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

April 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:

HWANG, Bu-Hyun   _____

KIM, Kyung Baek   _____

CHOI, Deok Jai   _____

April 2015

Table of Contents

LIST OF FIGURES

LIST OF TABLES

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. Our research purposes are to discover human behavior based on their smartphone life log data. Then we want to build behavior model which can be used for human identification. In this research, we have collected user personal data from 47 students during 2 months which consist of 19 kind of data sensors. There is still no ideal platform that can collecting user personal data continuously and without data loss. The data which collected from user's smartphone are heterogeneous because the data came from multiple sensors and multiple source information and sometimes one or more data does not available. We have developed a new approach to build human behavior model which can deal with those situations. 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). An example case, in the context of daily behavior: Bob 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 is living in dormitory, he walks from dormitory to his lab (campus) takes 10 minutes. Usually, he arrives 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 Bob's smartphone sensor data to define and build Bob's behavior model.

In terms of user personal data collection, 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 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

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 Funf[1] and

also Funf developer site[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

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/

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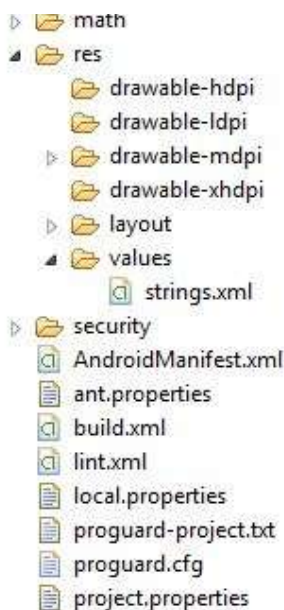
Figure 2-3. Strings.xml file in project directory

```
{
  "@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 site[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

---

[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 (℃).	
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,"mElapsed
RealtimeNanos":21989372000000,"mExtras":{"networkLocationSou
rce":"cached","networkLocationType":"cell","noGPSLocation":{
"mAccuracy":1625.0,"mAltitude":0.0,"mBearing":0.0,"mElapsedR
ealtimeNanos":21989372000000,"mHasAccuracy":true,"mHasAltitu
de":false,"mHasBearing":false,"mHasSpeed":false,"mIsFromMock
Provider":false,"mLatitude":35.1837595,"mLongitude":126.9052
379,"mProvider":"network","mSpeed":0.0,"mTime":1403484137091
},"travelState":"stationary"},"mHasAccuracy":true,"mHasAltit
ude":false,"mHasBearing":false,"mHasSpeed":false,"mIsFromMoc
kProvider":false,"mLatitude":35.1837595,"mLongitude":126.905
2379,"mProvider":"network","mSpeed":0.0,"mTime":140348413709
1,"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 site[4], in *"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 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 data.

---

[4] https://github.com/rischanlab/Rfunf

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 shows the list of probes that used by our application to collect users personal data. Not all of those data that we collected 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 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 used are 9 probes.

The total of students who participated are 47 students. From those data not all data are full available. Some 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 around 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 once 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. When we load all of those data in the same time it will spend computer resource especially RAM. To process data, R environment system 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 (temporary file), in this case, we use csv files.
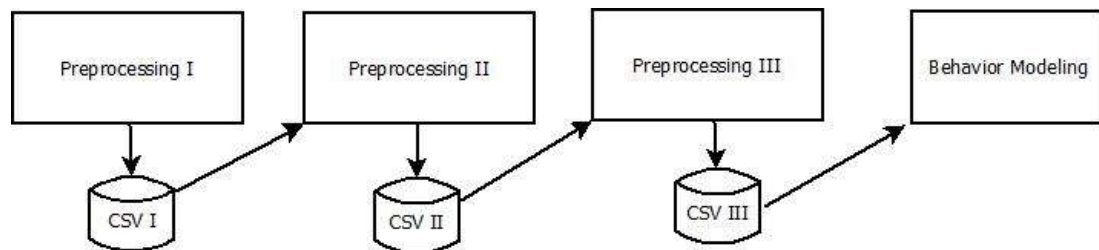

Figure 2-5. Data preprocessing flows

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 Bob's daily activities from he wakes up until arrives to his lab room in working day. We call that Bob's activities are Bob's behavior because that activities carried out continuously by Bob in his working day.

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

1. What kind of human activity (e.g. meeting, studying, exercising, and etc).

2. When the activity happened (e.g. around 9 AM).

3. Where the location is, 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.
    - o 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.
    - o We only have activity status (none, low, and high), these status based on accelerometer sensor activity.
    - o 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.
    - o 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.
    - o Every values in our dataset has timestamp value. The timestamp value following UNIX timestamp, we have to transform to human timestamp.
    - o Date and time are used as features in this research.
- Where the location is.

- ○ 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.
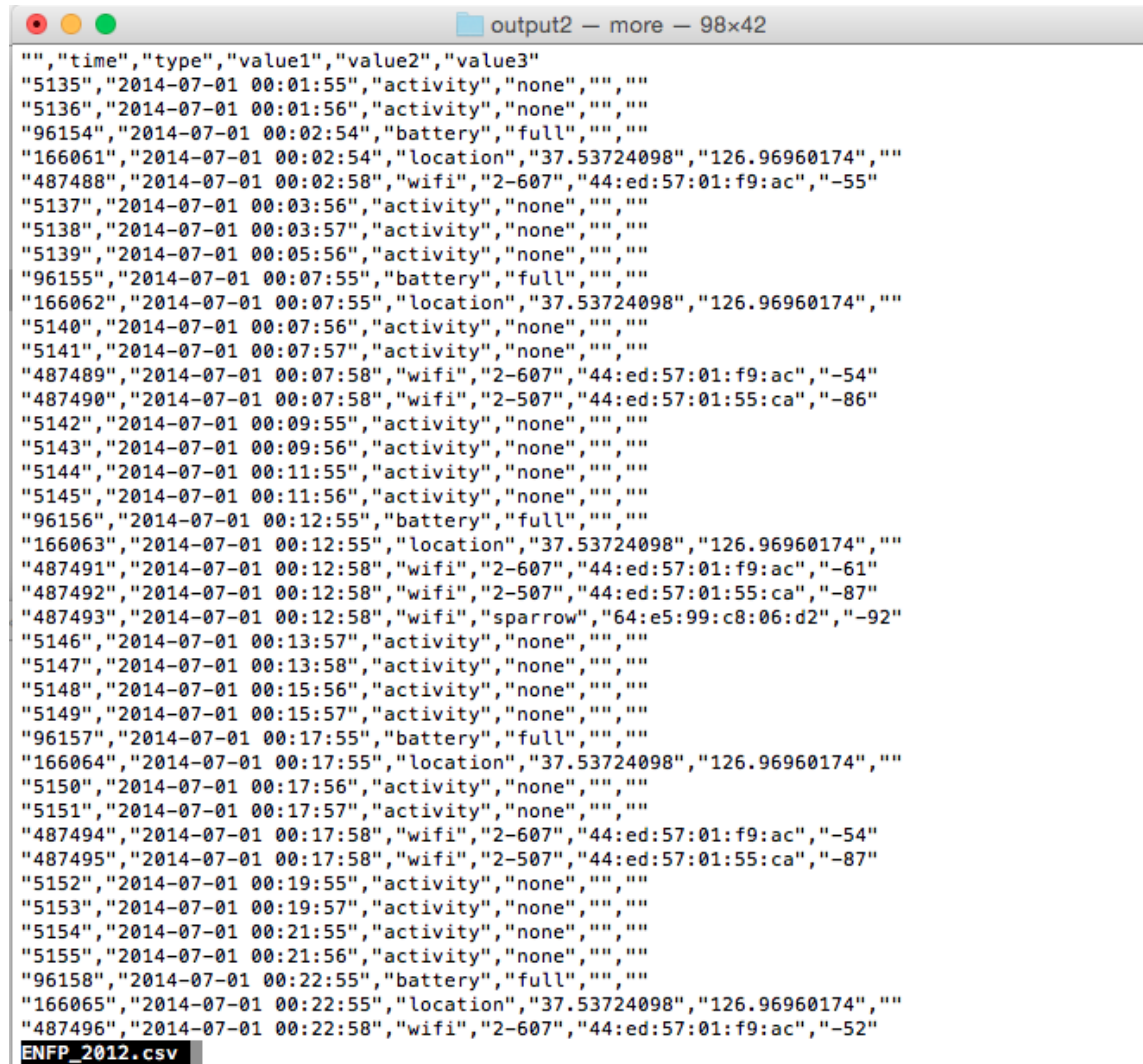
4. Bluetooth probe only has single value, *value1* is the name of nearby Bluetooth devices.

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 ("*none*","*low*", and "*high*")		
2.	SimpleLocationProbe	Latitude	Longitude	
3.	WifiProbe	List of nearby SSID	MAC	Signal strength (dB)
4.	BluetoothProbe	List of nearby Bluetooth devices		
5.	BatteryProbe	Status ("*discharging*","*full*", and "*charging*")		
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.

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
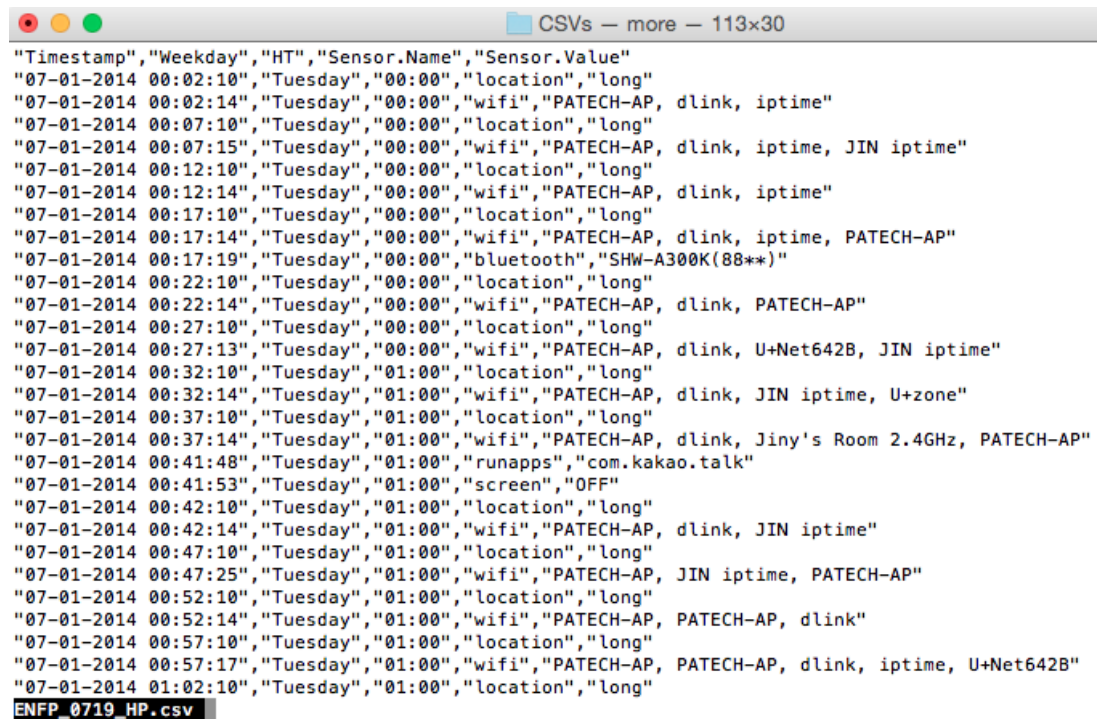
Figure 2.5 shows the result of features extraction from Pre-processing II module. We still have some problems on that result. We create Pre-processing III module to make our dataset fit enough before applying behavior modeling module. Another reason is more features mean more time to processes, light 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.

6. Aggregate the values of Call Log and SMS log. In this preprocessing, we use only two of values from call log and SMS log and we combine to one columns. The values of call log and SMS log that used are "type and number". An example of value of call log *"incoming 1bae527e84708183049d8e892a1c959a492ee6a9"*. Even the number was hashed but if the number is same, it has same hash value so we still have pattern information.

7. Removing values such as text length and duration from SMS log and call log,

duration from running application probe, MAC and signal strength from nearby

Wi-Fi probe. The reason why we did not use these features because our purpose

is to find the similarity of data pattern, the value of call duration, application

usage duration will make the data quite different. Probably, we will use those data

in different approach but not in this approach.

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.

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

3. HUMAN BEHAVIORS MODELING

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

Figure 3-1 shows the data visualization example in the same day for four days from two 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 sure that the data features in user personal data log can be used for many purposes such as user identification and classification, recommendation, and etc. In this section, we explain about

our research background and the problem statements, 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 focus to use accelerometer sensor for human gait identification [11], accelerometer sensor for basic activity recognition [12], and magnetic field sensor for location identification [13] and etc. Those approaches which are using one feature is good to know that feature is reliable or not. When we use only one feature, the problems are the lack of sensor accuracy and data loss. We have to realize that the data from user's smartphone are uncertainly data. Not all data are in good condition, sometime probably the sensor has problems 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. When users use their phone in real environment, they will use like in their natural life. We have to consider about realistic data, means the data which is the data based on real data in real environment. In this research, we define what the realistic data is, the explanation can be seen below:

1. In realistic environment, user has different types and brand of smartphone and each smartphone has different types of sensors and hardware specification and capabilities.

2. We could not expect the human actions and their activities, they will do 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 resources.

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

3.2 Proposed Methods

In this section, we explain about our proposed methods. First is about overall architecture of our system, the algorithm that we use to find similar patterns and also method that we use to create user behavior model.

3.2.1 Overall architecture

Figure 3-2. Finding similar pattern in different days same week (the window size is 2 days)

We have dataset around one months and 20 days (7 weeks). We use one month dataset to build user behavior model and then use the remaining data to testing our approach performance. Figure 3.2 sketches our proposed method to find similar pattern in all of our dataset, the explanation of that figure can be seen below:

1. First, we define the window size. In this research, the window size that we use is two, means two days.

2. We remove the last day of weekday (Sunday) because when the window size is two and the first day start from Monday, so the days in one windows are "Monday-Tuesday", "Wednesday-Thursday", "Friday-Saturday" the remaining is "Sunday", so we remove it.

3. We applied Algorithm 3.2 to find similar pattern between days inside the window.

Algorithm 3.2.

```
Data : D, w
Result : All Detected Group in a Window
grpAll, grpTemp, grpPrevious<- NULL
dataValue, dataValueNext <- NULL
while (D in w) for all of D do
     dataValue <- D.current.day
     dataValueNext <- D.next.day
     grpTemp <- findingSimilarPatterns(dataValue, dataValueNext)
     if (grpPrevious != NULL) then
       grpPrevious <- getSimilarBetweenGroups(grpPrevious, grpTemp)
       grpAll <- merge(grpPrevious)
     else
       grpAll <- add(grpPrevious)
return groupAll
```

3.2.2 Similar Patterns Detection

Group-1 = 13:00,location,same | 13:00,wifi,1-AP,iptime
Group-1 = 13:00,location,same | 13:00,wifi,1-AP,iptime
Group-2 = 15:00,runapps,kakao | 15:00, location, little
Group-2 = 15:00,runapps,kakao | 15:00, location, little

Figure 3-3. Find similar patterns algorithm overview

Figure 3-3 illustrates the algorithm that used to find similar data patterns. On that figure, we have two of days in one window. First data is the data of first day and the second data is the data of second day and both of data have six rows. We want to find the similar data between first data and second data. Based on an example in that figure, we have two groups of data which similar. First group in green rectangle and the second group in purple rectangle. To know the similarity between data in rows, we use simple strings matching method. The output of strings matching method is *true* when the strings is same/match or *false* when the strings not match. We have used *Levenshtein* distance also to measure the similarity score between two strings in rows to anticipate the data which not match but actually similar. We have mentioned that we applied aggregate function among strings in our dataset. We can imagine, when we use string matching, strings "D-Link AP" and "D-Link AP " is not match because the second string has *"space"* in the end of word. By using *Levenshtein,* we can handle these problem. Mathematically, the *Levenshtein* distance between two strings a, b is given by $lev_{a,b}(|a|, |b|)$ where

$$lev_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} lev_{a,b}(i - 1, j) + 1 \\ lev_{a,b}(i, j - 1) + 1 \\ lev_{a,b}(i - 1, j - 1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Finally, after we get the similar data patterns from the data which is in one window, we store those data to *"current data"* variable. Then the system will check whether the *"current data"* currently exists in the *"previous data"* or not. If yes the system will merge the *"current data"* (groups) with *"previous data"*, if not the system will identified the *"current data"* is new group data, more details see Algorithm 3.2.

4. EXPERIMENTAL RESULTS

4.1 Result and Discussion

In this section, we explain about our research result and analysis. The goal of our research are to discover human behavior from the user smartphone life log data and based on those behavior data we want to build behavior model which can be used for user identification. This section consist of two of subsections are behavior identification and performance evaluation.

4.1.1 Behavior Identification

```
G2,"19:00,location,same|19:00,bluetooth,DTVBluetooth|"
G2,"19:00,location,same|19:00,bluetooth,DTVBluetooth|"
G2,"19:00,location,same|19:00,bluetooth,DTVBluetooth|"
G2,"19:00,location,same|19:00,bluetooth,DTVBluetooth|"
G2,"19:00,location,same|19:00,bluetooth,DTVBluetooth|"
G3,"20:00,location,same|20:00,bluetooth,DTVBluetooth|"
G3,"20:00,location,same|20:00,bluetooth,DTVBluetooth|"
G3,"20:00,location,same|20:00,bluetooth,DTVBluetooth|"
G3,"20:00,location,same|20:00,bluetooth,DTVBluetooth|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G7,"06:00,screen,OFF|06:00,location,same|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
G9,"12:00,battery,charging|12:00,bluetooth,ESVH-PC|12:00,location,same|12:00,runapps,com.lge.launcher2|"
INTJ_8928_groups.csv
```

Figure 4-1. An example of output from our system (grouping result)

In previous section, we have explained about our system, how we find the similar pattern between days inside the window. Figure 4-1 is the one of example the output from our system. From those data we build behavior model. The details about our experiment as follows:

1. The average of number of days from our dataset around 1 month 20 days not fully two months. So based on those dataset, we divide all of dataset to two parts

a. First month for creating model (first dataset)

b. Remaining dataset for testing performance (second dataset)

2. Modeling user behavior based on first dataset (first month dataset). We applied our approach to our first dataset and build human behavior model/profile. We call that profile is B1 data.

3. Extract and process the second dataset.

a. Applying similarity detection to second dataset with same setting as that used in building behavior model.

b. We called the result from this process is B2 data.

4. Is the all of new behavior (B2) identified by behavior model (B1)?.

a. How many groups of activities (B2) which identified by behavior model (B1)?

b. Calculate the percentage of groups of activities (behavior) which identified.

5. Applying to all students data and observing the result.

Table 4-1. The result of user identification

		TEST					
		ENFP_0719	ENFP_2012	INTJ_5498	ISTJ_3052	ESFJ_2301	ESFP_4634
MODEL	ENFP_0719	**67.922**	0	0.4	2.187	0	1.943
	ENFP_2012	0	**83.582**	0	0	0	0
	INTJ_5498	2.178	0	**75.977**	2.087	0	3.401
	ISTJ_3052	2.289	0	0.4	**93.439**	8.232	1.943
	ESFJ_2301	0	0	0	0.099	**22.866**	0
	ESFP_4634	2.289	0	0.977	2.087	0	**89.686**

Table 4-1 shows the result of user identification. We applied to all student's data which are 37 students but that table only shows the data from 6 students. The full of data from 37 students can be seen in Appendix (full table of result user identification). Table 4-1 does not confusion matrix table, it just looks like confusion matrix table. The value means the percentage of B2 (behavior data from test dataset) which successfully identified by B1 (behavior model). We can see that our proposed features and our approach can be used for identification. Based on the result and our observation, our approach can achieved good enough accuracy even some of users has bad accuracy (under 30%).
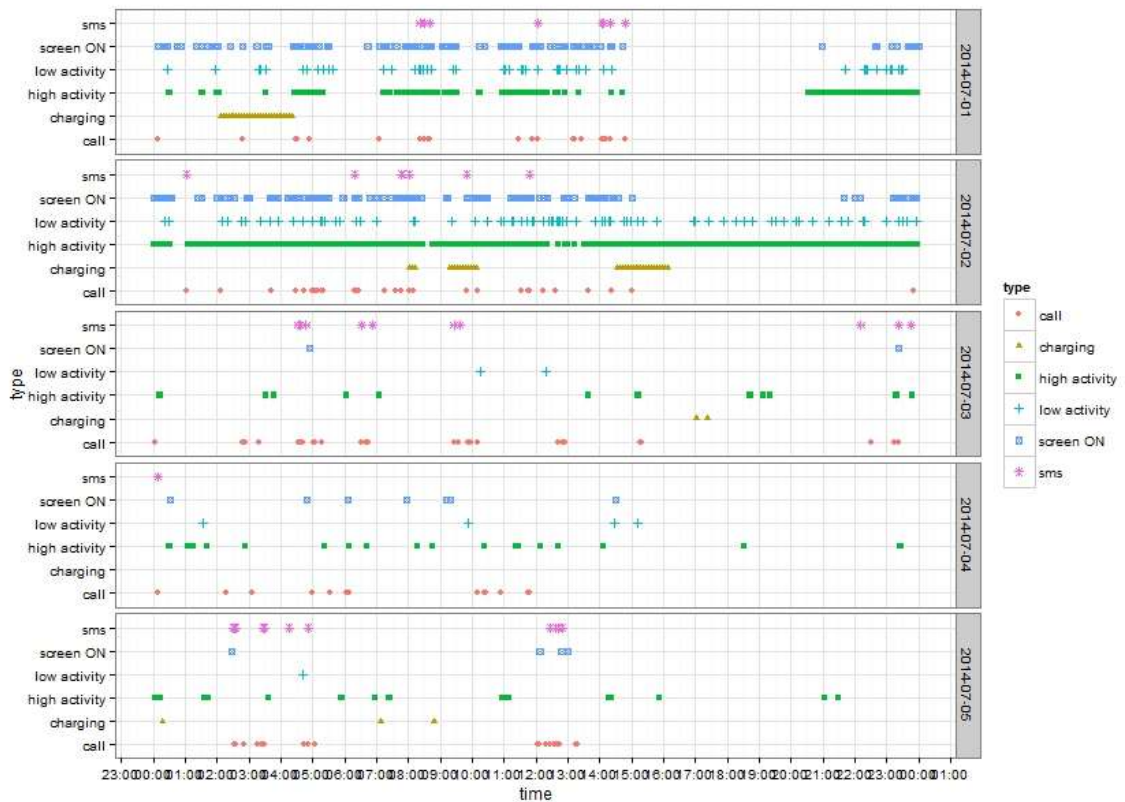
Figure 4-2. An example plot of data from student who has bad accuracy

We tried to looking the answer, why some of users have bad accuracy. After we observe and

investigate, we got the answer. The reason why some of students have bad accuracy is because of theirs dataset. An example is data from *"ESTJ_5190"* and *"ISFJ_2711"*, the dataset from those users after preprocessing III and splitting to two datasets (model and test), the size of those are 64 KB and 40 KB for the model dataset. The number of rows are less than 500 rows, whereas another data from students who has good accuracy, those data have number of rows around more than 50,000 rows. It means the problem is theirs dataset were not enough for creating their behavior model. We also tried to plot the activities from one user who has bad accuracy. Figure 4-2 shows the result of our plotting. We can compare this figure to Figure 3-1. The users who have bad accuracy, besides in some days they have few activities, they also have different behavior almost in every day which our approach could not handle it.

Despite some of users have bad accuracy (under 30 %) means only around 30% behavior data in test dataset which identified in behavior model, but the value is the highest one than other values. We can see from student who has ID "ESFJ_2301" only 22.866 % B2 which are identified by B1 (model), but this value is the highest than another values in the horizontal (same row) and vertical (same column), see appendix for full result. It means our approach still can be used for identification.

In chapter three, we have mentioned that we also use *Levenshtein* distance to measure the similarity score between two strings in rows. The reason why we used *Levenshtein* is to anticipate the data which not match but actually similar. Finally, we only use string matching method to find similarity data patterns. We did not use *Levenshtein* distance because whether use it or not, it does not affected the accuracy but only increasing time processing.

4.1.2 Testing Performance by Removing Some of Features
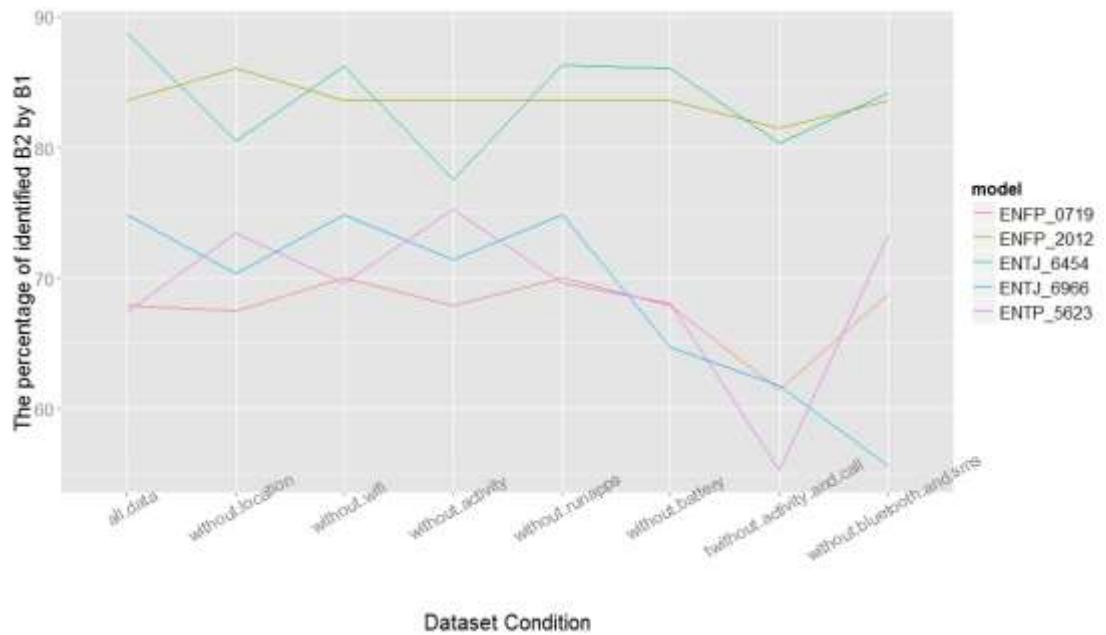
Figure 4-3. The percentage of identified B2 by B1 in different dataset condition

In our research background and problem statements, we have explained about the realistic data. We want our approach can dealing with realistic data. When we doing research in this field and want to collect personal user data, we cannot said that all of users have same smartphone brand which have same sensors. We have to realize that some sensors probably does not supported by users smartphone or probably user does not have any data in one of sensor such as user does not have SMS and call log. We have to consider about that, if we focus only one sensor, it will be problem. Based on the Table 4-1 (see appendix for full table), we see that our approach good enough for user identification but is it still good enough if we remove some features/sensors data?.

To answer that question, we tried to remove one and more features from our dataset and then we compare the result with previous result which is using all features. The cases that we tried are:

- Without GPS sensor data

- Without Wi-Fi sensor data

- Without Activity data

- Without Current running applications data

- Without Battery sensor data

- Without Activity data and Call log data

- Without Bluetooth sensors data and SMS log data

The result of our cases implementation can be seen on Figure 4-3. That figure only shows data from five students, all of data can be seen in appendix. When we see and observe the result, we can conclude that by removing one or two features our approach still good enough for user identification. It means by using our approach, we can handle the realistic data which sometimes the data from one or more sensor does not available.

5. LIMITATION AND FUTURE WORK

In this research, we realize that we have many of limitations. This section explain about our limitations that will be consider as the future work. The lists of our research limitations as follows:

1. Changing the size of window. Our approach is using similarity detection between days in each window size. In this research, we used two days as the size of window. Actually we can increase the window to three, four, or five, or probably we use six days means one week as our window size. We can use different window and then observe the accuracy, whether the size of window will

influence accuracy or not. Due to time limitation, we decide to using two days for the size of window. The reason, why we used two days as the size of window is because two is the minimum numbers when we want to compare two of data.

2. Using different time precision. In our approach, when we change the time machine to human machine, we used one hour time precision. Actually, we can use different time precision such as 15 minutes, 30 minutes, and one hour and compare the accuracy.

3. Comparing days in vertical method, means same day but different week. When we compare the days in one window, we are comparing the days between current day and the next day. We call this approach is horizontal approach. The next research, we can use different approach such as comparing same day but in different week, we called it vertical approach.

6. RELATED WORKS

User personal data log from smartphone can be used for many purposes such as user identification, user classification, recommendation system, mood detection and etc. In this section, we explain about previous works which related with exploring user personal data log that collected from user's smartphone. Smartphone log consist of many of data such as contact, call log, SMS log, GPS, Wi-Fi, Bluetooth, and many more as that we have explained in previous chapters. We can choose which data or information features that we want to explore. For example is contact data, from this data we can explore many thing. [11] they collected the contact lists and then analyzed using several features such as communication intensity, regularity, medium, and temporal tendency. By using machine learning techniques and their methods they can achieved up to 90 % accuracy to classify life facets/type of relation in contact (family, work, social). Another interesting research which based on smartphone contact 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.

Another previous works which used call log 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. The main purpose of this research is for business area, they said that the smartphone log could be used for predicting customer type such as loyal customers or overspend. They found that using their approach social features could be better predictors of spending behavior of a couple than personality variables. Previous work which based on location features, an example done by [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. Still based on location data, research which done by [14], they utilize location information which can obtained from phone sensors (GPS, Wi-Fi, GSM, and 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.

Not only those features that can be exploited, another example are, based on list of application installed in android devices which done by [15]. This paper, the authors 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 achieved over 90 % of precision.

From those that we mentioned above no one which care about user privacy. In this related works, we also found previous research which consider about user privacy, research by [16] . They proposed a different approach that use multimodal mobile sensor and log data to build framework called *mFingerprint*. *mFingerprint* is user modeling framework which can uniquely depict user. The thing 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. Our application also does not expose the sensitive information such as name and phone number in smartphone contact, SMS log, and call log, and etc.

We also can use user personal data for unique purposes such as to know the user happiness, mood and stress. Smartphone log can be used for happiness identification done by [6], stress identification done by [17], user mood identification done by [7], or we also can develop an application which can help human to do their daily routines [18]. [6] This paper provides the evidence that we can predict the happiness of human based on their phone log. In this paper, the authors 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 to classify 3-class daily happiness (happy, neutral, and unhappy). [17] 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). Their approach based on Random Forest and Gradient Boosted Machine algorithms. 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 proof that by using phone log, we can predict the user mood. The authors develop smartphone service called *MoodSense*. They observe 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.

Smartphone data log also can be used for personality classification. [19] They develop conceptual model that explains about relationship between user Big Five personality (Extraversion, Agreeableness, Conscientiousness, Emotional Stability and Openness to Experience) and their satisfaction with basic mobile phone services such as call, message, and 3G services. The main propose of this paper is several implications for design of mobile phone services. Another research done by [20]. 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. The last example proves that smartphone log can be used for personality classification done by [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.

7. CONCLUSIONS

In this thesis, we proposed approach that can used for user identification by building human behavior model. We use and combine of many sensors instead only focus on one sensors because we realize that sometimes user does not has data from one or more sensors. Based on our result, we can see that our approach is good enough for user identification. We have tried also to remove one or more features and then observe the accuracy values. The result shows that even one or more features have been removed but our system still can be used for identification. It means our system can handle the problem if one or more data sensors

from users smartphone not available. Some of result from our system can achieve up to more than 80 % accuracy but any four of them have less than 30 % accuracy. In this thesis, we have explained also why four students have bad accuracy. The reason is students who have bad accuracy, they have different behavior for almost each day which our approach does not capable to handle it. Despite some of accuracy values are under 30 % but those values still can be used for identification because those values are the highest one compared to others. It means that our approach still good enough for identification system.

Bibliography

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), volume 9, No 3, March 2015

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 (ACM IMCOM 2015), January 8-10, 2015, Bali, Indonesia.

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.*

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

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] T. Hoang, V. Vo, T. Nguyen and C. Deokjai, "Gait Identification Using Accelerometer on Mobile Phone," in *ICCAIS*, 2012.

[12] M. Ayu, T. Mantoro, A. Fariadi and S. Basamh, "Recognizing user activity based on accelerometer data from a mobile phone," in *2011 IEEE Symposium on Computers & Informatics (ISCI)*, Kuala Lumpur, 2011.

[13] C. Galvan-Tejada, J. Carrasco-Jimenez and R. Branea, "Location Identification Using a Magnetic-field-based FFT Signature," in *The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)*, 2013.

[14] 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.

[15] 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.

[16] 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*.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

Life-Log 정보를 감지하는 스마트폰으로부터의 인간 행동 모델링 및 발견

MAFRUR, Rischan

전남대학교 대학원 전자컴퓨터공학과

(지도교수: 최덕재)

(국문초록)

오늘날 개인의 정보는 새로운 경제적 자산이 되고 있다. 우리의 스마트폰에서 생서된 개인 정보는 추천서, 신분증 등 같이 많은 목적으로 사용될 수 있다. 우리의 연구 목적은 스마트폰 생활 log 정보에 근거한 사람의 행동 발견에 있다. 그래서 우리는 개인 인증서로 사용될 수 있는 행동 모델을 구축하길 원한다. 이 연구에서, 우리는 2 달동안 19 종류의 데이터 센서로 구성된 사용자 개인 데이터를 수집했다. 이것은 여전히 손실 없고 지속적인 사용자 개인 정보를 수집할 수 있는 이상적인 플랫폼이 아니다. 사용자의 스마트폰으로부터 수집된 정보들은 다수의 센서와 소스 정보로부터 얻어진 것이며 때때로 한개 혹은 그 이상의 종보가 사용 불가능 했기 때문에 이질적이다. 우리는 그러한 상황을 처리할 수 있는 행동 모델을 구축하기 위한 새로운 접근 방식을 개발했다. 게다가 우리는 이 논문에서 우리의 접근 방식을 평가하고 세부사항을 제시한다.

ACKNOWLEDGEMENT

APPENDIX

Result of User Identification

model	t.ENFP_0719	t.ENFP_2012	t.ENTJ_6454	t.ENTJ_6966	t.ENTP_5623	t.ESTJ_5190
ENFP_0719	67.922	0	0	0	0	0
ENFP_2012	0	83.582	0	0	0	0
ENTJ_6454	0	0	88.746	0.441	0.544	1.524
ENTJ_6966	0	0	0.712	74.846	2.069	13.11
ENTP_5623	0	0	0.712	1.853	67.392	12.805
ESTJ_5190	0	0	12.393	1.986	4.954	22.866
ESFJ_9284	2.436	0	0	0	0	1.22
ISFJ_2711	0.812	0	0	0	0	0
ESFP_4634	2.289	0	0	0	0	0
ESTJ_3022	0	0	0.712	2.913	3.593	17.378
ESTJ_5071	0	0	0	0	0	0
ESTJ_5824	0	0	0.855	1.942	4.083	12.805
ESTJ_6510	0	0	1.425	3.486	2.722	14.939
ESTP_4301	0	0	1.567	3.928	4.627	17.378
ESTP_5154	3.285	0	0	1.059	0	7.317
INFP_1993	0.221	0	0	0	0	0
INTJ_5498	2.178	0	0	0	0	0
INTJ_7906	0	0	0.712	1.942	2.722	7.622
INTP_3739	0.554	0	0.712	2.295	1.742	4.878
INTP_6399	0	0	0	0.397	1.198	3.354
INTP_9712	0	0	0.712	3.001	4.083	20.122
ISFJ_2057	0	0	8.262	3.133	4.79	20.122
ISFJ_7328	2.178	0	0	0.53	0.435	1.22
ISFP_4282	0	0	3.419	2.56	3.702	7.927
ISTJ_0178	0	0	5.983	2.207	3.266	9.146
ISTJ_2068	0.812	0	0	0	0	0
ISTJ_2837	0	0	0.712	3.928	2.94	14.939
ISTJ_3052	2.289	0	0	1.456	1.034	8.232
ISTJ_4667	0	0	0	0	0	0
ISTJ_4700	0	0	0.712	2.692	4.083	16.768
ISTJ_4753	0	0	0	2.692	0.435	8.537
ISTJ_4968	0	0	3.276	3.398	3.321	15.549
ISTJ_9139	0	0	0	0	0	0
ISTJ_9576	0	0	0.855	3.839	4.192	20.122
ISTP_3948	0.443	0	0.712	1.633	2.069	5.488
ISTP_7676	0.148	0	1.425	3.619	1.306	10.976
XXXX_XXXX	0	0	0.712	2.692	1.96	12.805

model	t.ESFJ_9284	t.ISFJ_2711	t.ESFP_4634	t.ESTJ_3022	t.ESTJ_5071	t.ESTJ_5824	t.ESTJ_6510
ENFP_0719	3.191	1.779	1.943	0	0.68	0	0
ENFP_2012	0	0	0	0	0	0	0
ENTJ_6454	0	0	0	1.381	1.701	0.75	0.811
ENTJ_6966	0	0	0	9.116	7.483	1.237	3.581
ENTP_5623	0	0	0	9.669	15.646	2.774	1.689
ESTJ_5190	0	0	0	9.116	8.503	2.699	1.689
ESFJ_9284	83.247	1.449	2.653	0	5.782	0	0
ISFJ_2711	0.469	12.912	0.897	0	0	0	0
ESFP_4634	4.27	0.988	89.686	0	0.68	0	0
ESTJ_3022	0	0	0	66.851	9.184	2.099	2.838
ESTJ_5071	1.642	0	0	0	31.293	0	0
ESTJ_5824	0	0	0	9.669	9.524	77.586	1.689
ESTJ_6510	0	0	0	14.088	11.224	1.724	86.892
ESTP_4301	0	0	0	16.851	9.524	2.586	7.365
ESTP_5154	4.317	0.329	2.018	3.315	0.68	0	3.649
INFP_1993	0.422	0.395	0.112	0	0	0	0
INTJ_5498	3.379	0.593	3.401	0	0.68	0	0
INTJ_7906	0.282	0.725	0	7.459	9.524	1.837	1.689
INTP_3739	4.786	0.988	0.486	4.42	7.823	0.75	0.541
INTP_6399	0	0	0	3.039	1.02	0.45	0.203
INTP_9712	0	0	0	12.983	9.524	2.811	5.338
ISFJ_2057	0.282	0.593	0	12.983	9.184	2.774	5.811
ISFJ_7328	2.816	0.593	2.28	1.105	2.041	0.225	0.27
ISFP_4282	0	0	0.71	7.735	7.143	1.687	4.797
ISTJ_0178	0	0	0	8.287	7.483	1.874	1.486
ISTJ_2068	3.05	0.988	0.486	0	3.401	0	0
ISTJ_2837	0.047	0.329	0	14.917	9.524	1.837	6.351
ISTJ_3052	2.816	0.593	1.943	4.144	1.701	0.15	3.851
ISTJ_4667	0	0	0	0	3.401	0	0
ISTJ_4700	0.563	0	0	9.669	9.184	2.774	2.297
ISTJ_4753	0	0	0	4.972	2.041	0.3	1.419
ISTJ_4968	0	0	0	12.155	3.401	2.249	6.081
ISTJ_9139	0	0	0	0	0	0	0
ISTJ_9576	0	0	0	12.983	9.524	3.261	2.838
ISTP_3948	2.252	0.593	0.374	5.525	9.864	1.387	1.757
ISTP_7676	0.375	0	0	11.05	7.143	0.787	5.946
XXXX_XXXX	0	0	0	8.84	7.143	1.387	2.365

model	t.ESTP_4301	t.ESTP_5154	t.INFP_1993	t.INTJ_5498	t.INTJ_7906	t.INTP_3739	t.INTP_6399
ENFP_0719	0	2.011	0	0.4	0	0	0
ENFP_2012	0	0	0	0	0	0	0
ENTJ_6454	3.401	0.776	0	0	0.566	0.136	0
ENTJ_6966	7.297	1.954	0	0	6.674	0.734	0
ENTP_5623	7.669	1.207	0	0	6.561	1.44	0
ESFJ_2301	7.05	0.776	0	0	4.751	1.033	0
ESFJ_9284	0	1.437	0	0.666	0.679	0.38	0
ESFP_0912	0	0.172	0	0.311	0	0	0
ESFP_4634	0	0.316	0	0.977	0	0	0
ESTJ_3022	5.69	1.063	0	0	7.353	0.842	0
ESTJ_5071	0	0	0	0	0	0.408	0
ESTJ_5824	7.174	0.747	0	0	5.204	1.114	0
ESTJ_6510	12.492	2.615	0	0	8.937	2.011	0
ESTP_4301	30.736	5.23	0	0	9.389	1.467	1.822
ESTP_5154	2.474	55.144	0	0.488	4.186	0.462	0
INFP_1993	1.793	1.121	75.966	0	0	0	0
INTJ_5498	0	0.891	0	75.977	0	0	0
INTJ_7906	5.257	0.402	0	0	14.367	0.652	0
INTP_3739	4.329	0.977	0	0.666	1.584	59.701	0
INTP_6399	3.463	0.086	0	0	1.018	1.087	69.636
INTP_9712	8.905	2.328	0	0	9.389	1.114	0
ISFJ_2057	11.07	3.793	0	0	10.068	1.44	0
ISFJ_7328	0.989	0.431	0	0.4	2.262	0.109	0
ISFP_4282	12.121	2.701	0	0	5.769	0.87	0
ISTJ_0178	5.751	0.833	0	0	3.733	1.495	0
ISTJ_2068	0	0.546	0	0.666	0	0	0
ISTJ_2837	7.978	2.931	0	0	10.181	0.652	0
ISTJ_3052	2.288	2.586	0	0.4	5.204	0.516	0
ISTJ_4667	0	0	0	0	0	0	0
ISTJ_4700	7.916	1.063	0	0	5.204	2.174	0
ISTJ_4753	2.783	1.58	0	0	3.507	0.163	0
ISTJ_4968	16.017	4.943	0	0	5.769	1.25	0
ISTJ_9139	0	0	0	0	0	0	0
ISTJ_9576	9.524	2.011	0	0	7.353	1.495	0
ISTP_3948	3.958	0.718	0	0.666	3.62	0.652	0
ISTP_7676	9.709	4.483	0	0	8.597	1.685	0
XXXX_XXXX	6.37	2.213	0	0	5.769	0.652	0

model	t.INTP_9712	t.ISFJ_2057	t.ISFJ_7328	t.ISFP_4282	t.ISTJ_0178	t.ISTJ_2068	t.ISTJ_2837
ENFP_0719	0	0	0.835	0	0	1.075	1.366
ENFP_2012	0	0	0	0.641	0	0	0
ENTJ_6454	1.802	2.74	0	0.087	0.436	0	0.131
ENTJ_6966	10.811	3.756	0.304	0.67	0.949	0	3.677
ENTP_5623	12.613	4.463	0.304	0.991	0.769	0	1.97
ESFJ_2301	9.73	5.038	0.683	0.932	1.974	0	1.602
ESFJ_9284	0	0.53	0.835	0	0	1.459	0.21
ESFP_0912	0	0	0	0	0	0.23	0.105
ESFP_4634	0	0	1.367	0.437	0	1.306	0.184
ESTJ_3022	13.153	4.728	0.304	0.524	0.769	0	3.178
ESTJ_5071	0	0	0	0	0	0	0.499
ESTJ_5824	10.631	4.463	0.304	0.554	0.769	0	1.996
ESTJ_6510	15.135	4.949	0.304	0.991	0.564	0	6.882
ESTP_4301	20.18	5.745	0.304	2.331	1.308	0	7.066
ESTP_5154	8.468	1.061	0.607	0.641	0	0.998	4.334
INFP_1993	0	0.884	0	0	0	0.307	0.079
INTJ_5498	0	0	0.835	0	0	1.075	0.105
INTJ_7906	7.568	4.198	1.367	0.087	0.564	0	1.996
INTP_3739	4.324	1.061	0.456	0.466	0.538	1.306	0.867
INTP_6399	4.685	0.398	0	0.466	0.436	0	1.051
INTP_9712	82.883	5.524	0.304	1.195	0.769	0	4.465
ISFJ_2057	21.081	89.395	0.304	1.369	0.949	0	5.726
ISFJ_7328	1.441	0.53	81.093	0	0.179	0.845	0.735
ISFP_4282	11.892	3.182	0.304	61.218	1.462	0	4.597
ISTJ_0178	7.027	3.182	0.304	1.632	61.128	0	1.602
ISTJ_2068	0	0	0.456	0	0	78.111	0.184
ISTJ_2837	16.036	6.673	0.304	1.311	1.103	0	62.044
ISTJ_3052	9.55	2.165	0.835	0.641	0	0.922	3.126
ISTJ_4667	0	0	0	0	0	0	0.499
ISTJ_4700	10.45	5.17	1.215	0.962	1.41	0	1.918
ISTJ_4753	6.486	1.679	0.304	0.67	0.641	0	2.758
ISTJ_4968	18.919	4.11	0.304	2.448	1.231	0	4.886
ISTJ_9139	0	0	0.228	0	0	0	0
ISTJ_9576	14.955	5.524	0.456	0.554	0.769	0	3.336
ISTP_3948	6.306	2.872	0.759	0.087	0.385	1.075	2.023
ISTP_7676	13.694	3.182	0.456	3.089	0.769	0.154	4.912
XXXX_XXXX	10.631	5.524	0.304	0.524	0.385	0	3.546

model	t.ISTJ_3052	t.ISTJ_4667	t.ISTJ_4700	t.ISTJ_4753	t.ISTJ_4968	t.ISTJ_9139	t.ISTJ_9576
ENFP_0719	2.187	0	0	3.066	0	0	0
ENFP_2012	0	0.949	0	0	0	0	0
ENTJ_6454	0	0	2.153	0	3.333	0	3.855
ENTJ_6966	0.099	0	4.144	2.3	11	0	11.338
ENTP_5623	0.497	0.949	7.535	0	8.5	0	8.844
ESFJ_2301	0.099	0	7.212	0	6	0	8.39
ESFJ_9284	2.087	0	0.215	3.237	0	0	0
ESFP_0912	0	0	0	0.256	1.833	0	0
ESFP_4634	2.087	0	0	3.066	0	0	0
ESTJ_3022	0.099	0	7.051	1.278	10	0	13.152
ESTJ_5071	0	1.002	0	0	0	0	0
ESTJ_5824	0.099	0	7.535	0	8	0	11.791
ESTJ_6510	0.099	0	6.566	1.278	20.667	0	10.884
ESTP_4301	0.099	0	7.966	2.3	29.833	0	17.687
ESTP_5154	1.789	0	0	3.748	7.833	0	6.916
INFP_1993	0.099	0	0	0.341	0	0	0
INTJ_5498	2.087	0	0	2.811	0	0	0
INTJ_7906	0.099	0	5.759	0	4.667	0	6.236
INTP_3739	0.696	0	3.552	1.022	6.667	0	6.349
INTP_6399	0.099	0	2.314	0	3.333	0	2.834
INTP_9712	0.099	0	7.535	1.278	18.333	0	14.399
ISFJ_2057	0.497	0	8.934	1.278	17.833	0	14.172
ISFJ_7328	2.087	0	0.861	2.811	0.667	6.954	0.454
ISFP_4282	0.099	0	6.835	1.022	16.667	0	6.463
ISTJ_0178	0	0	7.158	2.47	11	0	6.349
ISTJ_2068	0.696	0	0	1.278	0	0	0
ISTJ_2837	0.099	0	5.759	2.3	16	0	11.791
ISTJ_3052	93.439	0	0.484	4.174	7.833	0	6.349
ISTJ_4667	0	70.675	0	0	0	0	2.381
ISTJ_4700	0.099	0	77.826	0	10	0	11.565
ISTJ_4753	0	0	1.184	23.254	8.167	0	6.236
ISTJ_4968	0	0	4.252	2.3	55.167	0	13.946
ISTJ_9139	0	0	0	0	0	81.126	0
ISTJ_9576	0.099	0.738	7.535	1.278	13.833	0	85.374
ISTP_3948	0.696	0	4.629	0.767	4.667	0	4.875
ISTP_7676	0.099	0	3.014	2.47	17.333	0.662	11.111
XXXX_XXXX	0.398	0	4.629	1.278	7.5	0	9.977

model	t.ISTP_3948	t.ISTP_7676	t.XXXX_XXXX
ENFP_0719	0.937	0.608	0
ENFP_2012	0	0	0
ENTJ_6454	0.52	0.608	1.592
ENTJ_6966	2.393	3.066	14.65
ENTP_5623	2.81	1.031	6.369
ESFJ_2301	2.601	0.978	5.732
ESFJ_9284	1.561	0.053	0
ESFP_0912	0	0	0
ESFP_4634	1.561	0	0
ESTJ_3022	2.81	2.511	16.242
ESTJ_5071	0	0.899	0
ESTJ_5824	3.018	0.608	6.688
ESTJ_6510	3.434	3.859	17.197
ESTP_4301	3.018	5.736	21.019
ESTP_5154	1.145	1.903	11.465
INFP_1993	0	0	2.866
INTJ_5498	1.561	0	0
INTJ_7906	3.018	0.608	6.688
INTP_3739	2.706	1.269	2.548
INTP_6399	0.624	1.163	0.955
INTP_9712	3.018	2.511	18.153
ISFJ_2057	2.81	3.04	20.701
ISFJ_7328	1.769	0.846	1.274
ISFP_4282	3.018	2.458	9.554
ISTJ_0178	1.977	1.031	4.777
ISTJ_2068	1.561	0.053	0
ISTJ_2837	3.226	4.679	18.471
ISTJ_3052	1.561	2.881	12.42
ISTJ_4667	0	0	0
ISTJ_4700	2.81	0.846	6.369
ISTJ_4753	1.041	2.723	11.783
ISTJ_4968	2.185	5.26	14.65
ISTJ_9139	0	0.053	0
ISTJ_9576	3.226	3.489	16.561
ISTP_3948	81.27	0.608	6.688
ISTP_7676	3.122	65.953	17.834
XXXX_XXXX	3.018	4.229	95.223

The percentage of identified B2 by B1 in different dataset condition

(We split the output to four charts because we have many data)