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, Rischan

June 2015

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

Department of Electronics and Computer Engineering

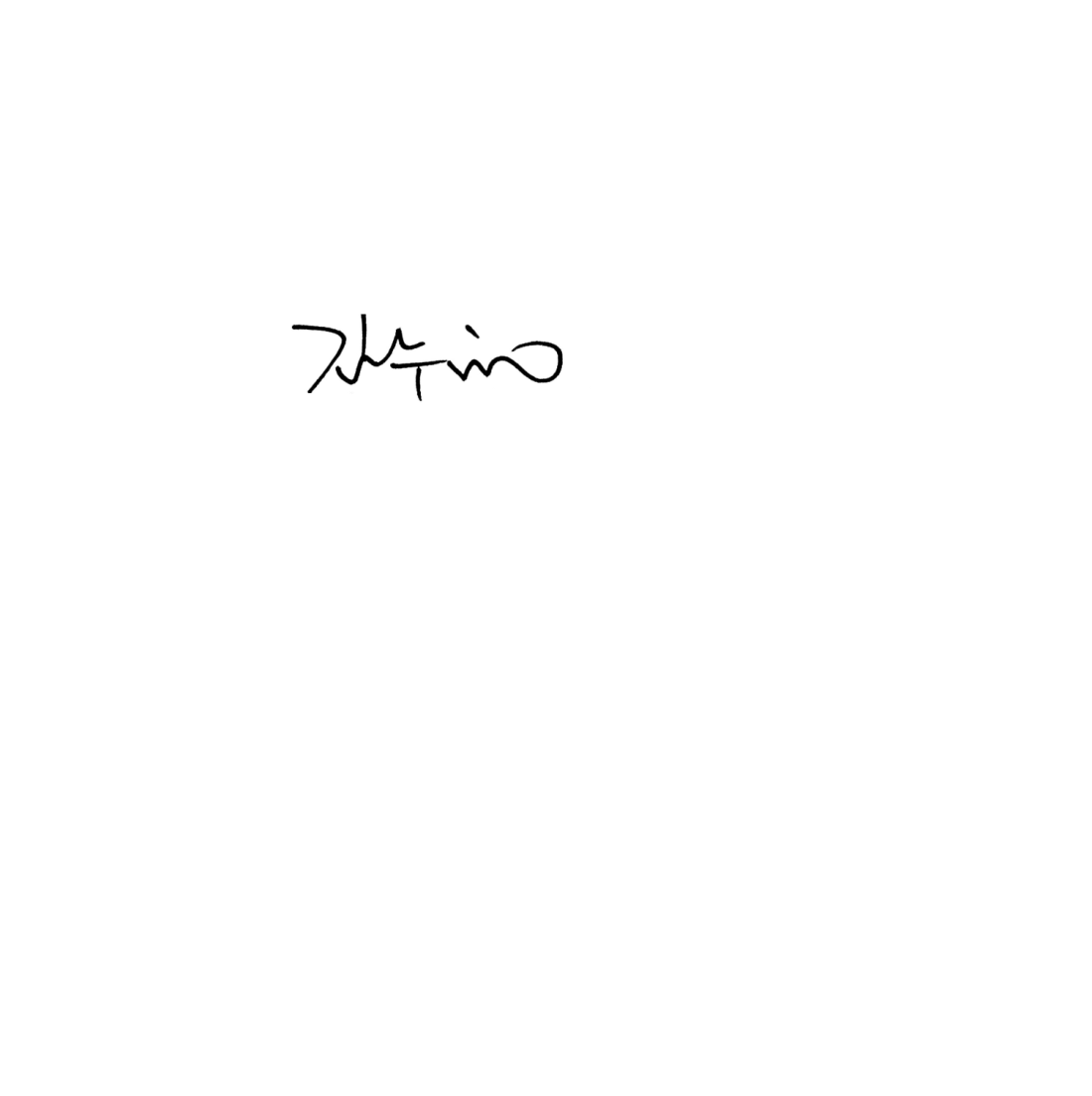
Graduate School, Chonnam National University

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 | C:\Users\ThangHoang\Desktop\Thesis\kbkim.png |
| CHOI, Deok Jai | C:\Users\ThangHoang\Desktop\Thesis\djchoi.png |

June 2015

Table of Contents

[LIST OF FIGURES iii](#_Toc413257797)

[LIST OF TABLES v](#_Toc413257798)

[(Abstract) vi](#_Toc413257799)

[1. INTRODUCTION 1](#_Toc413257800)

[1.1 Overview 1](#_Toc413257801)

[1.2 Contribution 2](#_Toc413257802)

[2. DATASET 2](#_Toc413257803)

[2.1 Data Acquisition 2](#_Toc413257804)

[2.1.1 Application Data Collector 3](#_Toc413257805)

[2.1.2 Dataset Description 4](#_Toc413257806)

[2.1.3 Dataset that used in this research 10](#_Toc413257807)

[2.2 Data Pre-processing 10](#_Toc413257808)

[2.2.1 Data Cleansing 11](#_Toc413257809)

[2.2.2 Dataset Transformation 11](#_Toc413257810)

[2.3 Feature Extraction 12](#_Toc413257811)

[2.3.1 Define Human Activity and Behavior 12](#_Toc413257812)

[2.3.2 Features Description and Extraction 13](#_Toc413257813)

[2.3.3 Human and Machine Time 16](#_Toc413257814)

[2.3.4 List of the Final Features 17](#_Toc413257815)

[3. HUMAN BEHAVIORS MODELING 18](#_Toc413257816)

[3.1 Background and Problem Statement 20](#_Toc413257817)

[3.2 Proposed Methods 21](#_Toc413257818)

[3.2.1 Overall architecture 21](#_Toc413257819)

[3.2.2 Similarity Pattern Detection 22](#_Toc413257820)

[4. EXPERIMENTAL RESULTS 25](#_Toc413257821)

[4.1 Result and Discussion 25](#_Toc413257822)

[4.1.1 Behavior Identification 25](#_Toc413257823)

[4.1.2 Validity of The Results 27](#_Toc413257824)

[4.2 Experimental Evaluation 28](#_Toc413257825)

[4.2.1 Time Execution and Performance 29](#_Toc413257826)

[4.2.2 Limitation 30](#_Toc413257827)

[5. RELATED WORKS 32](#_Toc413257828)

[5.1 Smartphone Personal Data 32](#_Toc413257829)

[5.2 Modeling and Discovering Human Behaviors 36](#_Toc413257830)

[6. CONCLUSIONS 38](#_Toc413257831)

[Bibliography 40](#_Toc413257832)

[References 41](#_Toc413257833)

[(국문초록) 43](#_Toc413257834)

[ACKNOWLEDGEMENT 45](#_Toc413257835)

# LIST OF FIGURES

[**Figure 2.1 (a)** Mobile coordinate system, **(b)** misplacement error, **(c)** disorientation error and **(d)** both cases **Error! Bookmark not defined.**](#_Toc369193731)

[**Figure 2.2** **(a)** Mobile orientation after applying rotation matrix, **(b)** the relative position between the phone and its wearer **Error! Bookmark not defined.**](#_Toc369193732)

[**Figure 2.3** Multi-level wavelet decomposition **Error! Bookmark not defined.**](#_Toc369193733)

[**Figure 2.4** Illustration of a gait cycle **Error! Bookmark not defined.**](#_Toc369193734)

[**Figure 2.5** **(a)** Auto-correlation coefficients, **(b)** detected marking points in Z-signal **Error! Bookmark not defined.**](#_Toc369193735)

[**Figure 3.1** The fuzzy commitment scheme proposed by A. Juels **Error! Bookmark not defined.**](#_Toc369193736)

[**Figure 3.2** The architecture of gait based BCS using fuzzy commitment scheme 22](#_Toc369193737)

[**Figure 3.3** Illustration of extracted real-valued gait templates of 3 users **Error! Bookmark not defined.**](#_Toc369193738)

[**Figure 4.1** **(a)** Confusion matrix of the gait recognition, **(b)** the classification accuracy of feature subsets by applying SFFS and SFS algorithms 26](#_Toc369193739)

[**Figure 4.2** the FRR at zeroFAR before and after applying voting scheme **Error! Bookmark not defined.**](#_Toc369193740)

[**Figure 4.3** The density distribution of real-valued and binary gait templates 29](#_Toc369193741)

[**Figure 4.4** **(a)** The Euclidean distance distribution of real-valued templates, **(b)** The Hamming distance of binary templates 30](#_Toc369193742)

[**Figure 4.5** **(top)** The Hamming distance distribution of binary templates, and **(bottom)** the FAR and FRR of our proposed BCS **Error! Bookmark not defined.**](#_Toc369193743)

# LIST OF TABLES

[**Table 2.1** Feature subset(s) selection using SFS and SFFS algorithms **Error! Bookmark not defined.**](#_Toc373313414)

[**Table 4.1** The performance of reducing feature dimension versus non-reducing 26](#_Toc373313415)

[**Table 4.2** **(a)** Improvements of segmentation based on gait cycles compared with fixed length, **(b)** the influence of disorientation error to the classification results 27](#_Toc373313416)

[**Table 4.3** Optimal length of the real-valued template corresponding with the requisite length of the binary gait template 32](#_Toc373313417)

[**Table 4.4** The FAR and FRR of our gait based BCS with different template length using binary BCH codes **Error! Bookmark not defined.**](#_Toc373313418)

[**Table 4.5** The performance of state of the art BCS using physiological and behaviroal modalities using various scheme such as Fuzzy Commitment scheme (FCS), Fuzzy Extractor, (FE), Feature Transformation (FF) and Password Hardening (PH) **Error! Bookmark not defined.**](#_Toc373313419)

[**Table 5.1** State of the art gait authentication using Standalone (S) and Mobile sensor (M) including Accelerometer (A), Rotation Sensor (R) by approaches: Template Matching (TM), Machine Learning (ML) and Hybrid (H) 37](#_Toc373313420)

**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 38 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.

# INTRODUCTION

## 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) human fall detection, transportation (monitoring road and traffic condition), personal and social behavior, environmental monitoring (pollution, weather) and etc. To develop such system, we have to collect the user personal data and then analyze it. In this research, we try to collect 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.

In term of personal data sensing, 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 try to identify human behavior and create their behavior model.

## Contribution

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

# DATASET

## Data Acquisition

This section explain about 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 section explain about our application which is used in this research for collecting user personal data. Dataset description section explain about our dataset itself, dataset that we have collected from user’s smartphone. Dataset that used in this research 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 some data that related with our research goal.

|  |  |
| --- | --- |
| C:\Users\rischan\Music\funf.png | D:\Dropbox\thesis\figures\dataviewinsmartphone.png |
| Figure 2.1. Funf Open Sensing Framework | Figure 2.3. Personal data in user’s smartphone |

### 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 Funf[[1]](#footnote-1) and also Funf developer site[[2]](#footnote-2).

**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 |

### Dataset Description

Our application follows opportunistic sensing because we do not want to bothering user much. To do that we must define the time (interval and duration), when the application will request the data from the smartphone. Interval means how many times in second system will send data request to the smartphone. The example, we set interval 300 seconds means 5 minutes, so application will 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 data because without duration is useless to collect the sensors data. The example of duration, when we set interval 120 seconds and duration 0.07 s menas 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 from each probes. Those interval and duration already tested and we thought those setting was optimum one but we can change those setting by change the value on the string.xml in android project. Figure 2.2a shows the string.xml file in the directory of android project and Figure 2.2b shows inside the string.xml file, we can change value of interval and duration in that file.

|  |  |
| --- | --- |
| D:\Dropbox\thesis\figures\ppt2\pptdata\sstringxml.JPG | D:\Dropbox\thesis\figures\ppt2\pptdata\funfsettingxml.JPG |
| (a) | (b) |
| Figure 2.2. (a) strings.xml file in project directory, (b) inside the string.xml file | |

To make easy for remembering, we classify the data to three of data categorization, are:

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

On request data means we ask current values (information) from android system such as location, battery, nearby Bluetooth and etc. Historical data means the data that already store 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 store in SQLite database format with (*\*.db*) extension, the view of data can be seen in Figure 2.3. To open those database, we can use SQLite browser that can be download in SQLite browser main site[[3]](#footnote-3). 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 data) that provided by our application. The total of probes which provided by our application are 19 probes and we use 9 probes for this research thesis.

**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 |

In this thesis, 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. Our application is collecting location information from the user’s smartphone, the value that returned by system is like in the box below:

That data which from location probes 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 documents about the data itself can be accessed in our projects site[[4]](#footnote-4), open “Data Documentation” directory. 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 for another purposes.

{"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}

The size of all of data after extracted is around 28.7 GB. Extracted data contain 47 directories in different name for each student 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 |

### 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 that 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 from 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 prefer to use application messenger such as Kakao, Whatsapp, etc instead of SMS application. In this research, we use data from 38 students which all of data are available during 2 months.

## Data Pre-processing

Not all of data from user’s smartphone are 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.

### Data Cleansing

Funf library which we used as base of our application for collecting user personal data has problem in historical data. Historical data is the data which are already stored in android database system such as contact, SMS log, call log, and etc. We use 86400 second interval, means we will copy those data from android database system to our application database one time every day. It will make duplication in our database, so 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 also clean the noisy data.

### Dataset Transformation

As we mentioned in data description section that the size of all of the data is around 28 GB. To process those of data, if 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 2.4. Data preprocessing flows

We have three kind of preprocessing modules and each module will store new data to csv file. Figure 2.4 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 fit format before creating behavior model. This way will reduce time processing and computer resource.

## 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.

### 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 arrived 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. With Whom (e.g. Call activity, with whom: his mother, friend, 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 activity happened but in different time and location, or maybe different activity but in same time and location, and vice versa.

### 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 any activity label.
  + 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 about the name of activity 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 define the user location in outside and nearby Wi-Fi and nearby Bluetooth can be used to define user location inside building.
* With Whom (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 application 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 to know the user interaction with other behavior.

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 will be fill with (*“none”,”low”,* and *”high”*).
2. The values of GPS are *value1* is latitude value and *value2* is the longitude value.
3. The values of Wi-Fi probe are *value1* is the list of nearby Wi-Fi SSID name, *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 list 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 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 row 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 |  |  |
| 2. | SimpleLocationProbe | Latitude | Longitude |  |
| 3. | WifiProbe | List of nearby SSID | MAC | Signal strength (dB) |
| 4. | BluetoothProbe | List of nearby Bluetooth devices |  |  |
| 5. | BatteryProbe | Status |  |  |
| 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 2.5. 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.



**Figure 2.5** Sample output of the features extraction in Pre-Processing II.

### Human and Machine Time

Machine is different with human, if machine can calculate 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 9.00 AM”. 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.

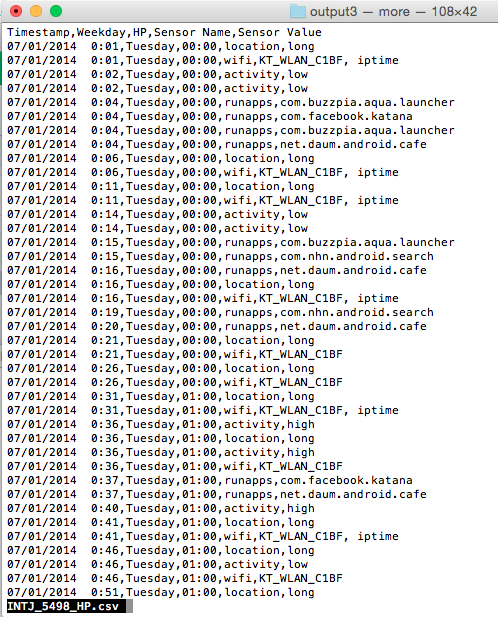
### List of the Final Features

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:
   1. If minute less than 30 minutes will be round down
   2. 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.
   1. If the previous value of GPS location not change, it means no movement, so the value filled by “*same*”.
   2. If the moving distance between 0.0001 ~ 0.0005, it means little movement, so the value filled by “*little*”.
   3. If the moving distance more than 0.0005, it means long movement, so the value filled by “*long*”.
   4. 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 2.5, in same time the value of Wi-Fi is one SSID in one row, and also for the Bluetooth. That is because every 5 minute our application store the lists of nearby access point and Bluetooth devices and the 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 2.6. 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.

# HUMAN BEHAVIORS MODELING

Biometric cryptosystems (BCS) (aka biometric encryption, biometric template protection) have been being developed to enhance both privacy and security of the conventional biometric systems [35-37, 58, 59, 61]. In these systems, biometric templates are bound with a cryptographic key, making it computationally challenging to retrieve either the key or the original template from the biometrically encrypted data. 

**Figure 2.6** Sample output of the features extraction in Pre-Processing III (Final Features).

A user would be authenticated / identified if he provided a new template which is sufficiently close to the original which is registered to the system before. The backbone of this system is based on conventional cryptography wherein a cryptographic key is employed as the authentication factor. However unlike conventional password-based system where key management remains security and utilization issues [38, 39], BCS links biometrics with cryptography to overcome such vulnerabilities by biometrically managing such keys with the more secure and more convenient way. In this thesis, we additionally introduce a gait based biometric cryptosystem. Our BCS relies on fuzzy commitment scheme [36] to deal with the natural variations of gait templates. The original gait template is always discarded and is not stored in the system so that the security and privacy are significantly enhanced. Such templates are acquired by an integrated accelerometer in mobile devices and then, are transformed for feasibly binding with cryptographic keys. Helper data supporting authentication are stored in the mobile storage and biometrically encrypted to prevent an attacker from retrieving either the key or original templates. Hence, they are still secure even though mobile devices are lost or compromised.

## Background and Problem Statement

The backbone of this system is based on conventional cryptography wherein a cryptographic key is employed as the authentication factor. However unlike conventional password-based system where key management remains security and utilization issues [38, 39], BCS links biometrics with cryptography to overcome such vulnerabilities by biometrically managing such keys with the more secure and more convenient way. In this thesis, we additionally introduce a gait based biometric cryptosystem. Our BCS relies on fuzzy commitment scheme [36] to deal with the natural variations of gait templates. The original gait template is always discarded and is not stored in the system so that the security and privacy are significantly enhanced. Such templates are acquired by an integrated accelerometer in mobile devices and then, are transformed for feasibly binding with cryptographic keys. Helper data supporting authentication are stored in the mobile storage and biometrically encrypted to prevent an attacker from retrieving either the key or original templates. Hence, they are still secure even though mobile devices are lost or compromised.

## Proposed Methods

### 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) 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

In the enrollment phase, gait signal of a user will be firstly acquired and then is pre-processed to eliminate the influence of the acquisition environment. Real-valued gait templates are then extracted and binarized. After that, reliable bits in the binary template are determined via estimating the error probability of each component using statistical analysis. Concurrently, a binary cryptographic key is generated randomly. On the one hand, this key will be encoded using binary BCH codes to mitigate the variations of gait characteristics. The encoded key will be bound with the binary template forming secured. On the other hand, a cryptographic hash function will be applied to keep safely under concealed form. Helper data used for binary template construction along with will also be stored for further use to replicate the key in the authentication phase.

C:\Users\ThangHoang\Desktop\res\biometric cryptosystem model_NO_ATTACK.emf

**Figure 3.1** The overall architecture of our proposed gait based BCS using fuzzy commitment scheme where ⊕ denotes the OR-exclusive operation

In the authentication phase, the user supposed to be will provide a fresh gait template. Such template is also preprocessed and binarized using helper data which is previously stored in the enrollment phase. After that, extracted binary template will be bound with the secured returning a string. This string will be decoded using BCH decoding to obtain a fresh key. Finally, the hash code of will be matched with for authentication decision. The milestones of our system are described in detail as in the following

### Similarity Pattern Detection

We use BCH to denote a binary BCH code, whereis the code length of bits, is the key length of bits and is the error correction capability. The binary cryptographic key of length is generated randomly corresponding to each user and then be encoded into the codeword of length using encoding scheme. After that, we bind the extracted binary gait template with yielding secured. 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 Identitication*

1. Select a by pre-defining parameters including the length of the codeword, the length of secret key.

For each user , biometric template is extracted using the method in Section 3.2.2

1. Determine the mean over all feature vectors and extract binary gait templates using the method in the Section 3.2.3. Then, discard
2. Determine the reliable bit indices and reducing the length of to by only selecting first bits among based on
3. Store as helper data for further use to construct fresh binary templates in the authentication phase
4. Randomly generate a binary secret key with the length of
5. Calculate the hash code of by using a cryptographic hash function (e.g. SHA, MD5, etc.). Then, discard and store .
6. Encode using encoding scheme to obtain the codeword
7. Bind with using *exclusive-OR* operator yielding , and then store

#### *Behavior Profiling*

For each user , a fresh biometric template is extracted using the method in the Section 3.2.2 same as in the enrollment phase

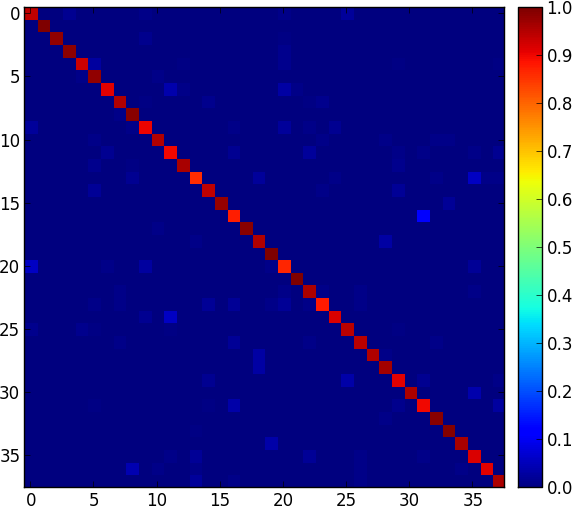
1. Extract binary gait templates with length of using the method in the Section 3.2.3 with the help of and which are previously stored.
2. Bind with using exclusive-OR operator to obtain a corrupted codeword
3. Employ BCH decoding algorithms to obtain the key from
4. Calculating hash code using the equivalent cryptographic hash function as in the enrollment phase
5. Matching with , if , the user is authenticated. Otherwise, it will be rejected.

# EXPERIMENTAL RESULTS

## Result and Discussion

### Behavior Identification

Total 8500 patterns are extracted from the dataset by using our segmentation algorithm. Around 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*). We used libsvm[[5]](#footnote-5) [26] as the tool to perform SVM with RBF kernel. The performance of RBF kernel fully depends on selecting parameters . In order to construct an optimal SVM model, we perform a strategy to find the good yielding the best classification result. Features described in section 3.4 are extracted on both *T-part* and *P-part*. To deal with over-fitting issue, 10-fold cross validation is applied on T-part with various . The yielding the best cross validation accuracy will be selected. According to [26], we tried exponentially growing sequences of and to identify the ‘coarse’ pair first (e.g. ). Subsequently, a more detailed search is performed to identify a finer () yielding an optimal cross-validation accuracy. The best is identified at the cross-validation accuracy of 98.71%. Then, whole *T-part* is trained again using () to obtain the final SVM model. An overall accuracy rate approximately 94.93% is achieved when using such model to predict *T-part*. Figure 4.1(a) illustrates the confusion matrix of prediction result.

 C:\Users\ThangHoang\Desktop\Authentication\Paper\unknown\image\SFFS_SFS_alg.emf

(a) (b)

**Figure 4.1** **(a)** Confusion matrix of the gait recognition using SVM and RBF kernel, **(b)** the classification accuracy of feature subsets by applying SFFS and SFS algorithms

Additionally, by applying the SFS and SFFS algorithm, the dimension of feature vectors is reduced and the classification accuracy is slightly increased as well. The processing time is also ameliorated significantly (Table 4.1 and Figure 4.1(b)). By applying SFFS, the prediction time only costs 411 milliseconds per sample using mobile resources. In authentication mode, a task requires to predict on 9 consecutive samples (as discussed in the Section 4.2.3). It costs less than 4 seconds to make a decision. This is an acceptable level compared to original case (≈ 20 seconds). Note that processing time is very important in mobile applications since we aim to deploy a lightweight authentication model running directly on mobile phone. Nowadays, it is likely to be optimized by its weight, power and size rather than computational power (e.g. CPU, memory). Hence reducing feature dimension will help the mobile device to perform classification task more quickly so that the interaction between the phone and its user is also improved.

**Table 4.1** The performance of reducing feature dimension versus non-reducing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scheme** | **No. of Subsets** | **Accuracy** | **Loading Time** | **Prediction Time** |
| **Original** | 29 | 94.34 % | 205897 ms | 2280 ms |
| **SFS** | 12 | 94.90 % | 86799 ms | 398 ms |
| **SFFS** | 13 | 94.93 % | 84223 ms | 411 ms |

### Validity of The Results

Before discussing the impact of mobile installation, we first compare the performance of segmentation based on gait cycles against previous studies used fixed size segmentation [15-17]. Since walking is a regularly cyclic activity, it is relatively easy to perceive that segmentation based on gait cycle always yields a better classification result compared with based on a fixed length (Table 4.2(a)).

**Table 4.2** **(a)** Improvements of segmentation based on gait cycles compared with fixed length, **(b)** the influence of disorientation error to the classification results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (a) | | |  | (b) | | |
| **Segmentation method** | | **Accuracy** |  | **Segmentation method** | **Fixing dis-orientation** | **Accuracy** |
| Fixed length | 3000 ms | 87.88 % |  | Fixed length | No | 79.53 % |
| 6000 ms | 87.78 % |  | Our algorithm | No | 84.03 % |
| 9000 ms | 84.73 % |  | Yes | 94.93 % |
| Gait cycle | 2 GCs | 92.26 % |  |  |  |  |
| 4 GCs | 94.93 % |  |  |  |  |
| 8 GCs | 90.94 % |  |  |  |  |

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 -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 -signal. Without rectifying such issues, the segmentation algorithm could not determine precisely the regularity of gait cycles caused by -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%).

## 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 consecutive gait cycles and each gait cycle is normalized to samples of length. Therefore, templates will have the equal length of samples where 3 is the number of dimensions in the acquired signal including 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.

### Time Execution and Performance

Figure 4.3 illustrates the normalized Euclidean distance distribution of real-valued gait templates and the Hamming distance distribution of binary template based on reliable bits selection. The Euclidean distance of two real-valued templates of length is calculated by

|  |  |
| --- | --- |
|  | (4.1) |

The Hamming distance of two binary templates of length is calculated by

|  |  |
| --- | --- |
|  | (4.2) |

C:\Users\ThangHoang\Desktop\fig1.emf

**Figure 4.2** The density distribution of real-valued and binary gait templates

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.

### **Limitation**

Spline interpolation is necessarily adopted to resample gait templates from the original length of to appropriate values of for extracting binary templates having sufficient length to bind with a binary BCH codeword (e.g. ). 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. The similarity of such templates does not changed significantly when they are resampled to various lengths.

C:\Users\ThangHoang\Desktop\res\euc_dist\63to4158interp.emf C:\Users\ThangHoang\Desktop\res\ham_dist\511bits_variousn.emf

**Figure 4.3** **(a)** The Euclidean distance distribution of real-valued templates when they are resampled to various length of , **(b)** The Hamming distance of binary templates of length extracted from when is resampled to be times of

However, although resampling process does not modify the similarity of real-valued templates, it adversely affects the reliable bits selection process which determines reliable bits out of to extract binary templates. If is much larger than, determining reliable bits based on their error function may be instable because of existing a large number of bits having a same error probability. Such bits could be selected arbitrarily. As a result, the extracted binary templates are more sensitive to error. Figure 4.4(b) illustrates the changes of distribution area when is resampled to a value of times of in term of . The distance of intra class templates are getting decreased and towards to 0. In other words, binary templates of the same user are getting more similar. That makes the number of errors need to be corrected in a template of the same user will be reduced. However, the distribution area of inter-class templates not only reduces to 0 but also be getting wider and mixes with the distribution of intra class templates. This phenomenon also happens in different cases of (e.g. ). Hence, an appropriate value of should be selected according to the requirement of to trade off the similarity of intra- and inter- templates.

In BCH codes, the length of information is inversely proportional to the number of correcting errors. The larger theis, the lower the information would be. For example, suppose the BCH codeword of length is 511 bits, if is up to 25 bits approximately 5% of , the length of key will be 157 bits. If is up to 121 bits ≈ 24% of , 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 to be approximately 12% for 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 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. 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

**Table 4.3** Optimal length of the real-valued template corresponding with the requisite length of the binary gait template

|  |  |
| --- | --- |
|  |  |
| 127 | 317 |
| 255 | 586 |
| 511 | 126 |

# RELATED WORKS

## Smartphone Personal Data

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. [1] 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 [2], 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 [3], 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, [4] 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, [5] 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, [6] 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 [7] 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, [8] 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 [9], stress [10], mood [11], or maybe we can create application which can help human doing daily routines [12]. [9] 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). [10] 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. [11] 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). [13] 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. [14] 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. [15] 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.

## Modeling and Discovering Human Behaviors

Human gait has been considered as a particular style and manner of moving human feet and hence contains the information of identity authentication. In a more detailed level view, the mechanism of human gait involves synchronization between the skeletal, neurological and muscular system of human body [4]. In 2005, H. Ailisto et al. were the first to propose the gait authentication using wearable sensor [13] and this area was further expanded by Gafurov et al. [10]. In general, sensors are attached to various positions on human body to record locomotion signal. Various sensors are experimented including gyroscope, rotation sensor but acceleration sensor (or accelerometer) is the most commonly used. In this field, there are two typical approaches: (1) Template Matching (TM) and (2) Machine Learning (ML). In (1), acquired signal is preprocessed and then split into patterns. Best patterns which represent the most characteristics of the subject are considered as representative gait templates. They are then stored as referred templates corresponding to individual. Various distance metrics such as Dynamic Time Warping (DTW) [9, 19, 14], Euclidean distance [8, 9], auto-correlation [13], nearest neighbors [11] are used for calculating the similarity score between a given pattern and referred templates.

Second method is the most popular approach used in pattern recognition areas. In this approach, gait signal is segmented into patterns. On each pattern, features are extracted in time domain, frequency domain, and wavelet domain or by special techniques such as time delay embedding [18]. Extracted feature vectors are then classified using supervised classifiers like HMM [16], SVM [17, 15, 20, 14, 18], ANN [5], LDA [5]. Some other works propose hybrid approaches in which either distance metrics such as DTW [7], Euclidean [10, 12], are used to measure the similarity scores of features extracted in time and frequency domains, or similarity scores of gait templates can be considered as features which are used for classification [6].

**Table 5.1** State of the art gait authentication using Standalone (S) and Mobile sensor (M) including Accelerometer (A), Rotation Sensor (R) by approaches: Template Matching (TM), Machine Learning (ML) and Hybrid (H)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Study** | **Sensor /**  **Sampling rate** | **Location** | **Methods** | **No. of Subjects** | **Results** |
| [14]  [6] | M–A / 27Hz  S–A / 50Hz | T Pocket  Ankle | TM, ML  H | 11  22 (16M 6F) | 79.1%, 92.7% RR  3.03% EER |
| [5] | 9 S–R | Body | ML (LDA) | 30 (25M 5F) | ~ 100% RR |
| [15] | M–A | T Pocket | ML (SVM) | 36 | HTER: 10.1% |
| [7] | S–A / 40Hz | Ankle | H | 22 | 3.27% EER |
| [16-17] | M–A / 120Hz  M–A / 45Hz | Hip | ML (HMM)  ML (SVM) | 48 (30M 18F) | 6.15% EER, 5.9% FAR, 6.3% FRR |
| [8]  [18] | S–A / 100Hz  M–A / 25 Hz | Ankle  T Pocket | TM (Euclidean)  ML (SVM) | 10  25 | 20% EER  100% RR |
| [9] | S–A / 100Hz | Hip | TM (PCA) | 60(43M 17F) | 1.6% EER |
| [19] | M–A / 45Hz | Hip | TM (DTW) | 51 (41M 10F) | 20% EER |
| [20] | M–A / 37Hz | Hip | ML ( SVM) | 6 | 90.3 ± 3.2% RR |
| [10] | S–A / 16Hz,  100Hz | Ankle Pocket Arm Hip | H (Euclidean)  H (Manhattan) | 21 (12M 9F)  100 (70M 30F) 50(33M 17F) 30 (23M 7 F) | 5% EER  7% EER  10% EER  13% EER |
| [11] | S–A / 100Hz | Body | TM(NN) | 30 | 96.7% RR |
| [12] [13] | S–A / 256Hz | Waist | TM(cross-corr.), H (FFT, histogram) | 36 (19M 17F) | 6.4 %, 10%, 19% EER |

In early stages, most of works used standalone sensors (SSs) have been implemented with a variety of success rate, they still have some restrictions. For example, SSs is relatively expensive and the interface of some special sensors needs to be developed separately. Thus, there is an increasing need to develop an easy-to-operate gait monitoring system within pervasive and ubiquitous environment. Recently, the developing of micro electromechanical (MEMs) technology helped such sensors to be miniaturized and integrated inside mobile devices (known as mobile sensors - MS). Gait authentication has been initially experimented on MS during recent years. In 2009, S. Sprager et al. used built-in accelerometer in Nokia cellphone positioned at the hip to collect and analyze gait signal [20]. Feature vectors for classification were built based on collected data using dimension reduction on cumulants by Principal Com-ponent Analysis (PCA). The classification in this module was accomplished by Support Vector Machines (SVM). They achieved about 90.3% accuracy. However, the number of experimental participants is rather small (6 persons). In comparison to SSs, MSs are designed to be cheaper, simpler and as a result the quality is not guaranteed as SSs. For example, the sampling rate is low and unstable (<50Hz vs.>100Hz), the noise is rather high. Derawi et al. [19] pointed up that impact by redid Holien’s work [21] using MS instead of SS and achieved EER of 20.1% compared to 12.9%. Table 5.1 summarized gait authentication approaches and their performances with various evaluation metrics such as Equal Error Rate – EER, Recognition Rate – RR, etc. on both SS and MS.

# 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

**Published**

**Twitter Mining: The Case of 2014 Indonesian Legislative Elections**

***Authors****:* ***Rischan 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*

**Awareness Home Automation System Based on User Behavior through Mobile Sensing**

***Authors****:* ***Rischan 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.*

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

***Authors****:* ***Rischan Mafrur****, I Gde Darma Nugraha, Deokjai Choi*

*International Conference on Ubiquitous Information Management and Communication (IMCOM/ICUIMC 2015), January 8-10, 2015, Bali, Indonesia.*

**Revision Being Processed**

**Developing and Evaluating Mobile Sensing for Smart Home Control**

***Authors****:* ***Rischan******Mafrur****, Priagung Khusumanegara, Gi Hyun Bang, Do Kyeong Lee, I Gde Dharma Nugraha, Deokjai Choi*

*International Journal of Smart Home (IJSH)*

**Under Review**

**Twitter Campaign in 2014 Indonesian Elections**

***Authors****:* ***Rischan Mafrur****; Priagung Kusumanegara; Deokjai Choi*

*Special Issue on Online Social Networks, Computer Communications, Elsevier*

# References

|  |  |
| --- | --- |
| [1] | 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. |
| [2] | 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. |
| [3] | V. K. Singh, L. Freeman, B. Lepri and A. Pentland, "Predicting Spending Behavior using Socio-Mobile Features," in *BioMedCom 2013*, 2013. |
| [4] | 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*. |
| [5] | 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. |
| [6] | 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. |
| [7] | 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. |
| [8] | 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. |
| [9] | A. Bogomolov, B. Lepri and F. Pianesi, "Happiness Recognition from Mobile Phone Data," in *BioMedCom 2013*, 2013. |
| [10] | 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. |
| [11] | R. LiKamWa, Y. Liu, N. D. Lane and L. Zhong, "Can Your Smartphone Infer Your Mood?," in *PhoneSense workshop*, 2011. |
| [12] | 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. |
| [13] | 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. |
| [14] | 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. |
| [15] | 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. |

**패턴 인식 및 생체인식 암호화 시스템을 이용한 모바일 폰에서의 보행 인증**

MAFRUR, Rischan

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 향상 시킨 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 분류 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 38명의 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 4초 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 FRR를 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 also would like to thanks also go to my thesis committee, including Professor Soohyung Kim, and Professor Kyungbaek Kim from whom I got much knowledge during studying their classes and in thesis review process. I also would like to thank 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 idea and supported me throughout my master course.

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 SAW who is the inspiration of my life, my dear parents, my beloved Indonesian community 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

1. http://www.funf.org/ [↑](#footnote-ref-1)
2. https://code.google.com/p/funf-open-sensing-framework/ [↑](#footnote-ref-2)
3. http://sqlitebrowser.org/ [↑](#footnote-ref-3)
4. https://github.com/rischanlab/Rfunf [↑](#footnote-ref-4)
5. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm [↑](#footnote-ref-5)