NANYANG
TECHNOLOGICAL
UNIVERSITY

Human Activity Data Analytics

A Final Year Project Report

by

Wee Jia Yi

Student ID: U1122725E

SCHOOL OF COMPUTER ENGINEERING

SCE14-0326

Human Activity Data Analytics

Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Computer Science of the Nanyang Technological University

by

Wee Jia Yi

(U1122725E)

Supervised By: A/P Tan Ah-Hwee

School of Computer Engineering

2015

Table of Contents

List of Figures

List of Tables

List of Equations

Abstract

Human activity recognition related technologies are enjoying a fuelled growth in investment as they gains popularity around the world, especially so for its application in home-based patient care, elderly living and many more. There exists a need to support this trending innovation with a robust and extensible motion recognition framework to serve as a widely applicable foundation for this research arena. Majority of the existing work face the problem of having a recognition system with limited generalisation capability (e.g. handling users from different age groups or handling situations when sensor devices are unfixed) as well as the inability to address high level motion transitions (e.g. Standing Up or Falling Down).

In this project, we propose for the establishment of a highly robust human activity recognition framework which can accurately detect both low level motion (e.g. Sitting, Standing) and high level motion transitions (e.g. Standing Up, Falling Down). This new framework will be referred to as the Robust Activity Recognition with Motion Transition (RARMT) framework in the rest of this report. Furthermore, an extensive set of experiments will be conducted to test for the performance of the RARMT framework, providing assurance on the reliability of the system.

The three major contributions in this project are listed as follows. Firstly, we established the low level motion recognition processing components of RARMT framework and used several benchmark datasets for performance testing. The benchmark results achieved are positive and there is even an accuracy improvement over what was achieved by the benchmark models. Secondly, we collected a new dataset covering two distinct age groups (i.e. youth and elderly) while also capturing human behavior in their most natural environmental settings. Extensive experimentations were performed and promising results were achieved, proving the RARMT's robustness in handling diverse datasets and noise. Lastly, we designed a novel way of detecting higher-level motion transitions. With that, we extended the RARMT framework to allow for motion transition recognition, enabling us to recognize a wider range of human activities (i.e. not limited to basic low level motions). Thus, providing the system with the extensibility to support many other activity recognition applications like those that can handles health-critical situations (e.g. falling down motion).

With the RARMT framework, we believe that it can provide a robust foundation for many more exciting technological innovations that hinges on reliable human motion recognition in the future.

Acknowledgement

First and foremost, I would like to thank my research supervisor, Associate Professor Tan Ah Hwee. Without his dedicated involvement in every step throughout the process, this project would have never been accomplished. His commitment towards excellence and constant improvement has encouraged me to go further and stay focused. The weekly meetings conducted by Professor Tan provided essential guidance for me to stay on track and generate new ideas for the work. I would like to express my gratitude for the support and understanding provided over the past year.

I would also like to show gratitude to Doctor Wang Di, the project manager for this batch of FYP students. His enthusiasm in the research field motivated me to never stop learning more about my area of study and his extensive knowledge in the field provided great help to this project. Furthermore, I am grateful for his dedication and time spent in helping me appreciate the ideas and concepts in my research work, which were critical for my project's progression.

Last of all, I would like to thank the group of volunteers for their commitment and patience in staying through the data collection activities. They were most understanding and willing to help in the project by providing their valuable time and effort.

None of this could have happened without the support and guidance of these people. I am thankful for your help and time spent in making the entire research journey in this project a smooth and fulfilling one.

1. Introduction

1.1 Background and Motivations

Human activity recognition related technologies are increasingly gaining popularity and importance around the world. This is especially so for its application in monitoring and alert systems used by people in need of constant medical attention. This population has been consistently growing due to the global phenomenon of an aging population [7], which affects countries across different levels of development, from developed countries like Singapore to developing ones like China. Singapore is among the fastest aging countries in the world. The number of those aged 65 and above will nearly double from 352,000 in 2011 to 600,000 by 2020. Even for a developing nation like China, it is predicted that its older population will likely swell to 330 million, or a quarter of its total population, by 2020.

These grey markets have become target gold mines for many eager investors; elderly friendly technologies and especially retirement villages are sprouting rapidly across the world. All of which aims to cater for the growing demand of elderly wishing to age alone, and to live their lives as individuals independently in a safe environment [9]. This can be achieved through ensuring sufficient monitoring and administration.

With that, reliable and robust monitoring technologies will be needed to ensure that the health-related needs of this grey population are met, especially so for those living alone or spending most of their time alone while their families are out. Currently, most of the medical monitoring and alert system tools used require significant user interactions and intrusive monitoring. For example, the use of emergency buttons and accessories [8] or surveillance cameras, to alert medical personnel when in need. There are growing demands for more robust and non-intrusive monitoring systems that can raise the necessary alerts in case of a suspected medical concern with minimal user interaction and intrusion to one's privacy [1, 2, 3].

Human Activity Daily Living (ADL) Patterns can meet this demand, as it is one technology that enables robust monitoring with minimal need for human intervention. Based on the daily human activity dataset of individuals, the system is able to generate a general lifestyle pattern of the person. Once the system detects abnormalities in the living pattern at any time, or detects activities which require immediate medical attention such as a fall; it will be able to raise an alert instantly. This removes the risk that human-interactions dependent monitoring tools have. For example, inability to manually press the emergency button after a fall. Furthermore, such technology tap into daily-used gadgets like the smartphone for monitoring which provides ease of use and can also lower one's concern about privacy intrusion as the sensors are well blended into the environment. However, there is still need for the advancement of such technology to become more robust and reliable in detecting these human activities, especially under different environment conditions and across various user profiles, in order to maximize accuracy and minimize false alarms. Hence, there is a call for greater contribution to this field of activity recognition, before it can be realized as a truly feasible solution.

1.2 Project Aims and Objectives

In this project, we aim to develop a robust and reliable activity recognition system. As such, there will be a strong focus on experimentation and analysis to extensively test for the performance of the developed system. In short, there are three main objectives in this project.

Objective 1

To build a new framework - **Robust Activity Recognition with Motion Transition (RARMT)** that is capable of recognising both basic low level motion (i.e. Lying, Sitting, Standing, Walking and Running) accurately, as well as determining transitions of these motion (e.g. Sitting Up, Falling Down) reliably. We will adopt concepts from work in [17] to develop RARMT's ability to recognise low level motion robustly; also we will design and implement a novel way of motion transition detection using a rule-based algorithm. This new model will provide us with immense improvement from the existing work done on human activity recognition by the previous student [4].

Objective 2

To establish a dataset that covers two distinct age groups (i.e. youth and elderly) and also captures human behaviour in their most natural environmental settings. This dataset will be playing a critical role in the following experimentations performed to ensure that the new RARMT framework implemented in this project is well able to cater for users from very distinct demographic profiles and has high tolerance to noise.

Objective 3

This objective focuses on conducting extensive experimentations to test for the performance of the new RARMT framework.

1. Since RARMT's low level motion recognition component is developed based on core concepts adopted from [17], we aim to replicate the experiment described in [17] and achieved similar results using benchmark datasets.
2. In order to test the RARMT's generalisation ability and its robustness, the dataset collected as mentioned in objective 2, which consists of user profiles from two very distinct age groups will be used to run the experiments.
3. Algorithmic parameter values and information input selection for RARMT's Motion Transition recognition processor will be determined through experimentations. Different statistical features and information sources like the Barometer readings will used in this motion transition recognition component. As this work is new and not seen before, we will conduct intensive experimentations to see how they can potentially generate additional meaningful information which can more accurately recognize certain activities such as the *falling down motion*.

1.3 Scope and Limitations

1.3.1 Scope of Project

The scope of this project focuses primarily on the establishment of the *Robust Activity Recognition with Motion Transition (RARMT)* framework based on the ideas adopted from [17] and also on items as described in *Section 1.2*. The core of this project will be driven by intensive back-end research and experimentation in the field of Human Activity Recognition.

This project will not be covering front-end human activity recognition application development as well as the integration of the new framework to the existing web application developed in [4].

1.3.2 Tools

Hardware Sensors

Accelerometer readings will be used as the main sensory information input for the new framework in recognising basic human motions. Additionally, the barometer readings will be used as a second layer check to perform more reliable fall detection.

Applications

- *Android Application:* App will be installed into an android smartphone and used as a front-end data collection device for the sensory values
- *Java Eclipse:* Used to develop the data pre-processor to sample raw motion sensory data at defined fix frequency; and to develop high level motion recognition processor.
- *Matlab:* Used to develop the SVM algorithm, Feature Extractor, Grid Search and testing components
- *SQLite Browser:* Used to retrieve sqlite files containing sensory data from android smartphone

Algorithms

In this project, the Support Vector Machine [18] will be used as the classification model for the new activity recognition framework. Also, Data Pre-processing Logic and the Motion Transition Recognition Algorithm will be implemented as part of the new RARMT framework.

1.4 Report Overview

Section 1 introduces the background and objectives of this project.

Section 2 presents the literature overview of a number of related work done in the field of human behavior modelling.

Section 3 dives into details of a particular related work which was a final year project [4] completed by a previous student, to evaluate potential improvement areas that this project's framework can address.

Section 4 defines the dimensions that this project will be investigating for the identified problem.

Section 5 presents the new human activity recognition framework designed and developed by this project. This is done by first introducing the methodology involved and analysing on how they were applied to the problem on hand.

Section 6 presents the various experimentations conducted in this project. It also contains detailed results analysis and discussion.

Section 7 concludes the report with a summary of work done as well as suggestions for future work in this research area.

2. Related Work

Existing work on human activity recognition has built a strong foundation on the usage of smartphone sensors for data collection and also the construction of a user model for recognising activities such as sitting, standing, walking, running, etc. Using algorithms like the Hidden Markov Model (HMM) and Support Vector Machine (SVM), many research work managed to categorise these readings and interpret the activity performed with high accuracy.

In many of the work like [4, 10, 11, 19], experimentation and data collection is performed strictly using a device that is at a fixed position and orientation on the body. In [4] and [19], data collection from the smartphone has to be carried out using a smartphone holder on the waist with exact setup conditions throughout all training and testing routines in order to achieve high performance (more to be discussed in *Section 3.1*). Furthermore, the location to place these sensors are usually strategically selected to be at specific locations where the intrinsic characteristics of the target activities can be well captured. For instance, a possible location can be on the shoulder [10] to detect activities like walking and climbing stairs; or somewhere near the pelvic region [11].

Moreover, many related work has also highlighted the concern of the inability to obtain a broad and diverse range of data sets. Most of the existing work done [4, 5, 6] uses very limited sample data sets that fail to capture a sufficient range of different population types, particularly data from elderly users. This raises the concern of the inability to sufficiently test for a classification model's generalisation capability to the fullest extent, since people of different demographic groups have very different activity styles.

In the above mentioned cases, we can see that many of the research done in the arena is conducted within fairly stringent experimental set-up and conditions. The case of unfixed and diverse experiment setup conditions, where the position and orientation of sensor device is varying and data collection covers a wide range of population groups, is still yet to be considered by most.

Unlike most of the prior work done, work presented in [17] has contributed tremendously in its effort to identify the effect of the varying position and orientation issues in accelerometer-based human activity recognition. The authors of [17] has achieved high performances in the recognition of human activity in natural settings where the mobile phone’s position and orientation are varying, depending on the position, material and size of the hosting pocket.

In this project, we aim to adopt the concepts in [17] and bring the investigation further to address the case of human activity recognition in handling user profiles belonging to diverse demographic groups.

Furthermore, few related literatures have expanded their work to cover complex motion transition recognition. Even in [17], research and experiments were only performed on basic low level motion (e.g. Sitting, Lying, Walking). Even preliminary existing work done in the field of transition recognition are still largely based on raw sensor readings which tend to be less robust. For instance, work in [22] depends on a complicated hierarchy binary structure to make prediction on a set of motion transitions based on raw accelerometer readings. We believe in the importance of building a widely applicable foundation framework that can offer extensibility to other human activity recognition related innovations, thereby creating a value-add to offer meaningful motion transition information (e.g. Falling down). As such, in this project we aim to extend the capability of the framework to handle high level complex motion transitions. We will propose a novel way of utilising accelerometer’s Signal Magnitude Area and barometer readings for high level motion transition recognition. Work presented in [6] has shown the usefulness of using the Barometer to detect activity like climbing the staircases and taking the elevator, where there is a change in altitude from sea level. This serves as a promising line of investigation to enhance the accuracy of activity recognition more accurate especially for important ones like falling down motions.

3. Analysis of Existing Work Done

3.1 Background on Existing Work

This project will be a continuation of effort from a previous work [4], which implemented activity recognition using the Hidden Markov Model (HMM). Multiple sources of raw sensory data (i.e. accelerometer data, linear accelerometer data, gyroscope data, magnetometer data) were employed as inputs into four corresponding HMMs and prediction of activity was done via a voting mechanism. In [4], activities are categorized into lower-level activities and higher-level activities (see Table 3.1). Lower-level activities will be determined directly using the vote results of the four HMMs while the simple higher-level activities will be predicted via a rule-based system together with Signal Magnitude Area (SMA) on top of the HMMs' vote results.

Table 3.1: Higher-level activity and Lower-level activity categorization excerpted from [4]

Higher-Level Activity	**Lower-level activity**	
	From	**To**
Standing Up	Sitting	Standing
Falling Down	Running	Lying
	Walking	Lying
	Standing	Lying
Sitting Down	Running	Sitting
	Walking	Sitting
	Standing	Sitting

3.2 Evaluation of Existing Work

An evaluation of [4] has led to the following conclusions regarding the various aspects of the activity recognition model implemented:

3.2.1 Robustness of Model used

The previous model [4] employs a set of Hidden Markov Models, which will be fed with raw sensory data. Using raw sensory data as an input for classification consequently results in an activity recognition model that is extremely sensitive to noise. In [4], the experiment and research was performed in a strictly controlled environment where a smartphone holder must be used for data collection and the model is trained and tested on only a single user.

Performance of the HMM model highly hinges on the cleanliness of the data used and as a result of these constraints, robustness of the activity recognition model is compromised:

- Smartphone (sensor device) has to be at a *fixed orientation and location* at all times
- Smartphone must not have direct contact with the body as noise introduced by the user's breathing action has proven to be too significant for the model to handle
- Data collection has to be performed in a stringent manner with no fidgetiness
- Recognition model in [4] has poor or little generalization capability across multiple user profiles
- Performance is low for High level motion transition. Accuracy score of only 76.25% is reported in [4] for simple high level motion transition detection and it is unable to handle complex transitions such as Walking to Running.

A constrained recognition model like this has very limited ability to provide a strong foundation for extension to future application in the field of human behavior recognition.

3.2.2 Usability of the model developed

Previous work in [4] has imposes a number of restrictions in their experiment work to help ensure only extremely clean data is collected. However, this comes at the high cost of usability, making the model less realistic to be used as a solution.

- Data collection can only be triggered by the computer and conducted in short time intervals. Due to the model's sensitivity to noise, data collection period must be kept short. This results in a cumbersome and inefficient data collection procedure.
- Smartphone cannot be placed anywhere on the body, it has to be strictly at the exact position on the waist at the exact angle. Also, it cannot be in direct contact with the user's body as breathing activity introduces noise to the data collected. This restriction causes extreme inconvenience, hence reducing usability of the model.
- Recognition model cannot be generalized across multiple users.

In this project, we aim to circumvent the above mentioned concerns especially regarding working with sensory noise and imperfect data, by developing a new framework that is robust and has high usability.

4. Problem Investigation Dimension

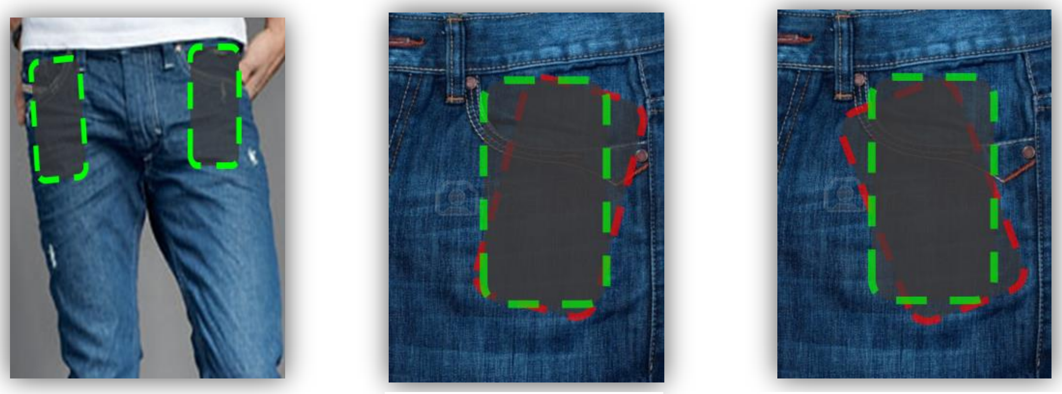
In *Section 2 and 3*, we reviewed a number of related literatures and discussed some of the limitations and problems that these work have. In this section, we set up the investigation dimensions for the mentioned problems that this project aims to overcome. The defined problem investigation space will be as follows:

1. Smartphone location and orientation
2. Human physical activities
3. User population age groups

Smartphone Location and Orientation

Two common pocket locations present in people's daily apparels that are the common preferred place to keep mobile phones are selected to be used for this project. The two locations chosen are the two front pockets on the trousers, as shown in Figure 4.1(a).

Also, we will find out if the activity recognition model designed in this project will be able to handle slips and slight angle tilt of the smartphone in users' pockets as shown in Figure 4.1(b) and 4.1(c).



(a) *(b)* *(c)*

Figure 4.1: Pocket locations and Phone Orientation

Human Activities

We identified five basic human activities: Lying Down; Sitting; Standing; Walking and Running. Based on these five basic low level motions, we are able to add another layer of analysis to recognize higher-level transitional activities like: Standing Up; Sitting Down and

Falling Down. The full list of motion transitions addressed by this project can be found in Table 6.21 in *Section 6.4.2*.

User Population Age Group

Most existing work done reviewed in *Section 2* have been performed on a narrow range of population age group. In this project we target to experiment on users from two distinct age groups (i.e. youth and elderly). The youth group consists of users age 18-30 years old, while the elderly group consists of users age 65-80 years old. We aim to test the ability of the model to generalize and predict motions of age group not present in the training data. For example, to train the model based on the youth user profiles and test on elderly user profiles.

In a nutshell, this project aims to develop a human activity recognition model that is able to generalize across varying smartphone location and orientation; varying physical human activity and varying demographic of user profiles. On top of that, performance of the proposed framework in terms of its robustness and generalization capability across these investigation dimensions will be tested via an extensive set of experimentations.

5. Activity Recognition Methodology

5.1 Overview of the Robust Activity Recognition with Motion Transition (RARMT) framework

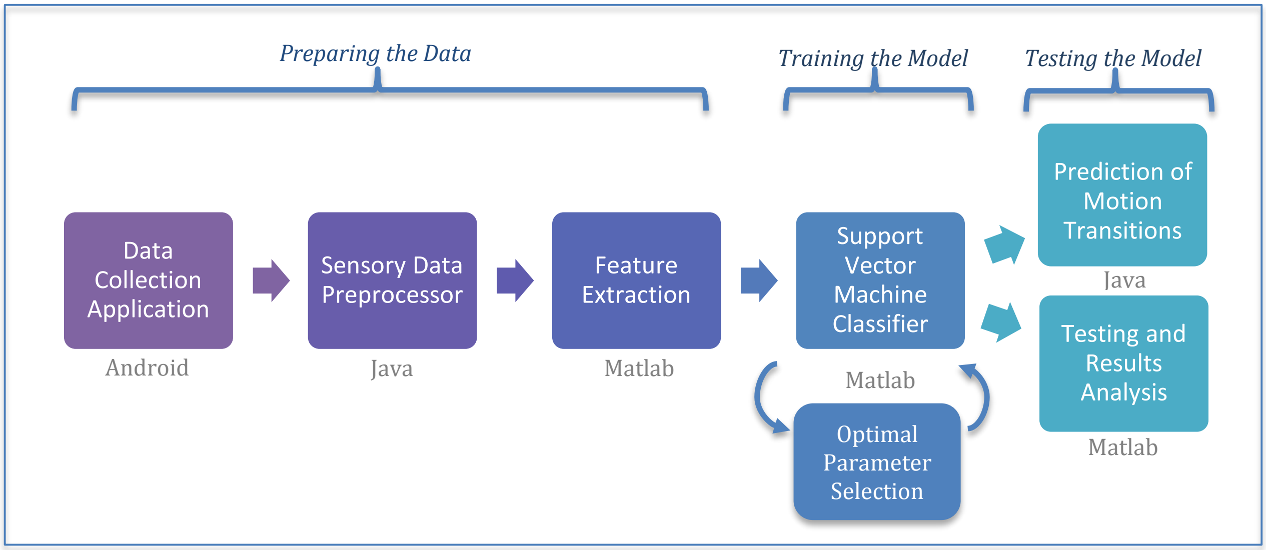

Preparing the Data
Training the Model
Testing the Model
Data Collection Application
Android
Sensory Data Preprocessor
Java
Feature Extraction
Matlab
Support Vector Machine Classifier
Matlab
Optimal Parameter Selection
Prediction of Motion Transitions
Java
Testing and Results Analysis
Matlab

Figure 5.1: The RARMT Framework

The *Robust Activity Recognition with Motion Transition (RARMT) framework* is made up of the following seven main components:

Data Collection Application: An android application that utilizes the Android Sensor Framework to collect raw activity sensory data for both training and testing purposes.

Sensory Data Preprocessor: Reads in raw sensory data and samples data at different fixed frequency.

Feature Extraction: Derives informative features in different domains that aims to help facilitate further learning and generalization steps.

Support Vector Machine (SVM) Classifier: Make use of the derived features and perform classification to predict basic low level human activities (e.g. Walking, Standing, Sitting).

Optimal Parameter Selection: Search for more suitable Cost and Gamma parameter values for the SVM classifier.

Prediction of Motion Transitions: Reads in the predicted low level motions and determine whether there exists any motion transitions (e.g. Falling Down, Standing Up).

Testing and Results Analysis: Employs the confusion matrix and evaluation metrics to determine performance of the model.

The core methodology of RARMT's low level activity recognition is adopted from [17], whereby a Support Vector Machine classifier is designed to handle the case of varying sensor device's location and orientation, resulting in a classification model that is robust and has high generalization capability. More details about how [17]'s work is used in this project can be found in *Section 5.4 and 5.5.*

In addition, other than being able to recognize only low level basic motions (e.g. Standing, Sitting), one of RARMT framework's highlight is in its ability to detect high level motion transitions (e.g. Falling Down) using a novel methodology. This processing logic will be designed and implemented in this project. More details about how the algorithm details of the logic will be discussed in *Section 5.6*.

5.2 Data Collection Application Using Native Android Sensor Framework

The android data collection application is implemented to collect two different kind of sensory information from a smartphone- the Accelerometer data and Barometer data. Data is collected at a suggested rate of about 0.2 seconds (application uses SENSOR_DELAY_NORMAL). In order to make the data collection process as efficient and user-friendly (e.g. battery usage, convenience) as possible, the android application is designed to allow data recording and labelling to be triggered directly from the application. Before data recording, users will be prompted to select the activity they are performing and the configurations used in that procedure. After completing the data collection setup, users will place the smartphone according to a preferred pocket (i.e. left/ right front pants' pocket) and orientation and wait for the sound signals.

Data that occurs at the start and end of the collected data file per data collection procedure is commonly more prone to corruption. This is because users require time allowance to put the smartphone in their pockets and also to take them out. To cater for this concern, the android application is designed to produce three sound signals (see Figure 5.2).

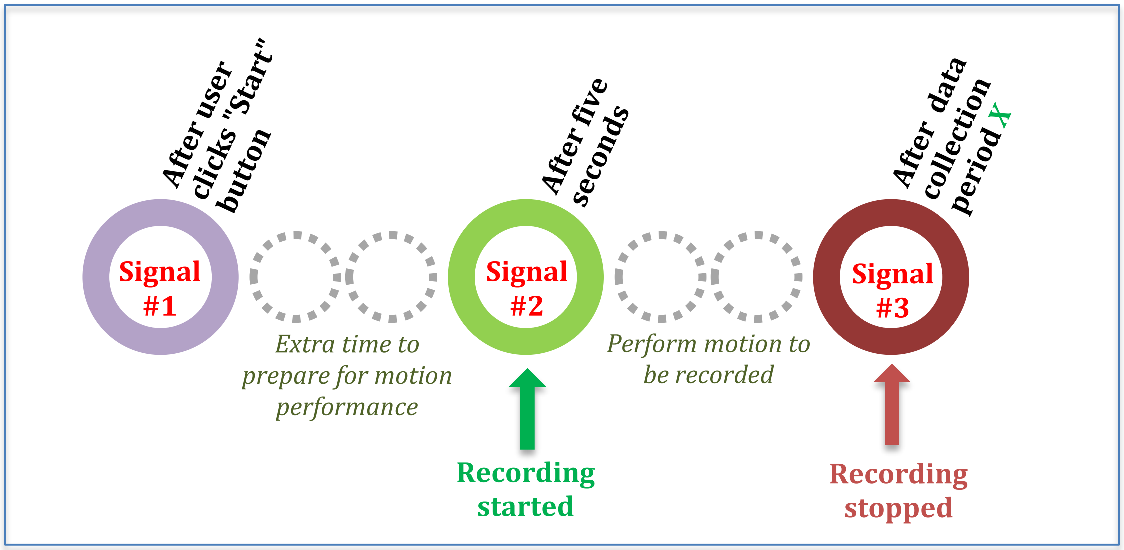

After user clicks "Start" button
After five seconds
After data collection period X
Signal #1
Signal #2
Signal #3
Extra time to prepare for motion performance
Perform motion to be recorded
Recording started
Recording stopped

Figure 5.2: Android Data Collection Application Sound signals

Signal #1: To acknowledge that the user has pressed the “Start Recording” button.

Signal #2: Happens five seconds after Signal #1. To inform user that recording has started.

Signal #3: Happens after data collection period. To inform the user that he can remove the phone from the pocket thereafter.

These sound signal reduces the possibility of collecting corrupted data and also improve the usability of the application. Data collected in the Smartphone will be stored within the phone as a SQLite database file. A SQLite browser will be able to retrieve the database file and export the data according as a csv file for data preprocessing.

5.3 Sensor Data Preprocessing

5.3.1 Limitation of Android Sensor Framework

It is currently not possible to define and determine the rate at which the android sensor framework is sending sensor events to an application. The sampling rate defined in the application using Android’s Sensor Manager can only serve as a suggestion and is not guaranteed because the android system and other applications can alter this rate [21]. Hence, there is a need for a Data Preprocessing component (see Figure 5.3) in the *Robust Activity Recognition with Motion Transition (RARMT) framework*. There will be two main steps for data preprocessing.

5.3.2 Sensor Data Preprocessor

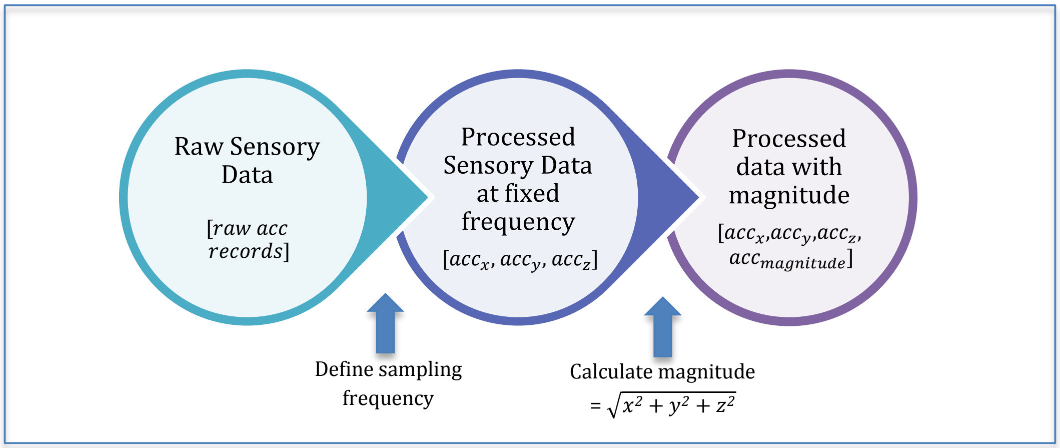

Raw Sensory Data
[raw acc records]
Processed Sensory Data at fixed frequency
$[acc_x, acc_y, acc_z]$
Processed data with magnitude
$[acc_x, acc_y, acc_z, acc_{magnitude}]$
Define sampling frequency
Calculate magnitude
$= \sqrt{x^2 + y^2 + z^2}$

Figure 5.3: Human Activity Recognition Data Preprocessor

<u>Step #1: Sample Raw sensory data at defined frequency</u>

The Sensor Data Preprocessor is built as a java program to read in comma-separated values (csv) files and process them to generate sensory data sampled at the user defined frequency.

Let us assume that the sampling frequency (f_s) is set at 10Hz (0.1 sec interval). There are two basic functions that the preprocessor performs to manipulate the raw sensory data- *duplicate* and *ignore*. See Figure 5.4 for illustration.

- *Ignore function:* Raw sensory data records are ignored if there are more than one data point within the same defined time interval (as illustrated by the grey dotted lines). Only the record that was most recently-read will not be ignored (e.g. between record "d" and "e", "e" is not ignored)
- *Duplicate function:* Raw sensory data records are duplicated from the most recently-read record if there are no data points within the same defined time interval.

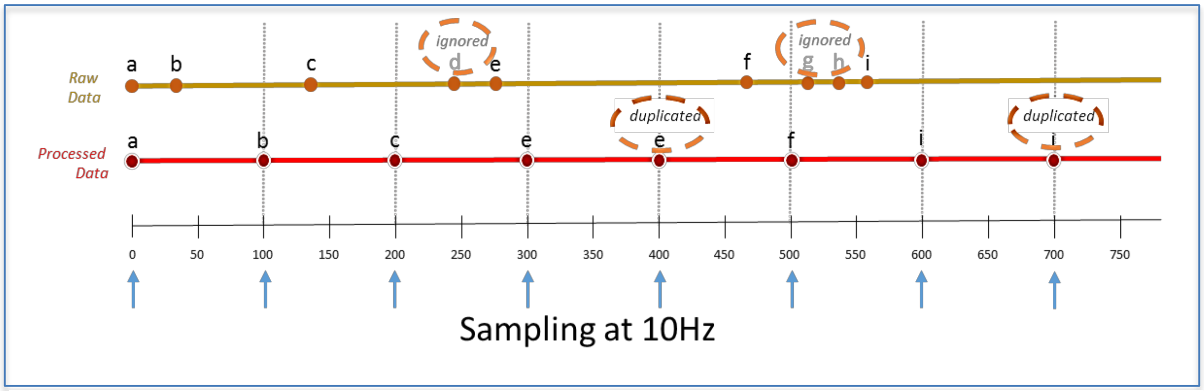


Figure 5.4: Sensor Preprocessor Example

Step #2: Extract Magnitude of the Raw Sensory Data

Work done in [17] proved the effectiveness of introducing an additional magnitude vector to sensor readings that are sensitive to directions.

Using accelerometer as an example, it was justified that since acceleration magnitude is a measure of quantity of acceleration and has no direction, it is insensitive to the orientations of the smartphone. In order to relieve the influences of the device orientation (exact orientation of acceleration is unknown), extracting the magnitude of sensor readings sensitive to directions allows the classification model to acquire much improved generalization capability. Figure 5.5 illustrates how the extracted magnitude justifies to be a discriminative feature in the experimental results achieved- as the dataset with magnitude consistently outperforms the dataset without magnitude.

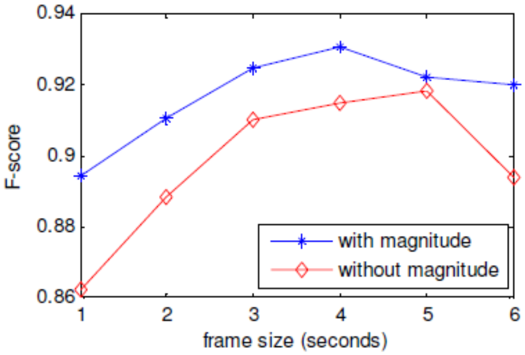


Figure 5.5: Performance improvement with 4th dimension to accelerometer data- excerpted from [4]

5.4 Feature Extraction

Features are derived values from raw sensory readings. Frames are windows of data that will be captured for each round of feature extraction. One resulting record after feature extraction will correspond to the set of features extracted for one data frame. In Figure 5.6, we can see an illustration of how windows of frames are first defined and then features can be extracted from these defined windows.

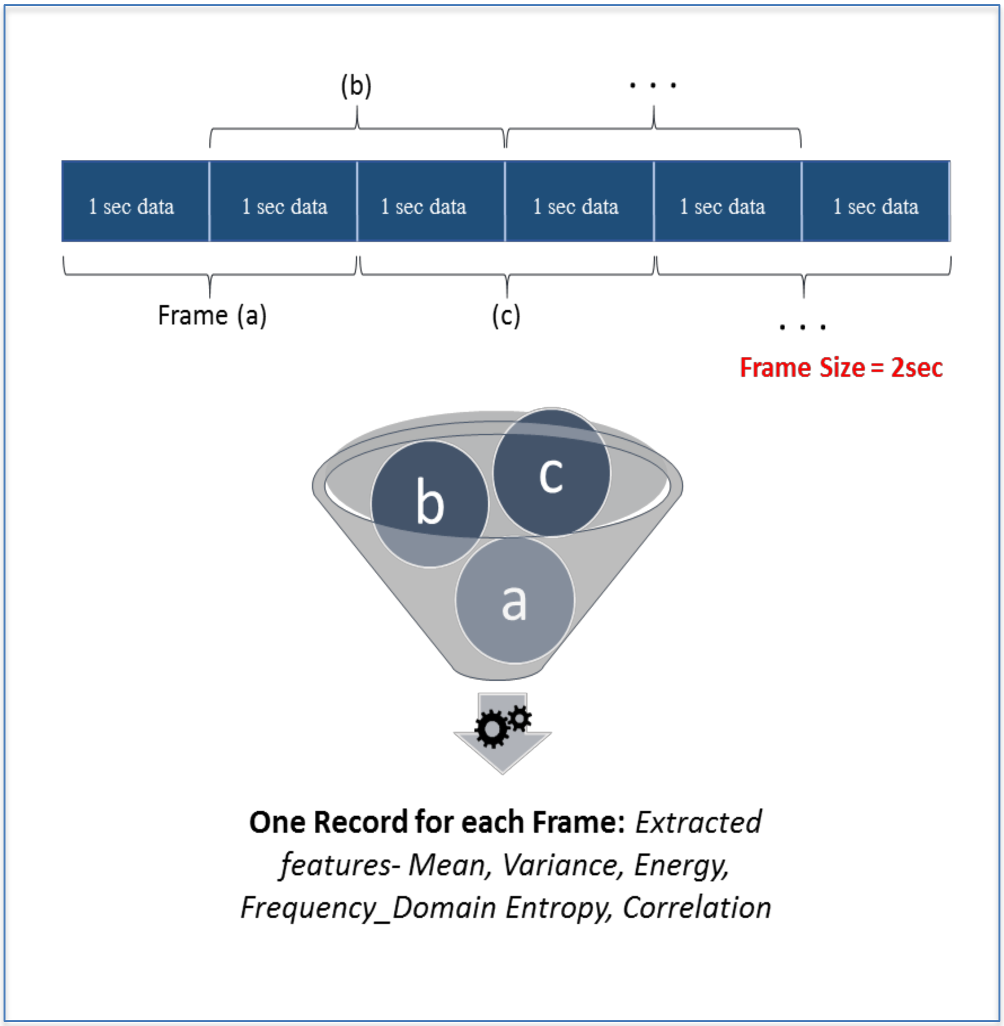

(b)
. . .
1 sec data
1 sec data
1 sec data
1 sec data
1 sec data
1 sec data
Frame (a)
(c)
. . .
Frame Size = 2sec
b
c
a
One Record for each Frame: Extracted features- Mean, Variance, Energy, Frequency_Domain Entropy, Correlation

Figure 5.6: Feature Extraction Illustration

Step#1: Define Frames of data

The Feature Extractor will first define Frames of data. Referring to Figure 5.6, the processed sensory data will be divided into frames of size two seconds. In this project we will be using a half overlapping sliding window, hence as illustrated in the Figure 5.6, the data frames defined will be overlapped with one another.

By defining a larger frame size, the classifier will be able to eliminate short term noises as it takes information from longer time to make a classification decision. However, this comes with the consequence of taking too much time to detect an activity when the application is run in real time. This tradeoff will be further analyzed in *Section 6*.

Step#2: Extract Features from the Data Frames

After defining the Frames of data, the following time and frequency domain features will extracted:

- Mean
- Variance
- Energy
- Entropy
- Correlation Features

In this project, only the accelerometer data will be used ($acc_x, acc_y, acc_z, acc_{mag}$). Therefore, there will be 22 features in total:

Table 5.1: Features Extracted

Mean	$MEAN_X; MEAN_Y; MEAN_Z;\ MEAN_{mag}$
Variance	$VAR_X; VAR_Y; VAR_Z;\ VAR_{mag}$
Energy	$ENERGY_X; ENERGY_Y; ENERGY_Z;\ ENERGY_{mag}$
Entropy	$ENTROPY_X; ENTROPY_Y; ENTROPY_Z;\ ENTROPY_{mag}$
Correlation	$COR_{XY}; COR_{XZ}; COR_{YZ};\ COR_{Xmag}; COR_{Ymag};\ COR_{Zmag}$

mag: Magnitude

The extracted features described in Table 5.1 are partially recommended in [12-16] to be highly discriminative features for the classification of human activities, and we will be adopting these ideas into the project.

When selecting the number or type of sensory data to be analyzed and features to be extracted, it is important to consider the tradeoffs in the consequential increase of resource consumption by the model, which can compromise usability. There is no direct relationship between the amount of information used for classification and prediction accuracy, meaning that having a larger information set does not necessary lead to better performance. Hence, it is important to identify discriminative features to use for the new RARMT framework. In *Section 6.2.4*, we can see how the selection of discriminative features has helped the RARMT framework to achieve better results in the benchmark tests.

Step#3: Data Normalization

All extracted features have to go through normalization to prevent attributes in greater numeric ranges from dominating those in smaller numeric ranges. As suggested by [18], it is recommended to normalize data to the range of [0,1]. In this project, the MIN-MAX normalization is used for each feature (e.g. $MEAN_X, VAR_Z$) in both the training and testing dataset.

The formula is as follows:

$$v' = \frac{v - min_A}{max_A - min_A} \tag{1}$$

Where v' is the normalised value, v is the unnormalised value, min_A is the minimum value and max_A is the maximum value of the value range per feature.

Both min_A and max_A are variables that will be dependent on the training dataset used. This set of variables will be used for during feature extraction for the testing dataset as well, to ensure that normalisation is done based on the same scale. Hence, the resulting values of v' will always be in the range of [0-1] for the training dataset but not necessary for the testing dataset.

5.5 Model Training

5.5.1 Model Selection

In this project we will be using LibSVM [18] as the classification model. The Support Vector Machine in general offers one of the most robust and accurate methods among all well-known algorithms, and the LibSVM allows multi-class classification which is the essential task for this project.

The SVM aims to find the optimal solution (C, **γ**) to this problem for a given classification task:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{t} \xi_i$$

$$subject\ to \quad y_i(w^T \phi(\mathrm{x_i}) + b) \geq 1 - \xi_i \tag{2}$$

$$\xi_i \geq 0$$

Furthermore, the Radial Basis Function (RBF) will be used as the Kernel Function in this project:

$$\mathrm{K(xi, xj\ ) = exp}(-\gamma \left\|x_i - x_j\right\|^2), \gamma > 0 \tag{3}$$

The main rationale behind choosing the RBF kernel is that it nonlinearly maps samples into a higher dimensional space so that it can easily handle the case when the relation between class labels and attributes is nonlinear.

5.5.2 SVM Hyperparameter optimization

The Grid Search method is used for the optimal selection of the tradeoff parameter Cost (C) and the bandwidth Gamma (G) in the Support Vector Machine RFB kernel.

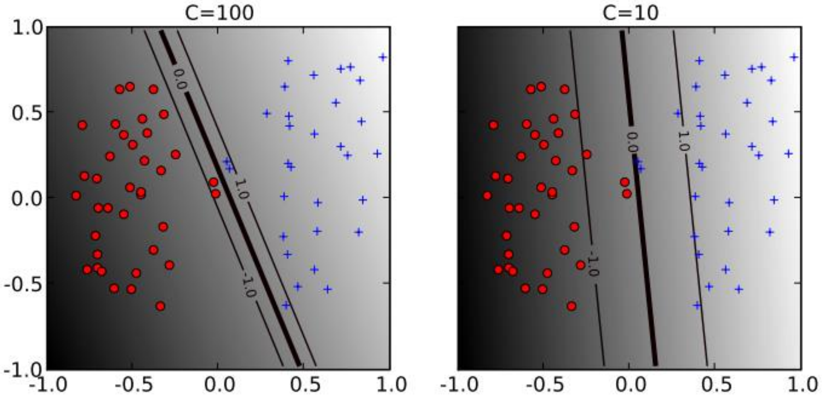


Figure 5.7: Cost Parameter Selection excerpted from [23]

Cost (C) is a trade-off between training error and the flatness of the solution. The larger the C value, the less the final training error will be. However, a high C will run the risk of losing the generalization properties of the classifier, because the model will try to fit as best as possible all the training points, including the possible errors of the dataset (see Figure 5.7).

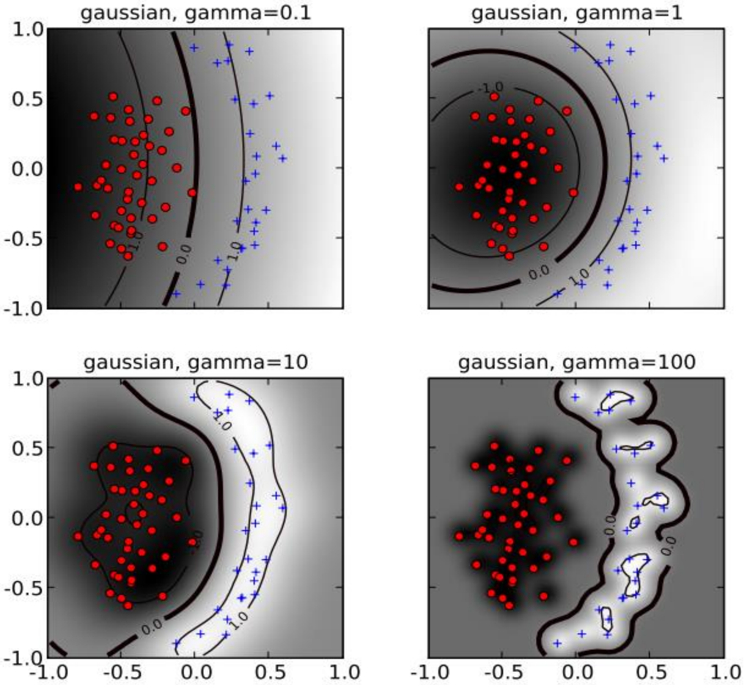


Figure 5.8: Gamma Parameter Selection excerpted from [23]

On the other hand, Gamma (G) is a kernel parameter that controls the width of the insensitive zone, which is used to fit the training data. It can also affect the number of support vectors used to construct the regression function. From Figure 5.8, we can observe that larger G values increases the flexibility of the decision boundary increases. However, this comes at a cost of over-fitting.

Hence, we can see that both C and G values affect the SVM model complexity in different ways. In order to obtain optimal performance, our framework must be able to determine the most suitable SVM parameters to use. This can be done when a grid search method is applied, in which given a set of parameter search range, the algorithm will do a grid search over the parameter space to find the optimal SVM setting. It will be guided by a performance metric measured by cross-validation on the training dataset.

However, the grid search method will only be effective when it has visibility of the user profiles in the testing dataset. In the case when the framework has no knowledge of the user profile at all, grid search will likely lead to lower performance due to over-fitting. This problem will be discussed further in the *Section 6.3.3*.

5.6 Motion Transition Recognition

5.6.1 Overview and Objective

The basic motion recognition model described in [17] has very limited extensibility to other possible human behavior recognition applications, which require more informative activity recognitions capability (e.g. motion transitions). It can only detect the basic low level motions such as Standing, Walking, Running, Bicycling, Ascending stairs, Descending stairs and Driving. Therefore in this project, the *Robust Activity Recognition with Motion Transition (RARMT)* framework is designed to be equipped with the ability to make complex higher level motion transition recognitions. RARMT's motion transition recognition processor is able to make predictions on transitions represented in the motion state diagram (see Figure 5.9). For instance, when a user transits from a sitting to a walking motion, the processor can infer the flow of action as- *"User stands up and walks".* This applies the same when a user transits from a walking to a sitting motion, which will be inferred as- *"User walks and sat down"*. Note that we determine a transition from Running to Lying or Sitting motion to be abnormal in a normal home-living scenario. This transition will be catered under Fall Detection (see Figure 5.10).

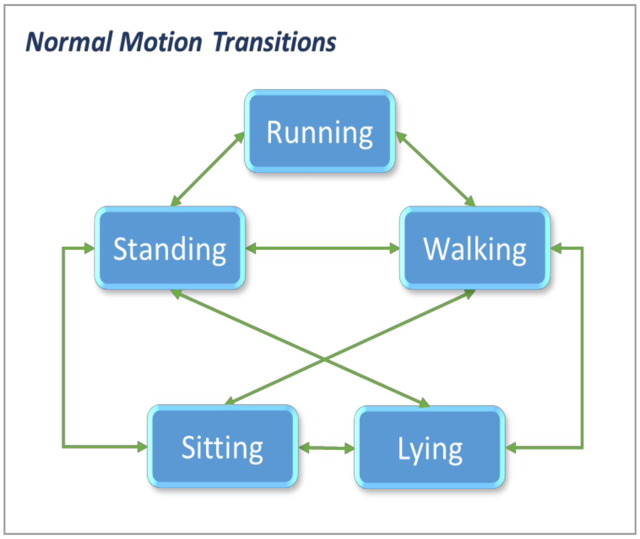


Figure 5.9: Normal Motion Transition State Diagram

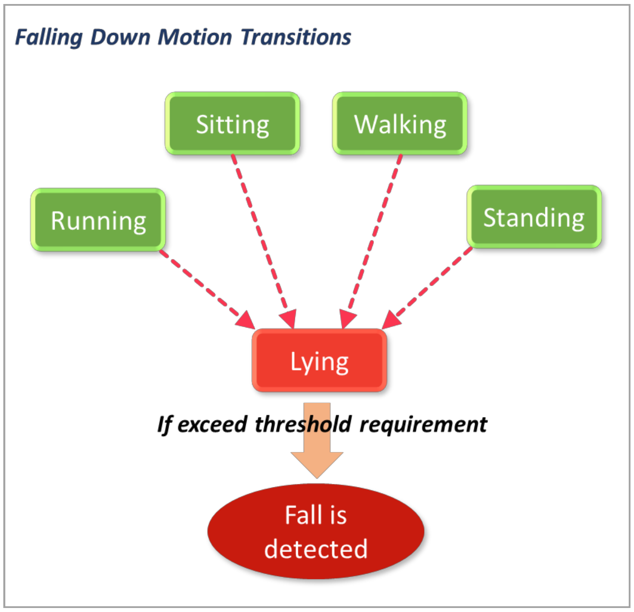

Falling Down Motion Transitions
Sitting
Walking
Running
Standing
Lying
If exceed threshold requirement
Fall is detected

Figure 5.10: Fall Motion State Diagram

One main highlight of RARMT's motion transition recognition is in its Fall Detection ability. In this case, a normal motion transition can be differentiate from a real falling motion with the usage of a threshold which will be determined from experimentations and analysis (see *Section 6.4*). For instance, when the processor detects that user's consequent state is a Lying motion, a possible fall has occurred and fall detection will be activated. The processor will investigate further by checking the other information inputs and determine if the threshold is exceeded, in order to finally determine if this is a real fall or a normal motion transition. Figure 5.10 summarizes the transitions catered for by the Fall Detection.

In the next section, we will elaborate on the logic flow of the algorithm designed to handle these motion transitions.

5.6.2 Algorithm Details

5.6.2.1 Assumptions

Three key assumptions are made in RARMT's motion transition processing logic:

1. Only Motion transition that happens within two seconds can be recognized
    - Human motion transitions tend to be transient in nature. Hence it is reasonable to assume that any transitions should happen within two seconds.
    - In the case when this does not happen, and the motion transition is nondeterministic, the system will identify that as a non-stable state.
2. System will not be able to make predictions for scenarios when more than one transition occur within two seconds
    - For example, user transits from sitting to standing and then standing to walking within two seconds. The system can only detect that user has transited from sitting to walking. Intermediate transitions will be ignored.
3. Users have to remain at a constant low level motion for at least two seconds
    - This means that users have to remain at steady low motion state (i.e. Lying, Sitting, Standing, Walking, Running) for at least two seconds in order for the algorithm to perform prediction accurately.

Note that the two seconds periods mentioned in this section translates to three of the *data rows* as referred by the algorithm in the rest of this section. This is due to due to the 50% overlap of raw data during RARMT's feature extraction.

5.6.2.2 *Logic Flow*

Figure 5.10 below shows the high level logic process flow of the algorithm:

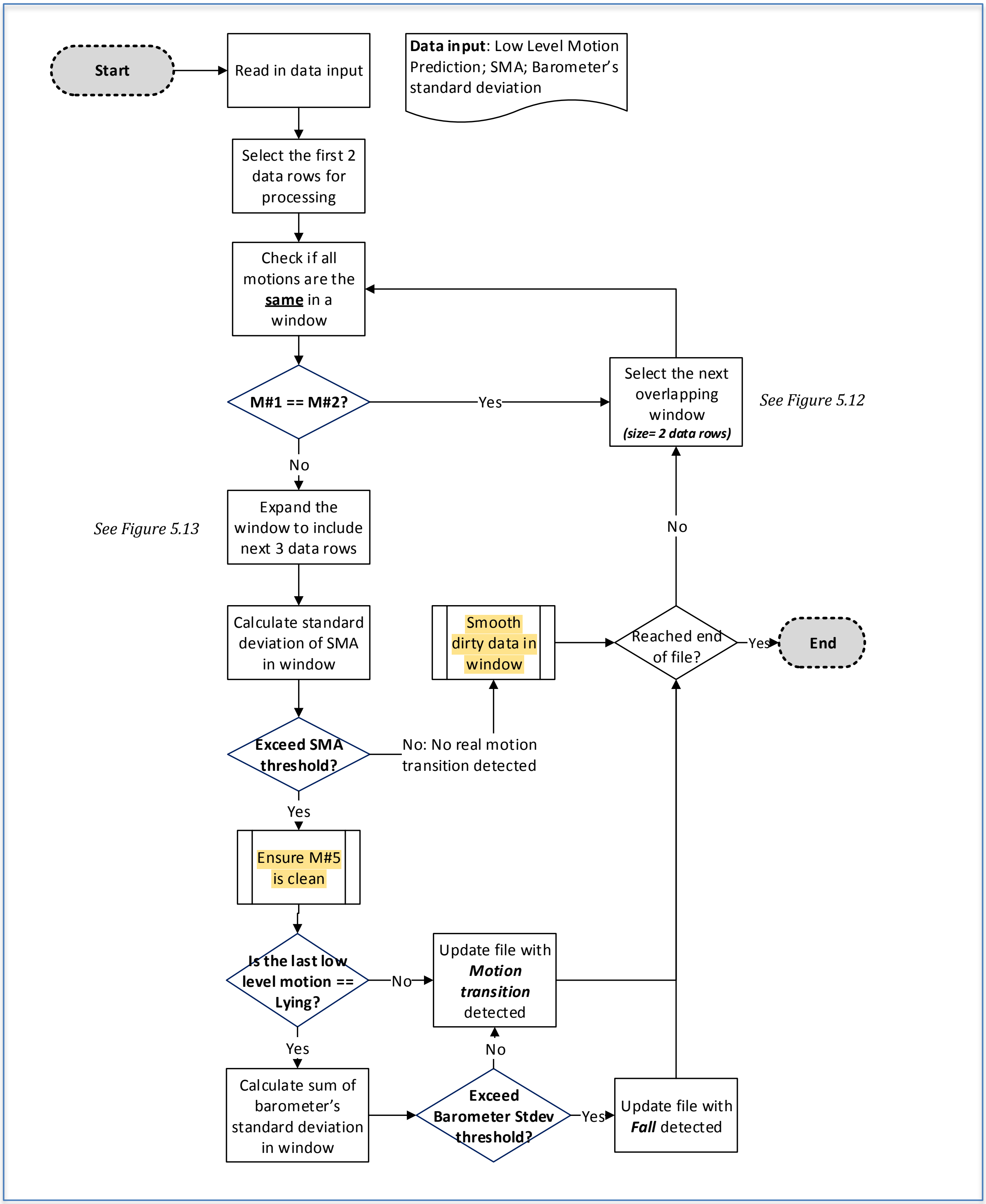

Start
Read in data input
Data input: Low Level Motion Prediction; SMA; Barometer's standard deviation
Select the first 2 data rows for processing
Check if all motions are the same in a window
M#1 == M#2?
Yes
Select the next overlapping window (size= 2 data rows)
See Figure 5.12
No
Expand the window to include next 3 data rows
See Figure 5.13
Calculate standard deviation of SMA in window
Smooth dirty data in window
Reached end of file?
Yes
End
No
Exceed SMA threshold?
No: No real motion transition detected
Yes
Ensure M#5 is clean
Is the last low level motion == Lying?
No
Update file with Motion transition detected
Yes
Calculate sum of barometer's standard deviation in window
No
Exceed Barometer Stdev threshold?
Yes
Update file with Fall detected

Figure 5.11: Motion Transition Recognition Algorithm Logic Flow

Flow Descriptions

i. Data input is a consolidation of information coming from RARMT SVM's *low level motion prediction labels* and RARMT Feature Extraction of accelerometer's *Signal Magnitude Area* as well as barometer's *Standard Deviation* (see *Section 5.6.2.2*).

ii. The input file will first be processed 2 rows at a time (see Figure 5.11 in *Section 5.6.2.1.1*).

iii. This will happen until the processor discover a difference in motion label between the two consecutive data rows [M#1 != M#2].

iv. When this happens, the processor will expand the window to include the next three data rows (see Figure 5.12 in *Section 5.6.2.1.1*).

v. The processor will calculate the Standard Deviation of all SMA values in that window.

vi. A real motion transition is determined if the calculated Standard Deviation value exceeds the established SMA threshold.
   - If no real transition is detected, data will be smoothed (see *Section 5.6.2.1.2*) and the process repeats - go to step ii.
   - If a motion transition is detected, the processor continue to check if the last motion (M#5) in the window is clean (see *Section 5.6.2.1.3*).

ii. To perform motion transition recognition, we follow a rule table.
   - If the last motion M#5 == Lying, there is a possible Fall Down. We will proceed with Fall Detection.
   - If the last motion is not Lying, we can move on to update the input file with the corresponding normal Motion Transition Label.

vii. To perform Fall Detection, we look at the Barometer's Standard Deviation information in the input file. We will calculate the sum of all that value in the window.

viii. If the calculated sum of Barometer's standard deviation exceed the established threshold, a Fall Down motion is detected and we will update the input file with the Falling labels.

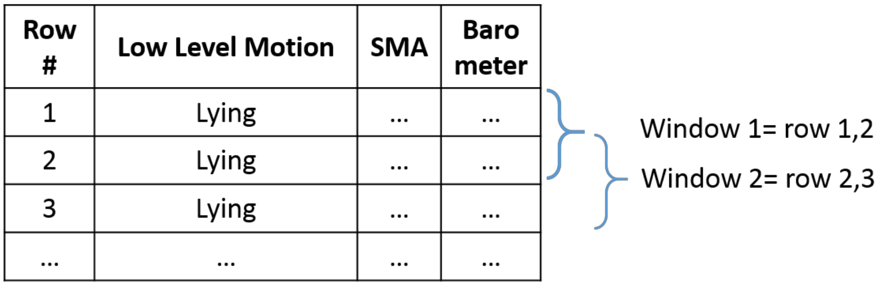
ix. Else, we will update the input file with the normal Motion Transition Labels.

x. If the processor has not reached the end of file, it will continue to step ii.

In the following sections, we will go into detail on how the algorithm select its processing window, how Smoothing is achieved and also how it ensures that Motion#5 is clean.

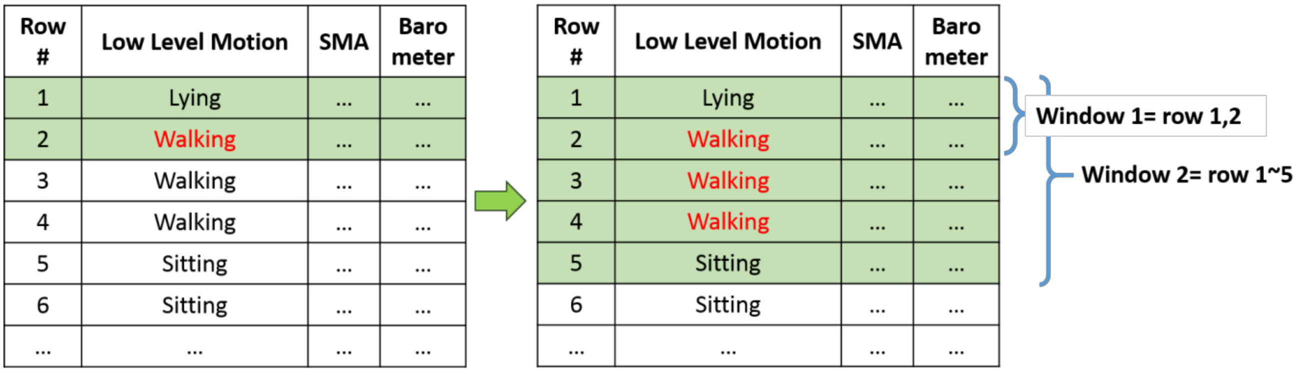
5.6.2.2.1 Window Selection

Default window size of two data rows are used from the start to compare between consecutive low level motion prediction labels for immediate identification of possible transition. In Figure 5.12, we can see how overlapping windows are selected for this purpose.



Row #	Low Level Motion	SMA	Baro meter
1	Lying	...	...
2	Lying	...	...
3	Lying	...	...
...	...	...	...

Figure 5.12: Motion Transition Algorithm- Selecting overlapping windows (size=2rows)



Row #	Low Level Motion	SMA	Baro meter
1	Lying	...	...
2	Walking	...	...
3	Walking	...	...
4	Walking	...	...
5	Sitting	...	...
6	Sitting	...	...
...	...	...	...

Row #	Low Level Motion	SMA	Baro meter
1	Lying	...	...
2	Walking	...	...
3	Walking	...	...
4	Walking	...	...
5	Sitting	...	...
6	Sitting	...	...
...	...	...	...

Figure 5.13: Motion Transition Algorithm- Selecting overlapping windows (size=5rows)

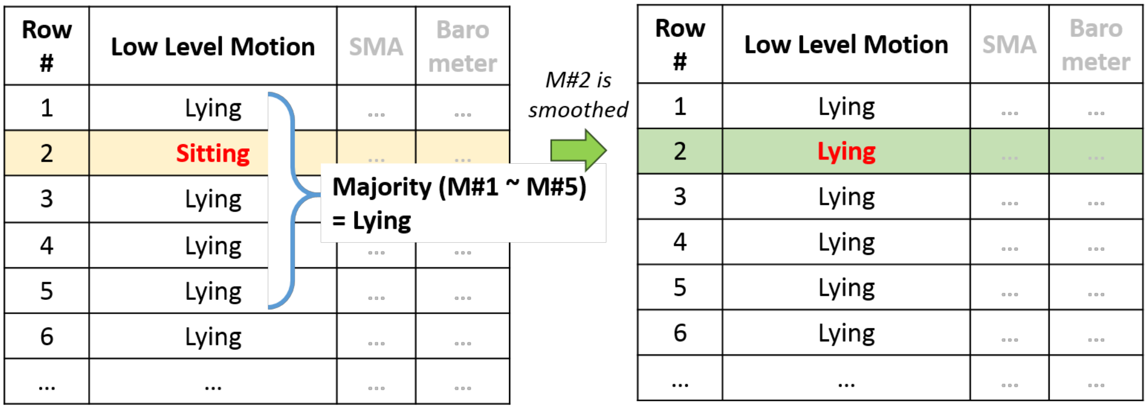
However, when the algorithm detects discrepancies in any two consecutive motions (see Figure 5.13), it will expand the window to include three following data rows into the processing window. Hence, when the transition logic is activated, processing window size becomes five data rows instead of two. The decision to set the processing window size at five data rows is based on the assumption made in this algorithm that all transitions will occur within two seconds, which translate to three data rows.

From Fig 5.13, we observe a motion transition from Lying → Sitting, in which the three middle "Walking" records in row 2 to 4 are transition motions. In this case as according to the assumptions made in *Section 5.6.2.1*, since transition happens within three data rows, the algorithm can recognize this transition and update the rows as transition motions.

5.6.2.2.2 Smoothing Function

Data smoothing is required when the algorithm detects a discrepancy in motion labels between consecutive data rows and determines that there is no actual motion transition happening later on in the flow. A voting by majority mechanism will be employed in the smoothing function.

For example, (see Figure 5.14) upon detecting a discrepancy between M#1 and M#2, the algorithm expanded the processing window to include (M#1~M#5). When the algorithm determined that there is no real motion transition, it will proceed to smooth the data in M#2. In this case, since majority of the motion labels from the window is "Lying", we will update M#2 with the Lying motion.



Row #	Low Level Motion	SMA	Baro meter
1	Lying	…	…
2	Sitting	…	…
3	Lying	…	…
4	Lying	…	…
5	Lying	…	…
6	Lying	…	…
…	…	…	…

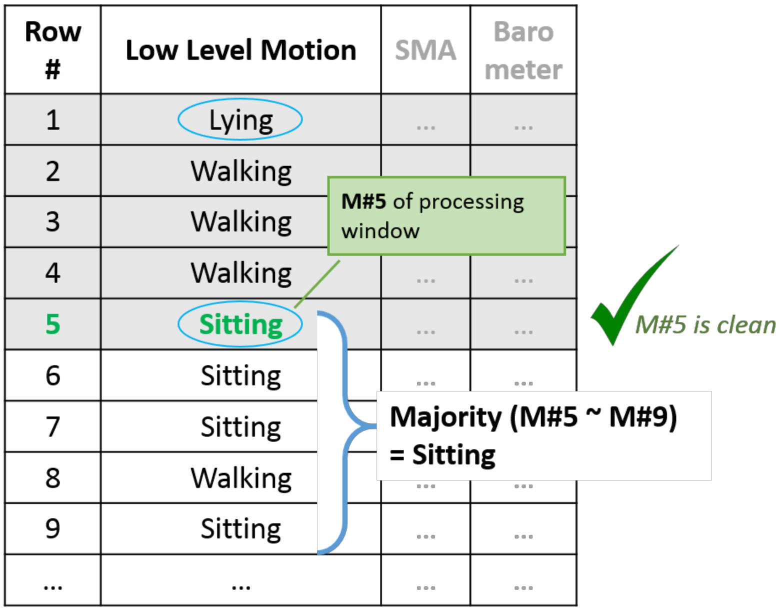
Row #	Low Level Motion	SMA	Baro meter
1	Lying	…	…
2	Lying	…	…
3	Lying	…	…
4	Lying	…	…
5	Lying	…	…
6	Lying	…	…
…	…	…	…

Figure 5.14: Smoothing Function Example

5.6.2.2.3 Cleaning Motion#5 Function

After the algorithm has confirmed that there is a motion transition, it will proceed to verify if motion#5 of the processing window is clean before it can make the final transition prediction. To determine the integrity of the motion, we use the same vote by majority mechanism mentioned in the previous section. The 5th motion will be cleaned up as according to the vote results.

Figure 5.15, shows an example where the 5th motion is clean (vote results == M#5). In this case, nothing will be done. However, in Figure 5.16 we see another example where the 5th motion is dirty (vote results != M#5). In this case, the algorithm will clean up M#5 by correcting it with the vote results label.



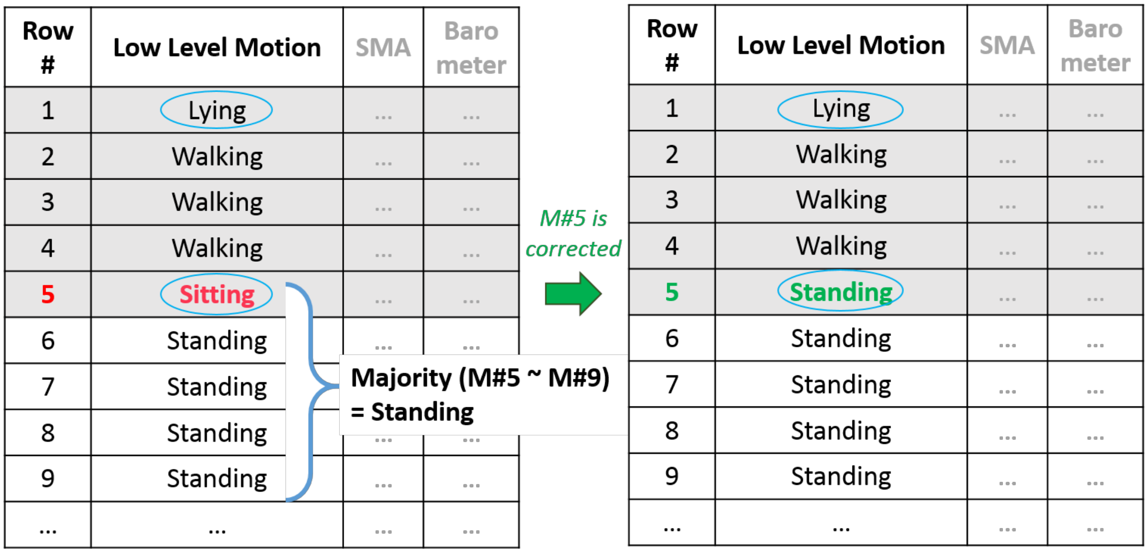
Row #	Low Level Motion	SMA	Baro meter
1	Lying	...	...
2	Walking		
3	Walking		
4	Walking	...	...
5	Sitting	...	...
6	Sitting	...	...
7	Sitting		
8	Walking	...	...
9	Sitting	...	...
...	...	...	...

M#5 of processing window

Majority (M#5 ~ M#9) = Sitting

M#5 is clean

Figure 5.15: Clean Motion#5 in processing window



Row #	Low Level Motion	SMA	Baro meter
1	Lying	...	...
2	Walking	...	...
3	Walking	...	...
4	Walking	...	...
5	Sitting	...	...
6	Standing	...	...
7	Standing		
8	Standing	...	...
9	Standing	...	...
...	...	...	...

Majority (M#5 ~ M#9) = Standing

M#5 is corrected

Row #	Low Level Motion	SMA	Baro meter
1	Lying	...	...
2	Walking	...	...
3	Walking	...	...
4	Walking	...	...
5	Standing	...	...
6	Standing	...	...
7	Standing	...	...
8	Standing	...	...
9	Standing	...	...
...	...	...	...

Figure 5.16: Dirty Motion#5 in processing window

In the following sections we will discuss the data inputs used (*Section 5.6.2.2*) and also the threshold values set in the algorithm (*Section 5.6.2.3*).

5.6.2.2 Data input

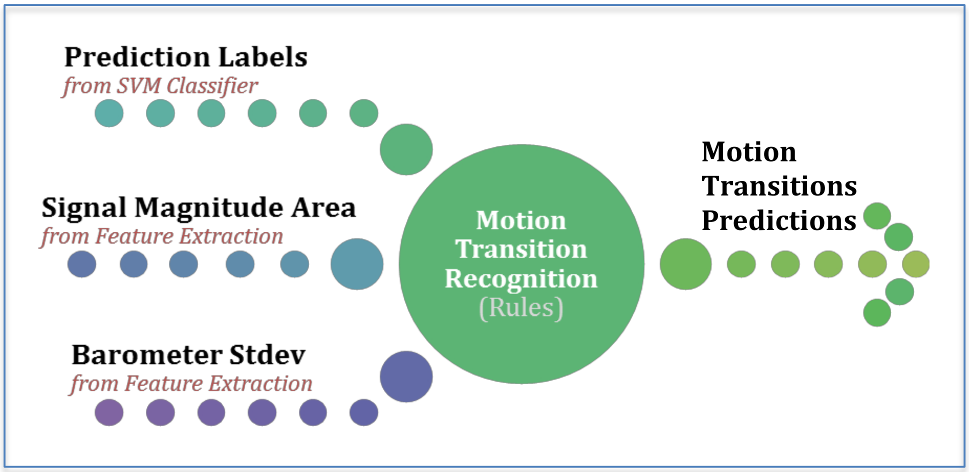


Figure 5.17: Motion Transition Recognition Algorithm Data input

<u>Low Level Motion Prediction Labels</u>

This information is obtained from the SVM classifier's prediction labels, where low level motions (i.e. Lying, Sitting, Standing, Walking, Running) are recognized. Based on these labels, the motion transition recognition processor can determine the correct motion transitions if any.

These predictions will be performed using frame size of one second. Due to the transient nature of motion transitions, the system will be expected to recognize motions within one second frame size in order to give robust prediction of motion transitions.

<u>Signal Magnitude Area (SMA)</u>

This information is obtained by using the RAMRT's Feature Extraction component (*Section 5.4*). Similarly as mentioned above, this feature will be extracted based on one second frame size. With that, the corresponding SMA values will be attached to each Low Level Motion Prediction Labels in the data input file.

The SMA values serve as a reliable differentiator between a stable state and an active state. Hence, it helps the algorithm to more accurately determine if an actual motion transition occurred, or if the user is at stable motion state. More information on why and how the SMA values are used can be found in *Section 6.4.3*.

Barometer Standard Deviation

Similarly to SMA, this information is obtained using Data Pre-processing and Feature Extraction. Raw barometer readings will be sampled at the same frequency and time interval as its corresponding accelerometer readings as discussed in *Section 5.3*. Next, in Feature Extraction, the standard deviation of the barometer readings within one second time frame will be calculated. This values will be attached to the each Low Level Motion Prediction Labels in the data input file.

The barometer sensor has the ability to detect an individual's altitude from the sea level. Hence when a fall is detected, the barometer readings can serve as a useful second layer check to verify the prediction, showing great promise in improving fall detection accuracy. More information on why and how the barometer values are used can be found in *Section 6.4.3*.

These three information sources will be consolidated and be fed into RARMT's Motion Transition Recognition component for motion transition predictions.

5.6.2.3 Threshold Values Selection

There will be two threshold values used in this algorithm- **SMA threshold value** for differentiating between a real motion transition and noise; and **Barometer threshold value** for differentiating between a fall and normal motion transition. These two threshold values will be determined via experimentations and will be discussed in *Section 6.4.4*.

6 Experimentation and Analysis

The *Robust Activity Recognition with Motion Transition (RARMT)* framework is established with the main goal of recognizing activity robustly, and being able to adapt effectively to different dataset profiles. The framework's recognition accuracy does not hinge on the cleanliness of data and is able to have high generalization capability across a diverse range of user profiles. Furthermore, RARMT must be able to detect motion transition and perform fall detection with high reliability. With these goals in mind, we established three main sets of experimentation in this project for the purpose of testing the performance of the RARMT framework.

1. **Performance Testing using Benchmark Datasets** *(Section 6.2)***:** This experimentation focuses on further proving RARMT's ability to achieve high performance results using externally sourced datasets as benchmarks.
2. **Performance Testing using Dataset on Different Age Groups (DAG)** *(Section 6.3)***:** This experimentation focuses on constructing a rich human motion sensory dataset covering two distinct age groups (i.e. youth and elderly), which also captures human behavior in their most natural environmental settings. This is then followed up with intensive testing on RARMT's performance in handling diversified data sources like the DAG.
3. **Motion Transition Recognition Performance Testing** *(Section 6.4)*: Unlike the previous two sets of experiment, this experimentation focuses on the RARMT framework's performance in recognizing high level motion transitions and doing fall detection. It aims to test the implemented system's ability in handling cases of motion recognition in a more realistic scenario, where data is collected in different time intervals and can consist a series of different low level motions.

In this section, we will first introduce the methodology behind this project's experimentation effort. Next we will address each of the three items mentioned above separately and report the performance test results of the RARMT framework.

6.1 Experimentation Methodology

Extensive experimentations are required in this project to effectively test for the robustness of the RARMT framework in handling a wide range of user profiles and its tolerance for sensory noises. In this section, we will:

1. Introduce the Testing Methods to be employed in this project's experiments
2. List down and explain the performance Evaluation Metrics to be used
3. Discuss possible factors that can affect the framework's performance

6.1.1 Performance Testing Method Used

There are two testing methods employed in this project:

Cross Validation

Cross-validation is a model validation technique that provides insight on how the prediction model will generalize to an independent and unknown dataset. The goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the validation dataset), in order to address problems like over-fitting. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

In this project, the 10-fold cross validation is used. The original dataset is randomly partitioned into ten equal size subsets. Out of the ten subsets, a single subset is retained as the validation data for testing the model, and the remaining nine subsets are used as training data. The cross-validation process is then repeated ten times, with each of the ten subsets used exactly once as the validation data. Finally, the ten results from the folds can then be averaged to produce a mean performance score.

Leave-One-[User/Group]-Out Testing (LOO)

In Leave-One-[User/Group]-Out testing, the recognition model will be trained on all of the user profiles except one user/ one user group, the remaining profile will be used to test the model. For example, to train on the youth data group and test on the elderly data group. In this testing method, the model would have zero visibility of the data pattern in the *left-out user/ user group* during training phase. The results of this testing method can thus provide us a better indication of the generalization capability of our activity recognition framework.

6.1.2 Performance Testing Evaluation Metrics

In this project, the balanced F-score is used as the performance index to evaluate the experiment results. The F-score is a derived value from the Precision and Recall value calculated from the confusion matrix. This section describes how the F-score is derived.

i. The confusion matrix will be constructed for all result analysis. Table 6.1 illustrates how a basic confusion matrix looks like:

Table6.1: Confusion Matrix

		Predicted Class	
		A	**B**
Actual Class	**A**	True Positive (TP)	False Negative (FN)
	B	False Positive (FP)	True Negative (TN)

ii. After constructing the confusion matrix, then we can calculate the Precision and Recall values for each class as follows:

$$\text{Precision (A)} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall (A)} = \frac{TP}{TP + FN} \tag{5}$$

iii. Lastly, the F-Score for each classes can then be calculated based on this formula:

$$\mathrm{F} - \mathrm{score}\ (\mathrm{A}) = \frac{2 * \mathrm{Precision(A)} * \mathrm{Recall(A)}}{\mathrm{Precision(A)} + \mathrm{Recall(A)}} \tag{6}$$

Note that calculation for each class (e.g. A & B) follows the same formula for precision, recall and f-scores. Also, if cross validation is used, the f-scores will be calculated for each fold and then averaged out to get the final F-score.

6.1.3 Possible Factors of Framework Performance

Frame size used to extract features

The frame size defined determines how many seconds of preprocessed sensory data is used as a window for feature extraction. For example, if the frame size is defined to be four seconds, it means that four seconds of sensory data is used to calculate a set of features (e.g. mean, variance). In essence, the frame size represents the time duration needed by the recognition framework to recognize a human activity.

Having a small frame size indicates that the decision is made with the information within a short time period. An advantage of a small frame size is that the model can predict activity quicker and this will be useful if the framework is to be used for real time applications. However, having a small framework may potentially give rise to the risk that the features extracted become insufficient and unreliable to describe a human activity [20].

On the other hand, having a large frame size indicates that the decision is made with the information within a longer time period, which can cause frequent changes in information within a frame. This can be a problem especially with limited training data, and may cause over-fitting problems [17]. Therefore, there should be a suitable window length that could achieve the best tradeoff between the two.

Cost and Gamma Parameters of the SVM classifier

In this project, effort has been spent to experiment on the impact of the SVM's Cost and Gamma parameters on the overall performance of the framework (see *Section 5.5.2*). We identified that there is possibility of over-fitting when using Grid Search in scenarios when the SVM model is dealing with unseen test data (i.e. LOO testing and Motion transition testing).

Grid Search finds the best SVM parameters based on the training dataset, hence when applying the same derived "optimal" Cost and Gamma parameters to a completely separate testing dataset, performance will drop due to over-fitting problem. As a result, we decided to use a recommended Cost (C=1) and Gamma ($G = \frac{1}{no.of\ features}$) value [18] in this project to overcome this problem for such cases. Table 6.2 summarizes the C and G parameter value used for the different testing methods in this project.

Table 6.2: C & G parameter selection for different testing

	Using Grid Search		**Fixed C & G values**		**Remarks**
	C range	G range	C value	G value	
Cross Validation Testing	$2^{-5} <$ **C** $< 2^{5}$	$2^{-5} <$ **G** $< 2^{5}$	-	-	Grid Search can be used to find optimal C & G values.
Leave-One-[User/Group]-Out Testing	-	-	1	1/22	Default (C=1) and ($G = \frac{1}{no.of\ features}$) values have to be used to avoid over-fitting problem since SVM classifier has never seen the testing dataset before.
Motion Transition Testing	-	-	1	1/22	

The impact of the selection of these parameters on the experimental results can be found in *Section 6.3.3*.

6.2 Performance Testing using Benchmark Datasets

6.2.1 Experimentation Objective

The core of *Robust Activity Recognition with Motion Transition (RARMT)*'s SVM model is developed based on concepts adopted from [17], which centers on building a robust classification system for recognizing basic low level human motions (e.g. Lying, Standing, Walking etc.). As such, this set of experimentations aims to test for the performance of RARMT in recognizing low level motions using benchmark datasets.

6.2.2 Experimentation Set-up

In this experiment, we adopted two externally sourced datasets [17] and [19] which will be referred as the Lin's Dataset and Davide's Dataset respectively. These two datasets will be used to serve as a benchmark to test for the performance of our RARMT framework. In the following sections we will extract prior results obtained from [17] and [19], and compare them to the reported results obtained from our own RARMT framework using these datasets.

6.2.3 Lin's Dataset

6.2.3.1 Overview

This dataset is obtained from [17] through personal communication, where a motion recognition model is built with the capability to perform well given that the position and orientation of the sensor device is unknown and varying in an accelerometer-based physical activity recognition. The core ideas and concepts used in the SVM model of this project's RARMT framework were largely adopted from [17]. These adopted ideas were enhanced with additional features in the RARMT framework and further experimentations and detailed analysis of the recognition model were conducted.

The motivation of using this dataset is to serve as proof that this project has succeeded in replicating the results obtained in [17]. Tables 6.3 and 6.4 summarize the dataset profile and experiment summary of [17].

Table 6.3: Lin's Dataset Profile

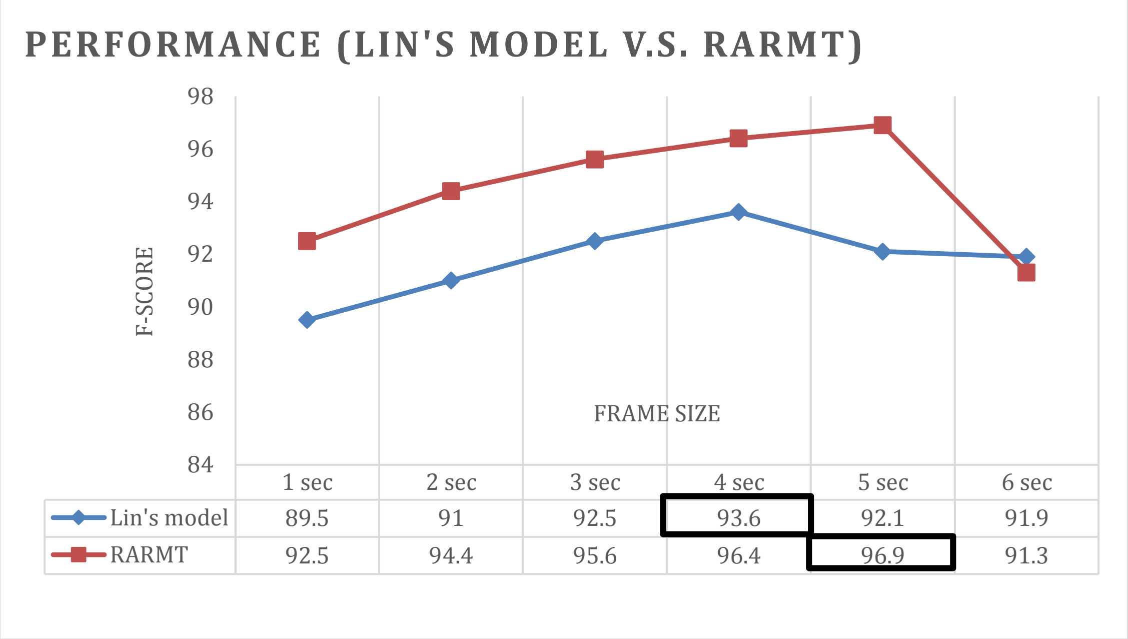
Generic SVM Dataset Profile	
No. of Users	7
User Profile (age range)	25-46
Activities Recorded	Stationary, Walking, Running, Bicycling, Ascending stairs, Descending stairs, Driving
Data Collection Device	Nokia N97- Python Program (placed in different
Sensor/s Used	Accelerometer
Sampling Frequency	10Hz
Frame Size	1-6 seconds
Hours	48.2

Table 6.4: Lin's Experiment Summary

Generic SVM Experiment Summary	
Classifier	SVM
Preprocessing procedure	No noise filtering done
Feature Types Used (x5)	Mean, Variance, Energy, Entropy, Correlation
Test Strategy	Cross Validation (10 Folds)
Additional Information	Data collection on 6 different pockets locations and 4 phone orientations.

We will replicate Lin's experiment as described in Table 6.4 closely and only cross validation for different frame sizes (1sec~6sec) will be tested. In the next section, we will combine prior results extracted from [17] and RARMT's performance result in the same chart for comparison and analysis.

6.2.3.2 Benchmarking Results and Analysis



	1 sec	2 sec	3 sec	4 sec	5 sec	6 sec
◆ Lin's model	89.5	91	92.5	93.6	92.1	91.9
■ RARMT	92.5	94.4	95.6	96.4	96.9	91.3

Figure 6.1: Lin's Dataset- Performance of Lin's model vs RARMT

Analysis

Figure 6.1 shows the benchmarking performance scores of the RARMT framework and Lin's model across different frame size used for feature extraction. We can see that the RARMT framework has consistently matched its performance to what is reported in [17]. This shows proof that we have successfully achieved the project objective of replicating the experiment based on concepts described in [17]. In fact, there is even a slight improvement in scores when using the RARMT framework. This is possibly due to the different C and G values used in this project's experiments.

Furthermore, from Figure 6.1, we can observe that the impact of frame size on the performance of the framework's performance is quite significant when using Lin's dataset. This is mainly due to the fact that Lin's dataset is a very large and substantial collection of sensory data. In general, having a larger frame size help to improve performance of the classifier. But if the training dataset is not big enough, having a large frame size will degrade performance instead. In this case, since Lin's Dataset comprises of data coming from 7 users and has the size of about 50 hours, it can overcome the limitation and achieve generally better performance with a larger frame size.

6.2.4 Davide's Dataset

6.2.4.1 Overview

In [19], a multi-class hardware friendly SVM is designed. The method used in [19] adapted the standard Support Vector Machine (SVM) and exploits the fixed-point arithmetic for computational cost reduction. Three sensor inputs, the acceleration, body-acceleration (i.e. acceleration subtract gravity) and gyroscope data, and a total of 561 features were used in this project. Furthermore, in Davide's experiment, 70% of the volunteers' data were used to train the model and the remaining 30% of the unseen data were used to test the model. Table 6.5 and 6.6 summarize the dataset profile and the model used in [19].

The motivation behind acquiring this dataset for benchmarking purposes is that it consist of a large group of user profiles (30 volunteers). Furthermore, a large information dimension is used in Davide's hardware friendly SVM model [19]. Hence, Davide's dataset will be useful to serve as a benchmark to test out the generalization capability of the new *Robust Activity Recognition with Motion Transition (RARMT)* framework established in this project, as well as its ability to work with limited information (i.e. using a single sensory input and only 5 different feature types).

Table 6.5: Davide's Dataset Profile

Hardware Friendly SVM Dataset Profile	
No. of Users	30
User Profile (age range)	19-48
Activities Recorded	Standing, Sitting, Laying, Walking, Walking upstairs and Walking downstairs
Data Collection Device	Samsung Galaxy S2 (worn with a holder on the waist)
Sensor/s Used	Accelerometer, Gyroscope
Sampling Frequency	50Hz
Frame Size	2.56 sec (50% overlap)

Table 6.6: Davide's Experiment Summary

Hardware Friendly SVM Experiment Summary	
Classifier	SVM- Laplacian Kernel
Preprocessing procedure	Noise filter was applied
Feature Types Used (x17) (561 features in total)	Mean value, Standard deviation, Median absolute deviation, Largest value in array, Smallest value in array, Signal magnitude area, Energy measure, Interquartile range, Signal entropy, Auto-regression coefficients, correlation coefficient, Index of frequency component with largest magnitude, Weighted average of frequency components, skewness of the frequency domain signal, kurtosis of the frequency domain signal, Energy of a frequency interval, Angle between to vectors
Test Strategy	70% collected data used for training and 30% collected data used for testing.
Additional Information	A lesser number of bits were used in the SVM parameter estimation.

6.2.4.2 Benchmarking Results and Analysis

In this section, we first represent the two result tables reported in [19] - the prior results of Davide's dataset with a standard SVM model and Davide's adapted SVM model. These classification results are depicted by means of a confusion matrix in Tables 6.7(a) and 6.7(b) respectively. These two tables can help serve as benchmark comparison for our RARMT framework performance score using the same Davide's dataset.

In this experiment frame size of three seconds will be used, gyroscope and linear acceleration information will be omitted, we will use only the body-acceleration and 22 features. We will first conduct a cross validation test on 100% of Davide's dataset to find out the most suitable Cost and Gamma parameter values for the dataset using grid-search (see Table 6.8). Next, we will replicate Davide's experiment using 70% of the volunteers' data to train and the remaining unseen data to test our RARMT framework, using the identified optimal C and G values (see Table 6.9). Through this experiment, we can fulfil our experimental aim in testing RARMT's ability in working with limited information, since a much smaller dimension of information is used in this case.

Table 6.7(a): Prior Results- Standard SVM

Activity	Walking	W.Up	W.Dowm	Standing	Sitting	Laying	Recall%
Walking	**109**	0	5	0	0	0	95.6
Upstairs	1	**95**	40	0	0	0	69.8
Downstairs	15	9	**119**	0	0	0	83.2
Standing	0	5	0	**132**	5	0	93.0
Sitting	0	0	0	4	**108**	0	96.4
Laying	0	0	0	0	0	**142**	100
Precision%	87.2	87.2	72.6	97.1	95.6	100	
F-Score	**89.8%**						

Table 6.7(b): Prior Results- Davide's model

Activity	Walking	W.Up	W.Dowm	Standing	Sitting	Laying	Recall%
Walking	**109**	2	3	0	0	0	95.6
Upstairs	1	**98**	37	0	0	0	72.1
Downstairs	15	14	**114**	0	0	0	79.7
Standing	0	5	0	**131**	6	0	92.2
Sitting	0	1	0	3	**108**	0	96.4
Laying	0	0	0	0	0	**142**	100
Precision%	87.2	81.7	74	97.8	94.7	100	
F-Score	**89.3%**						

Table 6.8: RARMT results- Cross Validation on 100% of Davide's dataset

Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall%
Walking	3606	33	136	0	0	0	95.52
Upstairs	51	3377	6	0	0	0	98.34
Downstairs	493	12	2606	0	0	0	83.77
Standing	0	0	0	3865	0	0	100
Sitting	0	0	0	0	4155	0	100
Laying	0	0	0	0	0	4230	100
Precision%	86.89	98.69	94.83	100	100	100	
F-Score	**96.50%**						

Table 6.9: RARMT results- 70%-30% testing on Davide's dataset

Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall%
Walking	786	59	0	0	0	0	93.02
Upstairs	281	520	1	0	0	0	64.84
Downstairs	4	70	641	0	0	0	89.65
Standing	0	0	0	495	135	206	59.21
Sitting	0	0	0	73	825	8	91.06
Laying	0	0	0	29	5	881	96.28
Precision%	73.39	80.12	99.84	82.91	85.49	80.46	
F-Score	**83.02%**						

Prior Results Analysis

Despite restricting the number of data bits used in the calculation in Davide's model processing, results reported [19], results shown in from Table 6.7(b) is decent. This accuracy score is in fact close to what is reported when a standard SVM is used (see Table 6.7(a)). However it is interesting to note that in both cases, a huge information dimension of - 561 features (from 17 feature types) and three sets of sensory data- linear acceleration, gyroscope and body acceleration (linear acceleration subtracted gravity) were used. Also, data cleaning methods were employed in Davide's approach.

Benchmarking Results Analysis

The benchmark results obtained using Davide's dataset is shown in the form of the confusion matrix in Tables 6.8 and 6.9. Unfortunately due to the limitation of the data source, we were not able to expand the experiment to test out the effect of other factors (e.g. frame sizes). In this case, the benchmark test with RARMT omits the acceleration and gyroscope information and uses only five different features (see *Section 5.4)*. Also, due to the different frame size used and possible discrepancies between their publicly published dataset and the actual one used in [19], the testing sample size is different.

We can observe a slight drop in performance of about 6% with the RARMT framework as compared to Davide's hardware friendly SVM model (see Table 6.7(b)). However, RARMT achieved a comparable results of 83.02% F-score despite working with a much smaller dimension of information - one sensory inputs (instead of three) and 22 features (instead of 561), as compared to Davide's model [19]. This suggests that it is possible for a model to perform well even with a smaller set of information, as long as the most informative features are selected for the classification problem. Hence, it is probable for the RARMT framework to be more hardware friendly than Davide's model although this experiment is not yet implemented and tested on smartphones.

Findings from this experiment serve as a strong indication that:

- Gyroscope readings may be omitted of features extraction already contains orientation information. In this case, with the 4th dimension (magnitude) of the accelerometer readings included, certain orientation information is captured. Hence, we can still achieve high performance without gyroscope data.
- There is no direct correlation between the number of features and performance. Using more features used does not necessarily equate providing better results. It is important to identify and use only distinguishing features that have been empirically proven to be relevant to the problem type.

6.3 Performance Testing using Dataset on Different Age Groups (DAG)

In this section, we will first discuss on the Different Age Groups (DAG) dataset regarding details like its collection procedures and users profile included. Next, we will explain the methodology behind this experiment and finally evaluate the experiment results.

6.3.1 Collection of DAG Dataset

6.3.1.1 Objective

As mentioned in *Section 2*, we realized that many of the experiments conducted by existing work like [4] and [19] are too stringent to reflect the natural settings and the way users normally behave in real life. Hence, in this project we collected a new set of data- Dataset on Different Age Groups (DAG) to serve this need. With the DAG dataset, we then can then further test for the robustness of the RARMT framework.

6.3.1.2 Data Collection Overview

The DAG dataset should reflect the most natural activity performed by users from two very distinct age group (i.e. youth and elderly) under normal circumstances. As such, strict restrictions on users' activity were not imposed, users are allowed to fidget or make small upper body movements as they normally do during the entire data collection procedure. Unlike data collected in work in [4] and [19], in this project it is not our aim to collect extremely clean data. The RARMT framework is expected to handle noisy data effectively.

Each data collection procedure last for a total duration of 5 minutes (which can be easily modified according to experiment needs). The user will perform the same activity continuously for 5 minutes each time data is collected. Setting a long collection period allows for more efficient data collection as there is less overhead efforts expensed to perform data tagging. However, this also increases the likelihood of collecting dirty data (e.g. User fidgets while standing). Furthermore, data collection is carried out under different environment settings (e.g. closed room/ opened field) and with various ground condition (e.g. smooth pavement/ rocky ground). The Samsung Galaxy S5 is used in this project for data collection and the Android application records data at a suggested rate of about 0.2 seconds (SENSOR_DELAY_NORMAL). At this frequency, we are able to sufficiently capture sensory data changes for each activity while not taking up too much resources on the smartphone.

6.3.1.3 Data Collected

Before users commence on recording their activity data, an introduction to how the android application can be used; what activity they will be performing and how to place the smartphone in their pockets will be given. Users are allowed to perform the data collection procedure in any sort of clothing with any pocket size. Note that the looseness of clothing/ size of pockets indicates how much freedom the smartphone has to deviate from its original position in the pocket during data collection. Also, no incentive are given to the volunteers.

Youth Dataset- User Profiles

Table 6.10: Youth User Profile (Data Collected)

User Gender	User Age
Female	18
Female	20
Female	22
Male	30
Mean	22.50
Standard deviation	5.26

Elderly Dataset- User Profiles

Table 6.11: Elderly User Profile (Data Collected)

User Gender	User Age
Male	67
Female	65
Male	70
Female	80
Mean	70.50
Standard deviation	6.66

No. of hours collected

About a total of 8 hours of data is collected for this project:

Table 6.12: Estimated No. of Hours (Collected Data)

Human Activity	No. of Hours Collected	
	Youth Group	**Elderly Group**
Laying	65mins	35mins
Sitting	65mins	35mins
Standing	65mins	35mins
Walking	65mins	35mins
Running	70mins	10mins

Note: Some elderly volunteers were unfit to perform the running motion. Hence, some running motions are not collected.

6.3.1.4 Data Collection Procedure

1. User indicates the activity they will be performing
2. User indicates the phone location (left/right pockets)
3. User press “Start Recording”
4. User place the smartphone is the preferred pocket
5. User waits for the smartphone’s start sound signal
6. User performs the activity until he/she hears the stop sound signal

6.3.1.5 Experimentation Set-up

1. **Cross Validation Testing**
   a. Single User: In this test, we pick a user profile and perform 10 folds cross validation on his data.
   b. Multiple Users: In this test, we combine all user profiles and perform 10 folds cross validation on the entire DAG dataset.
2. **Leave-One-[User/Group]-Out Testing**

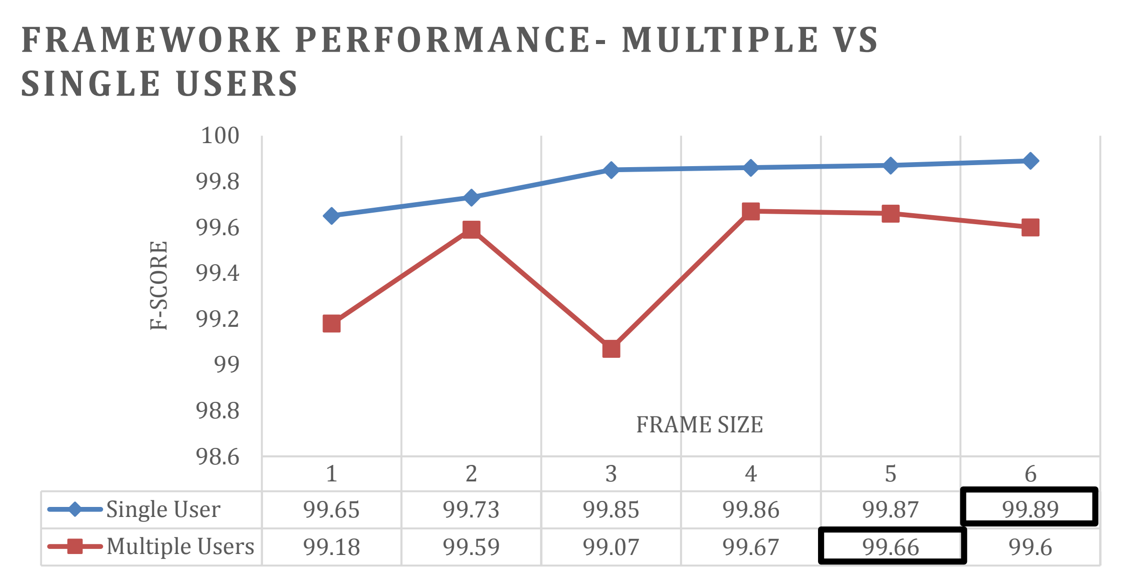
Table 6.13: LOO Testing Set-up

	Training Data	Testing Data	Description
a.	Entire DAG Dataset excluding User X	User X- a single unknown user profile	This test aims to test if the system is able to extend its generalisation capability and still perform well on its prediction for an unknown user profile.
b.	Youth Dataset	Elderly Dataset	The DAG dataset is divided into the Youth and Elderly dataset. This two datasets can then be used to test for the system’s generalisation capability across different age groups. For instance, to see if the system can still perform well on its prediction for elderly people when it only has knowledge of young people’s motion sensory patterns.
c.	Elderly Dataset	Youth Dataset	

6.3.2 Experimentation Results

6.3.2.1 Cross Validation Testing

We tested the RARMT’s performance in handling a single user profile vs handling a diversified dataset with multiple user profiles from very distinct age groups, using cross validation. The results are extracted and shown as a comparison chart in Figure 6.2. From both test scenarios (i.e. Single User and Multiple Users) we picked out the best performing frame size and constructed two confusion matrixes (Tables 6.14 and 6.15) respectively.



	1	2	3	4	5	6
Single User	99.65	99.73	99.85	99.86	99.87	99.89
Multiple Users	99.18	99.59	99.07	99.67	99.66	99.6

Multiple User Scenario Cross Validation is performed using the entire DAG dataset

Figure 6.2: Performance graph for different frame size- multiple vs single users

Table 6.14: Single User Cross Validation Confusion Matrix (6 seconds frame)

Activity	Laying	Sitting	Standing	Walking	Running	Recall%
Laying	**1324**	0	0	0	0	100
Sitting	0	**1319**	2	0	0	99.85
Standing	0	0	**783**	0	0	100
Walking	0	1	2	**781**	0	99.62
Running	0	0	0	0	**478**	100
Precision%	100	100	99.92	99.49	100	
F-Score	**99.89%**					

Table 6.15: Multiple Users Cross Validation Confusion Matrix (4 seconds frame)

Activity	Laying	Sitting	Standing	Walking	Running	Recall%
Laying	**2939**	5	0	0	3	99.73
Sitting	4	**3533**	1	0	1	99.83
Standing	0	1	**2429**	6	0	99.71
Walking	0	9	3	**2566**	5	99.34
Running	0	0	0	4	**1521**	99.74
Precision%	99.86	99.58	99.84	99.61	99.41	
F-Score	**99.67%**					

Analysis

There are two key points to note in this cross validation test performed on the Single User and Multiple Users test scenarios. Firstly, to see if there is a significant performance difference when RARMT handles a more diversified dataset. Secondly, to experiment the effect of different frame size (1 second to 6 seconds) on RARMT's performance. This can be analyzed from Figure 6.2.

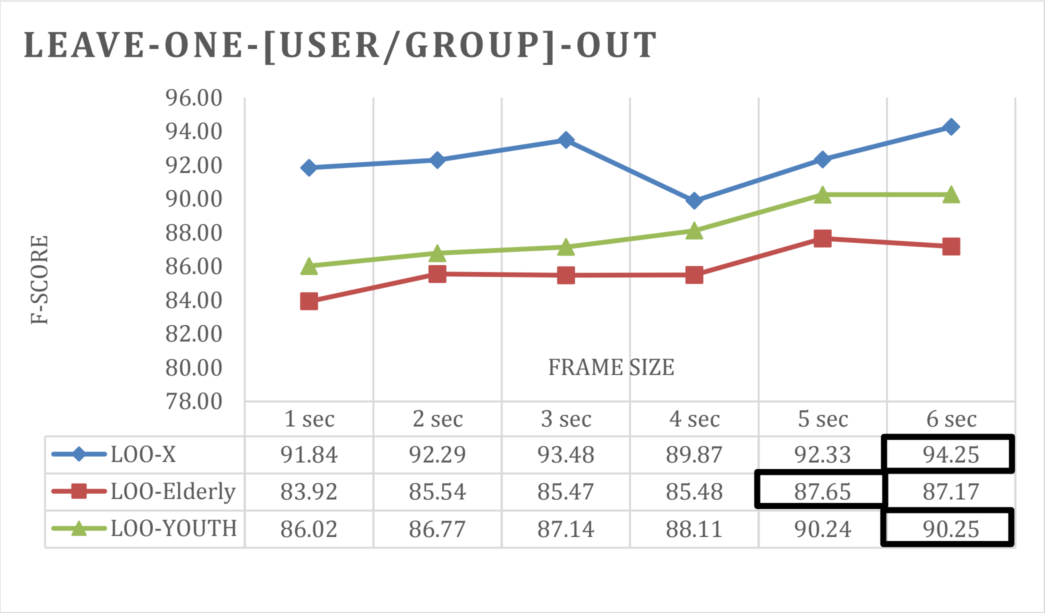
From Figure 6.2, we can see that the RARMT framework has consistently achieved very high performance F-scores of above 99% for **both** *Single and Multiple Users Scenario* across all frame sizes tested. This proves that the RARMT framework has high ability to recognize human activities accurately regardless of whether the system is trained and tested on a single user profile or diversified user profiles.

When comparing this result to that of the Single User Scenario, there is only a slight performance drop of less than 1%, which is insignificant. Hence, this indicates that the RARMT framework has high generalization ability and is highly robust as it can handle a mixture of user profiles from such a wide age group range.

In Tables 6.14 and 6.15, we report the confusion matrix of the optimal frame size cross validation testing for single user scenario and multiple users scenario. From both the confusion matrixes we can observe that (Running/ Walking) and (Laying/ Sitting) activities in particular are less distinguishable when testing on Multiple Users as compared to Single User.

6.3.2.2 Leave-One-[User/Group]-Out Testing (LOO)

We performed a more rigorous performance test on RARMT to test for its robustness and generalisation capability in handling users from very different profile. In this experiment, we separate the dataset into two groups (training and testing dataset) and test for RARMT's ability to predict activities for the unseen testing dataset. Results are extracted and displayed in Figure 6.3.



	1 sec	2 sec	3 sec	4 sec	5 sec	6 sec
LOO-X	91.84	92.29	93.48	89.87	92.33	94.25
LOO-Elderly	83.92	85.54	85.47	85.48	87.65	87.17
LOO-YOUTH	86.02	86.77	87.14	88.11	90.24	90.25

Figure 6.3: Performance graph for different frame size- Leave-One User Profile-Out

LOO-X: System is trained with the entire DAG dataset, leaving out one user profile. For example, we will train the system with [DAG without 22 years old female data] and test with [22 years old female data]. This experiment will be repeated for all users in the dataset. Then the results will be aggregated.

LOO-Elderly: System is trained with only motion data of Young people and tested with Elderly people data.

LOO-Youth: System is trained with only motion data of Elderly people and tested with Young people data.

Table 6.16: LOO-X Confusion Matrix (6 seconds frame)

Activity	Laying	Sitting	Standing	Walking	Running	Recall%
Laying	**1931**	25	0	0	0	98.72
Sitting	199	**2045**	2	1	0	91.01
Standing	0	14	**1584**	13	0	98.32
Walking	0	0	72	**1533**	5	95.22
Running	0	0	18	141	**947**	85.62
Precision%	90.66	98.13	94.51	90.82	99.47	
F-Score	**94.25%**					

Table 6.17: LOO-Elderly Confusion Matrix (5 seconds frame)

Activity	Laying	Sitting	Standing	Walking	Running	Recall%
Laying	**372**	106	0	0	0	77.82
Sitting	120	**477**	0	1	0	79.77
Standing	0	0	**474**	4	0	99.16
Walking	0	0	3	**474**	1	99.16
Running	0	0	0	76	**282**	78.77
Precision%	75.61	81.82	99.37	85.41	99.65	
F-Score	**87.65%**					

Table 6.18: LOO-Youth Confusion Matrix (6 seconds frame)

Activity	Laying	Sitting	Standing	Walking	Running	Recall%
Laying	**1556**	12	0	0	0	99.23
Sitting	133	**1229**	391	5	0	69.91
Standing	0	0	**1212**	11	0	99.10
Walking	0	0	70	**1124**	27	92.06
Running	0	0	18	66	**732**	89.71
Precision%	92.13	99.03	71.67	93.20	96.44	
F-Score	**90.25%**					

Analysis

Figure 6.3 is a graph chart showing the three LOO test results across different frame sizes. Tables 6.16, 6.17 and 6.18 are confusion matrices of the optimal frame sizes of the respective LOO tests performed in this experiment. From these statistics, we can measure the RARMT framework's generalization capability across different user profiles and groups to test for its robustness.

General Observations from Figure 6.3:

- LOO testing generally yields lower F-scores than Cross Validation testing.
  - This is expected as LOO testing is a much more robust way in accessing the generalization capability of a model. The model is dealing with unseen testing data, hence there will inevitably be more prediction errors.
  - However, it is encouraging to note that the results are still consistently in the 85-90% range.
- LOO testing performs better using a larger frame size
  - This is expected as unseen test data is used in LOO testing. Hence, the framework will require more information (larger frame size for feature extraction) in order to make a more accurate prediction.
- LOO-X (leave single user out) testing yield a consistently high F-scores (~90%) across the different frame sizes.
  - This is an optimistic observation. It serves as a good indication that the model is likely to perform well in making activity predictions for an unknown user profile, when given a substantial training dataset (like the DAG). This potentially increases the usability of the system as it means that a new user does not have to go through a tedious data collection period in order to use the motion recognition system.

General Observations from Confusion Matrix:

- We can observe that certain motions are less distinguishable. For instance, when the model is trained on the youth dataset and tested on the elderly dataset, the model wrongly categorized some of the elderly's running motion to be walking (see Table

6.17). This is expected, as an elderly person runs much slower, which can be similar to how a youth brisk walk.

- The same can be applied to the other less distinguishable motions like sitting. (see Table 6.18).

6.3.3 Experiment Results Evaluation Summary

1. **C & G parameter value Evaluation**

Table 6.19: C&G parameter Experiment results

		LOO- Elderly	LOO- Youth
Optimal F-score %	Fixed C=1; G=1/22	87.65	90.25
	Grid Search C & G search range [2^{-5} *to* 2^{5}]	79.98	87.81
	Performance difference	7.67	2.44

As discussed in *Section 6.1.1.3*, the selection of SVM Cost and Gamma parameters will have an impact on the performance of the SVM model. We identified the potential over-fitting problem of using grid search for tests where the testing dataset is not visible to the model during training phase (i.e. LOO) and hence, proposed to use default C and G parameters. In Table 6.19, we see the experimental results improvement for Leave-One-User-Group-Out tests achieved from this decision.

2. **Frame Size Selection**

Table 6.20: Test Results Summary

	Cross Validation	LOO-X	LOO-Elderly	LOO-Youth
Optimal Frame Size	4sec	6 sec	5 sec	6 sec
Optimal F-score	99.67%	94.25%	87.65%	90.25%
F-score of 1 sec Frame	99.18%	91.84%	83.92%	86.02%
F-score Difference	0.49%	2.41%	3.73%	4.23%

From Table 6.20, we can observe that different tests have different optimal frames. In general, LOO testing has larger optimal frame size. The RARMT framework requires more information (i.e. larger frames) in order to make an activity prediction when it has no knowledge of the testing data. However, this also indicates that more time is required to make a decision.

In applications that require real time analysis of human motions, using a larger frame size will not be realistic. Hence, in the following section, when we conduct experimentation for Motion Transition Recognition, frame size of 1 second will be used. From Table 6.20, we can see that RARMT is capable of high performance even with a smaller frame size, as its F-scores (1 second frame size) are still reasonably high and close to optimal score for most of the cases.

6.4 Motion Transition Recognition Performance Testing

6.4.1 Experimentation Objective

In this experiment, we aim to test for the Robust Activity Recognition with Motion Transition (RARMT) framework's performance in recognizing motion transitions. The list of motion transitions to be recognized by the framework can be found in Table 6.21 below. Also a consisted illustration of the transitions using a state diagram can be found in Figure 5.9 in *Section 5.6*.

6.4.2 Experimentation Set-up

In this experimentation section, we will first collect a series of motions that represents:

- **Stable States** (user remain at one of the low level motion- Lying, Sitting, Standing, Walking, Running)
- **Active States** (user transits from one low level motion state to another)
    - **Fall Down Motion**
    - **Normal Motion Transition**

Table 6.21: Motion Transition Experiment- Reference Table

Reference	Motion
A	Low level motion- Lying Down
B	Low level motion- Sitting
C	Low level motion- Standing
D	Low level motion- Walking
E	Low level motion- Running
F	Falling Motion- From Standing

G	Falling Motion- From Sitting
H	Falling Motion- From Walking
I	Falling Motion- From Running
J	Normal Transition- Standing to Lying
K	Normal Transition- Lying to Standing
L	Normal Transition- Sitting to Lying
M	Normal Transition- Lying to Sitting
N	Normal Transition- Walking to Lying
O	Normal Transition- Lying to Walking
P	Normal Transition- Walking to Running
Q	Normal Transition- Running to Walking
R	Normal Transition- Walking to Sitting
S	Normal Transition- Sitting to Walking
T	Normal Transition- Standing to Sitting
U	Normal Transition- Sitting to Standing
V	Normal Transition- Running to Standing
W	Normal Transition- Standing to Running

These series of sensory data will be collected using different time period, some test data may last for 15 seconds while others 30 seconds. Also the point of transition each test data will not be fixed. This is to simulate more closely to how users behaves in real life. For example, *Walk to Kitchen → Stand while pouring a cup of water* or *Walk to Living Room → Sit down on sofa.*

A total of 5 test data instances are collected for each motion. These instances will then be used for mainly 2 purposes:

1. To determine the threshold values to set in the algorithm
2. To test for RARMT performance in recognizing motion and transitions

In the following sections, we will discuss more on the information input for RARMT's motion transition recognition, the threshold value to set and finally report the experiment results.

6.4.3 Information input for Motion Transition Recognition

6.4.3.1 SVM low level motion prediction labels

Raw readings from the testing datasets will be fed into the RARMT. The data will be sampled at a fixed frequency, features will be extracted for frame size of one second, and the SVM will perform classification and output the prediction labels. These labels will then be used as an information input for motion transition recognition.

6.4.3.2 Signal Magnitude Area of Raw Accelerometer X, Y, Z

The SMA metric is used in this algorithm to distinguish between a stable state and user activity for recognition of basic daily movements. This means that the SMA feature is able to assist in the differentiation between a sequence of motion with no transition and a sequence of motion with transitions. For example, the Motion Transition Recognition processor will be able to detect if the user is at stable state in one of the low level motion (i.e. Lying, Sitting, Standing, Walking, Running) or is performing a motion transition (e.g. Standing Up). Eq. 7 shows the SMA calculation using acceleration signals from each axis (x, y, z) with respect to time t.

$$SMA = \frac{1}{t}\left(\int_0^t |x(t)|dt + \int_0^t |y(t)|dt + \int_0^t |z(t)|dt \right) \tag{7}$$

When a user makes a sudden movement (e.g. Falling Down), raw accelerometer x, y, z readings will consequently change drastically as seen from Figure 6.4. This will result in a significant change in the corresponding SMA values, which is calculated by the area under the three axis. As such, we can use SMA readings as an indication to differentiate between a stable and active state.

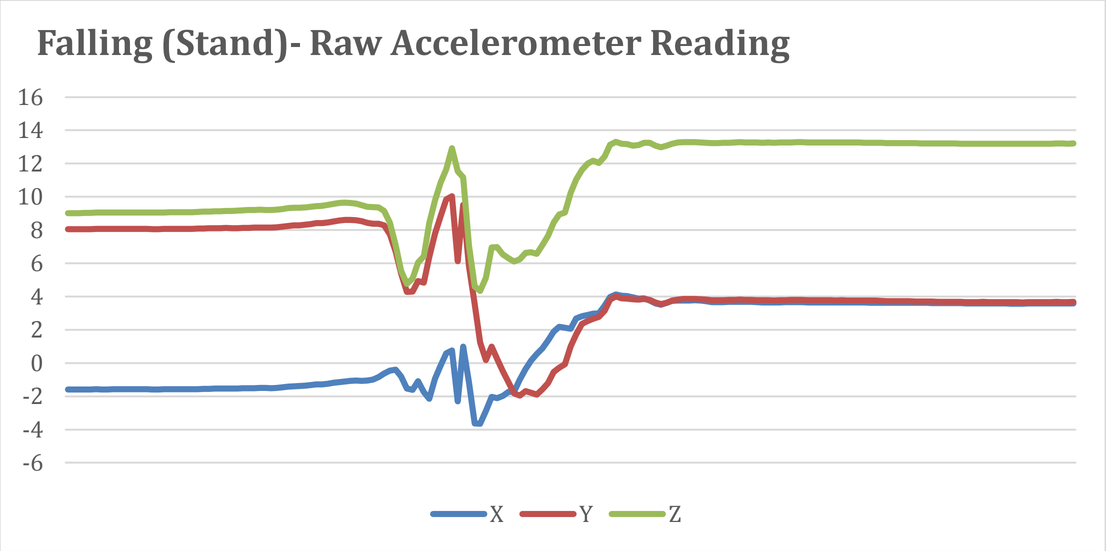


Figure 6.4: Falling Raw Accelerometer Reading Graph

From Figures 6.5 and 6.6 we can see how SMA provides valuable information in determining between a real motion transition and stable state. Figure 6.5 shows the SMA values of the five stable low level motions, we can see that the SMA values are generally stable with little variation. On the other hand, from Figure 6.6, we can see the SMA values of three example sequence of motion with transitions. In this case, we see that there is a significant variation of SMA values when user transits from one motion to another. Also, we can observe that after user reach a stable state at "Lying", the SMA values will also reach a steady level similar to what is shown in Figure 6.5.

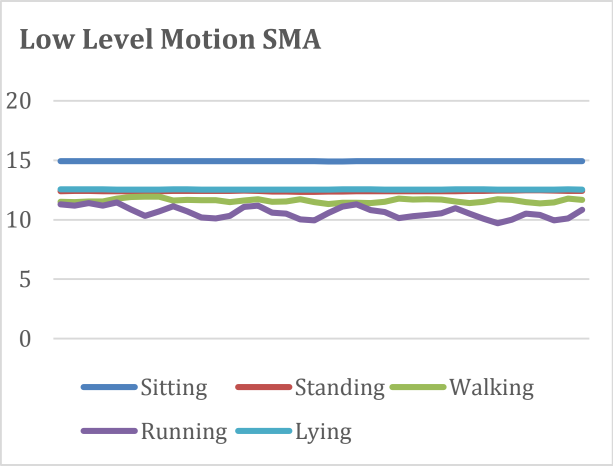


Figure 6.5: Low Level Motion SMA Graph

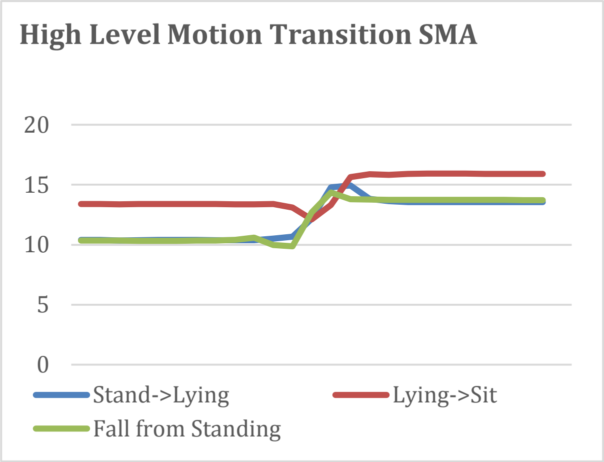


Figure 6.6: High Level Motion Transition SMA Graph

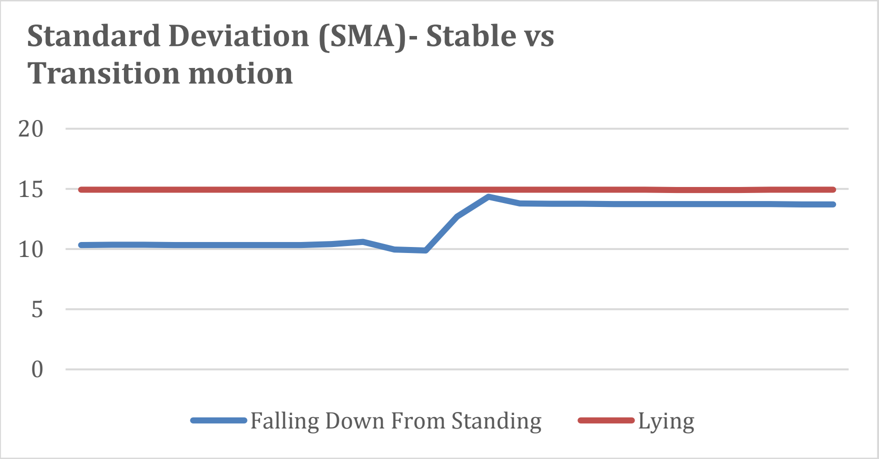


Figure 6.7: Standard Deviation (SMA) Graph

Furthermore, by extracting the standard deviation of these raw SMA values, we can determine if an actual transition has happened. When a transition occurs (e.g. Falling down), there will be a larger change in SMA values as compared to a stable motion (e.g. Lying) (see Figure 6.7). Therefore, we can use a SMA standard deviation threshold to differentiate between a stable state and a real motion transition. This threshold value will be determined via experimentations which will be elaborated in the following section.

6.4.3.3 Barometer Readings

A barometer is an instrument used primarily to test atmospheric pressure, which is directly related to elevation. Hence, a barometer can very easily be used as an altimeter to measure one's altitude. As such, we can employ barometer readings from our phone sensors to determine our change in altitude. This potentially gives as valuable insights in differentiating between a normal motion transition and a real Fall motion, which cannot be achieved using SMA readings.

SMA readings between a fall motion and a normal transition is not very different. From Figure 6.6, we can observe that the SMA values of a Fall Motion (Fall from Standing) follows a similar trend to a normal motion transition (Stand->Lying). SMA measures only the change in area under the raw accelerometer readings axis upon a change of motion state, it is not differentiating enough to determine what kind of motion state change occurred (i.e. Fall vs normal transition).

On the other hand, the barometer readings will be able to do the distinguishing. When a user falls, the barometer will record a sudden change in pressure. This is unlike when user make a normal motion transition, in which pressure change will be gradual. This is illustrated in Figure 6.8 below, where we can see an example of a Falling motion as compared to a normal motion transition.

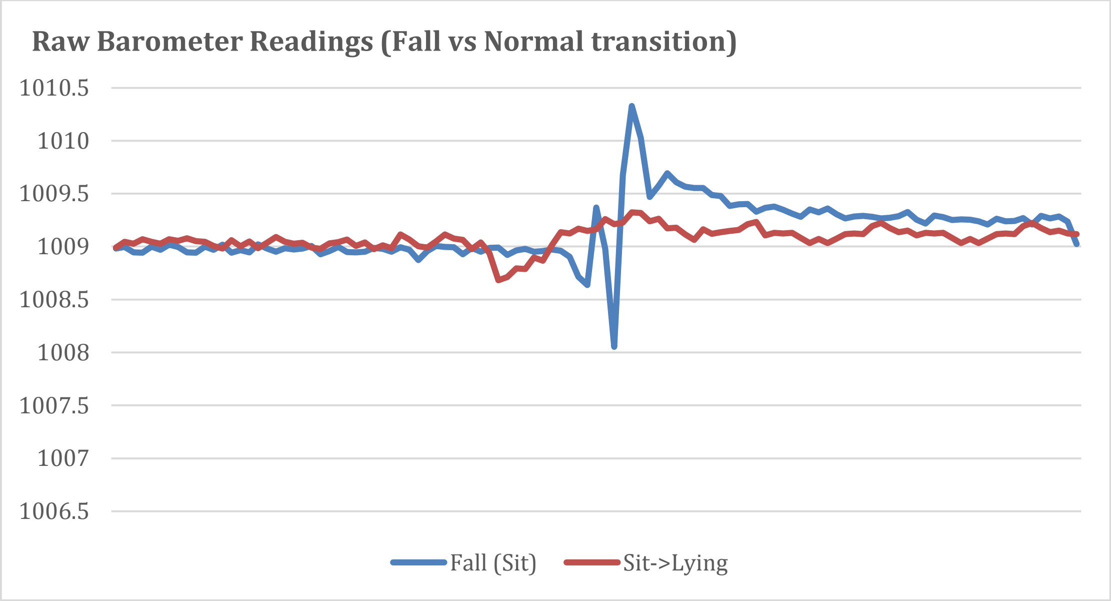


Figure 6.8: Raw Barometer Readings (Fall VS Normal transition)

Based on this train of thought, we derived ideas on how to utilize this piece of valuable information to perform Fall Detection more accurately. In this project, to perform motion recognition, we will first extract the *Standard Deviation of Barometer readings* in a frame (size = 1 second). It is only by extracting the barometer's standard deviation feature from a frame that allow us to retain the variation information of barometer readings, without any dilution of statistics.

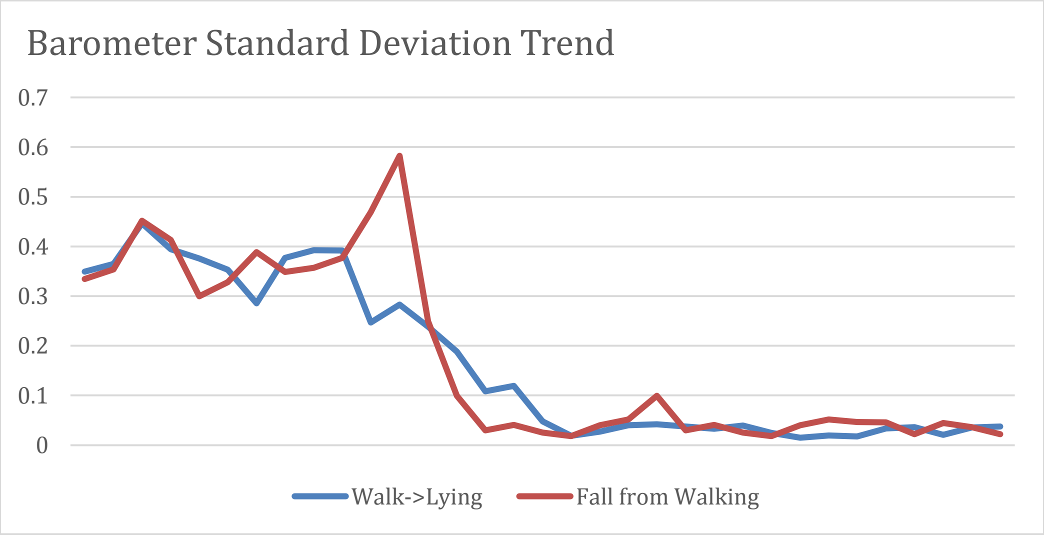


Figure 6.9: Barometer Standard Deviation trend

From Figure 6.9, we can see an example of a Falling motion and a normal motion transition's barometer standard deviation trend. We can observe that in general, barometer readings has a much larger variation for a falling motion, where the transition happens sharply, instead of a gradual variation as seen from a normal transition trend.

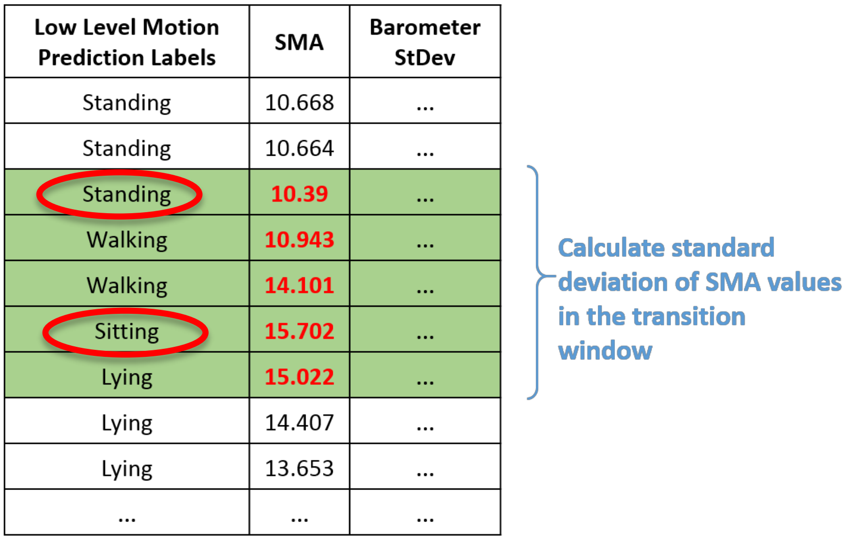
Therefore, by establishing a barometer standard deviation threshold we can distinguish effectively between a Falling Motion and a Normal Motion Transition. More details about the threshold will be discussed in the *Section 6.4.4.2*.

6.4.4 Determining Algorithm Thresholds

6.4.4.1 SMA threshold

The SMA values of the accelerometer raw readings will be extracted as part of RARMT Feature Extraction step, using frame size of 1 second and 50% overlap. At this stage of motion transition recognition processing, window size of 5 frames will be used.

The standard deviation of the SMA values in a processing window (5 frames) will be calculated and used to check against this threshold to differentiate between a stable (no transition) and an active state (with transition).



Low Level Motion Prediction Labels	SMA	Barometer StDev
Standing	10.668	...
Standing	10.664	...
Standing	**10.39**	...
Walking	**10.943**	...
Walking	**14.101**	...
Sitting	**15.702**	...
Lying	**15.022**	...
Lying	14.407	...
Lying	13.653	...
...	...	...

Calculate standard deviation of SMA values in the transition window

Figure 6.10: SMA standard deviation calculation

Experiments will be conducted to establish the *SMA threshold*. First, a set of high level motion transitions will be recorded. Next, SMA's Standard Deviation value of the identified "Possible transition" window (see Figure 6.10) will be calculated and reported. Second, record a set of stable low level motions and extract the SMA's Standard Deviation value from any selected window. Finally, consolidate the SMA's standard deviation results of all Transition Motions and the results of all Stable Motions separately. The resulting statistics for each transition category is summarized as shown in Table 6.22 below.

Table 6.22: SMA standard deviation distribution

SMA Standard Deviation Distribution		
	Mean	**Standard Deviation**
Active State (with transitions)	1.44	0.17
Stable State (no transitions)	0.11	0.02

From Table 6.22, we can observe that when a user is at a stable state (e.g. remain at sitting) the SMA's standard deviation in a window is about 0.11 with 0.02 deviation, while when a user is at an active state (e.g. falling down) the SMA's standard deviation in a window is about 1.44 with 0.17 deviation. This tells us that we can safely set the SMA threshold at 0.78 to differentiate between a stable state and active state.

$$SMA\ threshold = 0.11 + \frac{1.44 - 0.11}{2} = 0.78 \tag{8}$$

6.4.4.2 Barometer Threshold

The standard deviation of barometer readings will be extracted as part of RARMT Feature Extraction step, using frame size of 1 second and 50% overlap. At this stage of motion transition recognition processing, window size of 5 frames will be used.

The barometer's standard deviation in the processing window will be summed up. We will use this value to check against the Barometer Threshold to determine if transition category (i.e. Falling Down or normal).

Similarly, we will consolidate the barometer sum results of all Falling Down Motions and the results of all corresponding Normal transition Motions (i.e. Standing/ Walking/ Running/ Sitting → Lying) separately. The resulting statistics for each transition category is summarized as shown in Table 6.23 below.

Table 6.23: Barometer standard deviation distribution

Barometer's standard deviation Sum Distribution		
Transition Category	**Mean**	**Standard Deviation**
Falling Motion	1.67	0.08
Normal Motion	0.53	0.04

From Table 6.23, we can observe that when a user falls down the barometer's standard deviation sum in a window is about 1.67 with 0.08 deviation, while when a user makes a normal transition (e.g. Standing -> Lying) the barometer's standard deviation sum in a window is about 0.53 with 0.04 deviation. This tells us that we can safely set the Barometer threshold at 1.10 to differentiate between a real Falling Motion and a normal motion transition.

$$Barometer\ threshold = 0.53 + \frac{1.67 - 0.53}{2} = 1.10 \qquad (9)$$

6.4.5 Experimentation Results and Discussion

In this experiment, we will be using the DAG dataset as the training data and subsequently collect a series of transition motions to use as testing data. Table 6.24 shows the confusion matrix, representing the best guess results of RARMT motion transition recognition.

Table 6.24: Motion Transition Experiment Results- Confusion Matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
A	5																						
B		5																					
C			5																				
D				5																			
E					5																		
F						4				1													
G							4					1											
H								5															
I									5														
J						1				4													
K											5												
L							1					4											
M													5										
N														5									
O															5								
P																5							
Q																	5						
R																		5					
S																			5				
T																				5			
U																					5		
V																						5	
W																							5

***F:** Falling Motion- From Standing*
***J:** Normal Transition- Standing to Lying*

***G:** Falling Motion- From Sitting*
***L:** Normal Transition- Sitting to Lying*

Note that due to the sparse nature of the results, all zeros are omitted for presentation purposes.

Discussion

From Table 6.24, we can see that the RARMT framework has achieved high performance in differentiating between the different motion transitions with high accuracy.

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} = 0.965217$$

The transitions that proved to be harder to distinguish are:

- *Falling Down from Standing* **V.S** *Standing* → *Lying* (vice versa)
- *Falling Down from Sitting* **V.S** *Sitting* → *Lying* (vice versa)

The results obtained are within expectations as it is by nature, harder to differentiate between these motions with just the accelerometer and barometer readings. However, it is encouraging to observe that RARMT is accurate is most of the cases tested (100% F-score for the other transition tests).

6.5 Experimental Conclusion

In this section, we summarize how experimental observations in this project addresses the Problem Investigation Dimensions identified in *Section 4*.

Experiment		Data	Test Method	Frame Size/ sec	F-score
Basic Low Level Motion recognition (Lying, Sitting, Standing, Walking, Running)	Benchmark	Lin's Dataset	CV	5	96.90%
		Davide's Dataset	CV	3	96.50%
			LOO	3	83.02%
	Self-Collected	Dataset on different Age Group (DAG) Dataset	CV	6	99.67%
			LOO-X	6	94.25%
			LOO-Elderly	5	87.65%
			LOO-Youth	6	90.25%
High Level Motion Transition Recognition (Falling Motion, Normal transition)	Self-Collected	Motion Transition Dataset	Motion Transition testing	1	96.52%

6.5.1 Smartphone location and Orientation

The RARMT framework is able to predict human physical activities accurately regardless of whether the phone is placed in the left or right front pockets and whether the phone is placed in a fixed orientation. It is also not sensitive to the different type of clothing (hence, pocket size variation) worn during data collection and the smartphone deviation/orientation from its original position throughout the experiment. This capability is proven via the benchmark testing using Lin's dataset, where F-score in the range of 91-97% on average was achieved (see Figure 6.1 in *Section 6.2.3*).

6.5.2 Human physical activities

Experimental results obtained using the both the benchmark datasets and DAG dataset have shown that the RARMT framework is able to predict basic low level human motions with high accuracy. For the case of using DAG dataset, F-scores of about 99% for single user experiments and 98% for multiple user experiments were achieved in cross validation testing (see Figure 6.2 in *Section 6.3.2.1*). Its high performing ability is further justified with a number of Leave-One-User-Out test experiments, which obtained F-scores in the range of 90-95%.

Furthermore, the framework has a high capability to recognize various high level complex motion transitions as well (see *Section 6.4.5*). Using a series of testing datasets collected, we justified RARMT's robustness in predicting these transitions with high accuracy.

6.5.3 User population age groups

The RARMT framework demonstrated high generalization capability to recognize motion from user profiles belonging to widely different age groups. For instance, in the Leave-On-User/Group-Out testing, the model is trained only using user profiles in the elderly age group and tested with the youth user profiles in Leave-One-User-Out testing. Despite the large difference in the demographics of the user profiles, the model is still able to achieve high performance of 90.25% for its optimal frame size (see Figure 6.3 in *Section 6.3.2.2*).

7 Conclusion

7.1 Summary of work done

In this project we developed a **Robust Activity Recognition with Motion Transition (RARMT)** framework. It is a system that offers a highly accurate capability in recognizing human activities, for both stable basic motions, like *Sitting* and *Walking*, and complex high level motion transitions, like *Standing Up* and *Falling Down*.

Within the RARMT framework, we first developed a *Data Preprocessor* that helps sample motion data at a predefined fixed frequency to ensure consistency. Next, we developed a *Feature Extractor* to extract selected deterministic features for classification. Thirdly, we developed a *Support Vector Machine Classification Model* to make accurate predictions of low level motions (i.e. Lying, Sitting, Standing, Walking, and Running). Finally, we designed and implemented a novel way of recognizing human motion transitions using the *Motion Transition Recognition Processor*. Specifically, this method detects transitions and performs Fall Detection using barometer's Standard Deviation and accelerometer's Signal Magnitude Area information.

We also collected our own dataset (**Dataset on Different Age Groups- DAG**) which consist of users from two distinct age groups- youth and elderly. The DAG dataset also captures human behavior in their most natural environmental settings (e.g. noises occurring from breathing motions, shifting of phone's position in loose pockets were included in the dataset). Using the DAG dataset, we were able to test RARMT's generalization ability in handling a diversified dataset with user profiles belonging to distinct age groups, and its performance in handling noisy data.

One of the main contribution of this project centers around the extensive experimentations conducted. We first perform benchmark testing to verify the RARMT's performance using externally sourced datasets, like the Lin's Dataset and Davide's Dataset mentioned in *Section 6.2*. Next, we used our self-collected DAG dataset for more vigorous performance testing, using test strategy like Leave-One-Out. Finally, we collected a series of testing data comprising of various sequence of motion transitions to test on RARMT's performance in recognizing complex motion transition, as well as to help establish distinguishing algorithmic thresholds for its Motion Transition Recognition Processor.

7.2 Future Work

For future work, we plan to build a Human Activity Daily Living Pattern (ADL) analytical system on top of our RARMT framework. In recent years, we have seen numerous innovative breakthroughs in the field of Human Activity Daily Living Pattern related technologies. The global smart home market is growing rapidly [24] and exciting smart home security, energy and surveillance applications which leverage on advanced technology like *home installed wireless sensing network* are emerging in the market today. Furthermore, with the increased sophistication and popularity in smart wearables, like the *Smartwatches*, we can now explore the possibility of detecting more complex human daily activities (e.g. cooking, washing, brushing teeth etc.) [25].

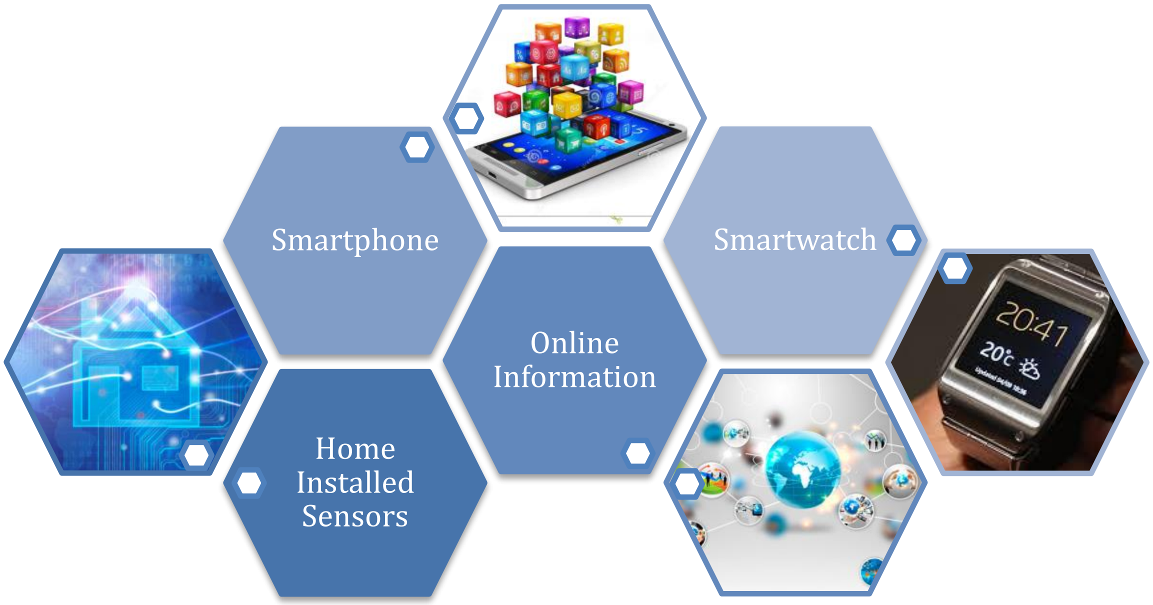
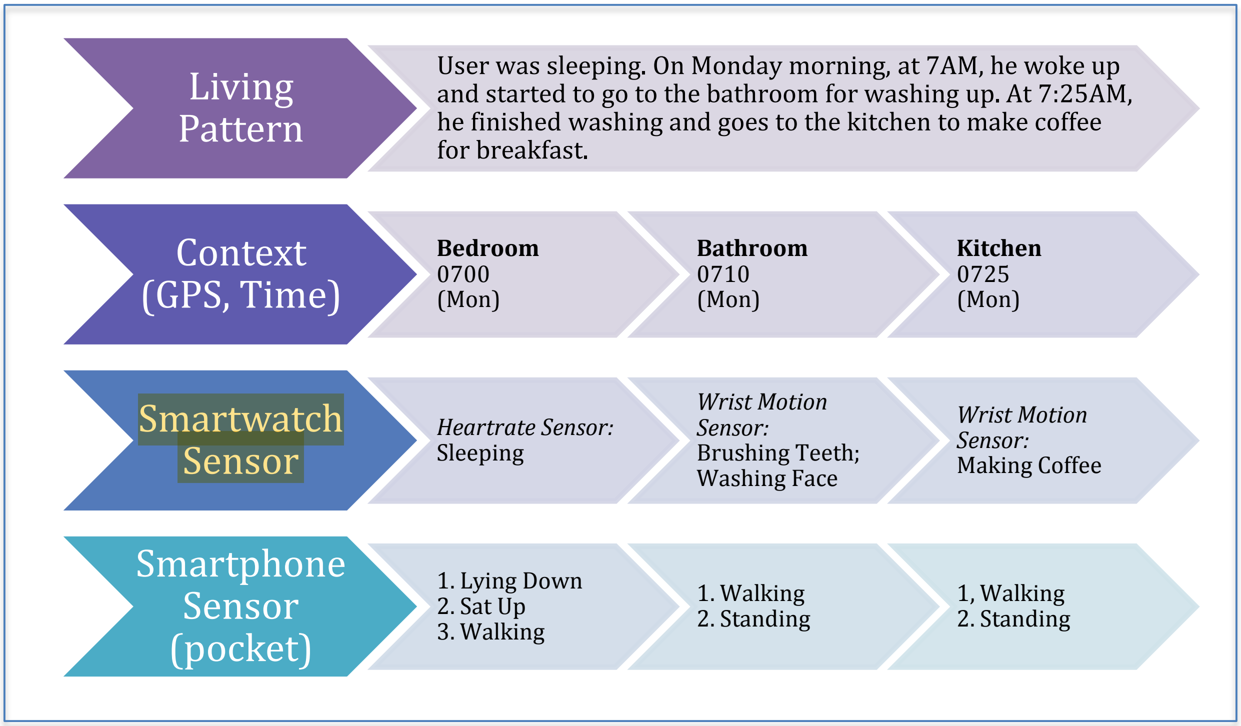


Figure 7.1: Activity Daily Living Pattern Synergy

The RARMT framework will serve as a strong foundation base that offers robust and accurate basic human motion recognition, providing reliable support for future application like the ADL system.

An ADL system uses various information sources (e.g. human installed sensors, smartwatch etc.) to recognize the living pattern of an individual (see Figure 7.1). For instance, the GPS location and Time of the day, coupled with RARMT's activity prediction, can allow the ADL system to infer a much richer context of the user who is performing the activity (see Figure 7.2). This gives us access to much more detailed information on the activities the user is actually performing. For example, the smartphone can only tell us that the user is standing. But with the smartwatch, the RARMT can detect that the user is actually brushing his teeth at the same time too.



Living Pattern	User was sleeping. On Monday morning, at 7AM, he woke up and started to go to the bathroom for washing up. At 7:25AM, he finished washing and goes to the kitchen to make coffee for breakfast.		
Context (GPS, Time)	**Bedroom** 0700 (Mon)	**Bathroom** 0710 (Mon)	**Kitchen** 0725 (Mon)
Smartwatch Sensor	*Heartrate Sensor:* Sleeping	*Wrist Motion Sensor:* Brushing Teeth; Washing Face	*Wrist Motion Sensor:* Making Coffee
Smartphone Sensor (pocket)	1. Lying Down 2. Sat Up 3. Walking	1. Walking 2. Standing	1, Walking 2. Standing

Figure 7.2: Activity Living Pattern Recognition Example

In this project, what we have achieved is to provide a robust activity recognition framework that served as a strong backend processor which can perform accurate activity recognition on a diverse set of user profiles from very distinct age groups. This in turn can support and fuel many more exciting breakthroughs in the application of human behaviour recognition arena.

References

1. Morris, M., Lundell, J., Dishman, E., Needham, B.: New Perspectives on Ubiquitous Computing from Ethnographic Study of Elders with Cognitive Decline. In: Proc. Ubicomp. (2003).
2. Lawton, M. P.: Aging and Performance of Home Tasks. Human Factors. (1990).
3. Consolvo, S., Roessler, P., Shelton, B., LaMarcha, A., Schilit, B., Bly, S.: Technology for Care Networks of Elders. In: Proc. IEEE Pervasive Computing Mobile and Ubiquitous Systems: Successful Aging. (2004).
4. Quak, June.: Human Activity Recognition and Tracking. FYP Report. Nanyang Technological University, School of Computing, Singapore. (2014).
5. Lester, J., Choudhury, T., & Borriello, G.: A Practical Approach to Recognizing Physical Activities. (2006).
6. Pei, L., Guinness, R., Chen, R., Liu, J., Kuusniemi, H., Chen, Y., et al.: Human Behaviour Cognition Using Smartphone Sensors. MDPI. Sensors. (2013).
7. Klinenberg, E., Torres, S., & Portacolone, E.: Aging Alone In America. Council on Contemporary Families, America. (2013).
8. Lifeline. Lifeline- Live A Bigger Life. Retrieved August 29, 2014, from *http://www.lifeline.com.sg/*
9. Tan, T.: More S'poreans living alone; trend seen rising. C3A. (2014, May 2).
10. Lester J., Choudhury T., Kern, N., Borriello G. Hannaford B.: A hybrid discriminative /generative approach for modelling human activities. In: Pro. of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 776–772. (2005).
11. Ravi N., Dandekar N., Mysore P., Littman M.L.: Activity Recognition from Accelerometer Data. In: AAAI, pages 1541–1546. (2005).
12. Bao L., Intille, S.: Activity Recognition from User-Annotated Acceleration Data. In: Proc. Pervasive pp. 1-17. (2004).
13. Parkka J., Ermes M., Korpipaa P., Mantyjarvi J. Peltolla J. Korhonen I.: Activity Classification Using Realistic Data From Wearable Sensors. IEEE Transactions on Information Technology in Biomedicine. pp. 119–128. (2006).
14. Lester J., Choudhury T., Kern, N., Borriello G. Hannaford B.: A hybrid discriminative/generative approach for modelling human activities. In: Pro. of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 776–772. (2005).

15. Maurer U., Smailagic A., Siewiorek D.P., Deisher M.: Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. In: Proc. Of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06) pp. 113–116. (2006).
16. Yang J.: Toward Physical Activity Diary: Activity Recognition Using Simple Acceleration Features with Mobile Phones. In: IMCE'09 Beijing, China (2009).
17. Sun, L., Zhang, D., Li, B., Guo, B., & Li, S. (2010). Activity Recognition on an Accelerometer Embedded Mobile Phone with Varying Positions and Orientations.
18. Chih-Chung Chang & Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.
19. Davide, A., Ghio, A., Oneto, L., & Xavi. (2012). Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.
20. Gupta, Jaafar, Fatimah & Bansal. (2013). Feature Extraction Using MFCC. In: Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.4, August 2013
21. Android Developer: Sensor Framework Overview. Retrieved from December 26, 2014, from *http://developer.android.com/guide/topics/sensors/sensors_overview.html*
22. Karantonis, Narayanan, Merryn Mathie, & Celler. (2006) Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring : IEEE Transations on Information Technology in Biomedicine Vol.10, No.1, January 2006
23. Asa, Weston. (2010) A User's Guide to Support Vector Machines.
24. R. Baburajan, "Home Automation Grows with the Evolution of Mobile Devices", Asia-Pacific Business and Technology Report, from *http://www.biztechreport.com/story/`730-home-automationgrows-evolution-mobile-devices*, 2011

25. Bruno, Mastrogiovanni, Sgorbissa, Vernazza & Zaccaria. (2013) Analysis of Human Behavior Recognition Algorithms based on Acceleration Data: IEEE International Conference on Robotics and Automation (ICRA) May 2013