
2. When the activity happened (e.g. around 9 AM)

3. Where is the location when activity happened (e.g. Lab's room)
4. Interaction with (e.g. Meeting with whom: his lab members, and etc)

We tried to extract the features from the raw dataset based on those four points. We also have to consider about possibilities, probably same activities happened but in different time and location, or maybe different activity but in same time and location, and vice versa.

2.3.2 Features Description and Extraction

Based on our raw dataset and after we define the human behavior itself, the features that we proposed are:

- What kind of human activity
 - The important thing that we have to know is because of our application follows opportunistic method to collect user personal data, so we do not have activity label in our dataset.
 - We only have activity status (none, low, and high), these status based on accelerometer sensor activity.
 - We use sum of variance to detect the user activity, if the variance sum more than or equal to 10 float it will be return high activity, if the variance sum value between 3 float and less than 10 float it will be return low activity and else is none activity.
 - We use this data to define the user activity, even though we do not know the name of activity (activity label) but we still now the user activity pattern (none, low, and high) these values can be used to detect user behavior.
- When the activity happened
 - Every values in our dataset has timestamp value. The timestamp value following UNIX timestamp, we have to transform to human timestamp.
 - Date and time are used as features in this research.
- Where is the location

- Rather than living in time domain we also live in place domain (location).
- In this research, we use three of features to define the location are GPS, nearby Wi-Fi, and nearby Bluetooth. GPS is used for defining the user location in outside and nearby Wi-Fi and nearby Bluetooth can be used to define user location inside building.
- Interaction with (user interaction)
 - We divide user's interaction to two of kind interactions, first is interaction between users and their smartphone, and second is interaction between users and another users (between human).
 - User -> Smartphone interaction
 - Battery, based on this data, we can know when the user usually charge their battery and etc.
 - Smartphone screen, this data can be used as base information about user's smartphone usage.
 - Running applications, means the list of current applications that user used (time(when), name of applications, and duration)
 - Human -> Human interaction
 - SMS Log
 - Call Log
 - SMS and Call log can be used as the base information as the user interaction with others people.

Table 2.4 shows the list of our features and the values. We select three of the most important values from each probes data.

1. The *value1* of Activity Probes filled by ("*none*", "*low*", and "*high*").
2. The values of GPS are *value1* is latitude and *value2* is the longitude.
3. The values of Wi-Fi probe are *value1* is name of Wi-Fi SSID, *value2* is the mac address of Wi-Fi hardware, and the *value3* is the signal strength of the access point.

4. Bluetooth probe only has single value, *value1* is the name of nearby Bluetooth devices.
5. Battery probe has only one value, *value1* filled by (“*charging*”, “*discharging*”, and “*full*”).
6. The *value1* of Screen probe filled by “ON” or “OFF”
7. Running application probe has two important values are *value1* is the application name and *value2* is the duration of the application’s usage.
8. Call Log and SMS Log has three of values, *value1* is the number of person who (call/receive call, sent SMS, or receive SMS), *value2* is the types, means incoming and outgoing for the call, and inbox or sent message for the SMS, and the last *value3* filled by time duration for the call and text length for the SMS log.
9. All of rows data values has timestamp.
10. We define these all features in Pre-Processing II.

Table 2-4. List of features and the values

No	Name of Probes	Value1	Value2	Value3
1.	ActivityProbe	Status (“ <i>none</i> ”, “ <i>low</i> ”, and “ <i>high</i> ”)		
2.	SimpleLocationProbe	Latitude	Longitude	
3.	WifiProbe	List of nearby SSID	MAC	Signal strength (dB)
4.	BluetoothProbe	List of nearby Bluetooth devices		
5.	BatteryProbe	Status (“ <i>discharging</i> ”, “ <i>full</i> ”, and “ <i>charging</i> ”)		
6.	ScreenProbe	ON/OFF		
7.	RunningApplicationsProbe	Apps name	Duration	
8.	CallLogProbe	Number	Types	Duration
9.	SmsProbe	Number	Types	Text length

The example output of the features extraction can be seen in Figure 2-6. First columns is an ID, and then the second column is the time with the format (yyyy-mm-dd hh:mm:ss). Third

column is type, means the name of probes, to make easy to read we change *ActivityProbe* to *activity*, *SimpleLocationProbe* to *location*, *WifiProbe* to *wifi*, and etc.

```

", "time", "type", "value1", "value2", "value3"
"5135", "2014-07-01 00:01:55", "activity", "none", "", ""
"5136", "2014-07-01 00:01:56", "activity", "none", "", ""
"96154", "2014-07-01 00:02:54", "battery", "full", "", ""
"166061", "2014-07-01 00:02:54", "location", "37.53724098", "126.96960174", ""
"487488", "2014-07-01 00:02:58", "wifi", "2-607", "44:ed:57:01:f9:ac", "-55"
"5137", "2014-07-01 00:03:56", "activity", "none", "", ""
"5138", "2014-07-01 00:03:57", "activity", "none", "", ""
"5139", "2014-07-01 00:05:56", "activity", "none", "", ""
"96155", "2014-07-01 00:07:55", "battery", "full", "", ""
"166062", "2014-07-01 00:07:55", "location", "37.53724098", "126.96960174", ""
"5140", "2014-07-01 00:07:56", "activity", "none", "", ""
"5141", "2014-07-01 00:07:57", "activity", "none", "", ""
"487489", "2014-07-01 00:07:58", "wifi", "2-607", "44:ed:57:01:f9:ac", "-54"
"487490", "2014-07-01 00:07:58", "wifi", "2-507", "44:ed:57:01:55:ca", "-86"
"5142", "2014-07-01 00:09:55", "activity", "none", "", ""
"5143", "2014-07-01 00:09:56", "activity", "none", "", ""
"5144", "2014-07-01 00:11:55", "activity", "none", "", ""
"5145", "2014-07-01 00:11:56", "activity", "none", "", ""
"96156", "2014-07-01 00:12:55", "battery", "full", "", ""
"166063", "2014-07-01 00:12:55", "location", "37.53724098", "126.96960174", ""
"487491", "2014-07-01 00:12:58", "wifi", "2-607", "44:ed:57:01:f9:ac", "-61"
"487492", "2014-07-01 00:12:58", "wifi", "2-507", "44:ed:57:01:55:ca", "-87"
"487493", "2014-07-01 00:12:58", "wifi", "sparrow", "64:e5:99:c8:06:d2", "-92"
"5146", "2014-07-01 00:13:57", "activity", "none", "", ""
"5147", "2014-07-01 00:13:58", "activity", "none", "", ""
"5148", "2014-07-01 00:15:56", "activity", "none", "", ""
"5149", "2014-07-01 00:15:57", "activity", "none", "", ""
"96157", "2014-07-01 00:17:55", "battery", "full", "", ""
"166064", "2014-07-01 00:17:55", "location", "37.53724098", "126.96960174", ""
"5150", "2014-07-01 00:17:56", "activity", "none", "", ""
"5151", "2014-07-01 00:17:57", "activity", "none", "", ""
"487494", "2014-07-01 00:17:58", "wifi", "2-607", "44:ed:57:01:f9:ac", "-54"
"487495", "2014-07-01 00:17:58", "wifi", "2-507", "44:ed:57:01:55:ca", "-87"
"5152", "2014-07-01 00:19:55", "activity", "none", "", ""
"5153", "2014-07-01 00:19:57", "activity", "none", "", ""
"5154", "2014-07-01 00:21:55", "activity", "none", "", ""
"5155", "2014-07-01 00:21:56", "activity", "none", "", ""
"96158", "2014-07-01 00:22:55", "battery", "full", "", ""
"166065", "2014-07-01 00:22:55", "location", "37.53724098", "126.96960174", ""
"487496", "2014-07-01 00:22:58", "wifi", "2-607", "44:ed:57:01:f9:ac", "-52"
ENFP_2012.csv

```

Figure 2-6. Sample output of the features extraction in Pre-Processing II.

2.3.3 Human and Machine Time

Machine is different with human, machine can calculates and shows the time in exactly time such as 00:22:44:34 (millisecond) but human could not do that. As a human, usually when we want to do activity in term of time we said on hour and minutes. An example is when we have agreement with someone, usually we said “OK, we have meeting at 9.30 AM”, we

never said “OK, we have meeting at 09:30:00:00 (until millisecond)”. In this research, we transform the time machine to human machine. We create the module to transform time machine to human machine in module Pre-processing III.

2.3.4 List of the Final Features

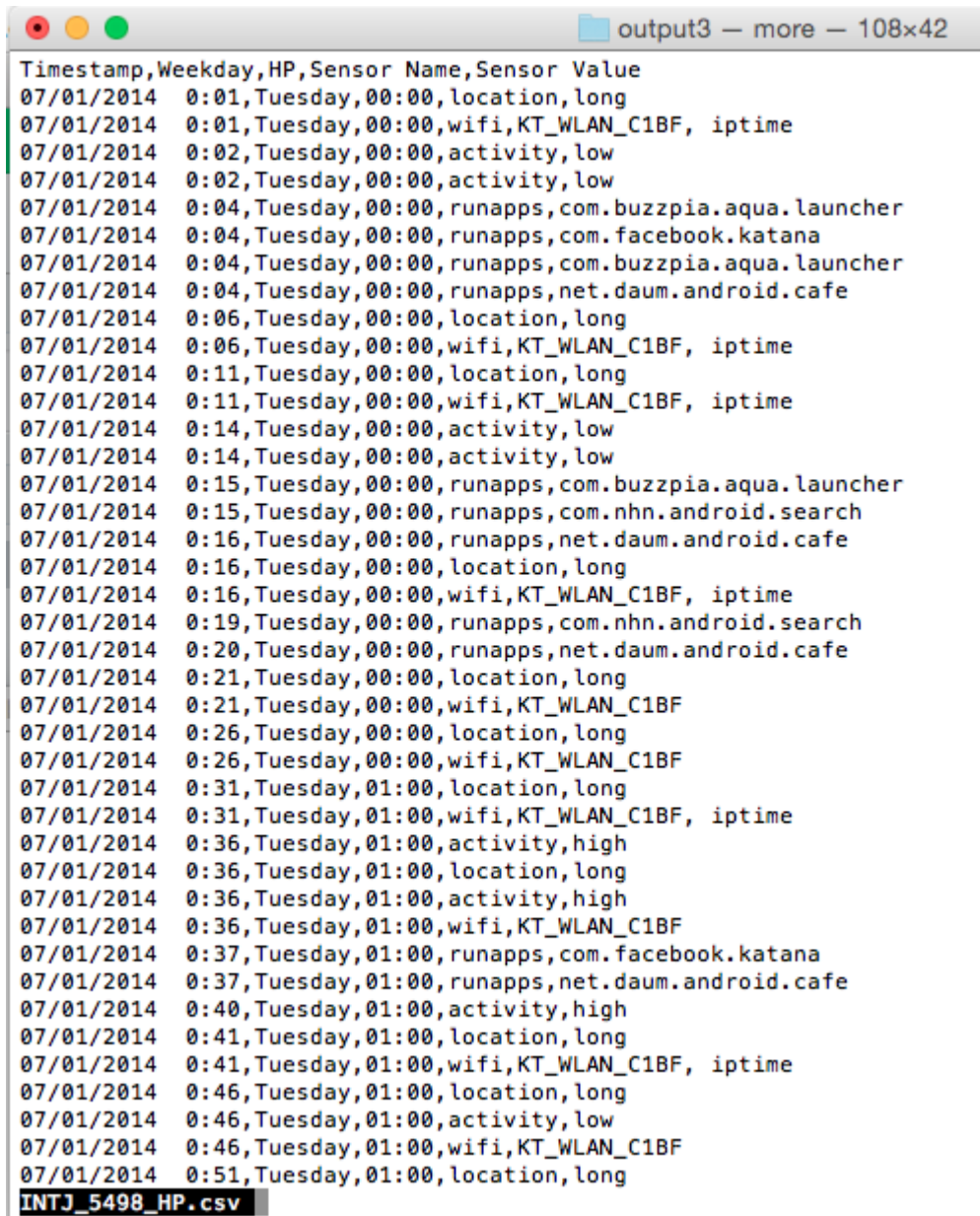
Figure 2.5 shows the result of features extraction from Pre-processing II module. We still have some problem on that result. We create Pre-processing III module to make our dataset fit enough before applying behavior modeling module. Another reason is more features mean more time to processes, light features means light time, so we try to find the most valuable features from all of those features. The process in the Pre-processing III module are:

1. Time change, from machine time to the human time. In this research, to convert machine time to human time we tried to round time with the setting:
 - a. If minute less than 30 minutes will be round down
 - b. If minute more than or equal to 30 minutes will be round up
2. Change GPS location value. We change the value of the GPS to “**moving status**” that value filled by “*same*”, “*little*”, or “*long*”. Note: 0.0001 degree= 11.1132 m.
 - a. If the previous value of GPS location not change, it means no movement, so the value filled by “*same*”.
 - b. If the moving distance between 0.0001 ~ 0.0005, it means little movement, so the value filled by “*little*”.
 - c. If the moving distance more than 0.0005, it means long movement, so the value filled by “*long*”.
 - d. To determine the value 0.0005 is based on experience of plotting, we

have tried to plot those point and we decide to use that value to distinguish little and long movement.

3. Remove “*discharging*” from the Battery value. The value of battery status are: “*charging*”, “*discharging*”, and “*full*”. We thought that default value is “*discharging*” because usually users use their phone in discharging mode so we remove this value and only use “*charging*” means when the user charge their phone and “*full*” means the battery was full.
4. Remove “*none*” from the Activity value. “*none*” value means idle, we tried to use “*low*” and “*high*” activity as our features.
5. Aggregate the values of Wi-Fi and Bluetooth. When we see Figure 2-6, in same time the value of Wi-Fi is one SSID in one row, and also for the Bluetooth. That is because every 5 minutes our application store the lists of nearby access points and Bluetooth devices and each value stored in rows. In this module, if the time is same the sensor values will be aggregate in one row.

The example of final features based on the result from Pre-processing III can be seen in Figure 2-7. The final features are: Timestamp with format (“*yyyy-mm-dd hh:mm*”) the time until minute, Day means the name of the day (weekday), HT means human time, filled by result from rounding of time, Sensor Name means the name of probes such as *activity*, *wifi*, *location*, *bluetooth*, and etc, Sensor value means the values of the sensors.



```
output3 — more — 108x42
Timestamp,Weekday,HP,Sensor Name,Sensor Value
07/01/2014 0:01,Tuesday,00:00,location,long
07/01/2014 0:01,Tuesday,00:00,wifi,KT_WLAN_C1BF, iptime
07/01/2014 0:02,Tuesday,00:00,activity,low
07/01/2014 0:02,Tuesday,00:00,activity,low
07/01/2014 0:04,Tuesday,00:00,runapps,com.buzzpia.aqua.launcher
07/01/2014 0:04,Tuesday,00:00,runapps,com.facebook.katana
07/01/2014 0:04,Tuesday,00:00,runapps,com.buzzpia.aqua.launcher
07/01/2014 0:04,Tuesday,00:00,runapps,net.daum.android.cafe
07/01/2014 0:06,Tuesday,00:00,location,long
07/01/2014 0:06,Tuesday,00:00,wifi,KT_WLAN_C1BF, iptime
07/01/2014 0:11,Tuesday,00:00,location,long
07/01/2014 0:11,Tuesday,00:00,wifi,KT_WLAN_C1BF, iptime
07/01/2014 0:14,Tuesday,00:00,activity,low
07/01/2014 0:14,Tuesday,00:00,activity,low
07/01/2014 0:15,Tuesday,00:00,runapps,com.buzzpia.aqua.launcher
07/01/2014 0:15,Tuesday,00:00,runapps,com.nhn.android.search
07/01/2014 0:16,Tuesday,00:00,runapps,net.daum.android.cafe
07/01/2014 0:16,Tuesday,00:00,location,long
07/01/2014 0:16,Tuesday,00:00,wifi,KT_WLAN_C1BF, iptime
07/01/2014 0:19,Tuesday,00:00,runapps,com.nhn.android.search
07/01/2014 0:20,Tuesday,00:00,runapps,net.daum.android.cafe
07/01/2014 0:21,Tuesday,00:00,location,long
07/01/2014 0:21,Tuesday,00:00,wifi,KT_WLAN_C1BF
07/01/2014 0:26,Tuesday,00:00,location,long
07/01/2014 0:26,Tuesday,00:00,wifi,KT_WLAN_C1BF
07/01/2014 0:31,Tuesday,01:00,location,long
07/01/2014 0:31,Tuesday,01:00,wifi,KT_WLAN_C1BF, iptime
07/01/2014 0:36,Tuesday,01:00,activity,high
07/01/2014 0:36,Tuesday,01:00,location,long
07/01/2014 0:36,Tuesday,01:00,activity,high
07/01/2014 0:36,Tuesday,01:00,wifi,KT_WLAN_C1BF
07/01/2014 0:37,Tuesday,01:00,runapps,com.facebook.katana
07/01/2014 0:37,Tuesday,01:00,runapps,net.daum.android.cafe
07/01/2014 0:40,Tuesday,01:00,activity,high
07/01/2014 0:41,Tuesday,01:00,location,long
07/01/2014 0:41,Tuesday,01:00,wifi,KT_WLAN_C1BF, iptime
07/01/2014 0:46,Tuesday,01:00,location,long
07/01/2014 0:46,Tuesday,01:00,activity,low
07/01/2014 0:46,Tuesday,01:00,wifi,KT_WLAN_C1BF
07/01/2014 0:51,Tuesday,01:00,location,long
INTJ_5498_HP.csv
```

Figure 2-7. Sample output of the features extraction in Pre-Processing III (Final Features).

3. HUMAN BEHAVIORS MODELING

Figure 3-1 shows the data visualization example in the same day for four days from two of students. Look at the different pattern from both of the users and if we observe the result of plot for more than one weeks we will see the pattern obviously. Based on our observation, we

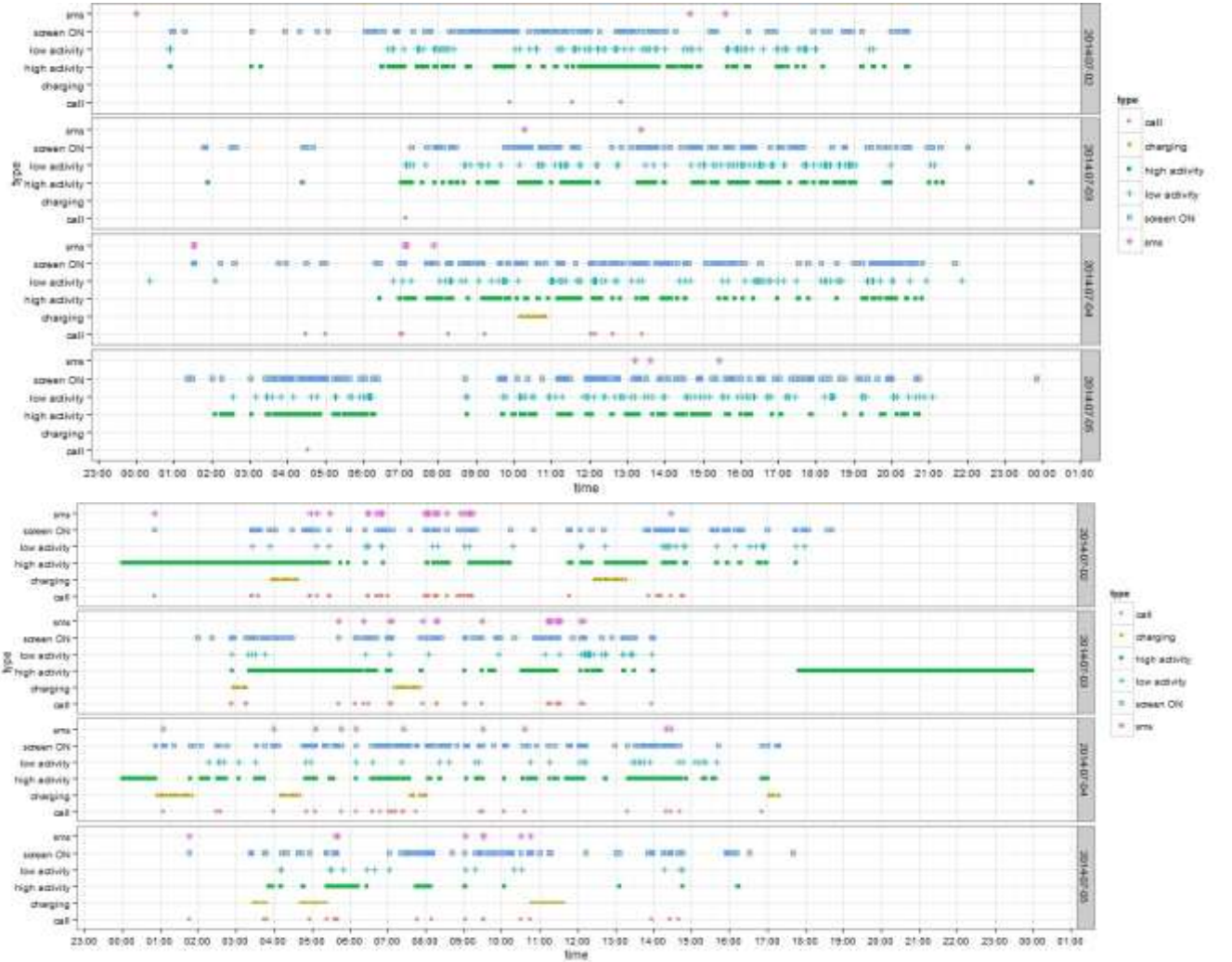


Figure 3-1. Example data visualization from two of students in the same day for four days.

sure that the data features in user personal data log can be used for many purposes such as users identification and classification, recommendation, and etc. In this section, we explain about the our research background and the problem statement, and our proposed methods to achieved our goals.

3.1 Background and Problem Statement

As we mentioned before that many of researchers focus on one feature such as Thang, etc who focus to use accelerometer sensor for human gait identification or for basic activity recognition and etc. That approach which is using one feature is good to know that feature is can reliable or not. The problem when we use only one feature is the lack of sensor accuracy and data loss. We have to aware that the data from user's smartphone are uncertainly data. Not all data are in good condition, sometime maybe the sensor has problem so sensor does not return the value and etc. Another problem is many of researchers mentioned that their approach can achieved good accuracy but they forget if they use experiment environment to collect their research data. In real environment, people, when they use their phone they will use like in his natural life, so we have to consider about realistic data. We explain the realistic data in the points below:

1. In realistic environment, user has different type and brand of smartphone and each smartphone has different type of sensors and hardware specification and capabilities.
2. We could not expect the human actions and their activities, they will doing actions and activities as they want.
3. There is no ideal data collection that can record user personal data for every day 24 hour non-stop, it will drain the battery and spend smartphone resource.
4. There is no ideal data collection that can record all of data without any data loss.
5. When we decide to use many of sensors rather than focus only one sensor, we have to realize that the data from smartphone are heterogeneous data because the data came from multiple sensors and multiple source information.

Based on those reasons, we proposed our approach which is modeling human behavior based on user smartphone data log by combining of many sensors data. In this approach, we tried to develop our system which can deal with realistic data.

3.2 Proposed Methods

3.2.1 Overall architecture

Figure 3.2 sketches our proposed gait based BCS using fuzzy commitment scheme based on binary BCH codes. The objective of this system is to biometrically encrypt a cryptographic key (i.e. symmetric key) m using user's biometric gait. This key will be successfully replicated if the user provides a fresh template which is sufficiently close to the original which has been registered before according to the Hamming distance metric. The system consists of two phases including enrollment phase and authentication phase which are briefly described as following

3.2.2 Similarity Pattern Detection

We use BCH $\text{BCH}_2(n_c, k, t)$ to denote a binary BCH code, where n_c is the code length of bits, k is the key length of bits and t is the error correction capability. The binary cryptographic key m of length k is generated randomly corresponding to each user and then be encoded into the codeword c of length n_c using $\text{BCH}_2(n_c, k, t)$ encoding scheme. After that, we bind the extracted binary gait template with c yielding secured δ . The method used to bind these two binary strings is *exclusive-OR* operation. We summarize all of essential steps both in enrollment phase and verification phase in our system as following

Motif Identification

1. Select a $\text{BCH}_2(n_c, k, t)$ by pre-defining parameters including the length n_c of the codeword, the length k of secret key.

For each user i , biometric template $T_i \in \mathbb{R}^{n_r}$ is extracted using the method in Section 3.2.2

2. Determine the mean over all feature vectors $\vec{\mu}$ and extract binary gait templates $\omega_i \in \{0,1\}^{n_r}$ using the method in the Section 3.2.3. Then, discard T_i

3. Determine the reliable bit indices rel_idx_i and reducing the length of ω_i to n_c by only selecting first n_c bits among n_r based on rel_idx_i
4. Store $\vec{\mu}$, rel_idx_i as helper data for further use to construct fresh binary templates in the authentication phase
5. Randomly generate a binary secret key m_i with the length of k
6. Calculate the hash code of m_i by using a cryptographic hash function h (e.g. SHA, MD5, etc.). Then, discard m_i and store $h(m_i)$.
7. Encode m_i using $BCH_2(n_c, k, t)$ encoding scheme to obtain the codeword c_i
8. Bind c_i with ω_i using *exclusive-OR* operator yielding δ_i , and then store δ_i

Behavior Profiling

For each user i , a fresh biometric template $T'_i \in \mathbb{R}^{n_r}$ is extracted using the method in the Section 3.2.2 same as in the enrollment phase

1. Extract binary gait templates ω'_i with length of n_c using the method in the Section 3.2.3 with the help of $\vec{\mu}$ and rel_idx_i which are previously stored.
2. Bind ω'_i with δ_i using exclusive-OR operator to obtain a corrupted codeword c'_i
3. Employ BCH decoding algorithms to obtain the key m'_i from c'_i
4. Calculating hash code $h(m'_i)$ using the equivalent cryptographic hash function as in the enrollment phase
5. Matching $h(m_i)$ with $h(m'_i)$, if $h(m_i) = h(m'_i)$, the user i is authenticated.

Otherwise, it will be rejected.

$$verify(U_i) = \begin{cases} \text{true}, & \text{if } h(m_i) = h(m'_i) \\ \text{false}, & \text{if } h(m_i) \neq h(m'_i) \end{cases}$$

4. EXPERIMENTAL RESULTS

4.1 Result and Discussion

4.1.1 Behavior Identification

Total 8500 patterns are extracted from the dataset by using our segmentation algorithm. Around $\frac{8500}{38}$ patterns corresponding to each volunteer are split into two separated parts. The first part is used for training (*T-part*) and the remaining is used for prediction (*P-part*). Second, we analyze the impacts of installation errors to segmentation algorithm and the classification accuracy. Note that a perfect accuracy rate of segmentation is achieved when using our algorithm with the transformed *Z*-signal. All gait cycles are detected and segmented correctly. Table 4.2(b) illustrates the performance of the segmentation task with/without fixing disorientation error. As discussed above, the periodicity of walking is only represented well in transformed *Z*-signal. Without rectifying such issues, the segmentation algorithm could not determine precisely the regularity of gait cycles caused by *Z*-signal's instability. Therefore, each segmented pattern could not only represent a sequence of consecutive gait cycles well. That leads features extracted from these patterns could not represent the characteristics of walking style of individuals as well. As a result, the classification accuracy rate is contaminated. Even with using segmentation based on fixed length, the best achieved classification rate at length of 3000 ms is also worse (79.53%).

4.2 Experimental Evaluation

The variation of biometric gait could be influenced by acquisition conditions. Since this is the early approach of gait based BCS not using PR-ML algorithms to handle natural variations of biometric gait, we only consider gait signals not to be influenced by many

environmental conditions such as the influence of footgear, the installation errors, etc. Hence, we exclude gait signals which are significantly influenced by these conditions. Only signals acquired when the phone is placed vertically inside the trouser pocket with a fixed orientation and position are selected. Totally, we obtained 34 out of 38 users satisfying our conditions above and having at least 16 gait templates extracted by our proposed segmentation and template extraction method. Each extracted template consists of $n = 4$ consecutive gait cycles and each gait cycle is normalized to $n_{gc} = 32$ samples of length. Therefore, templates will have the equal length of $32 \cdot 4 \cdot 3 = 384$ samples where 3 is the number of dimensions in the acquired signal including X, Y, Z as described in the Section 3.2.2. After that these real-valued gait templates are resampled using interpolation to appropriate lengths for binarization and key binding scheme. Finally, such resampled gait templates will be equally divided into two parts used for training and testing.

4.2.1 Time Execution and Performance

Looking at the case of the Euclidean distance distribution of real-valued templates, the discrimination is likely to be low. In a more details, the distribution areas of same and different users mostly distribute from 0 to 0.2. That means gait templates between users are likely to be similar. Therefore, applying a threshold-based classification on such templates will result in a high error rate. From our experiment, we observe that extracting binary templates using the quantization method not only makes such templates be applicable to binding with binary BCH codewords but also increase the discrimination property. This is because binary templates only contains bits having high reliable. As shown in the Figure 4.3, the Hamming distances of intra- and inter- class templates are more discriminant and distribute mostly around 0.2 and 0.5

respectively. Templates between users are more dissimilar so that determining an appropriate threshold to classify such templates is more straightforward to achieve an acceptable error rate.

4.2.2 Limitation

Spline interpolation is necessarily adopted to resample gait templates from the original length of $n_0 = 384$ to appropriate values of n_r for extracting binary templates having sufficient length to bind with a binary BCH codeword (e.g. $n_c = 511$). Hence, we analyze the impacts of resampling process on the gait template similarity. Figure 4.4(a) shows that the variation of real-valued templates is not influenced by the resampling process.

In BCH codes, the length of information is inversely proportional to the number of correcting errors t . The larger the t is, the lower the information would be. For example, suppose the BCH codeword of length n_c is 511 bits, if t is up to 25 bits approximately 5% of n_c , the length of key k will be 157 bits. If t is up to 121 bits $\approx 24\%$ of n_c , k will be reduced significantly to 10. Applying cryptographic hash functions to conceal the cryptographic key at this length is insecure. Hence, in our system, we set t to be approximately 12% for k to be large enough.

As discussed above, the FAR and FRR reflect the security and friendliness of a BCS, respectively. In our system, we prioritize the security so that our objective is to make the FAR always equal to 0% and the FRR is as low as possible. To do that, the appropriate value of n_r is selected based on analyzing the distance distribution of intra- and inter- class binary templates as already illustrated in the Figure 4.4(b). Table 4.3 specifically shows our selected values of n_r . At such values, the normalized Hamming distance of extracted binary templates between users is always larger than 12% so that the expected FAR of 0% could be achieved,

whereas the normalized Hamming distance of gait templates of the same users would be mostly lower than 12%, hence a low FRR could be achieved.

5. RELATED WORKS

In this section we will explain about previous work which related with exploring user personality and user smartphone log. Smartphone log consist of many of data such as contact, call log, SMS log, GPS, Wi-Fi, Bluetooth, etc. So, we can choose which data or information features that want to explore. For example is contact data, from this data we can explore many thing. [11] they collected the contact list and then tried to analysed using several features such as communication intensity, regularity, medium, and temporal tendency. By using machine learning techniques and their method they can achieved up to 90 % accuracy to classify life facets/type of relation in contact (family, work, social). Another interesting research conducted by [12], they proposed SmartPhonebook, it is like an artificial assistant which recommends the candidate callees whom the users probably would like to contact in a certain situation. The approach is they used social contacts based on the contact patterns, while it extracts the personal contexts based on the contact patterns, the personal contexts means such as the user emotional states and behaviors from the mobile log. They use Bayesian networks for handling the uncertainties in the mobile environment. The example work based on call and SMS log, such as [9], they tried to predict the spending behavior for couples in terms of their tendency to explore diverse businesses, become loyal customers, and overspend. They use the social features such as face to face interaction, call, and SMS logs. So, this research related with business, they said that the smartphone log could be used for predicting customer type such as loyal customers or overspend and in this research they found that using their approach social features could be better predictors of spending behavior of a couple than personality variables. Example work based on location features, [13] They said how proximity, location, and user personality such as friendship could play important role in understanding user behavior. They found three things : friendship (SMS contacts and facebook friendship) in

proximity has a significant impact on traffic consumption, personality tends to impact application preference and consumption, applications can have different contextual usages based on the location. Another research which is focus on location, [14] in this paper they utilizing location information which can obtained from phone sensors (GPS, WiFi, GSM, accelerometer sensors). They proposed a new framework to discover places of interest based on location where the user usually goes and stays for a while.

From the data which mentioned before, we see that we can exploit call log, SMS log, contact, GPS, and smartphone sensor for many purposes. We still have many of android features that we can explore, another example except that already we mentioned, such as the list of application installed in android devices, [15] This paper, the author tried to investigate how user traits can be inferred by single snapshot of installed apps. They use SVM with minimal external information such as the religion, relationship status, spoken languages, and countries of interest, and the user is parent of small children or not. They collected data from over 200 smartphone user, and the list of installed apps, by using their approach, they can achieve over 90 % of precision. All of previous work which we mentioned, they focus on relation between user personality or user behavior with smartphone data, but on the other side we have to consider about user privacy, so research from [16] they are proposed a different approach that uses multimodal mobile sensor and log data to build framework called mFingerprint. The things that make this framework different with others is this framework does not expose raw sensitive information from the mobile device such as the exact location, Wi-Fi access points, or apps installed so it will save user privacy. By testing on 22 users during 2 months, with their approach they can achieve 81% accuracy across 22 users over 10 day intervals. We also have the data from previous research which was doing research related with user personality but in different directions such as, [17] the authors use virtual world (secondlife.com) to examine how satisfaction in the virtual world was affected by personality differences. They are involving 297 students engage in a virtual tutorial group in Second life

and they found that small variations in personality between the virtual and real world groups such as being helpful, sociable, seeking recognition, or submissive could lead to greater satisfaction of the discussion.

Not only user personality that we can predict based on smartphone log data but also happiness [6], stress [18], mood [7], or maybe we can create application which can help human doing daily routines [19]. [6] This paper provides the evidence that we can predict the happiness of human based on their phone log. In this paper, the author proposed approach using Random Forest classifier to recognize daily happiness of person which obtained from the mobile phone usage data (call log, SMS, and Bluetooth proximity data), and background noise. They can achieve 80.81% of accuracy for classify 3-class daily happiness (happy, neutral, and unhappy). [18] This paper proposed new approach for daily stress recognition based on human behavior metrics derived from the mobile phone activity (call log, SMS log, and Bluetooth interaction). The approach is based on Random Forest and Gradient Boosted Machine algorithms, their approach not only on the term of recognition but also for features extraction, selection, and the ensemble recognition model which combines a number of models for each different weather conditions and personality dispositions. They use two class classification problem (stressed and unstressed) and with theirs approach, they can achieved 72.39% of accuracy, it is could be proof that individual daily stress can be predicted from smartphone data. [7] This research is proof that by using phone log we can predict the user mood. The author in this paper tried to develop smartphone service called MoodSense. On this research they studying from 25 iPhone users and using only six information features from mobile log (SMS, email, phone call, application usage, web browsing, and location). By using simple clustering classifier can achieved 61% accuracy on average and improved to 91% when inference is based on the same participant's data.

We also have the data from previous research which focus on personality classification but most of them use the Big Five personalities (Extraversion, Agreeableness,

Conscientiousness, Emotional Stability and Openness to Experience). [20] They develop conceptual model that explains about relationship between user Big Five personality and their satisfaction with basic mobile phone services such as call, message, 3G services. The main propose of this paper is several implications for design of mobile phone services. [21] They said by using smartphone log and their approach, they can predict Big five personality types of users. The authors said, by using their approach they can achieved 42% better than random and on this research they found that Extraversion and Neuroticism were the traits that were best predicted in their study. [8] This paper shows the evidence that any relationship between Big Five user personality traits and users smartphone data log. They collected data from 117 Nokia N95 smartphone users during 17 months period in Switzerland, they use statistical and machine learning approach to classify the user's smartphone data log based on personality.

6. CONCLUSIONS

In this thesis, we proposed two approaches of gait authentication using PR-ML algorithms and biometric cryptosystem, respectively. In the PR-ML based authentication system, although the quality of built-in sensors is low (the sampling rate is only 27Hz), the achieved results are very considerable. It reflects high potentials to deploy our mechanism to support current active mobile authentications such as PIN or password in reality. Since there is currently no public dataset in this field, the comparison between related works is only relative. Therefore, a more realistic dataset is also constructed to evaluate our mechanism fairly. Nevertheless, many environment factors such as human emotion, time effect, disease and ground materials which could be affected to the human gait is not explored yet. Hence, such issues will be considered deeper in future.

Looking at the case of the biometric cryptosystem, we introduce a novel system using gait combined with fuzzy commitment scheme. The achieved performance in terms of FAR,

FRR as well as the key length and the security level are relatively comparative with other state of the art BCSs. The results show the potentials to construct an effective BCS especially on mobile devices since we use mobile sensors to acquire biometric gait and a lightweight model which only require low storage capability and computational complexity. Moreover, gait could be considered as a new modality for multi-modal BCSs. The drawbacks of our work are that the FRR is still rather high which could causes inconvenient for users. Hence, our further work will focus on reducing the rate of FRR by constructing higher discriminant templates as well as finding an optimal quantization scheme for binarization.

Bibliography

Developing and Evaluating Mobile Sensing for Smart Home Control

Authors: Rischana Mafrur, Priagung Khusumanegara, Gi Hyun Bang, Do Kyeong Lee, I Gde Dharma Nugraha, Deokjai Choi
International Journal of Smart Home (IJSH), volume 9, No 3, March 2015

Concept, Design and Implementation of Sensing as a Service Framework (S²aaS)

Authors: Rischana Mafrur, I Gde Dharma Nugraha, Deokjai Choi
International Conference on Ubiquitous Information Management and Communication (ACM IMCOM 2015), January 8-10, 2015, Bali, Indonesia.

Awareness Home Automation System Based on User Behavior through Mobile Sensing

Authors: Rischana Mafrur, M Fiqri Muthohar, Gi Hyun Bang, Do Kyeong Lee, Deokjai Choi
The 9th KIPS International Conference on Ubiquitous Information Technologies and Applications (CUTE 2014), December 16-20 2014, Guam, USA.

Twitter Mining: The Case of 2014 Indonesian Legislative Elections

Authors: Rischana Mafrur, M Fiqri Muthohar, Gi Hyun Bang, Do Kyeong Lee, Kyungbaek Kim and Deokjai Choi
International Journal of Software Engineering and Its Applications (IJSEIA), volume 8, Issue 10, page 191-202, December 2014

References

- [1] T. Faetti and R. Paradiso, "A Novel Wearable System for Elderly Monitoring," *Advances in Science and Technology*, vol. 85, pp. 17-22, 2013.
- [2] P. Pierleoni, L. Pernini, A. Belli and L. Palma, "An Android-Based Heart Monitoring System for the Elderly and for Patients with Heart Disease," *International Journal of Telemedicine and Applications*, vol. 2014, p. 11, 2014.
- [3] L. Tong, Q. Song, Y. Ge and M. Liu, "HMM-Based Human Fall Detection and Prediction Method Using Tri-Axial Accelerometer," *IEEE, Sensors Journal*, vol. 13, no. 5.
- [4] O. Aziza, E. J. Parkc, G. Morid and S. N. Robinovitch, "Distinguishing the causes of falls in humans using an array of wearable tri-axial accelerometers," *Gait and Posture*, pp. 506-512, 2014.
- [5] P. Zhou, Y. Zheng and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *MobiSys '12 Proceedings of the 10th international conference on Mobile systems, applications, and services*.
- [6] A. Bogomolov, B. Lepri and F. Pianesi, "Happiness Recognition from Mobile Phone Data," in *BioMedCom 2013*, 2013.
- [7] R. LiKamWa, Y. Liu, N. D. Lane and L. Zhong, "Can Your Smartphone Infer Your Mood?," in *PhoneSense workshop*, 2011.
- [8] G. Chittaranjan, J. Blom and D. Gatica-Perez, "Mining large-scale smartphone data for personality studies," *Personal and Ubiquitous Computing*, pp. Volume 17, Issue 3, pp 433-450, 2013.
- [9] V. K. Singh, L. Freeman, B. Lepri and A. Pentland, "Predicting Spending Behavior using Socio-Mobile Features," in *BioMedCom 2013*, 2013.
- [10] N. Maisonneuve, M. Stevens, M. E. Niessen and L. Steels, "NoiseTube: Measuring and mapping noise pollution with mobile phones," in *Information Technologies in Environmental Engineering*, 2009.
- [11] J.-K. Min, J. Wiese, J. I. Hong and J. Zimmerman, "Mining Smartphone Data to Classify Life-Facets of Social Relationships," in *CSCW '13 Proceedings of the 2013 conference on Computer supported cooperative work*, 2013.
- [12] J.-K. Min and S.-B. Cho, "Mobile Human Network Management and Recommendation by Probabilistic Social Mining," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, pp. VOL. 41, NO. 3, 2011.
- [13] L. Meng, S. Liu and A. D. Striegel, "Analyzing the Impact of Proximity, Location, and Personality on Smartphone Usage," in *2014 IEEE INFOCOM Workshop on Dynamic Social Networks*.

- [14] R. Montoliu, J. Blom and D. Gatica-Perez, "Discovering Places of Interest in Everyday Life from Smartphone Data," *Journal Multimedia Tools and Applications*, pp. Volume 62, Issue 1, pp 179-207, 2013.
- [15] S. Seneviratne, A. Seneviratne, P. Mohapatra and A. Mahanti, "Predicting User Traits From a Snapshot of Apps Installed on a Smartphone," *ACM SIGMOBILE Mobile Computing and Communications Review*, pp. Volume 18 Issue 2, Pages 1-8, 2014.
- [16] H. Zhang, Z. Yan, J. Yang, E. Munguia Tapia and D. J. Crandall, "mFingerprint: Privacy-Preserving User Modeling with Multimodal Mobile Device Footprints," *Social Computing, Behavioral-Cultural Modeling and Prediction Lecture Notes in Computer Science*, pp. Volume 8393, pp 195-203, 2014.
- [17] J. Sutanto, C. W. Phang, C. H. Tan and X. Lu, "Dr. Jekyll vis-a`-vis Mr. Hyde: Personality variation between virtual and real worlds," *Journal of Information & Management*, p. 19-26, 2011.
- [18] A. Bogomolov, B. Lepri, M. Ferron, F. Pianesi and A. Pentland, "Pervasive Stress Recognition for Sustainable Living," in *The Third IEEE International Workshop on Social Implications of Pervasive Computing*, 2014.
- [19] V. Antila, J. Polet, A. Lämsä and J. Liikka, "RoutineMaker: Towards End-User Automation of Daily Routines Using Smartphones," in *PerCom 2012*, Lugano, 19-23 March 2012.
- [20] R. DE OLIVEIRA, M. CHERUBINI and N. OLIVER, "Influence of Personality on Satisfaction with Mobile Phone Services," *ACM Transactions on Computer-Human Interaction*, pp. Vol. 20, No. 2, Article 10, 2013.
- [21] Y.-A. de Montjoye, J. Quoidbach, F. Robic and A. Pentland, "Predicting people personality using novel mobile phone-based metrics," in *Social Computing, Behavioral-Cultural Modeling and Prediction (2013)*, 2013.

패턴 인식 및 생체인식 암호화 시스템을 이용한 모바일 폰에서의 보행 인증

MAFRUR, Rischana

전남대학교 대학원 전자컴퓨터공학과

(지도교수: 최덕재)

(국문초록)

본 논문에서는 휴대 전화에 내장된 센서 자원을 활용하여 두 가지의 새로운 보행 인증 시스템을 제안하였다.

첫 번째 방법으로 전처리 단계에서 정밀한 검토를 실행함으로써 성능을 향상시킨 패턴 인식 및 기계 학습(PR-ML) 알고리즘을 기반으로 하는 시스템 구축에 초점을 맞추었다. 효과적이며 새로운 분할 알고리즘은 분할 신호를 완벽한 정확성을 갖는 분리된 보행 주기로 제공한다. 그 다음으로, 특징은 시간 및 주파수 영역으로부터 추출된다. 이 시스템은 간단하면서 신뢰성이 높은 모델의 구축을 목표로 하기 때문에 특징 부분 집합 선택 알고리즘은 특징 벡터의 크기뿐만 아니라 분류 태스크의 처리 시간을 최적화하기 위해 적용되고, 최적의 특징 벡터는 SVM 및 RBF 커널을 이용하여 분류된다.

이러한 최적화 방안에도 불구하고 PR-ML 기반의 생체 인식 인증은 여전히 시스템 보안 및 사용자의 개인정보 보호 문제가 남아있다.

본 시스템에서, 인증에 사용되는 본래의 생체 인식 템플릿이나 추출된 특징은 끊임없는 손실을 갖는 데이터 결과를 절충하기 위해서 안전하지 않게 저장된다.

두 번째 방법으로는 사용자의 개인정보 보호와 더불어 시스템의 보안을 보장하기 위해서 Fuzzy Commitment Scheme 방식을 이용한 생체 인식 암호화 시스템을 기반의 보행 인증을 연구하였다. 사용자 확인을 위한 본래의 생체 인식 템플릿의 사용을 대신에 생체 측정 통합 모바일 가속도 센서에 의해 취득한 보행 템플릿을 사용하여 암호화 된 암호 키를 기반으로 한다.

관련 분야의 연구에서 모바일 가속도 센서에 의해 취득한 공용 보행 데이터 집합이 없기 때문에, 자체적으로 38 명의 피험자(남 10, 여 28)로부터 보행 신호를 취득하여 성능을 평가하였다.

PR-ML 방식을 구현함으로써 식별 모드에서 94.93%, zeroFAR, FRR 3.89%에 가까운 정확도를 달성하였으며, 인증 모드에서 4 초 미만의 처리 시간을 달성했다.

또한 보행 기반의 생체 인식 암호화 방식에서, 139 및 50 비트의 키 길이를 갖을 때, 거의 16.18%과 14.71%에 해당하는 최적의 zeroFAR 및 FRR 를 달성하였다.

따라서 본 연구의 결과는 모바일 센서 기반의 보행은 홍채, 지문, 음성 등의 생체 요인과 비교했을 때, 생체 암호 시스템을 구축하는 효과적인 요소로 활용 할 수 있음을 보여준다.

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to my supervisor and advisor Professor Deokjai Choi for his useful comments, remarks and engagement through the studying process of this master thesis. His kind guidance supported me a lot in all the time of research and writing this thesis. I would also be grateful to him not only for enlightening me the first glance of research but also for his valuable advice about the future of my life. I would like to thank also to members of Advanced Network Lab mates: Ki Hyun Bang, Dokyeong Lee, Muhammad Fiqri Muthohar, Gde Dharma Nugeraha, Priagung Khusumanegara, Alvin Prayuda Juniarta who shared their ideas and supported me throughout my master course. I would like to thank also to two of my best Korean friends (Danmbi and Su Hyun) who always help me and teach me many things, you are very nice girls. ☺

I also deeply appreciate the generous support from BK21(+), NRF, ITRC and NIPA in two years for bringing me a great financial support and academic opportunities.

Finally I would like to thank to Allah Almighty God, my beloved prophet Muhammad ﷺ who is the inspiration of my life, my dear parents, my beloved Indonesian community (Kak Wawa, Kak Mimi, Kak Tonton, Mei, and many more) and Muslim community for their endless love and care during my period away from home. Without them, this work could not be done.

June 2015

Rischan Mafrur