International Journal of Software Engineering and Its Applications
Vol. 9, No. 1 (2015), pp. 67-86
http://dx.doi.org/10.14257/ijseia.2015.9.1.06

Medical Alert System Using Fall Detection Algorithm on Smartphone

Irwan[1] and Benfano Soewito[2]

[1]Graduate Program Bina Nusantara University, Jakarta, Indonesia
[2]Faculty Member of Magister of Information Technology, Bina Nusantara University, Jakarta, Indonesia
[1]wan.lynnfield@gmail.com, [2]benfano@gmail.com

Abstract

The purpose of this paper was designing a fall detection algorithm as a solution so victims who are unable to move after accident could get help. Data were gathered by experimenting using accelerometer to find standby threshold and fall threshold. Data for standby threshold were processed by calculating all the differences and thus use the maximum difference for a threshold. Data for fall threshold were also processed by calculating all the differences and thus use the minimum difference from activities that were ended with a fall. Those thresholds were implemented into prototype application and tested again to find its accuracy. Results were 1 on sensitivity and 0.96 on specificity. It could be concluded that this fall detection algorithm has a 0.98 on its accuracy.

Keywords: *Fall detection algorithm; Android; Medical Alert System; Accelerometer*

1. Introduction

Accident is around us every moment. Random people can be victims from an accident. Help is needed for those victims, even though a small cut. But, what if victims suffer severe injuries? There is only two results from this condition, victims will either ask help by themselves or asking help from someone else.

Victims who are asking help from others are tend to still be able to move. But what if victims are not conscious? Will they be able to ask for help? Of course it is an impossible task for them to ask for help. They will need help from others.

So from this case, someone who had an accident and is not conscious has a problem. How can they ask for help? Will others give help to them immediately?

This research is trying to solve this problem. This research is focusing in combining technology and communication to solve this problem.

The use of technology is started by a researcher [1]. He developed a ring-shaped device. This ring had a sensor to detect heartbeat. It detected heartbeat and stored it on smartphone. This ring was used to monitor the user's heartbeat. It was connected to a smartphone, so it would display the result in a graph. From this research, three aspects are considered to be used is as follows:

1) The used of smartphone,
2) The used of wireless connectivity or internet on exchanging information, and
3) The used of client and server architecture in processing information.

Later in 2011, a researcher from Orebro University, Gregory Koshmak [2] developed a technology to monitor its user's health condition. Smartphone was used in this research and it was using a sensor from that smartphone. Accelerometer was the sensor that was used in this research. Accelerometer detected and gathered heartbeat data and stored it.

ISSN: 1738-9984 IJSEIA

After that, the stored data would be sent to server in hospital. In server, that data will be processed and mapped into graph information.

This information would be used by hospital in decision making. The used of Android operating system smartphone will be considered to use. It has the function to detect motion of smartphone using accelerometer.

Then, in 2011, a researcher completed the same research with a complete step for developing the system. Fuchao Zhou [3] from Iowa State University, completed that research with a detail documentation for the system, including storage, infrastructure design, and database. The method used will be considered to be followed in this research.

Another technology used is GPS [4]. Researcher in 2009 had already conducted a research about the comparison of GPS (Global Positioning System) and RF Tags on devices. Those devices were worn by elderly. From this research, there was a suggestion from researcher. It was said that elderly doesn’t need many devices to wear. They only need one device that can do all things, including monitoring their health and tracking their location. This suggestion was based because elderly are tending to forget their device. So according to this suggestion, there will be only one devices used in research.

Gonzales, on 2011 [5], conducted a research in designing and implementing a fall detection algorithm. There were three algorithms with different threshold. This algorithm was evaluated by calculating its sensitivity and specificity. The evaluation was started by calculating its true positive, true negative, false positive and false negative. Then, there was formula to calculate sensitivity and specificity using those four values. From this research, there are aspects to be considered to be used in research, as follows:

1) The evaluation method,
2) The calculation of four values (TP, TN, FP, FN), and
3) Sensitivity, specification and accuracy formula.

The case model for this research is illustrated in Figure 1. The case itself has a problem; it is when victims had an accident where there is no one around and can’t move or unconscious. What can they do to get help? It will be a serious problem if they don’t get help soon; maybe they suffer severe injuries and need treatment immediately.

So to be simple, the research question is “How an unconscious victim could get help even if there is no one around?” This research is conducted to solve that problem by combining smartphone technology and network technology.

A fall detection algorithm will be the main focus of this research to detect the fall of a person. But before that, the definition of a fall in this research must be clear. A fall is a sudden position change of a human into lying on ground for five seconds. Fall detection algorithm, the use of Android smartphone, the use of accelerometer, and the development of a prototype application will be conducted in this research.

This paper will describe method used in designing a fall detection algorithm.

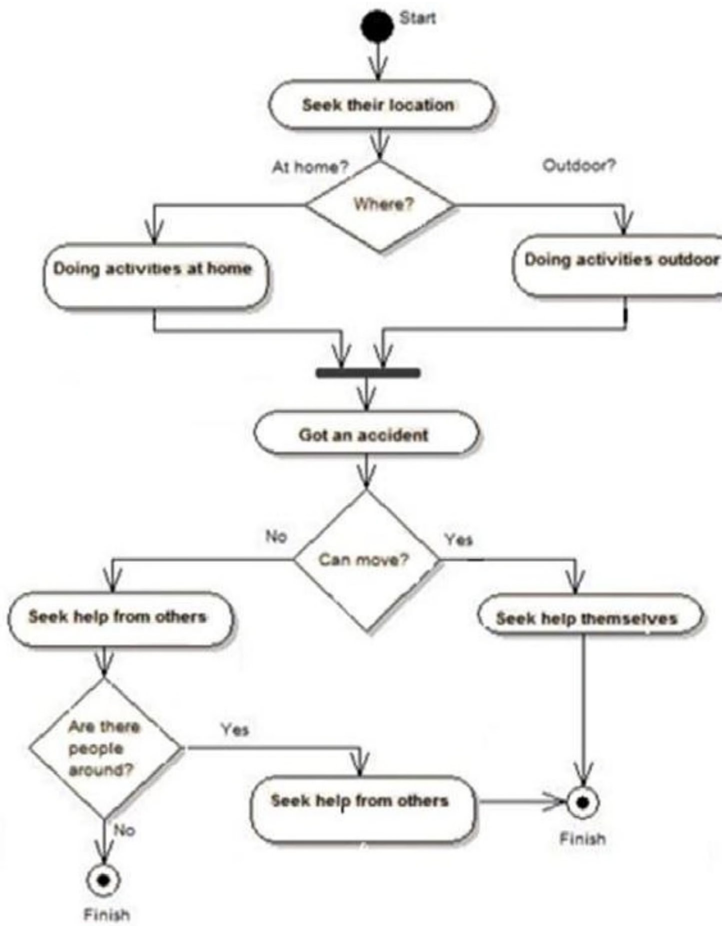

Figure 1. Case Process Model

2. Methodology

There are seven steps in research.

1) Problem statement, this step is where problem is discovered and analyzed.

2) Literature review, this step is where theory is gathered and corresponding research is found. The goal is to find the right method for solving problem.

3) Design solution, after knowing what method to use, this step is designing solution for solving problem.

4) Design prototype system, this step is part of developing solution. A prototype is designed and developed. A fall detection algorithm will be implemented in this step.

5) Testing prototype, after prototype is completed, then prototype is tested to find bugs and fixing bugs.

6) Analysis research result, this step is focusing on fall detection algorithm's result. All the result from the test of algorithm is described and evaluated here.

7) Conclusion is the step to define whether the problem is solved or not. This step is also will give a quantitative result as a conclusion.

2.1. Problem Statement

As described before, problem in this research is, “How an unconscious victim could get help even if there is no one around?” Designing a solution for this question becomes the focus of this research.

2.2. System Design

The following figure describe model of proposed design.

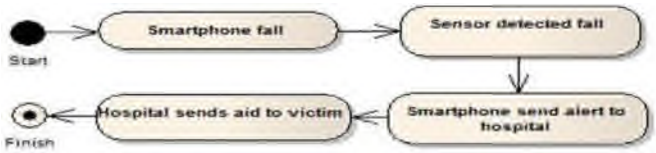

Figure 2. System Design

This proposed design is aiming to solve research problem. This proposed solution is using a client and server technology. The client is Android smartphone and server is a computer that sits in hospital.

The process of Figure 2 is as follows:

1) Victim falls because accident and smartphone falls too, assuming that smartphone is placed in pocket.

2) Smartphone will detect fall with the fall detection algorithm and generate a signal when fall is detected.

3) Smartphone will send that alert to server (hospital).

4) Server (hospital) will send aid to victim.

Above solution will become a new process from the problem process. This new process will replace some of old process. Figure 3 is the model of new process.

2.3. Prototype System

Prototype is developed based on solution. Prototype will have architecture like Figure 4. Figure 4 shows that the system used in this research is using client and server architecture. Android devices act as client and computer act as server. Each component has its own functions. Here are client functions:

1) Login,

2) Automatically activate sensor, active sensor means fall detection algorithm is working, and

3) Sending alert to server if fall detected.

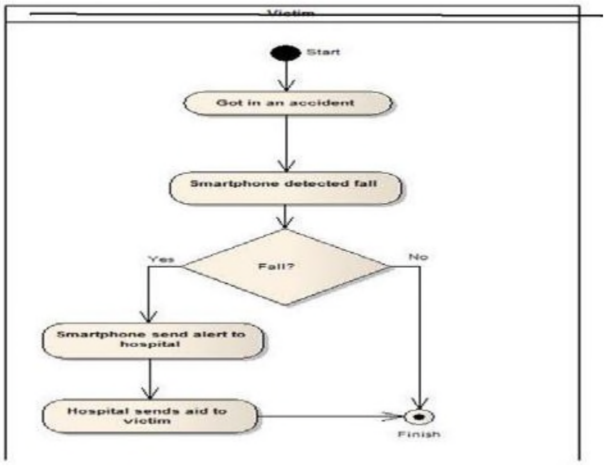

Figure 3. New Process Model

Here are server functions:

1) CRUD (Create, Read, Update, Delete) patient's data,
2) CRUD user admin data,
3) RU (Read, Update) alert,
4) Receive alert,
5) Display alert list, and
6) Display patient information. There

are also two databases in this system:

1) Patient database on client, and
2) Patient information database on server.

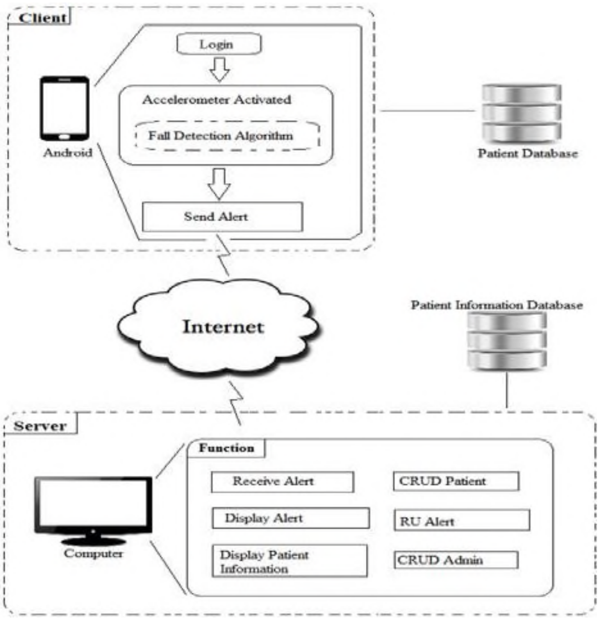

Figure 4. System Architecture

First database, patient database, will be developed on Android and based on SQLite.

This database will contain patient ID that is using this prototype application. Schema of the database shows in Figure 5. Patient database will have the corresponding description:

1) “Id”, Auto-increment, and
2) “Pid”, represent Patient ID.

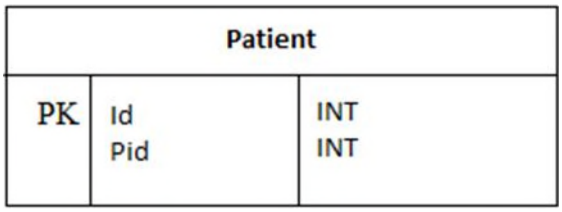

Figure 5. Client Database Schema

Record on Patient database only has one record, application prototype user’s record. When fall is detected, Pid will be sent to server along with smartphone location.

Second database, patient information database, will be developed on server. This database will store information of registered patient. It will also store alert sent from client. Schema of this database shows in Figure 6.

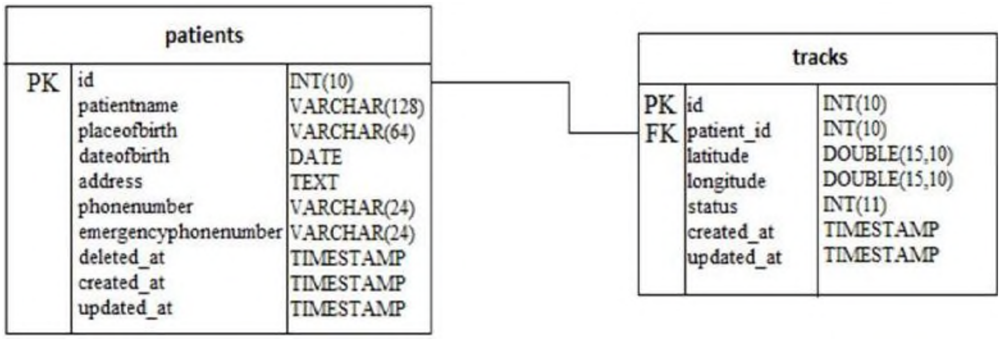

Figure 6. Server Database Schema

Record on table “patients” will store patient information with description as follows.

1) “id” represent patient ID,
2) “patientname” represent patient’s name,
3) “placeofbirth” represent patient’s place of birth,
4) “dateofbirth” represent patient’s date of birth,
5) “address” represent patient’s address,
6) “phonenumber” represent patient’s phone number,
7) “emergencyphonenumber” represent other people emergency phone number,
8) “deleted_at” represent time and date this record is deleted,
9) “created_at” represent time and date this record is created, and
10) “updated_at” represent last time and data this record is changed.

Record on table “tracks” will store information of alert sent from client to server. It has description as follows.

1) “id” represent ID from track,

2) “patient_id” represent patient’s ID that send alert,

3) “latitude” represent latitude value when alert is sent,

4) “longitude” represent longitude value when alert is sent,

5) “status” represent alert status, starting with “1” as pending, “2” as processed, and “3” as finished.

6) “created_at” represent time and date this record is created, and

7) “updated_at” represent last time and date this record is changed.

Figure 7 shows the process model for the system.

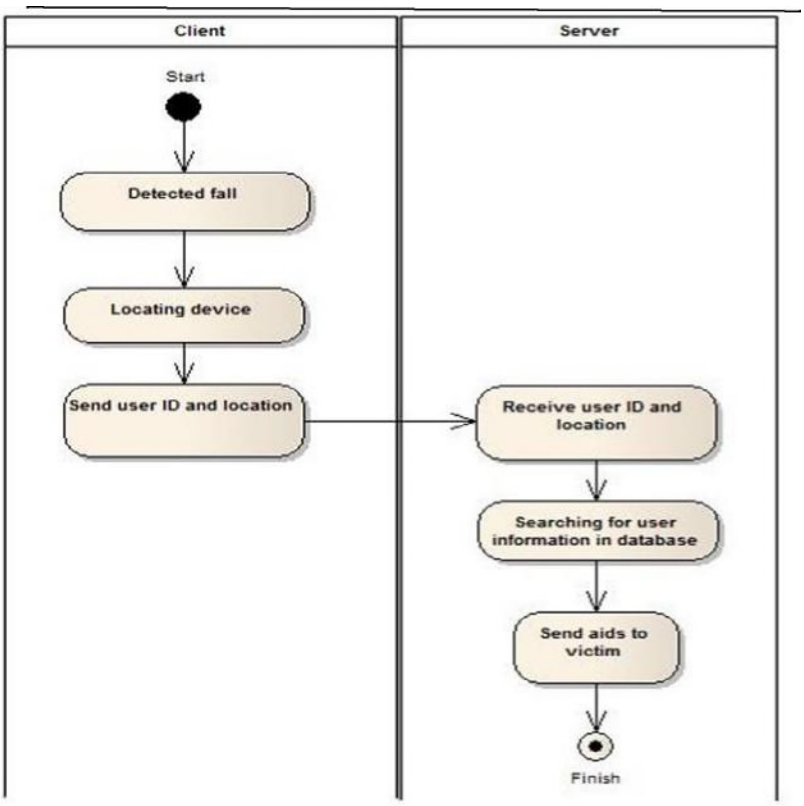

Figure 7. System Process Model

After designing system, system process must also be designed. These are system process for client and server.

1) Client prototype application detected fall.

2) Then, application locating device using GPS. If GPS can’t get location, then user location will be located using communication network automatically.

3) Alert, containing user ID and user location, is sent to server.

4) Server then receive alert sent by client.

5) User ID sent from client is used to find information regarding patient in the server database.

6) The host of server, hospital, will make decision regarding the complete information from the alert sent.

2.4. Fall Definition

Fall definition in this research must be defined for a clear goal. So in this research, "A fall is a sudden position change of a human into lying on ground for five seconds".

2.5. Data Gathering

Data is gathered by calculating g-force value. G-force value is calculated by following formula.

(1)

where G is G-force, a is acceleration and g is gravity.

Stored g-force value is highest and lowest g-force. There are three activities category in data gathering.

1) Frequency-based data,

2) Duration-based data, and

3) Frequency-duration based data.

First category activities are recorded for 100 times. The activities are as follows.

1) Smartphone is dropped from a height of 20cm and data is recorded when smartphone is on the ground.

2) Smartphone is dropped from a height of 50cm and data is recorded when smartphone is on the ground.

3) Smartphone is dropped from a height of 100cm and data is recorded when smartphone is on the ground.

4) Smartphone is placed in pocket; user sits on chair with a height of 50cm and stands. Data is recorded when user is standing.

5) Smartphone is placed in pocket; user squat and data is recorded when user is standing.

Second category activities are recorded for 100 seconds with an interval of 1 second. The activities are as follows.

1) Smartphone is placed on table with a height of 70cm.

2) Smartphone is placed on the ground.

3) Smartphone is placed in pocket and user sits on a chair with a height of 50cm.

4) Smartphone is placed in pocket and user stands still.

5) Smartphone is placed in pocket and user lying on ground.

6) Smartphone is placed in pocket and user walks.

7) Smartphone is held by user while typing message and walks.

8) Smartphone is placed in pocket and user skipping.

9) Smartphone is placed in pocket and user plays hula hoop.

Third category activities are recorded for 30 times with duration of 10 seconds. The activities are as follows.

1) Smartphone is placed in pocket; user walks and sits.
2) Smartphone is placed in pocket; user walks and stops in a standing position.
3) Smartphone is placed in pocket; user walks and falls in a lying position.
4) Smartphone is placed in pocket; user jumps and falls in a lying position.

Two values will be calculated, they are highest g-force value and lowest g-force value. Another value is calculated too, it is difference value. It has the formula as follows.

(2)

where *H* is highest g-force and *L* is lowest g-force.

2.6. Fall Detection Algorithm

This research will be using an algorithm to detect fall. It is called Fall Detection Algorithm. This algorithm is developed base on fall definition. It has three conditions as follows to classified fall:

1) Devices has a minimum height of 50cm from the ground,
2) Devices moves to ground, and
3) Devices standby on ground for a minimum time of 5 seconds.

If all three conditions above is achieved, then a fall is detected.

Fall detection algorithm will be using accelerometer as its sensor. This algorithm will be implemented into Android devices, because all Android devices have accelerometer. There will be three values retrieved from accelerometer when it is activated, they are:

1) X coordinates acceleration,
2) Y coordinates acceleration, and
3) Z coordinates acceleration.

Those following coordinates are retrieved based on this formula.

$$a \,\square\, a_x^{2} + a_y^{2} + a_z^{2}$$

(3)

where is device acceleration (x, y, or z), g is gravity acceleration constant, F is force, and m is mass.

From that formula, we will achieve three values; they are ax, ay, and az. They stand for acceleration for x coordinate, y coordinate, and z coordinate. By knowing these three values, we still won't know when devices fall. So we simplify these values into one value using Pythagorean Theorem as follows.

(4)

Where a is total acceleration, a_x is acceleration in x coordinate, a_y is acceleration in y coordinate, and a_z is acceleration in z coordinate.

This acceleration value will be used to calculate its g-force value using the first formula. After knowing its g-force, we can now determine if it is accelerating towards ground or not. If g-force value is moving close to 0, then it is accelerating towards ground.

If it is not, then it is accelerating towards other position.

G-force value will always be used in this research. Figure 8 shows fall detection algorithm model.

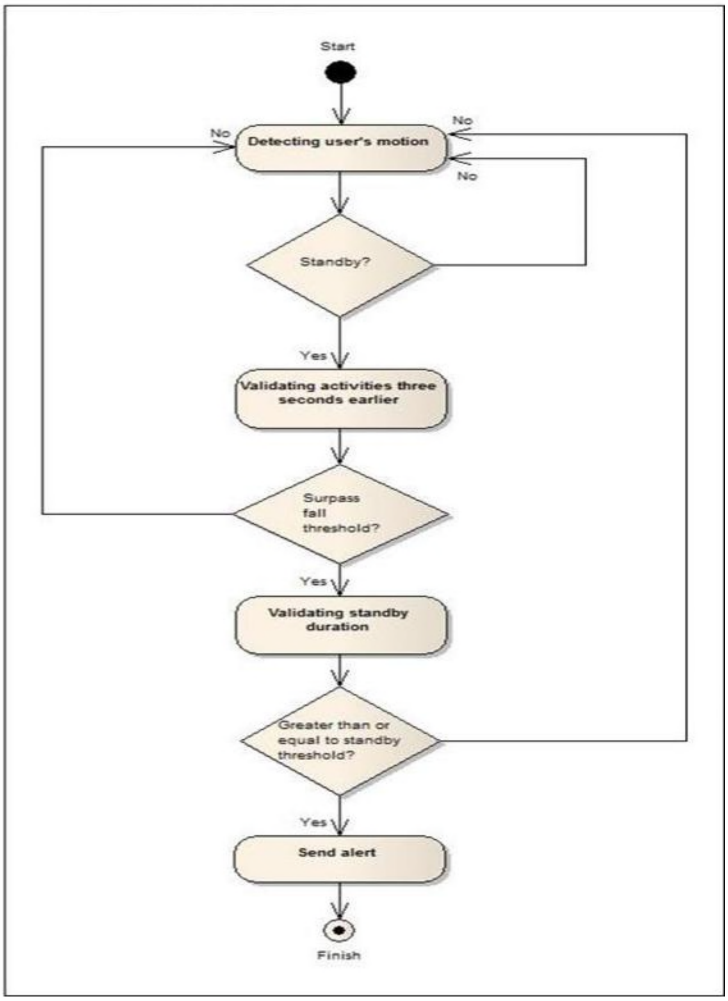

Figure 8. Fall Detection Model

The above fall detection algorithm is designed based on fall definition and it is designed solely for this research. This algorithm follows these steps as following.

1) Detecting user's motion, checking for if user standby or not.

2) If user is standby, then check for activities three seconds earlier, is it surpassing fall threshold or not.

3) If it is surpassing fall threshold, then validate if user is standby for five seconds or more.

4) If user is standby for five seconds or more, then send alert.

5) But if user is standby for less than five seconds, then no alert is sent.

6) Also, if user is not surpassing fall threshold, then no alert is sent, too.

7) Also, if user is not standby after surpassing fall threshold, then no alert is sent, too.

We can be concluded that this research will be using two thresholds to detect fall and standby from above steps.

2.7. Evaluation Method

Evaluation is conducted after fall detection algorithm has been implemented into client prototype application.

Evaluation is conducted by calculating TP (True Positive), FP (False Positive), TN (True Negative) and FN (False Negative). Those four values are used to calculate sensitivity, specificity, and accuracy.

Here is how the value will increase.

1) TP will increase if user is falling and smartphone sends alert to server,
2) TN will increase if user is not falling and smartphone doesn't send alert to server,
3) FP will increase if user is not falling but smartphone sends alert to server, and

FN will increase if user is falling but smartphone doesn't send alert to server.

3. Research Result and Discussion

3.1. System Development Result

Two applications are developed in this research as a prototype. It is used as a client and a server. Client application will be implemented into Android devices, and server application will be implemented into computer.

Server application is named MAS (Medical Alert System) which is developed using PHP 5.4.16. Client application is named FallClient which is developed using Java programming language and is targeted to be used in Android 4.2. Table 1 shows Android devices specification.

3.2. Data Processing Result

There are three categories of activities in this research. Data is gathered from those categories. The first categories will gather the following values.

1) Highest g-force,
2) Lowest g-force, and
3) Difference value between highest and lowest g-force.

After those values above are gathered, then the maximum, minimum, and average of those values are calculated. The second categories will gather the same values as the first categories.

The third categories will gather the following values.

1) Initial standby difference value,
2) Maximum difference from three seconds before initial standby difference, and
3) Difference between value number one and number two.

Table 1. Smartphone Specification

Criteria	Description
Mass	159g
Display	5.2 inches, 720x1280 pixels
Data	GPRS Class 12, EDGE Class 12, Wi-Fi

	802.11 a/b/g/n, dual band
Sensor	Accelerometer
OS	Android OS v.4.2.1 (Jelly Bean)
Chipset	Mediatek MT6589
CPU	Quad-core 1.2 GHz Cortex A7

Table 2 shows maximum, minimum, sum, and average data from first category. Table 3 shows maximum, minimum, sum, and average data from second category. Table 4 shows maximum, minimum, sum, and average data from third category.

Those results can be concluded into this following.

1) Maximum difference value of standby activities is 0.01092676. It means that if there is an activity with difference value greater than that value, it is not standby.

2) Activities that are in the state of free fall will have difference value lower than 1. It is tend to have difference value around 0.

3) Activity changes will have distinction between activities that end with fall and activities that do not end with fall. The distinction lies on difference value between initial standby difference and difference from three second earlier before standby. The difference value will be greater to activities that are ended with fall.

Table 2. First Category Data

Activities	Data	Sum	Min	Max	Average
Drop from 20cm	Highest g-force	110.417296	2.355708128	0.991721697	1.10417296
	Lowest g-force	13.61046714	0.528406564	0.044811162	0.136104671
	Difference	96.80682888	1.926863041	0.474568231	0.968068289
Drop from 50cm	Highest g-force	129.3409332	3.052207128	0.994608461	1.293409332
	Lowest g-force	9.426788275	0.109477204	0.083964323	0.094267883
	Difference	119.9141449	2.954942556	0.900584546	1.199141449
Drop from 100cm	Highest g-force	140.5919053	3.250369613	0.990222729	1.405919053
	Lowest g-force	9.443136774	0.108361674	0.076093655	0.094431368
	Difference	131.1487685	3.161179354	0.886116214	1.311487685
Sit	Highest g-force	140.3612524	1.98831039	1.126453527	1.403612524
	Lowest g-force	90.49484778	0.996106809	0.622299137	0.904948478
	Difference	49.86640462	1.191068049	0.208236628	0.498664046
Squat	Highest g-force	149.1186455	1.876311453	1.290869091	1.491186455
	Lowest g-force	88.54328931	1.060626756	0.39909234	0.885432893
	Difference	60.57535619	1.435730512	0.239155349	0.605753562

3.3. Threshold value

Standby threshold value is taken from five activities that are included in standby activities. Those are:

1) Smartphone is placed on table with a height of 70cm.

2) Smartphone is placed on the ground.

3) Smartphone is placed in pocket and user sits on a chair with a height of 50cm.

4) Smartphone is placed in pocket and user stands still.

5) Smartphone is placed in pocket and user lying on ground.

Table 3. Second Category Data

Activities	Data	Sum	Min	Max	Average
Put on desk	Highest g-force	99.16019337	0.995689441	0.988717476	0.991601934
	Lowest g-force	98.91449135	0.99361985	0.984060121	0.989144913
	Difference	0.245702026	0.010523511	0	0.00245702
Put on floor	Highest g-force	99.19993877	0.994613746	0.987624317	0.991999388
	Lowest g-force	98.95495192	0.991530492	0.982922582	0.989549519
	Difference	0.244986843	0.008741301	1.52577E-05	0.002449868
Sit on pocket	Highest g-force	96.27909801	0.968526544	0.960086504	0.96279098
	Lowest g-force	96.11236949	0.963724413	0.958487956	0.961123695
	Difference	0.166728519	0.008219222	0	0.001667285
Stand on pocket	Highest g-force	87.28232307	0.877598535	0.868778687	0.872823231
	Lowest g-force	87.05217867	0.873528178	0.867804255	0.870521787
	Difference	0.230144395	0.007198482	1.18737E-05	0.002301444
Lay on ground	Highest g-force	96.79487882	0.974645442	0.964756165	0.967948788
Walk test	Lowest g-force	96.56212875	0.968499834	0.959985448	0.965621288
	Difference	0.23275007	0.010926764	0	0.002327501
	Highest g-force	120.3970988	2.479915781	0.98833948	1.203970988
Walk-texting test	Lowest g-force	103.9465565	1.574996615	0.730941252	1.039465565
	Difference	16.4505423	1.222875931	0.000370092	0.164505423
	Highest g-force	121.2601944	1.583205272	0.940454613	1.212601944
Walking up stairs	Lowest g-force	77.13043474	1.044882822	0.356949553	0.771304347
	Difference	44.12975969	0.972478618	0.136933533	0.441297597
	Highest g-force	123.1196634	2.787400417	0.54836115	1.231196634

Table 4. Third Category

Activities	Data	Sum	Min	Max	Average
Walk Sit	Initial Standby Difference	1.317746348	0.019903971	0.002138196	0.013177463
	Highest 3 Seconds Before Standby	75.61878729	1.532114083	0.411658267	0.756187873
	BD - ISD	74.30104094	1.512347145	0.398415204	0.743010409
Walk Stand	Initial Standby Difference	1.087974284	0.019872262	0.002771964	0.010879743
	Highest 3 Seconds Before Standby	63.71201614	1.109117402	0.259538273	0.637120161
	BD - ISD	62.62404186	1.092805031	0.254507614	0.626240419
Walk Fall	Initial Standby Difference	1.030797045	0.019667231	0.002206147	0.01030797
	Highest 3 Seconds Before Standby	185.7148297	3.211097585	1.034462665	1.857148297
	BD - ISD	184.6840327	3.197259435	1.025697539	1.846840327
Jump Fall	Initial Standby Difference	1.165599352	0.019785582	0.00209277	0.011655994
	Highest 3 Seconds Before Standby	236.0620528	3.41365409	1.433921323	2.360620528
	BD - ISD	234.8964534	3.403780289	1.425310416	2.348964534

Maximum difference value from all of those five activities data is chosen as standby threshold. Maximum difference value is chosen because all of the activities are categorized as a standby, so if maximum value is chosen then all those data will become true positive for standby. Thus, standby threshold for this research is 0.01092676.

Fall threshold value is taken from the third categories activities. There are four activities in third categories. Those activities are divided into activities that ended with fall and activities that do not end with fall. The first and second activities are activities that do not end with fall. The third and fourth activities are activities that end with fall.

There are distinctions between difference values from activities that are ended with fall and not ended with fall. The difference values from all four activities are described in the figure 9.

Fall threshold will be chosen from activities that are ended with fall. It is activities number three and four; all of it is categorized as activities that ended with fall.

Minimum difference value from all the differences is chosen as fall threshold so all differences from activities number three and four can be detected as true positive.

True positive means that it is categorized as fall. Fall threshold for this research is 1.025697539.

3.4. Algorithm Implementation

Implementation algorithm will be conducted right after two threshold values are chosen. This implementation will be represented as a pseudocode. Figure 10 is pseudocode used in developing Android prototype to detect fall.

3.5. Algorithm Testing Result

Pseudocode describes in Figure 11 is used for implementation on Android. It is implemented and tested. It is tested using scenarios as follows.

1) Activities that are ended with fall, and
2) Activities that are not ended with fall.

Each category is tested for 100 times. Here are activities in first category.

1) Walk-fall 25 times,
2) Jump-fall 25 times,
3) Run-fall 25 times, and
4) Stand-fall 25 times.

Here are activities in second category.

1) Walk-sit 25 times,
2) Walk-stand 25 times,
3) Walking up stairs-stand 25 times, and
4) Walking down stairs-stand 25 times.

Table 5 is results for above activities.

Table 5. Alert Result

Activities	Alert Sent
Walk-sit	1/25
Walk-stand	1/25
Walking up stairs-stand	1/25
Walking down stairs-stand	1/25
Walk-fall	0/25
Jump-fall	0/25
Run-fall	0/25
Stand-falll	0/25

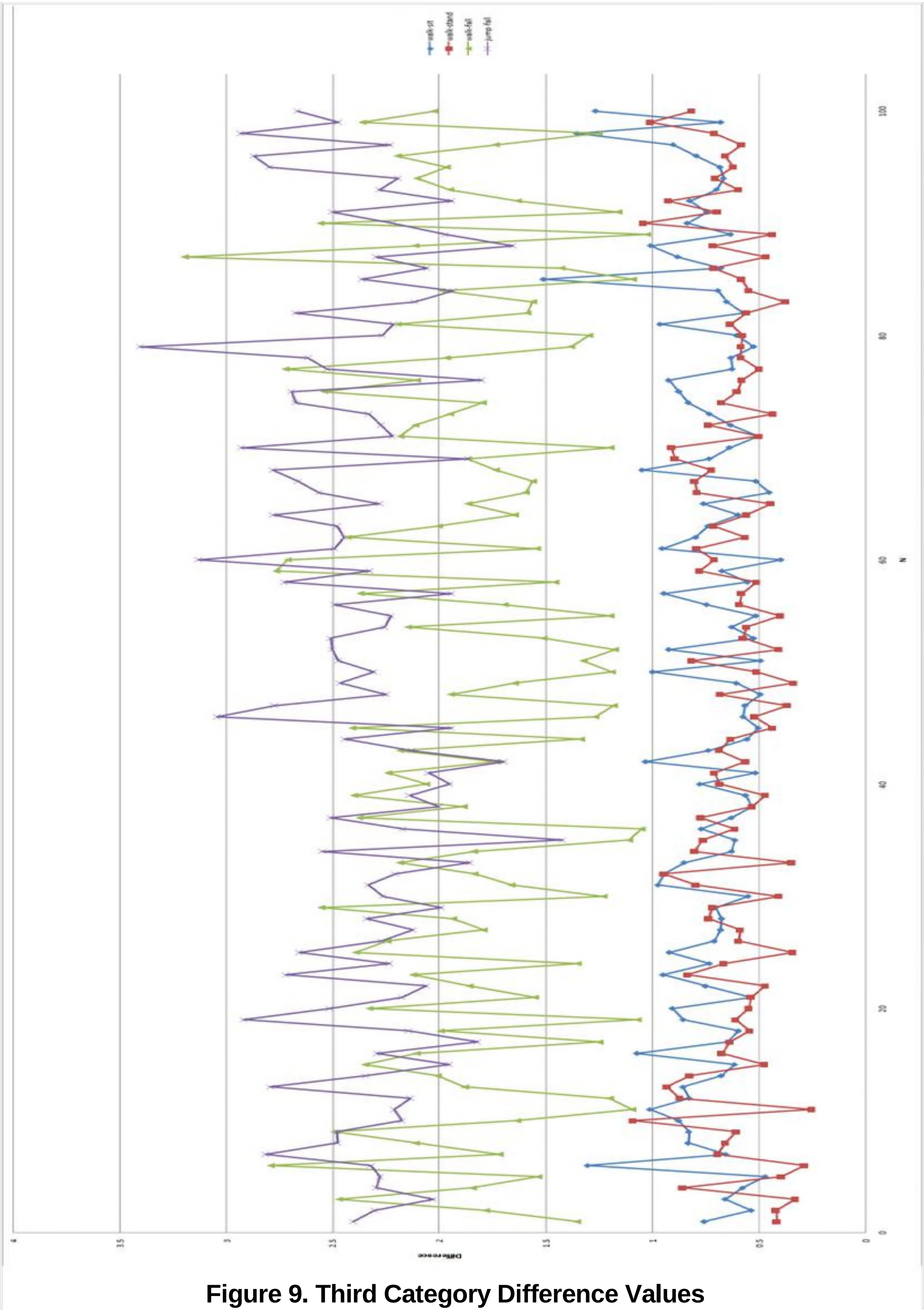

Figure 9. Third Category Difference Values

```
BEGIN
        Initialize accelerometer;
        Initialize variable STANDBY_THRESHOLD with value of 0.01092676;
        Initialize variable FALL_THRESHOLD with value of 1.025697539;
        Initialize variable SECOND with value of 1000;
        Initialize variable LOWEST_GFORCE with value of 0D;
        Initialize variable HIGHEST_GFORCE with value of 0D;
        Initialize variable DIFFERENCE[] with array of three index;
        Initialize variable curTime with value of current time in millisecond;
        Initialize variable STANDBY_FLAG, FALL_FLAG, START_FLAG, ALERT_AMOUNT each with value of 0;

        FOR(Every sensor's changes)
                Initialize variable x with value of sensorevent.values[0];
                Initialize variable y with value of sensorevent.values[1];
                Initialize variable z with value of sensorevent.values[2];

                Initialize variable gX with value of (x / SensorManager.GRAVITY_EARTH);
                Initialize variable gY with value of (y / SensorManager.GRAVITY_EARTH);
                Initialize variable gZ with value of (z / SensorManager.GRAVITY_EARTH);

                Initialize variable gForce with value of (ROOT(gX * gX + gY * gY + gZ * gZ));

                IF(START_FLAG == 0)
                        DIFFERENCE[0] = 0D;
                        DIFFERENCE[1] = 0D;
                        DIFFERENCE[2] = 0D;
                        START_FLAG = 1;
                ENDIF

                IF(START_FLAG == 1)
                        HIGHEST_GFORCE = gForce;
                        LOWEST_GFORCE = gForce;
                        START_FLAG = 2;
                ENDIF

                IF(LOWEST_GFORCE > gForce)
                        LOWEST_GFORCE = gForce;
                ENDIF

                IF(HIGHEST_GFORCE < gForce)
                        HIGHEST_GFORCE = gForce;
                ENDIF

                Show message "Detector is active" on smartphone;

                Initialize variable now with value of current time in millisecond;

                IF(curTime + SECOND <= now)
                        IF(HIGHEST_GFORCE - LOWEST_GFORCE <= STANDBY_THRESHOLD)
                                IF(FALL_FLAG == 0)
                                        IF(DIFFERENCE[2] >= FALL_THRESHOLD
                                        || DIFFERENCE[1] >= FALL_THRESHOLD
                                        || DIFFERENCE[0] >= FALL_THRESHOLD)
                                                FALL_FLAG = 1;
                                        ENDIF
                                ENDIF

                                IF(FALL_FLAG == 1)
                                        STANDBY_FLAG = STANDBYFLAG + 1;
                                ENDIF

                                IF(STANDBY_FLAG != 0 && STANDBY_FLAG % 5 == 0)
                                        ALERT_AMOUNT = ALERT_AMOUNT + 1;

                                        IF(ALERT_AMOUNT <= 3)
                                                Send alert to server;
                                        ENDIF
                                ENDIF
                        ELSE
                                STANDBY_FLAG = 0;
                                FALL_FLAG = 0;
                                ALERT_AMOUNT = 0;
                        ENDIF

                        DIFFERENCE[2] = DIFFERENCE[1];
                        DIFFERENCE[1] = DIFFERENCE[0];
                        DIFFERENCE[0] = HIGHEST_GFORCE - LOWEST_GFORCE;
                        curTime = now;
                        START_FLAG = 1;
                ENDIF
        END
END
```

Figure 10. Pseudocode

First until fourth activities are activities that not ended with fall. Fifth until eight activities are activities that ended with fall.

If alert is sent from activities that are not ended with fall, false positive value is increased. If alert is sent from activities that are ended with fall, true positive is increased.

If alert is not sent from activities that are not ended with fall, true negative is increased. If alert is not sent from activities that are ended with fall, false negative is increased.

Table 6 shows that four results.

Table 6. Implementation Result

Categories	Value
True Positive	100
False Positive	4
True Negative	96
False Negative	0

3.6. Testing Evaluation

Sensitivity, specificity, and accuracy are calculated using those four values above. Table 7 are three results of evaluation.

Table 7. Evaluation Result

Categories	Value
Sensitivity	1
Specificity	0.96
Accuracy	0.98

Value of 1 means it is 100% and value of 0 means it is 0%.

3.7. False Positive Anticipation

User who does not fall but the sensor still detects and sends alert to server is considered as a false positive. Maybe the fall is not serious so hospital does not need to send ambulance. So, this section is anticipating false positive to be counterproductive. Figure 11 is model to anticipate false positive.

Following are steps taken according to model above.

1) The first time alert is received, it means that fall is detected from victim's smartphone. This can mean that victim is falling or maybe it is a false alert.

2) Then hospital will send ambulance and try to call victim at the same time.

3) If there is response from victim, then hospital can ask if victim needs help or not.

4) If victim doesn't need any help, then hospital will have to stop sending ambulance.

Those steps are taken considering if victim fell but they can still move or receive call. To make sure if victim needs help, hospital call victim to know more information whether victim needs help or not.

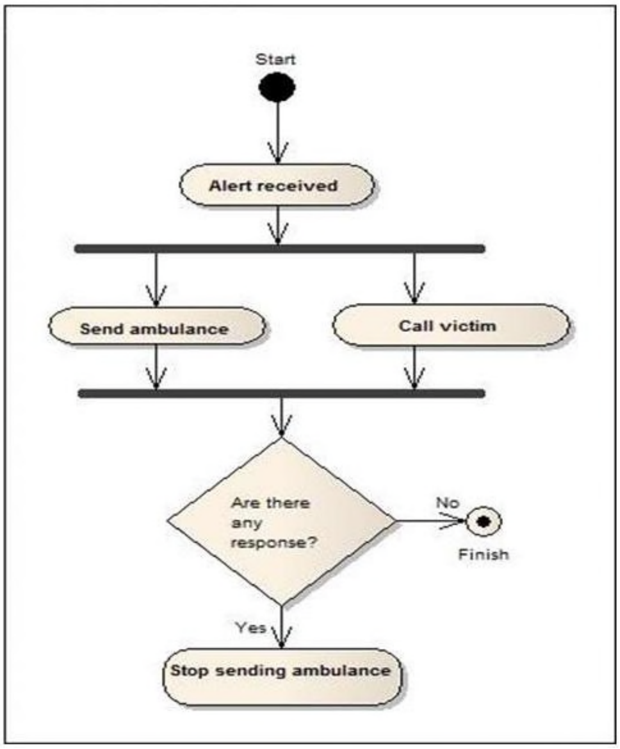

Figure 11. False Positive Anticipation Model

4. Conclusion

This research results are as following.

1) There are 104 alerts sent to server from a total of 200 experiments. 100 alerts are sent because fall is detected, and 4 alerts are sent even though activities are not ended with fall.

2) There are 96 alerts not sent to server because all of them are activities that are not ended with fall.

There are some improvements needed to be done in the next research. Those improvements are as following.

1) False negative anticipation is needed for false negative occurrence.

2) Activities frequencies in gathering data should be added to increase its accuracy in choosing threshold value.

3) Variance in activities should be added when gathering data to give variance of difference value.

4) An accurate threshold value is becoming focus when searching for this value because it will reduce the number of false positive.

5) Another sensor could be added to increase its ability to detect fall, for example gyroscope.

References

[1] Y. C. Wu, "A Mobile-Phone-Based Health Management System", Master of Science Thesis. Taiwan: National United University, **(2011)**.
[2] G. Koshmak, "An Android Based Monitoring and Alarm System for Patients with Chronic Obtrusive Disease", International Master's Thesis. Sweden: Orebro University, **(2011)**.
[3] F. Zhou, "Mobile Personal Health Care System for Patients with Diabetes". Graduate Theses and Dissertations. Ames, Iowa: Iowa State University, **(2011)**.
[4] M. W. Munir, "Tracking Devices for Elderly Care System by Using GPS and RF Tags", Master of Science Thesis. Finland: Tampere University of Technology, **(2009)**.
[5] I. J. D. Gonzales, "Fall Detection Using a Smartphone", Master's Thesis in Media Technology. Gjøvik: Gjøvik University College, **(2011)**.
[6] S. Abatte, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini and A. Vecchio, "A Smartphone Based Fall Detection System", Pervasive Mobile Computer, vol. 8, **(2012)**, pp. 883-899.
[7] T. Connoly and C. Begg, "Database Systems: A Practical Approach to Design", Implementation, and Management. 5th ed. Boston: Addison Wesley, **(2009)**.
[8] F. I. Darwin, "Android Cookbook", 1st ed. California: O'Reilly, **(2011)**.
[9] J. Freisen, "Learn Java for Android Development", 1st ed. New York: Apress, **(2010)**.
[10] M. Gargenta, "Learning Android", 1st ed. California: O'Reilly, **(2011)**.
[11] N. Jia, "Detecting Human Falls with 3-Axis Digital Accelerometer", Analog Dialogue, vol. 43-07, **(2009)**, pp. 1-7.
[12] M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg and T. Jämsä, "Sensitivity and Specificity of Fall Detection in People Aged 40 Years and Over", Gait Posture, vol. 29, **(2009)**, pp. 571-574.
[13] R. Matthews, "Beginning Android Tablet Programming", 1st ed. NewYork: Apress, **(2011)**.
[14] Z. Mednieks, L. Dornin, G. B. Meike and M. Nakamura, "Programming Android", 1st ed. California: O'Reilly, **(2011)**.
[15] R. Meier, "Professional Android 4 Application Development", 1st ed. Indianapolis: John Wiley & Sons, Inc, **(2012)**.
[16] R. Parikh, A. Mathai, S. Parikh, G. C. Sekhar and R. Thomas, "Understanding and Using Sensitivity, Specificity and Predictive Values", Indian J Ophthalmol, vol. 56, **(2008)**, pp. 45-50.
[17] R. Pressman, "Software Engineering: A Practitioner's Approach", 7th ed. New York: Mc Graw Hill, **(2009)**.
[18] Z. Roupa, M. Nikas, E. Gerasimou, V. Zafeiri, L. Giasyarani, E. Kazitori and P. Sotiropoulou, "The Use of Technology by the Elderly", Health Science Journal, vol. 4, **(2010)**, pp. 118-126.
[19] I. Sommerville, "Software Engineering", 9th ed. Boston: Pearson Education, Inc., **(2011)**.
[20] A. S. Tanenbaum, "Computer Networks", 5th ed. Boston: Pearson Education, Inc., **(2010)**.
[21] B. N. Taylor and A. Thompson, "The International Systems of Units (SI)", Maryland: National Institute of Standards and Technology, **(2008)**.