System (for) Tracking Equilibrium and Determining Incline (STEADI)

Agastya Silvina

Dissertation 2014

Erasmus Mundus MSc in Dependable Software Systems

NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

Department of Computer Science

National University of Ireland, Maynooth

Co. Kildare, Ireland

A dissertation submitted in partial fulfilment

of the requirements for the

Erasmus Mundus MSc Dependable Software Systems

Head of Department : Dr Adam Winstanley

Supervisor : Joseph Timoney

July, 2013

European Commission

ERASMUS MUNDUS

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Master of Science in Dependable Software Systems, is entirely my own work and has not been taken from the work of the others save and to the extent that such work has been cited and acknowledged within the text of my work.

Agastya Silvina

Acknowledgement

I would like to thank supervisor, Dr. Joseph Timoney from National University of Ireland, Maynooth for all the help and support on the project.

Abstract

The goal of this project was to design and implement a smartphone-based wearable system to detect fall events in real time. It has the acronym STEADI. Rather than have expensive customised hardware STEADI was implemented in a cost effective manner using a generic mobile computing device. In order to detect the fall event, we propose a fall detector that uses the accelerometer available in a mobile phone. As for detecting a fall we mainly divide the system in two sections, the signal processing and classification. For the processing both a median filter and a high pass filter are used. A Median filter is used to amplify/enhance the signal by removing impulsive noise while preserving the signal shape while the High pass filter is used to emphasise transitions in the signal. Then, in order to recognize a fall event, our STEADI system implements two methods that are a simple threshold analysis to determine whether or not a fall has occurred (threshold-based) and a more sophisticated Naïve-Bayes classification method to differentiate falling from other mobile activities. Our experimental results show that by applying the signal processing and Naïve-Bayes classification together increases the accuracy by more than 20% compared with using the threshold-based method alone. The Naïve-Bayes achieved a detection accuracy of 95% in overall. Furthermore, an external sensor is introduced in order to enhance its accuracy. In addition to the fall detection, the systems can also provide location information using Google Maps as to the whereabouts of the fall event using the available GPS on the smartphone and sends the message to the caretaker via an SMS.

Contents

1 Introduction

This chapter provides a high level description of the thesis work. The motivation behind developing a fall detector is introduced. The technical problems, challenges, and the research questions derived are described here. Lastly, the objectives and structure of this thesis are given.

1.1 Motivation for Fall Detection System Implementation

Fall detection system has become a popular research topic because of the effects and problems that are caused by falls on the increasing population of elderly people, that is, 65 years of age and above, throughout the world [1].

Every year, one in every three adults aged 65 and older falls. However, it's not the event itself that we are most concerned with. The impact of the fall itself is the backbone of the problem. As people gradually growing older, they typically become frailer, more unsteady, and have slower reactions then their younger counterparts. Thus, they are more likely to be injured than other people, such as athletes or toddlers, who are also considered to fall regularly [41].

From all the fall events that have been reported among adults, 40-60% of those lead to major injuries, 30-50% are minor injuries, and 5-6% result in fractures .Unfortunately, many older adults who sustain fractures due to a fall never regain their previous level of mobility. Worst of all, 20% of fall-related hip fractures result in death within a year [1].

Speaking of which, the details as to how fall events can lead to other social and economic concerns are mentioned below:

"An injurious fall in a person over 65 can cost the healthcare system US$1049 in Australia (Hendrie et al.,2003) or US$3611 in Finland (Nurmi I. & Lüthje P., 2002). If falls rates are not reduced in the immediate future, the number of injuries caused by falls is projected to double by 2030 (Kannus et al., 2007). " [1]

Another common problem is that 54% people over 70 express a fear of falling, which results in a reduction in their physical and social activities that could possibly lead to a decline in their physical performance as well as decline in their cognitive performance. [42]

Other than the elderly, falls can also be caused by other circumstances, for example to epilepsy patients who are at risk of falling during a seizure due to loss of consciousness, as well as to people with the high risk of getting a heart attack or stroke. [43]

A fast medical treatment is desired in all such fall situations. However, the fall might immobilize the victim to the extent that they are not able to reach a phone to call for help. Moreover, if the person ends up in an unconscious state, they may not be able to inform their relatives/caretakers. Thus, an automatic fall detection system that requires no human witness to raise the alarm would be very beneficial.

1.2 Technical Problem and Challenges

As people live longer and want to retain their mobility, there is a need for assistive technologies to support them. Falls are a very common reason for injuries in geriatric patients.

A system that is able to detect a fall event is highly desirable. The crux of the problem is designing an algorithm that can differentiate reliably between falling and other mobility events [44]. Furthermore, it should be dependable to prevent false alarms but also not fail to raise alarm.

For the recognition of a fall event, some of the current technological solutions are costly to implement. We think that the best approach is to have some form of wearable sensor equipped with an accelerometer for collecting and processing movement data real- time. An accelerometer is an electromechanical device that will measure acceleration forces. If they are dynamic, they are by moving or vibrating the accelerometer [9]. There are three main challenges with this as described below:

- The first challenge is to determine when a fall happens. The accelerometer signal associated with a fall should be known. However, a fall can vary in terms of its direction and speed depending on the circumstances. Needless to say, the accelerometer signal can also vary across different people depending on their size.
- The second challenge is the system accuracy. The ideal system will able to accurately to detect every fall and should not give a false alarm. This is challenging because people have different activities of daily living (ADL), such as jogging, walking, sitting. These will also produce a signal on the accelerometer but should not be misclassified as a fall.
- The last challenge is the budget and time needed to develop fall detection system because this system should be implemented in a cost effective manner within a limited timeline and should be in a form that is affordable for many people. Thus, the reason behind choosing the type of sensor, the computing device, and the detection algorithm selected becomes another challenge.

1.3 Research Questions

From the motivation, technical problems and challenges mentioned above, we are able to derive the research question for this thesis work:

- Is it possible to have a technology for fall detection that is: affordable, robust, portable, reliable? Can this be implemented in a cost effective manner using generic computing device? That is, can it satisfy the requirements to be an assistive technology?
- Are we able to use a simple signal processing algorithm that is fast and can accurately detect a fall? Will another method, such as a statistical or machine learning technique, be needed to enhance the system accuracy?
- Can we use smartphone as a fall detection device that will reliably give an accurate assessment as to whether a fall has taken place?
- How can we contact a relative where the fall event has occurred? Will this work in both external and internal environmental scenarios? How reliably can our technology provide the information of the fallen person’s whereabouts?

1.4 Objective of The Project

The system developed intends to design and implement a device that can differentiate between the activities of daily living and falling. The device is categorized as a wearable device that should satisfy the basic requirements as mention below:

- The system shall be able to process the signal in real time.
- The system must be portable, light weight and easy to wear.
- The system must be implemented in a cost effective manner using a generic computing device.
- The system should be able to inform relatives/caretaker the information of the fallen person whereabouts.

Thus, the algorithm developed in this thesis is intended to differentiate between activities of daily living includes sitting, standing, walking and jogging. Since the system is design for geriatric patients, the activity of running vigorously may be excluded. The recognition accuracy should be high.

The algorithm should be implemented using an affordable, portable technology. The solution that immediately comes to mind here is a modern smartphone-based application that has the facility of an accelerometer sensor, either internally or externally, and would also be of reasonable cost.

1.5 Structure of Thesis

Chapter 1 introduces the motivation of this thesis work, along with some general information about the problem as well as a brief introduction to the approach used for our solution. Chapter 2 is the review, which critically evaluates related work in this area. Chapter 3 describes software development techniques as well as the system algorithm and the reason for using it as a solution. Chapter 4 gives the result of several tests conducted during the development of the system. Chapter 5 presents a summary of the work and conclusions along with some recommendations for future work.

2 Related Work

There are several approaches to implementing a fall detection system, and all of them have one similar thing. They use some type of sensor to continuously monitor the condition of the target user. This includes (a) video based fall detection, in which the environment is continuously monitored by a video Camera and the captured image is digitally processed. [8] (b) audio-based fall detection where the system will distinguish between sound frequencies, and detect the audio signature of a fall event in comparison to those of ADL, (c) wearable sensor based fall detection where the fall is detected by sensor devices attached using a generic computing device or bespoke/custom designed hardware to the person [2], and (d) an indoor smart floor pressure sensor that will differentiate between different pressures made by the person [1]. All these are reviewed in the following sections.

2.1 Video-based Fall Detector

Due to the recent advancements in image processing techniques and the availability of low cost cameras, video based far detection become more feasible and has turned into one of the important areas for development [8]. There are several different approaches for implementing such systems. Typically, a video-based sensor uses video surveillance to monitor the user’s condition as well as digital image processing applied to the real-time video-captured imaging in order to detect whether or not a falling event has occurred as shown in figure 2.1 where the movement of the person in the video is captured by the form and position of the shape representation in the panel below. A reliable system and efficient surveillance video system needs to be robust. Therefore, the correct choice of camera, the position of the camera, and appropriate video compression method are important aspects to be considered. [9]. One simple method is based on the analysis of a moving object by monitoring its bounding box's aspect ratio. However, the ratio itself can also be altered due to blocking by other objects as well as the relative position of the person with respect to the camera. Others including, Lee, [10] detects a fall by analysing the shape and 2D velocity of the person. Vaidehi in [8] uses another alternative model for detecting fall. They implement the system by using static parameters of the person under observation such as their aspect ratio and then monitor the continuous change of their inclination angle. This method is computationally less intensive as it does not involve any velocity computation.

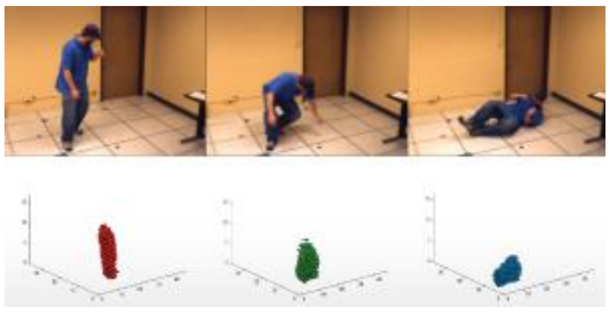


Figure 2.1: Example of video based fall detection system.

The advantages of the video surveillance system is that it provides a secure and quick intervention for senior citizens, and it's claimed that the video-tape images before fall occurrences can supply important information to give a better understanding of the origins of the falls [11]. However, because of the importance of camera position it makes this approach difficult to be portable. Furthermore, not all the users are in favour of this type of intelligent video-monitoring system. Some of them are concerned about the safety and privacy of the transmission of the video images [11].

2.1 Audio-based Fall Detector

Another technique that is used for detecting a falling event is based on audio processing. One of the research, papers by B. Uğur Töreyin, [12] used both audio and video signals to discriminate between falling and other ADL activities. A typical stumble and fall produces high amplitude sound as shown in figure Y.

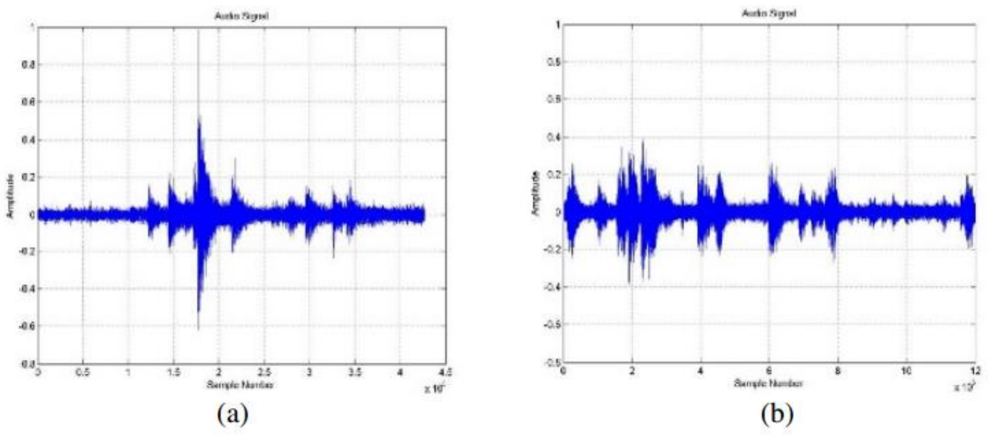


Figure 2.2. (a) a typical amplitude of fall signal, and while a normal ADL signal, person talking is shown in figure (b)

In this particular work, a wavelet transform is applied to the audio data. Their results showed that the wavelet coefficients of a fall sound are different to those for ADL. Hence, it's possible to do some signal processing to extract wavelet features, and then by using one of machine learning techniques (i.e. HMM) classification of wavelet values can be achieved to say whether a fall has happened or not.

Other approaches of using an audio signal for detecting fall is done by Xiaodan, [13]. They use the Gaussian Mixture Model as their baseline. The system will identify the existence and approximate occurrence time of falls. Each audio signal then is classified into several categories, mainly fall or various type of noise. To better distinguish falls from all competing noise sounds, they model falls and classes of noise in the living environment. Each class should have a sufficient number of instances in the training data. Each class is relatively distinguishable from others. The classes are chosen to better distinguish fall from noise. Though the signal generated from fall event can be distinguished from ADL, unfortunately this approach will not effective in an outdoors environment where audio signals can come from various different sources that will confuse the detection algorithm.

2.2 Wearable Sensor-based Fall Detector

In order to implement a fall detection system, a number of algorithms based on a wearable accelerometer and gyroscope have been proposed. One approach in common use is to discriminate between ADLs and falls by a threshold value (for acceleration and angular velocity). Nyan et al.[14] monitored fall events by utilizing a 3-axis accelerometer and 2-axis gyroscope strapped into thigh and waist. Other, such as Bourke and Lyons used a gyroscope mounted on the torso to measure the pitch and roll angular velocity data. Thresholds were again used to distinguish between falls and ADL. In [15] the PerfallD team measured the fall event by using a smartphone with a magnetic accessory. Before a fall happens, the magnetic field (MF) values are on a relatively stable high level. At the beginning of a fall, the MF value decreases slowly, but suddenly goes higher, making a salient convex shape, and then decreases quickly. After it reaches the minimum, it turns back to higher value that is much lower than the value before falling, and then remains relatively stable afterward[15]. Similarly, Yavuz et al. implement the system using an accelerometer and apply the discrete wavelet transform (DWT) to the signal captured rather than solely using a threshold-based method. They claim that, compared to a simple threshold-based algorithm, using wavelet transforms achieve a better true positive performance while decreasing the rate of false positives drastically [16]. Jay, use a slight different algorithm that was based on the observation that when a person falls, he or she will have different position before the fall to after [11]. The simplest example would be from standing upright to lying flat, give an orientation change in around 90 degrees. From the orientation and time of falling information, some information can be extracted. So far, the wearable sensor fall detector is considered to be a better approach to detect the fall event both indoors and outdoors. However, the current systems that are developed are limited only to differentiate ADL with lower impact such as walking, sitting and standing to an actual fall event.

2.3 Indoor Smart Floor Fall Detector

An indoor smart floor sensor is another option for detecting fall [1]. One possible implementation of this smart floor sensor is to do gait assessment. Human gait is a complex balance challenge, in which a person must initiate a fall forward and recapture their momentum through the appropriate placement of their leading foot [1]. Deficits in balance and gait are the most predictive risk factors for falls. The potential for loss of balance during walking is significant, and it is not surprising that over 50% of falls in older adults occur during walking [45]. The *GAITRite* system from CIR System Inc is one of the most popular methods for the objective measurement of Gait parameters in a clinical setting (using devices such as a force plate and EMG system) The *GAITRite* system automates the measurement of the temporal (timing) and spatial (distance) parameters of Gait via an electronic walkway connected to a PC. [46] The standard *GAITRite* system is approximately 366cm long and contains 13824 sensors spaced at 1.27 cm apart. As the patient walks across the walkway the system dynamically captures pressure data with respect to each footstep and calculates both temporal and spatial parameters for the walk.

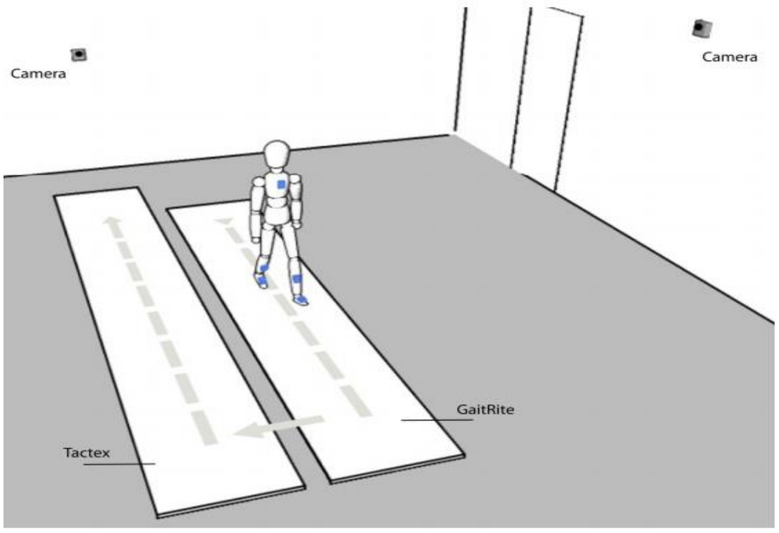


Figure 2.3. Overview of the TRIL Analysis Platform

The TRIL Centre developed a gait analysis system (Figure 2) which uniquely combines a floor-mat sensor, body worn sensors, video capture, and a software user interface for clinicians [1]. In addition to fall detector, this system is able to give feedback about user human gait and it would be possible to train a person to have better gait posture [1]. However, this system is not portable and hard to develop in a cost effective manner.

2.4 Summary

From the analysis in the previous sections it is clear that for the detection of a fall, a wearable device is much better than a camera-based system as well as the floor-mat based system due to the free user mobility that is being provided. The wearable device is very suitable for both indoor and outdoor uses. Thus, we will adopt the same approach for your system. Furthermore, it is definitely less expensive if this wearable device is no custom made but rather an off-the-shelf system. Thus, we choose a mobile smartphone as it has a built-in accelerometer but can also be extended to connect with external sensors. The next question is where to place the device and some research has been conducted to determine the ideal location on the body for the wearable sensor to achieve the highest accuracy. Therefore, we should choose the sensor placement base on previous related work by PerfallD team [9].

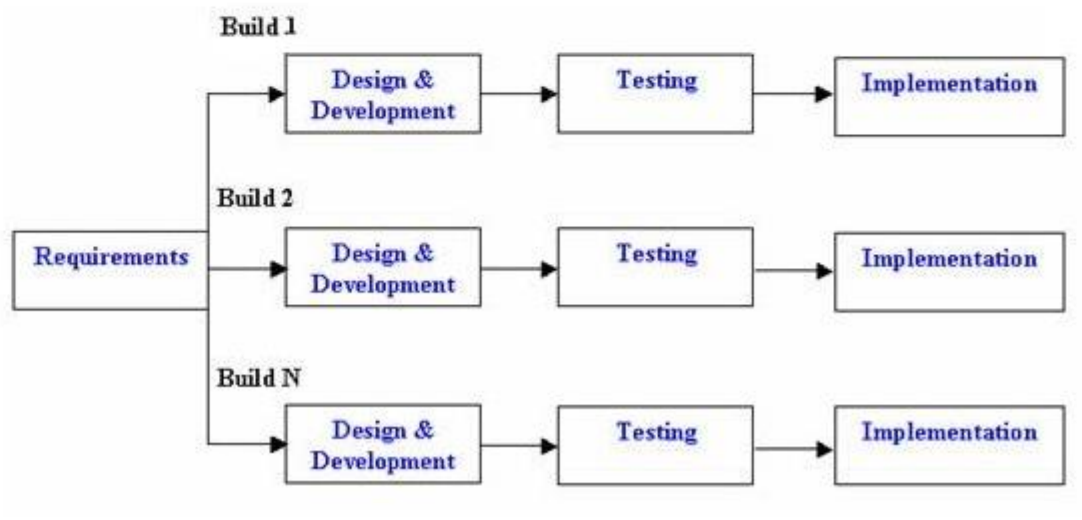
3 Solution

In this Chapter, both the software and the hardware components for STEADI are introduced. We describe how the accelerometer data was collected from the sensor is processed to determine the fall event. This includes the algorithms to clean the signal and find the features that are sent to the classifier to determine if a fall has occurred. Two types of classifier are discussed. The software development approach is explained ahead of these details.

3.1 Software Development

In order to fully build the system, an Iterative and Incremental development process is used. The idea behind this method is to create a complete system through repeated cycles (iterative) of development of small features (incremental) of the full system [14].

In this model, the whole requirements are divided into different incremental steps. Here, the waterfall model is repeatedly applied to enhance STEADI as a fall detection system. Each increment produces a new feature and progression in the work [47]. An iteration has several phases: requirements, design, implementation and testing phases as shown in Figure 3.1 below. A working version of software is produced during the first module. The complete system is achieved by doing several increments.



Source: [47]

Figure 3.1: Incremental Life Cycle model

The incremental model is used as it will have several advantages compared to the other software development process models as follows [15]:

- The working software can be generated quickly and early during the software life cycle
- It is easier to test and debug during a smaller iteration

Our detector system, STEADI, which was developed is divided into 5 increments:

1. The first increment was building a system that is able to capture the user activity signal in real time
2. The second increment was building the software needed to do the signal processing and implement it in the real time generic computing device. At the end of this increment the system should be able to classify the user activity using a threshold-based classification method
3. The third increment is to enhance the fall detector system. However, instead of using the threshold-based classification method, another classification technique called Naïve-Bayes classification is included.
4. After the system was applied on a mobile device, other features such as location capture and to inform a caretaker/relative that the fall event has occurred are implemented.
5. Lastly, after the system was implemented in a mobile device. We built a similar system that also takes information from another external sensor device and compared the result with the original system built in android.

3.2 Requirement specifications

The application was developed to provide an easy access, user-friendly system to detect if a fall event happens to the user. The activity signal is captured using the built-in accelerometer the device provides. If the fall event has been detected, the system informs the registered caretaker/relative of the time and location of the occurrence. Next, the system and user activities are shown using UML representations.

3.2.1 Use case

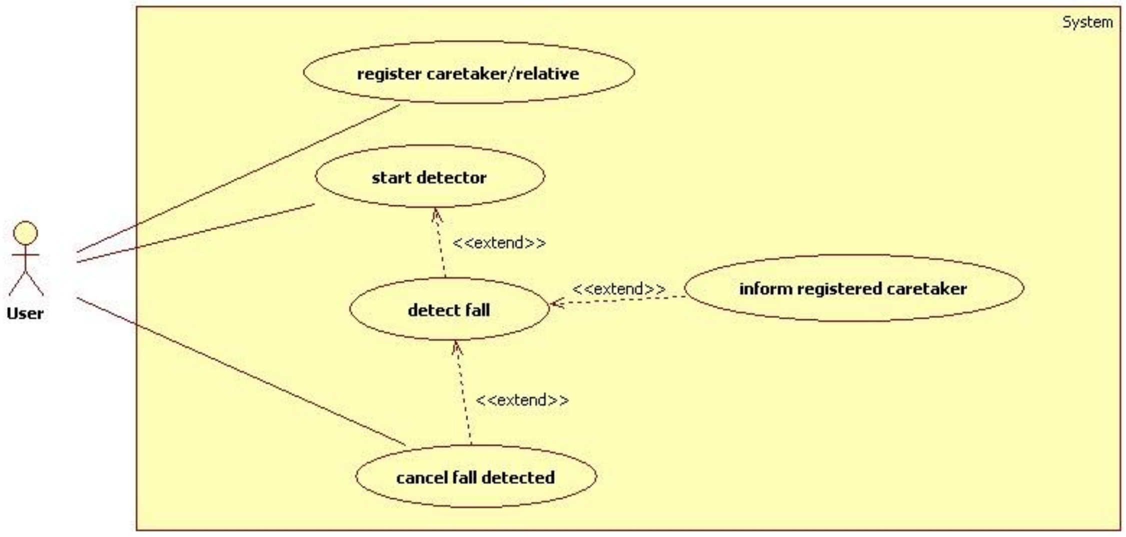


Figure 3.2 Use case for STEADI fall detection system

The use start detector represents the initialisation of the sensor activity of the system. It is assumed that the user has already launched the application. The system also consists of other menus such as registering the care taker. These are shown in the diagram in Figure 3.2.

3.2.2 Analysis and application design

The interaction with the system can be visualized using a sequence diagram. The application acquires the real-time signal from the built-in accelerometer. Then, the signal captured is analysed using some type of classification technique. Every time there is a change on android sensor, the system will try to detect an impact event, and as the impact happens further signal processing is needed to determine if the impact was a falling event. The following activity diagram in Figure 3.3 shows an outline of how the system detects falling events.

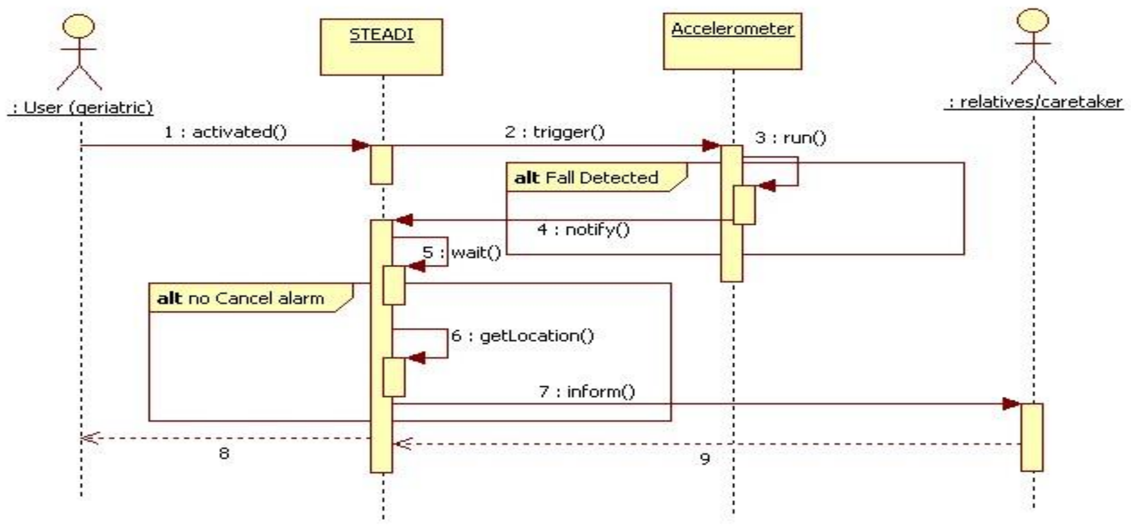


Figure 3.3: STEADI sequence diagram

First, the user will have to launch the application before interacting with the system. Here, the user needs to activate the sensor to start STEADI reading from the accelerometer. While the STEADI service is running in the background it keeps doing signal filtering and classification of each buffered signal into either a non-fall (ADL) or fall event. Once a fall event is detected it triggers the alarm to notify the user. Once the fall has been detected, the user is able to push a cancel button within 20 seconds if it is a false alarm. We implemented STEADI with duration of 20 seconds because it is considered to be enough time for users who are slow at operating their phone because of their age [10].

3.3 Application Design

This section describes how STEADI algorithm along with the signal filtering and classification are implemented.

3.3.1 Determining a Fall

Due to the in-built accelerometer being solely used as the source of data to determine fall events, we rely greatly on the quality of data post-processing techniques such as signal filtering to enhance the noisy signal coming from the accelerometer. An outline of the process is shown in Figure 3.4.

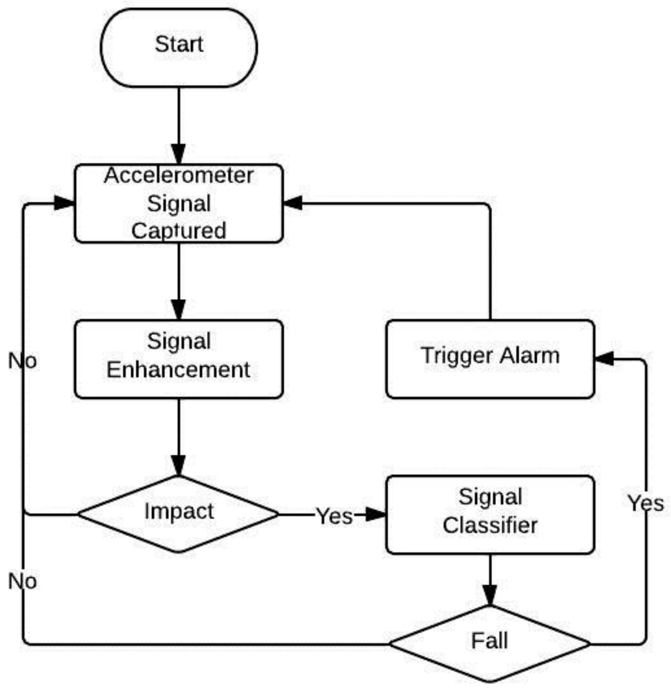


Figure 3.4: Fall detection Algorithm

3.3.2 Choosing a classifier

The main function of the system is to detect a fall. In order to do so, we need a classification system to determine the state of the user in terms of fall and non-fall. Classification is the problem of identifying to which of a set of categories (sub-population) a new observation belongs to [16]. Once the sensor signal is collected, the raw data is then processed. This will be then scanned for particular features that can be categorized to a class. This section describes the different kind of classifiers and outlines the reasons we chose the Threshold-based technique and also the Naïve-Bayes as our classification methods.

3.3.3 Threshold-based Classification

This method is most often used to differentiate between each state by a simple amplitude value. This method is very easy to implement. However, the presence of noise in the signal, and if a fixed threshold applied, make it hard to adapt for different individual cases [21]. It is used in our system as it can serve as a starting point for other more complicated methods.

3.3.4 Statistical classification

While threshold-based classification can be useful a classification algorithm that involves some statistical properties of the signal should improve the accuracy. Such classification methods varies from threshold -based classification to more complex algorithm e.g. Naïve-Bayes classification, artificial neural network or Hidden Markov Model (HMM).

In machine learning, the classification algorithm are categorized into two classes: supervised and unsupervised. Supervised means that labelled training data is provided to the classifier so that it can learn how to find the classes before the classifier is given new data. Unsupervised means that the classifier must analyse the data itself to find the classes that are present in it. In the subsections below are descriptions of some of the typical machine learning techniques for classification.

3.3.4.1 Artificial Neural Networks

Artificial Neural Networks are computational models that are inspired by the connections between natural neurons in the brain. Artificial neutral networks, ANNs basically consist of inputs (like synapses), which are multiplied by the strength of respective signals [48], and then are evaluated by a mathematical function which determines the activation of neuron. The advantage of ANNs includes that they requiring less formal statistical training, and that they are quiet flexible to classify a large range of different activities with a highly accurate result if correctly implemented. However, some of the networks are difficult to implement and sometimes can involves a great computational burden.

3.3.4.2 Support Vector Machine (SVM)

The SVM is a machine learning algorithm which solves classification problems using a flexible representation of class boundaries. SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces [17]. Though it is quiet powerful to work with a noisy datasets, the SVM is slow to train with large dataset as well as sometimes having difficulties to set their parameters.

3.3.4.3 K-Nearest Neighbour (KNN)

KNN is one of the simplest classification methods and works well when there is little or no prior knowledge about the distribution of data. It uses the training set directly to classify an input when an input is given [18]. There are some points to note for the advantages and disadvantages of KNN: this method can be developed fairly quickly to cluster a range of different activities and the learning progress is simple. However, classification is often time consuming.

3.3.4.4 Hidden Markov Models (HMM)

HMM is a statistical technique in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. In a HMM, the state is not directly visible, but the output is [19]. Despite of its wide usage for classification, a HMM needs to be trained on a set of seed sequences and generally requires lots of sample data. The training involves repeated iteration of the Viterbi algorithm, which can be quiet slow.

3.3.4.5 Naive Bayes Classifier

A Naïve-Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with the (naive) independence assumption. A naive Bayes classifier can be trained very efficiently in a supervised learning setting [20]. The classifier assumes that the existence (presence or absence) of some particular features of a class has no relation to the existence of other features. Even if these features are not independent with each other, a naïve-Bayes classifier considers all of these properties to independently contribute to the probability [20]. The advantage of using the naïve-Bayes classifier for a system lies in the fact that the naïve-Bayes is able to classify data only with a small amount of training data. This Naïve-Bayes method is considered to be a better classifier than other in an environment where only a low level of training data is available. Thus, in spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have been shown to work well in many complex real world situations [27].

3.3.5 Accelerometer Signal Processing

"Signal processing is an area of engineering and mathematics that related to operations on analog signals as well as digitized signal." [49].

In general, signals can be captured through various devices and methods based on the system specification and needs. The term noise in signal processing is a general term for unwanted, unknown modification to a signal during capture, storage, transmission, processing, or conversion. Noise will also carry no useful information and interferes significantly with the integrity of the signal that's being observed. Noise reduction, the recovery of the original signal from the noise, is a very common procedure in signal processing systems and this process of removing noise is usually done by implementing filters. The following subsections will look at the signal processing implemented in the STEADI system before the classifier to enhance the signal from the accelerometer.

In figure 3.5, the block diagram shows the two step of pre-processing the signal before extracting the signal feature.

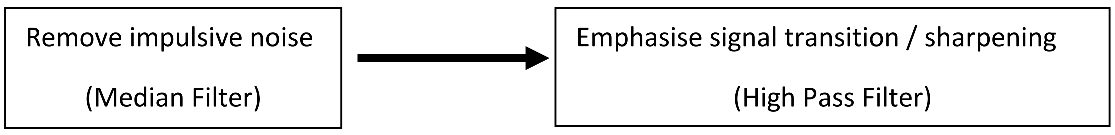


Figure 3.5 STEADI signal filtering's block diagram

3.3.5.1 Buffering technique

Before we process the signal captured by the built-in accelerometer, the raw signal need to be pre-processed to in order to extract some unique features from the signal. Here, a buffering technique is used to divide the signal captured into smaller segments, then the filter and classifier will be applied independently to each buffer. Each processed buffer gives the features that need to be classified. Without the buffering technique, the algorithms would process each data sample in succession. This could be considered to be ineffective and computationally wasteful as data from more than a few seconds past is irrelevant in determining whether a fall event occurred.

3.3.5.2 Buffering

There are several different buffering techniques. We use sliding window technique due to its simplicity. The sliding window approach does not required complex processing of the signal, thus is ideally suited to real-time application. With this method, the signal is divided into fixed length frames that are overlapping. Due to its simple method of implementation, most activity classification studies have employed this windowing technique [22]. In figure 3.5

below is shown an example of raw accelerometer data with a windowed frame of data highlighted by vertical bars at 750ms.

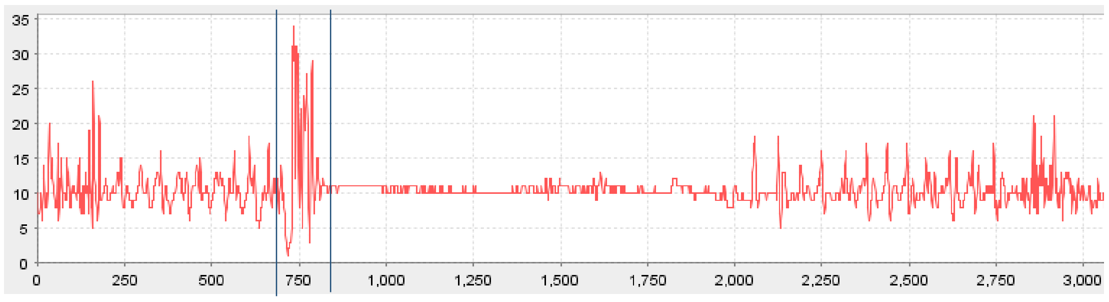


Figure 3.6 Sliding window technique applied to raw accelerometer data

3.3.5.3 Signal Filtering

Filter is a procedure to help removing unwanted component, such as noise from a signal [50]. Filter are usually specified in the time domain by their impulse response or in the frequency domain by their frequency response.

3.3.5.3.1 Median Filtering

Median Filters are a popular signal processing tool that are used in various applications like image and Speech processing, sound analysis, vocal separation and audio noise reduction [23]. One of widely use of median filter is to remove crackle sounds from vinyl recordings [39]. Median filter are usually used for reducing impulsive noise, enhancing the signal while preserving the signal shape.

The algorithm for median filtering consists of the following steps [64]:

1 Take an array of data determined by the length of the median filter.

2. Order the elements by their numerical values.

4. Take the middle element of selected array as the median value.

One issue is to deal with edge effects when applying the filter to a set of data. As the filter engages and disengages with the signal, padding with zeros needs to be done at the beginning and end of the signal as the filter buffer is just filling up with signal values.

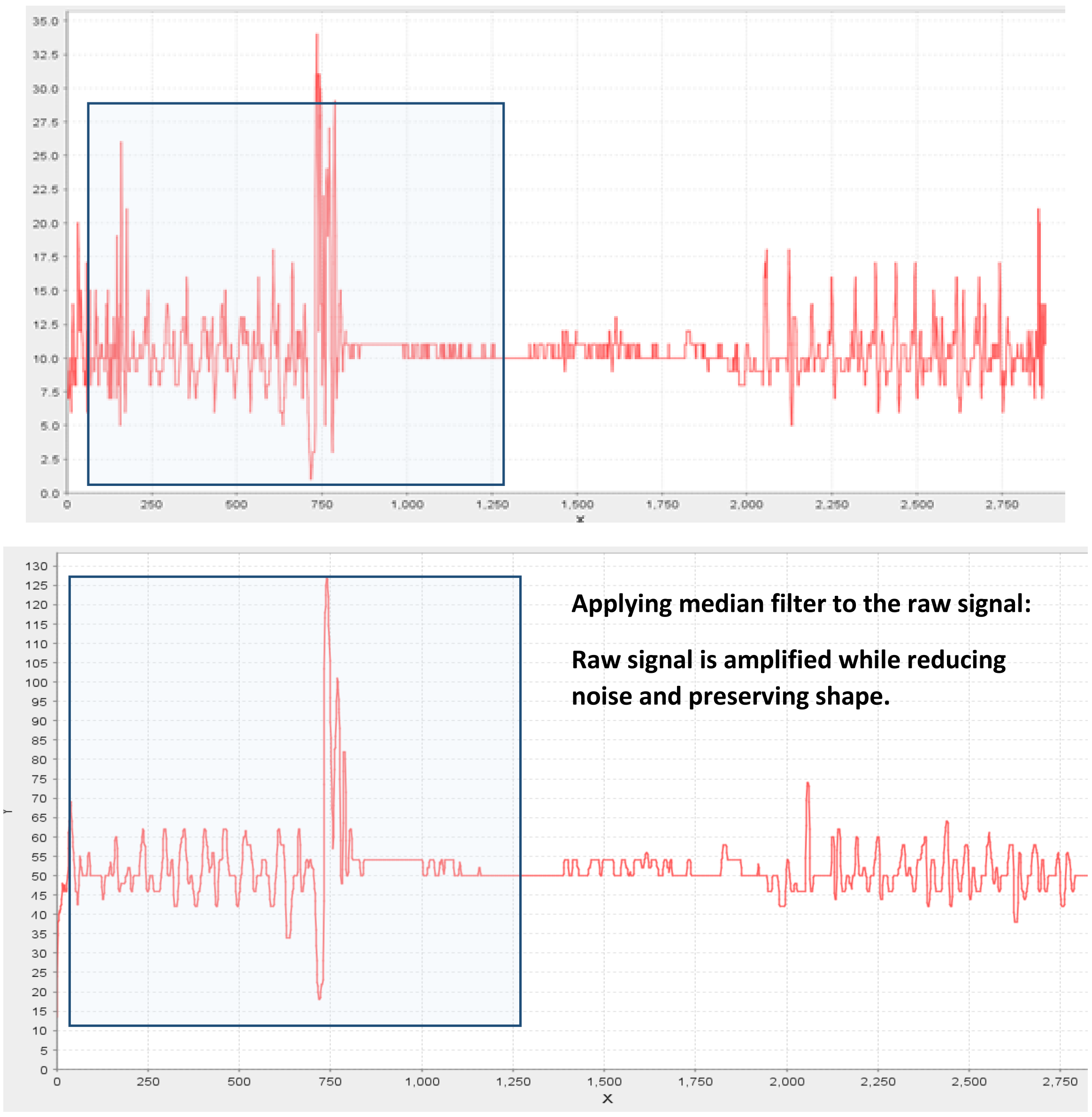


Figure 3.7: Application of the median filter to the raw accelerometer data

After the signal is enhanced using median filter, it's become easier to find impact features. In figure 3.6, it is shown that the median filter enhances the noisy signal in the upper panel in the region enclosed by the box, by removing impulse noise while preserving the signal giving the filtered version in the lower panel. Hence, this technique helps to decrease any false positives and false negatives in the fall detection as the signal features become more distinct between those that indicate a fall event and those that are due to noise.

3.3.5.3.2 High Pass Filter

Another filter that we used for processing our captured signal is a High-pass filter (HPF). A High pass filter is a filter that will block any signal components which exist below its cut-off frequency [50].

The accelerometer in android is influenced by the gravity. Here, high pass filter was used to reduce the noise from the gravity as well as sharpening the signal by removing the unwanted low frequency noise [67]. Figure 3.8 show that we are able to reduce the noise and sharpen the fall signal even further by reducing the low frequency noise.

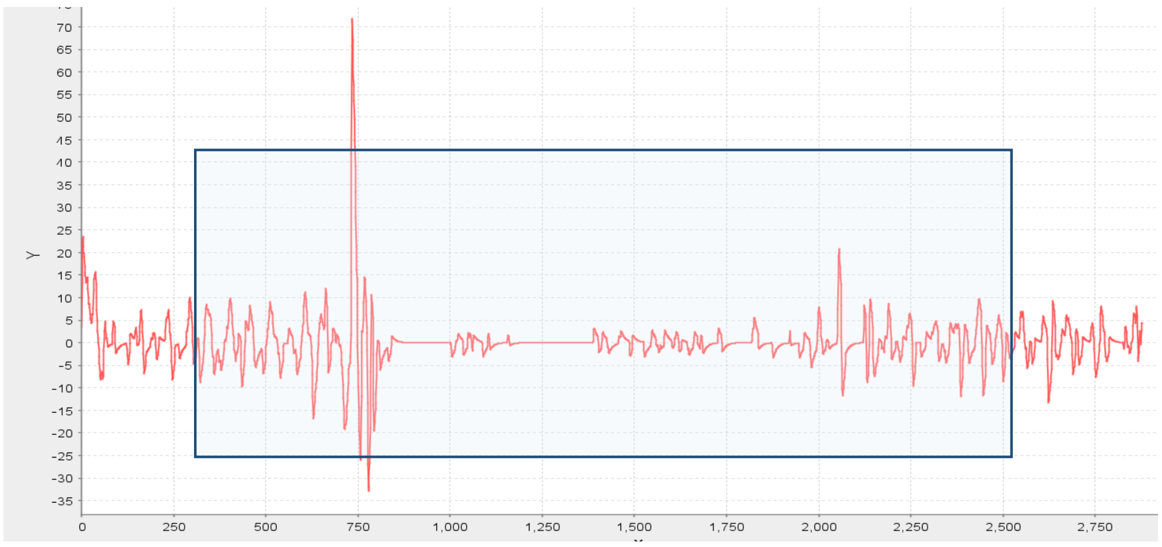


Figure 3.8: Application of the median filter and high pass filter to the raw accelerometer data

3.3.6 Extracting features

Following the signal enhancement procedure, we can extract signal features from each buffer of signal data. The intention behind this approach is to reduce the information complexity from that of the complete signal to a set of easily computed features, and hence we can increase generalization capability of the classifier [36]. The following figures show three different features we could get from the enhanced accelerometer signal given in Figure 3.8: the maximum amplitude value in each buffer (Figure 3.9), the mean value of the amplitudes in each buffer (Figure 3.10), and the distance between local maxima of signal maximum value (Figure 3.11).

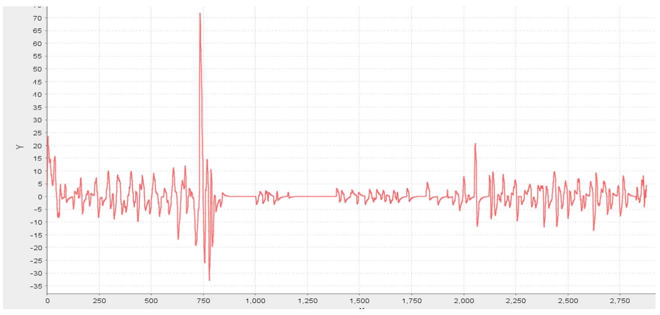


Figure 3.9: Enhanced Accelerometer Signal

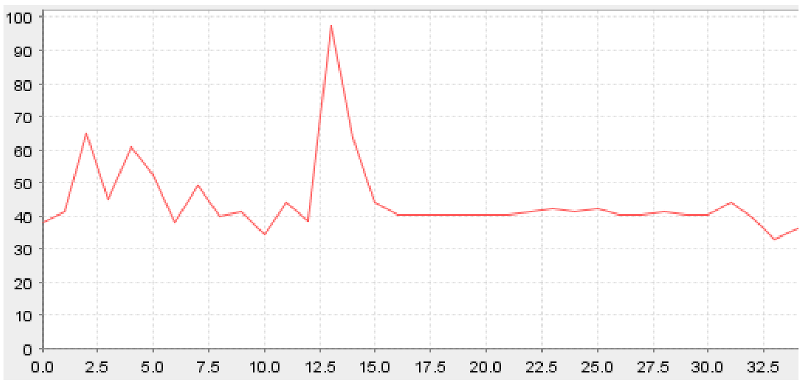


Figure 3.10: Maximum amplitude for each buffer

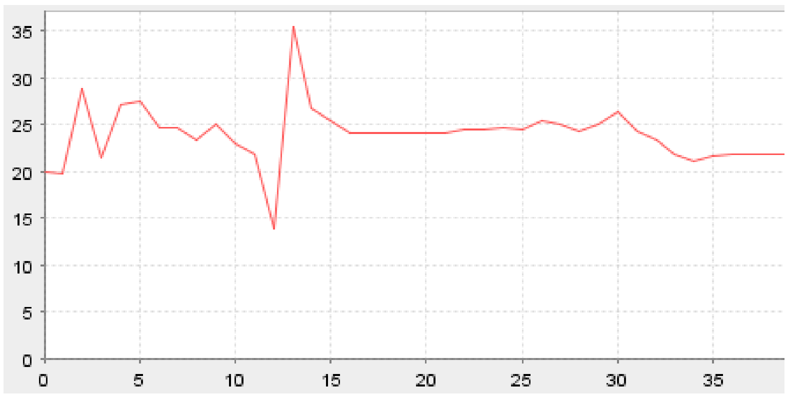


Figure 3.11: Mean amplitude value for each buffer

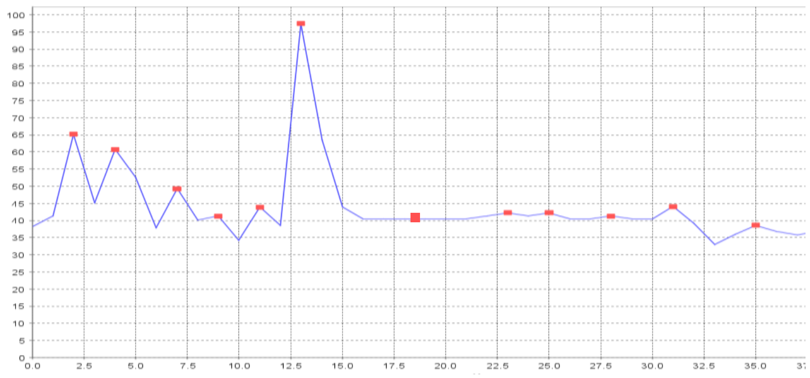


Figure 3.12: Local maxima from the Maximum buffer amplitude feature

A strong candidate feature set was found to be the mean and maximum amplitude value of each buffer. The maximum value of the buffered signal is being used because its value essentially captures the amplitude spike in the accelerometer signal associated with a severe movement such as a fall. The mean amplitude of the buffer is used to find the typical amplitude value within one buffer [51]. The mean value summarises the array of amplitudes in the buffer into a single value. This helps to differentiate between a sudden and transient amplitude spike in the buffer amplitudes against a consistently stronger set of amplitude values in a particular buffer that may be associated with a vigorous ADL activity, such as jogging and walking down stair that might produce spikes that are similar to, but not exactly the same as those associated with falls [11]. The last feature is the distance between the local maxima of the maximum value of the buffered signal feature. With the peaks identified in figure 3.11, STEADI will be able to compare the feature before and after an impact has been detected. Again, it assists in checking whether the user is still engaging in some type of ADL.

3.4 Analysis

The characteristic features of a fall event and activity daily living (ADL), using the features of maximum buffer amplitude, mean buffer amplitude and distance between the local maxima peaks can also be used to differentiate among a number of ADL events such as standing, walking, small jogging, walking to sitting, or using the stairs. This section shows how the feature set for several different activities has the potential to produce an impulse similar to fall events and how we differentiate those activities with fall event itself.

3.4.1 ADL Signal Features

Due to time constraints and practical limitations in carry out the tests, the variety of fall types between forward fall, backward fall, leftward fall, and rightward fall have not been examined. During different falls that were measured, the maximum value of buffer is always very distinct. Thus, it is possible to solely use the maximum buffer value feature with the threshold-based classifier [26]. However, sometimes other events produce a similar magnitude of maximum buffer values to fall events. Thus, the other features are also used to differentiate the event of falling from other ADL.

Table 3.2 below shows a typical enhanced accelerometer signal along with features extracted from various activities. The graphs show how the spike in the features associated with an impact detected can be used to make sure that the user has fallen. In the graphs for the fall event it is shown that the amplitude of the local maxima peak will decrease significantly compare to the other ADLs.

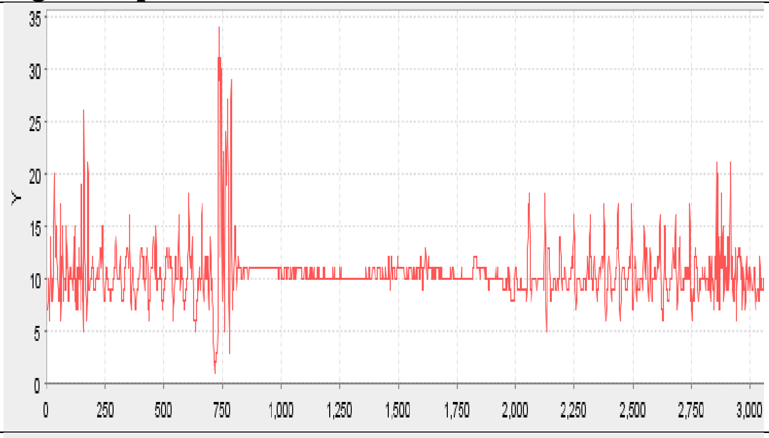
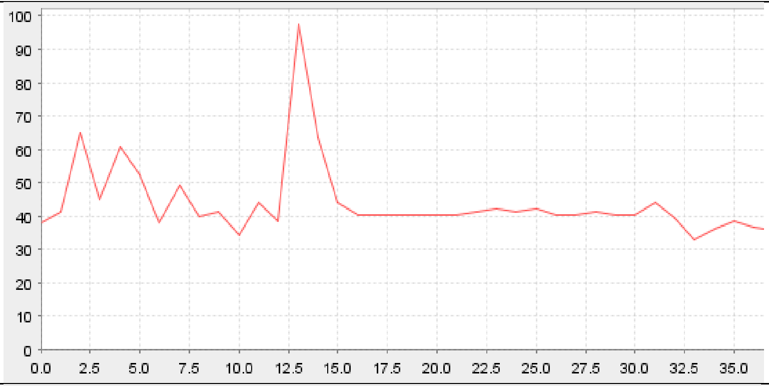
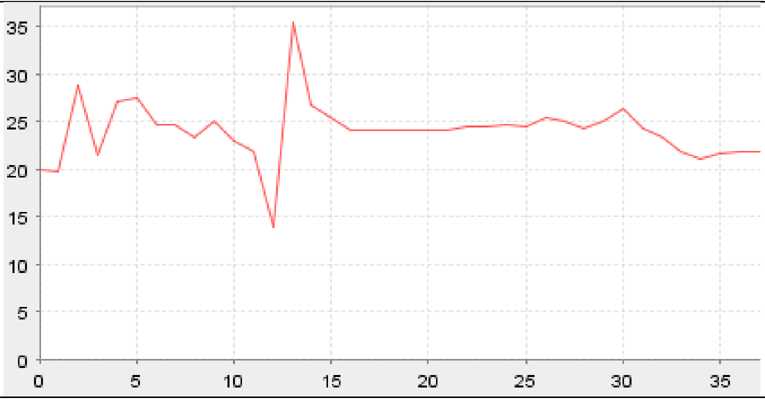
Activity	Signal Captured	Remark
Falling		Typical signal captured from android accelerometer
		Maximum value extracted from signal captured
		Mean value extracted from signal captured

Table 3.2 (a) fall signal features.

Table 3.2 (a) clearly show the anatomy of falling. First the user engages in activities of daily living, then the impulse signal show that the user was in free-fall and makes an impact on the ground.

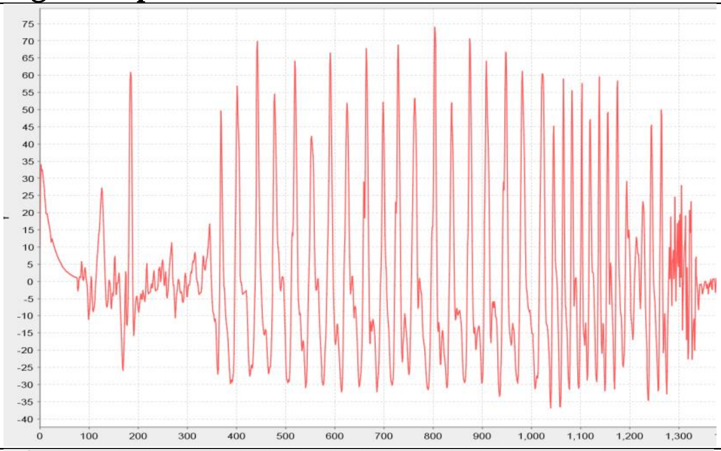
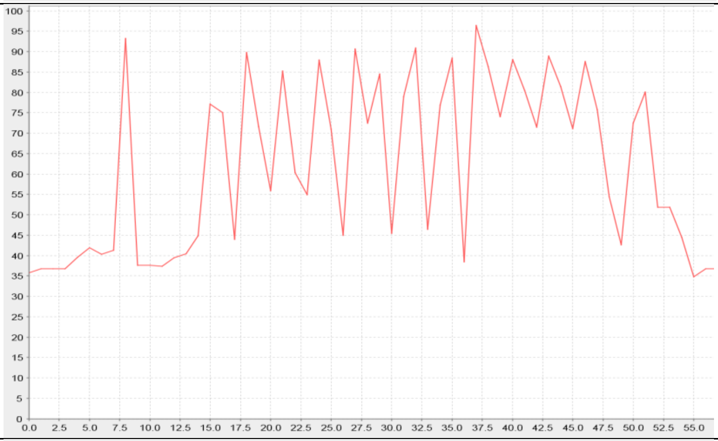
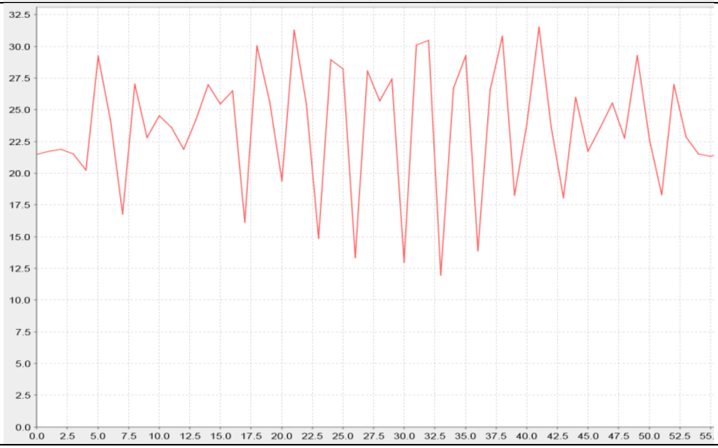
Activity	Signal Captured	Remark
Jogging		Typical filtered signal captured from android accelerometer
		Maximum value extracted from signal captured
		Mean value extracted from signal captured

Table 3.2 (b) Jogging signal features.

Here, the waveform clearly shows that running generates almost as high an acceleration signal as falling. However, the jogging signal high acceleration looks to be continuous compared to the fall signal that has lower acceleration value after impact.

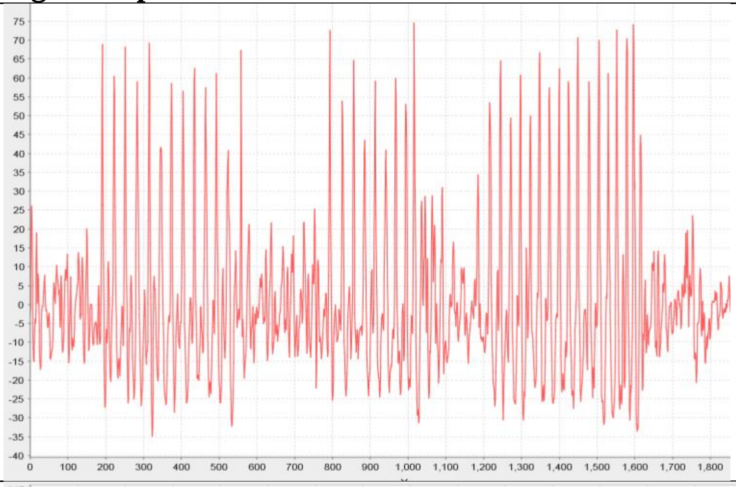
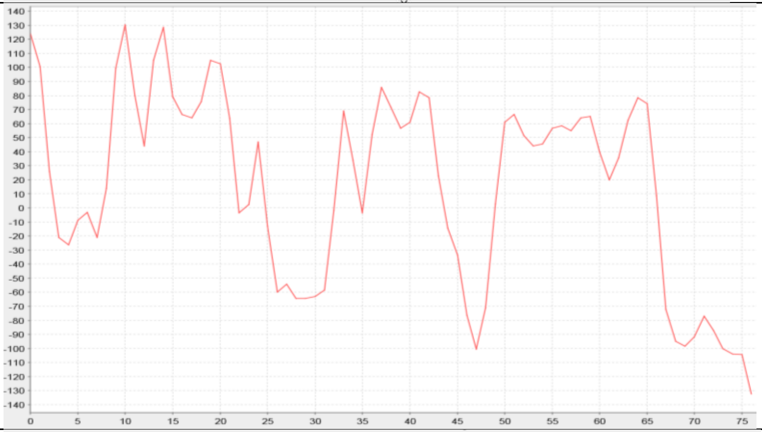
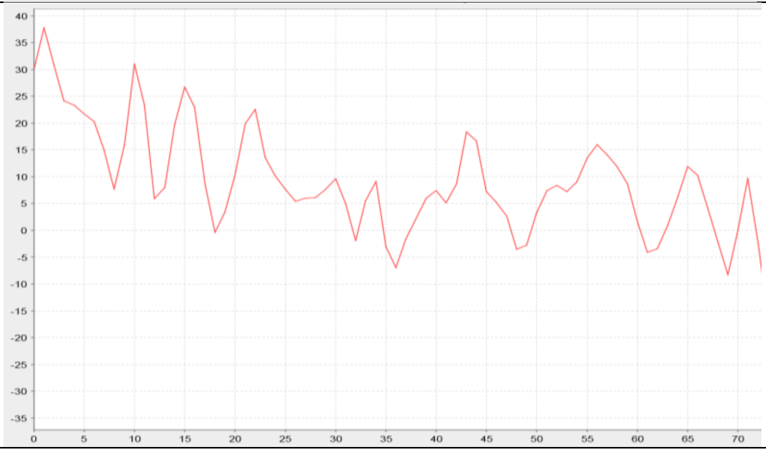
Activity	Signal Captured	Remark
Walking down stair		Typical filtered signal captured from android accelerometer
		Maximum value extracted from signal captured
		Mean value extracted from signal captured

Table 3.2 (c) Walking down stair signal features.

Although the signal clearly also generates almost as high an acceleration as falling. However, the high acceleration was followed by a brief low acceleration, unlike the fall signal that has a longer low acceleration after impact.

3.5 Classification

In this work we implemented both Threshold-based classification and Naïve Bayes Classification. Then, we compared both of them to determine if it was necessary to implement the statistical classification method within application. As for the statistical analysis approach, the Naïve-Bayes classification was implemented because of its high accuracy given a small set of training data [20].

3.5.1 Threshold Based Classification

The threshold-based classification is the simplest way to categorize the event of falling. Here, we set the Threshold based on the maximum amplitude of each buffer. Once we recorded the training data, we could simply use the value directly to determine the event of falling. Initially, we set the threshold to 80 m/s^2 as our training data showed that the average maximum value of falling event is around 85 to 90m/s^2. If the data captured exceed that value, we determined that a fall event has taken in place. Additionally, because different individuals might require a different threshold [9], for example, a person with a greater weight and height might need to have higher threshold value [26], In STEADI, we have the option to set the sensitivity of the application. This sensitivity option is used to decrease and increase the threshold.

3.5.2 Naive Bayes Classifier

Under the Naive- Bayes classification algorithm in STEADI, it is assumed that for the fall event the features are independent of each other. These feature variables will contribute evenly to the computation of the probability that a fall has occurred. For the system implemented, the Naive-Bayes is trained using the supervised learning setting [28]. The next subsection gives an outline of how the classifier works.

3.5.2.1 The Naive Bayes Probabilistic Model

The probability of event X given the evidence Y can be represented as below:

$$P(X|Y) = \frac{P(Y|X).P(X)}{P(Y)} \tag{3.1}$$

where P(X) is the prior probability, that is the probability of event before the evidence is applied, and P(X|Y) is the probability of event after incorporating the evidence. Provided that P(Y|X), P(Y), and P(X) are known, then P(X|Y) can be calculated.

To simplify the detection, we only have two classes, fall and non-fall. All other ADL such as walking, using the stair, jogging, and sitting are classified as being non-fall. As for the attribute, for the evidence, we use the 3 described in Section 3.3 and 3.4. A summary is given in Table 3.2.

Parameter	Number of Parameter	Remark
Class	2	Fall and Non-fall
Evidence/attribute	3	Max buffer value, Mean buffer value, local maxima distance

Table 3.3 STEADI Naïve-Bayes classification parameter

The probability model for the classifier is a conditional model as shown below:

$$P(X|Y_{1,} \dots, Y_n) \tag{3.2}$$

The Bayesian Probability terminology can be written as

$$posterior = \frac{prior\ x\ likelihood}{evidence} \tag{3.3}$$

In practice, only in the numerator (3.3) is of interest because the denominator does not depend on *X* and the values of the features *Y* are given, so that the denominator is effectively constant [28].

The numerator is equivalent to joint probability model, which can be written as follows,

$$P(X, Y_1 \dots, Y_n) = P(X)P(Y_1|X) \dots\dots P(Y_n|X, Y_1 \dots . Y_n) \tag{3.4}$$

As for the naive conditional independence, we assume that each *Y* is conditionally independent of every other feature given the category *X*, thus the conditional probability over the class variable *X* is:

$$P\big(X\big|Y_{1,} \dots, Y_n\big) = \frac{1}{Z} P(X) \prod_{i=1}^{n} P(Y_1|X) \tag{3.5}$$

Where *Z* = $P(Y_1 \dots, Y_n)$ is a scaling factor dependent only on the features $F_1, \dots, F_n$ [28].

With (3.5) it is possible to apply the Naïve-Bayes classification in STEADI. STEADI uses the value of 0.5 for the prior for both fall and non-fall event because we assume that the probability for such event is unknown and thus they are equally likely. From (3.5) in order to get the probability of falling event, the training data is used to calculate the evidence and likelihood for each feature. The training data were taken from two material arts student, they were asked to do the following activities:

- Walking followed by simulated falling
- Standing followed by simulated falling
- Running followed by walking
- Walking up/down stair
- Walking followed by sitting

The training data was then collected and is shown in Table 3.3 for the three features over the two classes.

Class	Maximum value	Mean Value	Local Maxima distance
Fall Event	97.44628 110.9488 108.3926 112.5929 92.29689 98.21989	35.41202 33.24051 28.39338 39.16586 27.13851 33.61478	55.36282394418663 50.54269711509716 59.88731678644783 51.36883782689136 56.91362
Non-Fall	90.26046 94.7635 89.67175 95.53383 84.05313	23.69591 29.08734 24.52235 24.28613 22.43171 24.10916 19.75125 28.9545 24.59877 20.44753 27.46918 23.69591 19.11767	24.31397930818467 23.44238364185715 39.3385238046875 40.88731678644783 25.179158581382765 35.24026

Table 3.4 STEADI training data for Naïve-Bayes classification

From the training data then we can measure the mean value and variance of each feature which is shown in Table 3.4.

Class	Mean value	Standard deviation	Remark
Fall	103.31621061166668	70.3872	Maximum value
	32.82751127666666	19.9341	Mean value
	54.81505877941129	15.1403	Gap value
Non-fall	90.85653335399029	21.2966	Maximum value
	24.012876193846157	10.0283	Mean value
	31.400270077557185	64.0022	Gap value

Table 3.5 STEADI training data for Naïve-Bayes classification

Using (3.3) the numerator for each class can then be calculated. Due to fact that the *evidence* between a fall and non-fall event given the specific value of maximum, mean and local maxima distance extracted from signal acceleration are equal , we are able to determine the event (fall/non-fall) by solely comparing the numerators [28].

3.6 Implementation on Android

In this section we describe the detail of the system implementation on an actual android mobile device.

3.6.1 Background to Android

In order to detect the event of falling, we need to be able to design an algorithm that can differentiate reliably between falling and ADL. In implementing such system, a real-time processing application is needed. Real-time signal processing demands an extremely high computational performance. At one time it was done by using a specialized microprocessor [52]. This would pursue performance at the expense of other system features such as generality, maintainability, and portability among other things [52]. Fortunately, nowadays real time signal processing can be done with generic computing device.

There were two main motivations behind the work in this thesis. First, we wanted to develop an assistive technology for detecting fall that was robust, portable and reliable. This implementation needed to be done in cost effective manner using a generic computing device. Thus, we developed the system for one of most common devices that is widely used and affordable: the Android smartphone [29]. Android is the first open source mobile application development platform based on Linux kernel. [29] The popularity of android is increasing year on year since it was unveiled in 2007 [29] along with the founding of the Open Handset Alliance—a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices [30]. Android is capable of doing multitasking, creating an *intents* to integrate each application installed on the device, using navigation, developing apps and utilizing various kind of sensors.

3.6.2 Android OS Architecture

The Android OS is a software system that is divided into several components as illustrated in Figure 3.12 below:

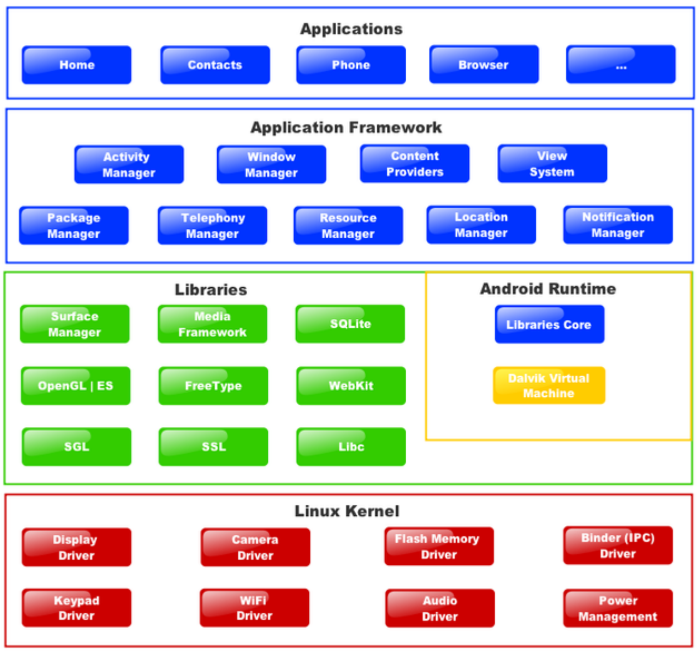


Figure 3.13: Android Architecture [53]

Figure 3.12 shows that Android is running on Linux. The Linux kernel provides system functionality like memory management, process management as well as device management [54]. Android is provided with various libraries including an open-source Web browser engine, a SQL database which is used for storage and sharing data between applications. It is also equipped with libraries to play and record audio and video, and has SSL libraries that are responsible for its internet system and security among other things.

During run time, Android uses another key component called the *Dalvik Virtual Machine* which is similar to Java Virtual Machine [65]. As for running various applications, Android provides an application framework with lots of services that can be used by any application built for it.

3.6.3 Android Application and Development Environment

Android applications are written in Java language. Once it is compiled into bytecode, it will be converted to a .dex file, a Dalvik executable file, using a dx converter. Further, it will be compiled into an Android package file that can be installed on any Android device [54]. Each android application is treated as single Linux user, each application is identified by a unique Linux user id with its own virtual machine. Android applications usually composed of four basic components [54]. The main building blocks of an android application can be listed as follows [54].

1. Activity

   Activity is usually categorized as the User interface (UI) of its application which the user will interact with. An application usually consists of multiple activities that are bound to each other [31] and the main activity will first appear in the UI when the application is launched. An Activity can also be stopped, suspended, paused, resumed and destroyed.

2. Service

   Services are designed to allow one of an application's features to run in the background. Services are responsible for tasks like updating, remote processes and also continuous monitoring of sensors. However, service does not provide UIs.

3. Content Provider

   The Content Provider is a mechanism that allows applications to share data between each other. There is no specific type of data structure implemented by the system. The Content provider is uniquely identified by its authority and contains many types of object. The most common method of accessing an object is to query the Content provider with the specific content of an android Uniform Resource Identifier (URI) as the basis for requesting data.

4. Broadcast Receiver

   Broadcast receiver is an android component that can receive an event and react to events. Each broadcast receiver will register a particular event that they are interested in, for example listening for an outgoing MMS message. When events occur, they are represented as Intents. Those intents are then broadcast to the system.

3.6.4 Android Implementation

This section describes the STEADI implementation using Android. The source file and the resources required by the application are explained here.

3.6.4.1 Android "manifest.xml"

The android manifest is a file consisting of important information related to the application that is required by the Android system for launching the application. The minimum API level is declared here as well as all the components that are used in the application. The manifest also consists of different kinds of permissions such as camera access, messaging, and location manager [31].

3.6.4.2 Android Component

We need an interface that enables the user to do important activities as mentioned in the use case in Section 3.2.1. Therefore, because this application is targeted at various users from different kinds of backgrounds, a clear, direct UI that is easy to use is important. The UI can be simple and minimal. The STEADI application comprises of three activity classes and one service class. The first activity class is implemented as a welcome screen that has two menus: *Start detector* and *Register user*. The main view is defined in the main.xml layout file. Figure 3.13 picture shows the main layout along with its menu.

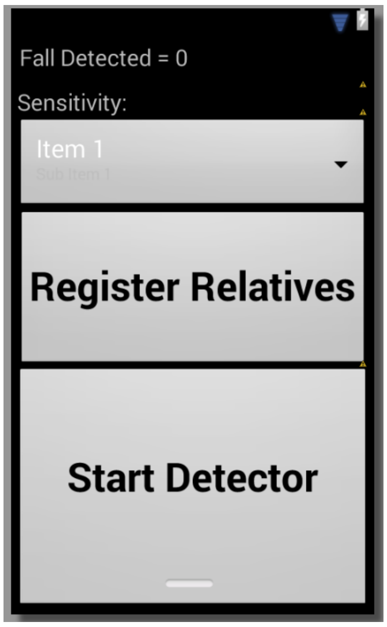


Figure 3.14: The main layout view of the application

The *register relative* button on the main layout will direct the user to the next activity that enables the user to register their caretaker along with the name and phone number. The UI is shown in figure 3.14. All the data are saved in the Android shared preferences; hence, the data can be shared throughout all activities.

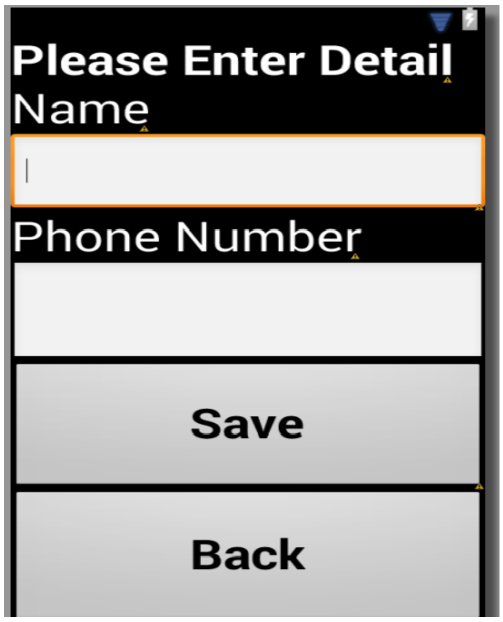


Figure 3.15 Register a relative UI layout

The last activity class is triggered when a fall event is detected. This activity will inform the registered care taker by sending the information message with the location of the occurrences to desired phone.

As for the service class, STEADI consist of four main classes: *FallDetector.java*, *FallListener.java*, *FallService.java*, and *GPSTrackker.java*. The *FallListener* is an interface that consists of the method *onFall*() that needs to be overwritten by the *FallService* class. The *FallDetector* class implement the *SensorEventListener*. Here, the algorithm for detecting a fall event is implemented, including the signal processing and the data classifier.

3.6.4.3 Android Location Tracker

In order to detect the location of the devices Android uses GPS and Android's Network Location Provider to provide the user location coordinate.

Although GPS is most accurate, it only works outdoors, it quickly consumes battery power, and doesn't return the location as quickly as users want. Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power. [55]

Hence, STEADI implements both the GPS and the Network Location Provider so that the whereabouts of the fall event can be detected both for indoors and outdoors locations. The relevant class is *GPSTrackker.java*. The *GPSTrackker* object is called in the *TestActivity.java* which includes the method to automatically inform the caretaker after the fall detected. In the *TestActivity.java* we implement the *Runnable* interface which do the time counting in the background and will automatically trigger the *sms* if the user didn't push any cancel button during certain timeline as shown in figure 3.16.

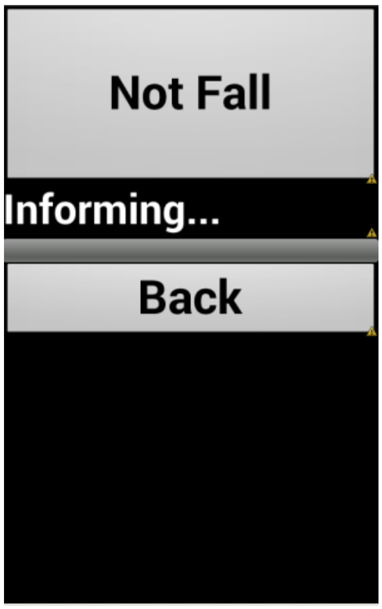


Figure 3.16 After Fall detected UI layout

3.7 Adding an External Sensor

They are several ways to improve STEADI’s accuracy to detect fall, one of which is including another peripheral device with a gyroscope and accelerometer and integrating the data it generates on the mobile device. With this peripheral device we were able to put our mobile phone anywhere, in the user’s bag for example [61]. In this section, we describe the STEADI implementation using an Arduino-based sensor as the peripheral device.

3.7.1 Implementation of a Peripheral Device

The Arduino uno is a microcontroller board based on the ATmega328 [59]. This device is highly accessible and very prevalent offering developers and casual users a chance to get started in experimenting with microcontroller projects. Arduino projects can be developed in several programming languages through an IDE such as Processing [59].

For STEADI, our Arduino was equipped with an IMU (Inertial Measurement Unit) that had 6DOF (degrees of freedom), consisting of an ADXL345 accelerometer and ITG-3200 gyroscope [66] as shown in figure 3.17.

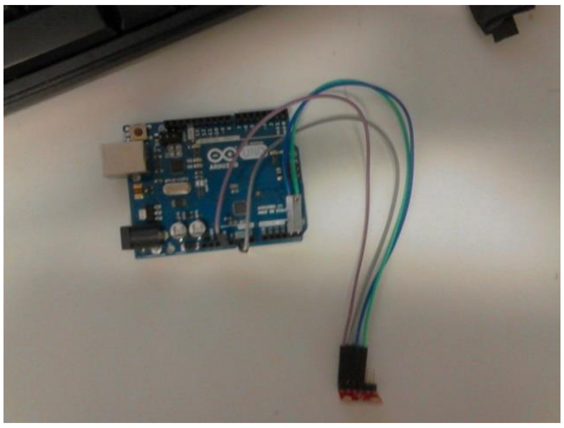


Figure 3.17 Arduino with IMU

The gyroscope is used to measure the device rotation while accelerometer is used to determine its orientation [60]. A fusion filter that has been implemented in the *freeIMU* Arduino library by varesano et al [60]. is available to combine the data streams from the accelerometer and gyroscope function into the features of Pitch, Yaw, and Roll.

In order to detect the fall event, some modifications of several methods from the *freeIMU* code are required. The system implements the threshold-based classification from the data captured for both the acceleration signal and rotation as shown figure 3.18 to 3.19.

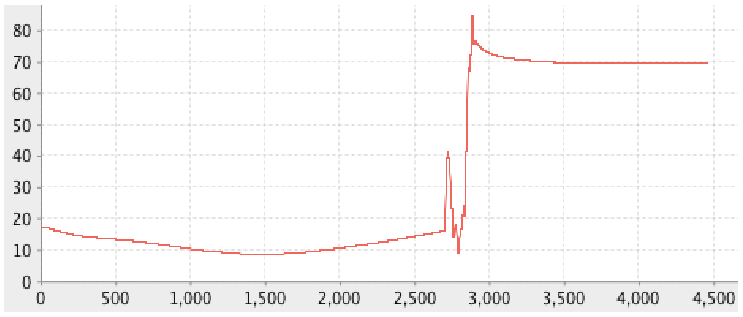
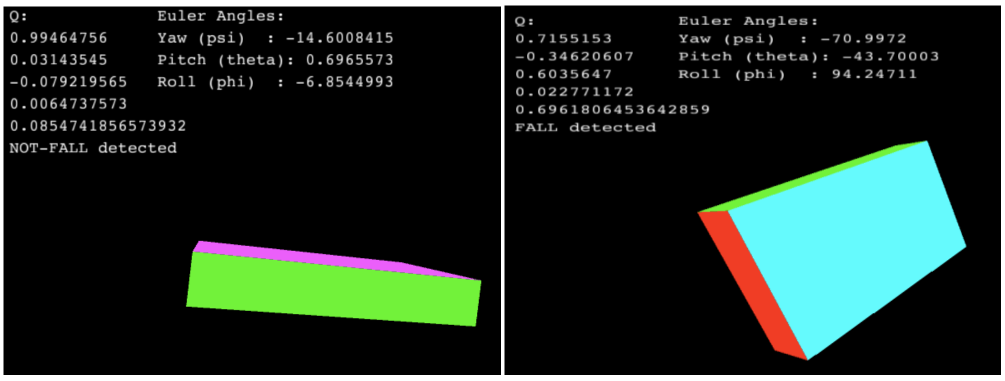


Figure 3.17Arduino acceleration signal of the fall event

When we simulate the fall event using a mannequin, figure 3.17 shown how the impact appears in terms of the acceleration signal as the mannequin hit the ground. As for the gyroscope signal, an event is count as a fall event when the rotation (roll) change is more than 90° [61] as shown in figure 3.18.



```
Q:                  Euler Angles:
0.99464756          Yaw (psi)    : -14.6008415
0.03143545          Pitch (theta): 0.6965573
-0.079219565        Roll (phi)   : -6.8544993
0.0064737573
0.0854741856573932
NOT-FALL detected
```

```
Q:                  Euler Angles:
0.7155153           Yaw (psi)    : -70.9972
-0.34620607         Pitch (theta): -43.70003
0.6035647           Roll (phi)   : 94.24711
0.022771172
0.6961806453642859
FALL detected
```

Figure 3.18Arduino Rotation signal of the fall event

3.8 Summary

STEADI is an application built for an Android mobile phone to detect falls. It is categorized as wearable-device for determining fall. It was implemented on an Android mobile phone that uses the built-in accelerometer, GPS, and other services that the phone offers. The STEADI algorithm can be divided into two main parts: the signal processing section and the classification. Here we use the signal processing to enhance the accelerometer signal and then convert it into features that represent the activities performed. As for the classification, both the threshold-based classification and the Naïve-Bayes were used. Because of the high accuracy that could be obtained regardless of size of the training data available, the Naïve-Bayes is most preferable than the threshold-based classification. Lastly, another sensor, a combined 6DOF accelerometer and Gyroscope sensor running on an Arduino platform, was introduced and it was shown how it can detect a fall event.

4 Evaluation

In this chapter we describe in more detail the testing and evaluation of the STEADI application. For the fall detector system, we conduct several tests such as unit testing, integration testing, as well as system testing. We also conducted a survey for evaluating the user interface of the application.

4.1 Testing

The purpose of software testing is to verify its correct operation to the specifications [32]. Two complementary ways of achieving this has been developed: by generating the tests from the specification (Black-Box testing), and by generating tests from the implementation (White-Box) testing [32]. Black box testing is a testing technique based entirely on the specification, and uses no knowledge of the inner working of the program code [32] while White box testing does use the code.

4.1.1 Unit testing

Unit testing is the testing certain functions or methods of code to verify that it works as expected. These units are generally small in size. Here, unit testing was done for the classifier method and signal filtering method as it is the backbone for the application. Firstly, we implemented the unit test using the black-box testing technique. Black-box tests are characterized by how the input test data is selected using the specification [32]. We performed the Equivalence Partitioning (EP) Black-Box test technique on several methods given in Table 4.1 below:

ID	Method Name	Result	Remark
	HannSmoother	Pass	Signal Filtering
	MedianValue	Pass	Signal Filtering
	padSignal	Pass	Signal Filtering
	IIRHighPass	Pass	Signal Filtering
	MedianFiltering	Pass	Signal Filtering
	getFallDetected	Pass	Classifier (Naïve-Bayes)

Table 4.1. Application method unit testing result

We use the Equivalence Partition method because we assume that the input accelerometer data and the buffer feature values given to the relevant method during run time will vary only within a certain range. If any value within a particular partition does not expose a fault in a particular method, we can assume that the method operate according to specification because the other values that belong to a same partition will be treated in the same way. From Table 4.1 all the EP tests carried out passed with the actual output matching the expected output so we are sure that the methods were implemented correctly. The detail of tests listed in the appendix.

4.1.2 System testing

System testing of an Android mobile device can usually be done using the development tool only. However, to test the accelerometer, it's required to do the testing on the Android device itself [56]. Here we use low cost android device, an Acer Liquid Z3 with an Android OS ver 4.4, with its details given in the Appendix. For the application black box testing is also applied. For the fall detection, we have separate test cases that focus on the detection of fall events specifically.

The following Table 4.2 show a list of tests for the application:

ID	Test Description	Steps taken	Result
1	Installation and starting up the application	Open ADT Install the application to the desired device Start/launch the application	Application was successfully installed and launched. (PASS)
2	Accelerometer testing	Start the detector activity Simulate falling System should detected the fall event	Fall was successfully detected. (PASS)
3	Register and location testing	Start the program Register the caretaker along with their real phone number Start the detector Simulate fall Idle until the message was send to the desired phone. Open Google maps and display the location with a marker	Message with location information was successfully sent. (PASS)
4	Robustness test, to test that the application runs smoothly [56]	Start the application Run the accelerometer for more than 4 hours Check whether the application still run correctly by simulating a fall each hour.	Fall was successfully detected. (PASS)

Table 4.2 STEADI system testing description and result

4.1.3 Application Testing and Evaluation

We evaluate the STEADI prototype by conducting experiments. In this section, we introduce how the data was collected so that we could compare the algorithms being used, that is, the threshold-based classifier and the Naïve-Bayes classifier, and also make another comparison with other fall-detector applications implemented on a mobile device.

During this evaluation, we didn’t measure the resource consumption of STEADI as it is not within the scope of our work. We focused only on evaluating a fall occurrence with the sensor. It’s a best practice if all the testing is performed on different subjects as well as mannequins for testing the fall detector implemented on Arduino [9]. The sensor was mainly attached in the waist for the reason that in the previous related work it was found to give the best accuracy [9] because the waist is the nearest point of people centre of gravity.

4.1.3.1 Data Collection

As we didn't specifically study different directions of fall, we simplify the fall event by simulating only one direction of falling. Then, we asked the subject to repeat certain actions, such as: walking, jogging, walking then falling, walking then sitting down, jogging then jumping, and walking up and down stairs.

We conducted the tests on 8 volunteers from 20 - 30 years, and one 65 years old martial arts teacher. The height is range from: 155 – 185 cm, with weight from: 50 – 85 kg. We could not conduct the test with elderly people as subjects as it was not feasible and high risk [10]. Each subject was asked to perform various actions as described in Table 4.3. The device was set to detect the event of falling with both methods at the same time, (1) the Threshold-based classifier and (2) the Naïve-Bayes classifier. Training had been done using just one subject alone already as mentioned in Section 3.5.1.

Category	Detail Activity	Test units
Fall	walking then falling	40
Non-fall	walking	10
	walking then sitting	5
	walking up stair	5
	walking down stair	5
	Jogging	10
	jogging then jumping	5

Table 4.3 Testing activities

Below are more details of the tests performed.

1. The term of one unit activity varies through different types of activity, such as:
    - Each time the subject walks up/down stairs for one storey is considered as one unit activity
    - Each time the subject runs a lap, for approximately one minute (100 – 150 m) is one unit activity
    - Each occurrence of a subject walks for 1 minute (50 – 100 m) is one unit activity
    - Each time where the subject does the activity of walking and then sitting is considered one unit activity
    - Each occurrence of subject performing a jump is one unit activity
    - Each occurrence of subject performing a fall event is one unit activity
2. We didn’t test the event of falling when the subject was walking up/down stairs due to safety reasons.
3. To test the detection all participants put the phone near waist area in their pocket jacket or their pants for accuracy and user convenience [9] as shown in the picture below. Here, the use of extra strap is not necessary to place the mobile phone. The red circle is the place where the subject put the mobile phone.

The testing activities were done in sequence to get more feel for the real situation. The activities and the associated features captured were saved to the device.

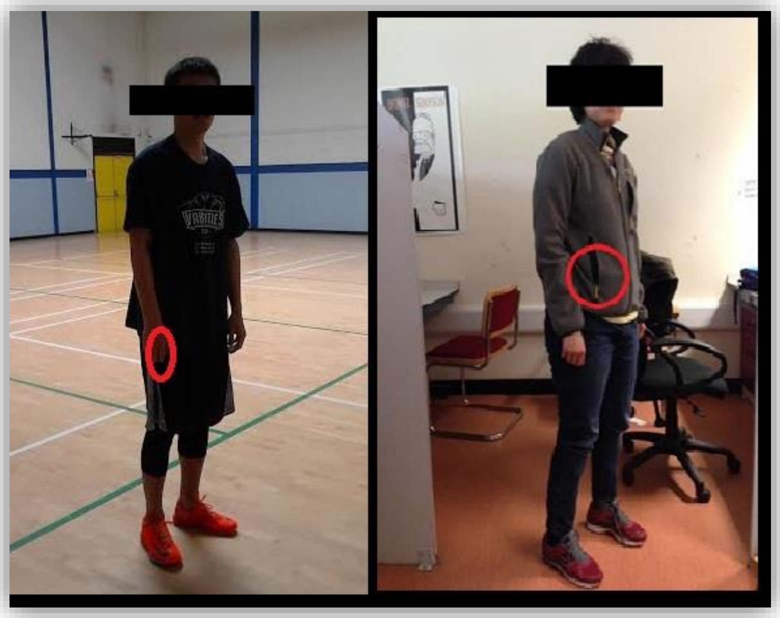


Figure 4.1 Subject of STEADI testing, the red circle shows the place where the mobile phone is placed

Figure 4.2 Mannequin for STEADI testing, the red circle shows the place where the arduino system is placed

4.1.3.2 Detection Evaluation

In detecting the fall event for the test, we tested our application for the Threshold-based method without the filter to compare the effectiveness of using signal filtering on the raw signal captured. Then, we implemented both the threshold-based (Th) and Naïve Bayes (NB) classification algorithm on the mobile device. The application recorded both results separately. Table 4.4 show the detail of the test results.

Category	**Detail Activity**	**Total**	**detection**					
			Thnf non-fall	**Thnf Fall**	**Th non-fall**	**Th Fall**	**NB non-fall**	**NB fall**
Fall	walking then falling	40	4	6	1	39	1	39
Non-fall	walking	10	7	3	10	0	10	0
	walking then sitting	5	-	-	4	1	5	0
	walking up stair	5	-	-	5	0	5	0
	walking down stair	5	-	-	2	3	4	1
	Jogging	10	-	-	2	8	0	5
	jogging then jumping	5	-	-	0	5	1	4

Note: for the Thnf we only test both the fall event and walking activity 10 times.

Table 4.4 Detail of the STEADI activities test result

After we have collected the data we measured the performance reliability of the system in terms of the proportion of false positives (FP) and false negatives (FN). A False positive happens when the system triggers the fall alarm without a real fall event. A False negative happens when as fall occurs, however the system misses the detection of such an event. The performance metrics we then use to evaluate STEADI are Precision, Recall (Negative predictive value), Sensitivity, Specificity, and Accuracy, which can be calculated as follows [57].

$$Precision = \frac{TP}{TP+FP} x\ 100 \tag{5.1}$$

$$Sensitivity = \frac{TP}{TP+FN} x100 \tag{5.2}$$

$$Negative\ predictive\ value = \frac{TN}{TN+FN} x100 \tag{5.3}$$

$$Specificity = \frac{TN}{TN+FP} x100 \tag{5.4}$$

$$Accuracy = \frac{TP+TN}{TN+FP+TP+FN} x100 \tag{5.5}$$

For the evaluation, there are four main cases that need to be observed:

- True positive: A fall event is correctly identify as a fall
- False positive: A non-fall event that’s incorrectly identify as a fall
- True negative: A non-fall event that’s not categorized as a fall
- False negative: A fall event that’s categorized as fall

From the equation and the test result we can calculate the performance matrix shown in Table 4.5 (a) and (b) for the threshold classification and Table 4.6 for Naïve-Bayes classification.

Result\Case	Fall	Non-fall	(a)	
Fall	6 (TP)	4 (FP)	60 %	Precision
Non-fall	3 (FN)	7 (TN)	70 %	Recall
	66.67%	63.64%	65 %	Accuracy
	Sensitivity	Specificity		

Result\Case	Fall	Non-fall	(b)	
Fall	39 (TP)	17 (FP)	69.64%	Precision
Non-fall	1 (FN)	23 (TN)	95.83%	Recall
	97.50%	57.50%	77.50%	Accuracy
	Sensitivity	Specificity		

Table 4.5 Performance matrix of threshold-based classification: (a) without filter (b) with filter

Result\Case	Fall	Non-fall		
Fall	39 (TP)	2 (FP)	95.12%	precision
Non-fall	1 (FN)	38 (TN)	97%	Recall
	97.50%	95%	96.25%	Accuracy
	Sensitivity	Specificity		

Table 4.6 Performance matrix of Naïve-Bayes classification

The test results show that signal filtering improves the accuracy by more than 15%. Further, the Naïve Bayes method is better than the threshold method for Fall detection by itself. Next, when the subject performs activities like jogging, jumping, and walking down the stairs if the fall detection algorithm is based on the threshold value method, it gives lots of false positives because those activities can easily cause the accelerometer signal to exceed the threshold value. However, the use of the Naïve Bayes algorithm gives far better results than threshold-based classification and achieved more than 20% accuracy than the threshold-based classification for both Fall and non-Fall activity detection.

After we tested STEADI mobile application, we tested the Arduino and external sensor using a mannequin. We performed 10 fall events and 10 non-fall (moving the mannequin to the next chair) events by moving the mannequin as shown in figure 4.3 and figure 4.4. The outcome of the tests is shown in table 4.7 below.

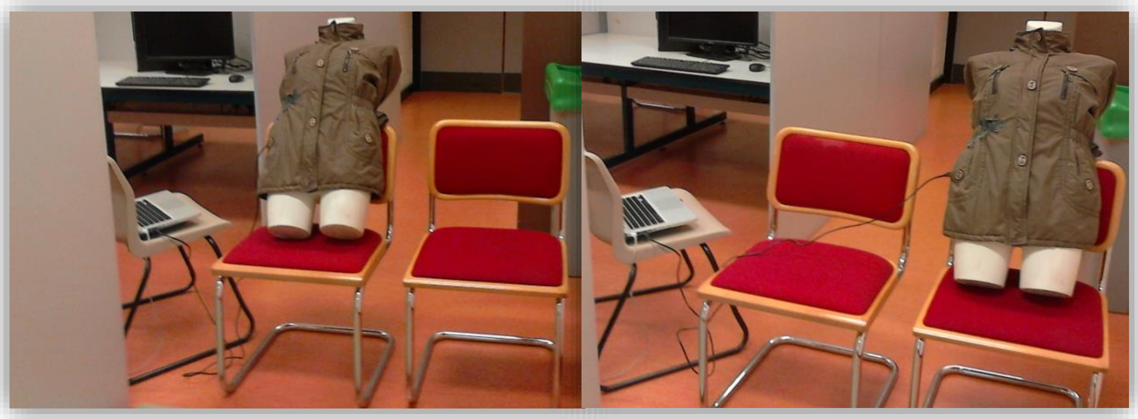


Figure 4.3 Arduino non- fall testing

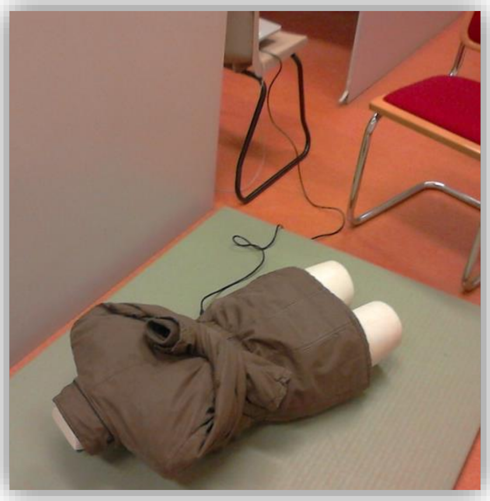


Figure 4.4 Arduino fall testing

Result\Case	Fall	Non-fall		
Fall	10 (TP)	0(FP)	100%	precision
Non-fall	0 (FN)	10(TN)	100%	Recall
	100%	100%	100%	Accuracy
	Sensitivity	Specificity		

Table 4.7 Performance Matrix for using an External Sensor with an Arduino

By adding the gyroscope to record the rotation of the device, the accuracy of the device is high even though we only implement the threshold-based classification on the accelerometer signal. It also adds the possibility to determine the actual orientation of fall, such as forward, backward, or lateral falling [61]. However, in order to determine if the use of external sensor is much more preferable another extra set of tests should be conducted.

4.2 Performance Comparison

In this section we compare the performance of STEADI to the other existing wearable fall detection devices. Here, we can only use the data provided from their research papers. Table 4.10 illustrates a comparison between each device for a mobile phone detection device and a custom-made wearable device.

Other Mobile Application	**Accuracy**	**Precision**	**ADL Activity**
Sun SPOT [21]	87.55%	NA	standing, sitting, lying and walking
Mark V. Albert and co. [33]	98.20%	NA	not mentioned
sAfe [34]	98.91 - 99.45%	NA	standing, sitting, lying and walking
perFallD (waist + forward falling event) [9]	94.35%	NA	Standing
Gokhan Remzi Yavuz and co.[10]	NA	95%	walking, sitting down, lying and jumping
STEADI	96.25%	95.12%	jogging, walking, walking up/down stair, sitting down, lying and jumping

Table 4.10 Performance comparison of STEADI with other wearable fall detection devices

The results also show that STEADI is quite robust in comparison to the others. The tests show that STEADI is able to give highest accuracy, more than 95%, while the subjects engage in a wider variety of ADL.

4.3 User Interface Evaluation

There are many possible ways to evaluate the user-interface of a system [41]: Accuracy, speed, and user friendliness are some of performance attributes that should be measured because they affect a person attitude toward a system. Additionally, the time it takes to learn a system is associated with how effectively a system can be used [41].

One of the methods by which it is possible to measure attributes of the user interface is by using a questionnaire [41]. Here, we use a measurement tool called the Generic User Interface Questionnaire (QUIS) which based on research in the Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface [41]. The original questionnaire consisted of a total 90 questions. However, the short version of QUIS (2.0) has only 20 main questions [41].

As for the STEADI survey, the questionnaire was also tailored to the system developed. The STEADI questionnaire has several sections as follows: (a) screen, (b) terminology and system information, (c) learning and system capabilities. Each section has several questions.

4.3.1 Method
The participants consisted of 10 people raging from the age of 20-30. Two of the participants were assumed to have a better level of computer experience while the other ten participants were from various different backgrounds such as arts and psychology. First, the participants were given a brief explanation about the system. Then, they were asked to do various tasks using the system as provided. After they have used the system for about 5 - 10 minutes, they needed to fill the questionnaire. The questions asked are available in the Appendix.

4.3.2 Survey Results
After conducting the survey, we calculated the scores. The result of the survey showed that the average user interface score for STEADI application was 169 out of 200, that is, 80%. Even though we can’t measure the user friendliness using objective measures of the software [58], we concluded from the subjective results that the application is easy to use and is user friendly as the survey showed a score of more than 70% on average. However, due to the lack of a help menu, some of the subjects suggested the addition of a help menu on the STEADI application would make it easier to learn the features of the application.

4.4 Summary
The tests and survey that were conducted on STEADI shows that we answered our research questions from the first chapter. First, the tests showed that STEADI is robust because it has achieved an accuracy of more than 80% overall. Being implemented on an Android mobile phone makes STEADI not only portable but also implemented in a cost effective manner. With an average score to 169/200 from the survey illustrates that STEADI is easy to use. STEADI can also contact a relative/caretaker when a fall event is determined to have taken place via a SMS message.

Lastly, we know that by solely doing signal processing to enhance the accelerometer signal before applying the threshold method can drastically improve the precision of the system. However, because more vigorous activity can confuse the threshold-based classifier it was necessary to implement a further algorithm to differentiate these fall-like events with a real fall. Thus, the Naïve-Bayes classification was found to be extremely useful.

We can conclude that the experimental results have shown that a fall detection system that built on generic mobile phone in a cost effective mannered has satisfied the requirement to be an assistive technology.

5 Conclusion

This chapter draws conclusions from the work that has been carried out. Then we discuss the other work that could be developed in the future related to implementation of the fall detector system.

5.1 Summary

The advancement of technology in mobile and wireless healthcare systems is growing significantly and one particular application is to recognize physical activity and detect falls. For the problem to detect falls, we have considered several solutions. This project has proposed a mobile application development on the android platform. The android application uses the built-in accelerometer sensor the device. In term of our algorithm, we implemented a signal processing algorithm to enhance the captured accelerometer data and then converted it into features. To do the classification we examined a threshold-based approach, followed by a statistical classification technique, Naive Bayes. All of the processes of filtering and classification were done in the real time.

We have also conducted some testing and a user survey. We first tested the application to recognise a falling event and we found that the combination of the signal processing and classifier gives an accuracy of 96.75% overall. In addition to the fall detection, we also included a system to inform the caregiver/relatives as to the whereabouts and time of a falling event.

Although we have successfully implemented a real-time system that is robust, reliable and cost efficient, there are still limitations. Taking into considerations time and resource limitations the training and testing data was collected from young subjects. Thus, more work would need to be done to prepare the system for deployment with its target users of people over 65 years old.

5.2 Future work

In the future, other than implementing more testing, one feature we could improve is that the algorithm should have lower power consumption. It's also possible to determine the fall direction by using the use of an external sensor (gyroscope).

Another possibility is to test the system for more other vigorous activities so that it can be targeted not only to the elderly but also at people from other categories that have a high probability of falling while they are doing vigorous activity and need immediate treatment such as those at risk of a heart attack, stroke, or who have Parkinson's disease [10]. Another aspect for the future development is to conduct extensive testing using other mobile devices and at different locations to test the robustness of the system.

6 Appendix

6.1 Mobile Device Specification [62]

GENERAL	2G Network	GSM 900 / 1800 / 1900
	3G Network	HSDPA 900 / 2100
	SIM	Optional Dual SIM
	Announced	2013, August
	Status	Available. Released 2013, September
BODY	Dimensions	109 x 60 x 10.4 mm (4.29 x 2.36 x 0.41 in)
	Weight	120 g (4.23 oz)
DISPLAY	Type	TFT capacitive touchscreen, 16M colors
	Size	320 x 480 pixels, 3.5 inches (~165 ppi pixel density)
	Multitouch	Yes
SOUND	Alert types	Vibration, MP3, WAV ringtones
	Loudspeaker	Yes, with stereo speakers
	3.5mm jack	Yes
MEMORY	Card slot	microSD, up to 32 GB
	Internal	4 GB, 512 MB RAM
DATA	GPRS	Yes
	EDGE	Yes
	Speed	HSDPA, 21 Mbps; HSUPA, 5.76 Mbps
	WLAN	Wi-Fi 802.11 a/b/g/n, Wi-Fi hotspot
	Bluetooth	Yes, v3.0 with A2DP, EDR
	USB	Yes, microUSB v2.0
CAMERA	Primary	3.15 MP, 2048 x 1536 pixels
	Features	Geo-tagging
	Video	Yes
	Secondary	No
FEATURES	OS	Android OS, v4.2 (Jelly Bean)
	Chipset	Mediatek MT6572
	CPU	Dual-core 1 GHz Cortex-A7
	GPU	Mali-400

	Sensors	Accelerometer, proximity
	Messaging	SMS (threaded view), MMS, Email, Push Email
	Browser	HTML
	Radio	Stereo FM radio
	GPS	Yes, with A-GPS support
	Java	Yes, via Java MIDP emulator
	Colors	Rock Black, Classic White
		- Separately sold exchangeable back covers in Rock Black, Classic White, Sakura Pink, Pop Yellow and Lagoon Turquoise - SNS integration - MP3/WAV/WMA/AAC player - MP4/H.264/H.263 player - Organizer - Document viewer - Photo viewer/editor - Voice memo/dial - Predictive text input
BATTERY		Li-Ion 1500 mAh battery
	Stand-by	
	Talk time	

6.2 Equivalence Partitioning Unit Testing (EP)

Test Case	Parameter	Range	Test no.	Remark
1	Input	0.. Integer.MAXIMUM	1	double [], length of array
2	value(input)	Double.MINIMUM.. Double.MAXIMUM	1	double[], value inside array
3	Return Value	Double.MINIMUM.. Double.MAXIMUM	1	double[], value inside array
4		0.. Integer.MAXIMUM	1	double [], length of array

Table B.1 HannSmoother EP Test Cases

ID	Test Cases Covered	Input	Expected Output
1	1,2,3,4	{1,2,3,2,1}	{1.0, 4.0, 8.0, 10.0, 8.0}

Table B.2 HannSmoother EP Test Data

Test Case	Parameter	Range	Test no.	Remark
1	Signal	0.. Integer.MAXIMUM	2	double [], length of array
2	value(signal)	Double.MINIMUM.. Double.MAXIMUM	2	double[], value inside array
3	Return Value	Double.MINIMUM.. Double.MAXIMUM	2	double

Table B.3 MedianValue EP Test Cases

ID	Test Cases Covered	Input	Expected Output
2	1,2,3	{1,2,3,2,1}	2

Table B.4 MedianValue EP Test Data

Test Case	Parameter	Range	Test no.	Remark
1	Input	0.. Integer.MAXIMUM	3	double [], length of array
2	value(signal)	Double.MINIMUM.. Double.MAXIMUM	3	double[], value inside array
3	filtDim	Double.MINIMUM.. Double.MAXIMUM	3	integer
4	Return Value	0.. Integer.MAXIMUM	3	double[], length of Array

Table B.5 padSignal EP Test Cases

ID	Test Cases Covered	Input	Expected Output	Remark
3	1,2,3,4	{1,2,3,2,1}, filtDim = 4	{0.0, 0.0, 1.0, 1.0, 2.0, 2.0, 3.0, 0.0, 0.0}	length: 9

Table B.6 padSignal EP test data

Test Case	Parameter	Range	Test no.	Remark
1	Signal	0.. Integer.MAXIMUM	4	double [], length of array
2	value(signal)	Double.MINIMUM.. Double.MAXIMUM	4	double[], value inside array
3	Cutoff	Double.MINIMUM..Double.MAXIMUM	4	-
4	Dt	Double.MINIMUM..Double.MAXIMUM	4	-
5	Return Value	0.. Integer.MAXIMUM	4	length of array

Table B.7 IIRHighPass EP test case

ID	Test Cases Covered	Input	Expected Output
4	1,2,3,4	{1,2,3,2,1}, cutoff = 1, dt = 1	length: 5

Table B.8 IIRHighPass EP test data

Case	Parameter	Range	Test no.	Remark
1	Input	0.. Integer.MAXIMUM	5	double [], length of array
2	value(signal)	Double.MINIMUM.. Double.MAXIMUM	5	double[], value inside array
3	filtDim	Double.MINIMUM.. Double.MAXIMUM	5	integer
4	Return Value	0.. Integer.MAXIMUM	5	double[], length of Array
5		Double.MINIMUM.. Double.MAXIMUM	5	double[], value inside array

Table B.9 MedianFiltering EP Test Case

ID	Test Cases Covered	Input	Expected Output	Remark
5	1,2,3,4,5	{1,2,3,2,1}, filtdim = 4	{0.5, 1.5, 2.0, 2.0, 1.5}	length: 5

Table B.10 MedianFiltering EP test data

Test Case	Parameter	Range	Test no.	Remark
1	maximum	Double.MINIMUM.. Double.MAXIMUM	N1	-
2	Mean	Double.MINIMUM.. Double.MAXIMUM	N1	-
3	Gap	Double.MINIMUM.. Double.MAXIMUM	N1	-
4	return value	TRUE	N1	-
5		FALSE	N2	

Table B.11 getFallDetected EP Test Cases

ID	Test Cases Covered	Input	Expected Output
N1	1,2,3,4	maximum = 103; mean = 32; local minima distance = 54	TRUE
N2	1,2,3,5	maximum = 90; mean = 24; local minima distance = 30	FALSE

Table B.12 getFallDetected EP test data

6.3 STEADI Survey

STEADI is an application designed to monitor the human activity and respond to certain event. This application is designed to observe the event of falling. If falling event is detected, the system will inform (via sms) to other user that registered as a care taker when the event has happened along with the location.

Thank you for taking the time to complete the survey. Your feedback is important to us in how we can measure the user friendliness of our system.

This survey should only take about 5 minutes of your time. Your answer will be completely anonymous.

Screen

		1	2	3	4	5	6	7	8	9	10	
Reading character on the screen	Terrible	●	●	●	●	●	●	●	●	●	●	Wonderful
Highlighting / simplifies task	Not at all	●	●	●	●	●	●	●	●	●	●	Very much
Organization of information	Confusing	●	●	●	●	●	●	●	●	●	●	Clear
Sequence of screen	Confusing	●	●	●	●	●	●	●	●	●	●	Clear

Terminology and System Information

		1	2	3	4	5	6	7	8	9	10	
Use the terms throughout system	Inconsistent	●	●	●	●	●	●	●	●	●	●	Consistent
Position of message on screen	Inconsistent	●	●	●	●	●	●	●	●	●	●	Consistent
Prompts for user input	Confusing	●	●	●	●	●	●	●	●	●	●	Clear
Computer informs about its progress	Confusing	●	●	●	●	●	●	●	●	●	●	Clear

Learning

		1	2	3	4	5	6	7	8	9	10	
Learning to operate the system	Difficult	●	●	●	●	●	●	●	●	●	●	Easy
Exploring features by trial and error	Difficult	●	●	●	●	●	●	●	●	●	●	Easy
Remembering names and commands	Difficult	●	●	●	●	●	●	●	●	●	●	Clear
Performing task is straightforward	Never	●	●	●	●	●	●	●	●	●	●	Always

System Capabilities

		1	2	3	4	5	6	7	8	9	10	
System speed	Slow	●	●	●	●	●	●	●	●	●	●	Fast enough
System Reliability	unreliable	●	●	●	●	●	●	●	●	●	●	reliable
System noise	noisy	●	●	●	●	●	●	●	●	●	●	quiet
Designed for all level of users	disagree	●	●	●	●	●	●	●	●	●	●	Agree

Based on: Chin, J.P., Diehl, V.A., Norman, K.L. (1988) *Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface.*

7 References

[1] Kenny, Rose A., et al. "Falls Prevention in the Home: Challenges for New Technologies." Hershey, PA: Information Science Reference. September 2010.

[2] V.Vaidehi, Kirupa Ganapathy, K.Mohan, A.Aldrin, K.Nirmal. "Video Based Automatic Fall Detection In Indoor Environment." IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011.

[3] C. Rougier, A. St-Arnaud, J. Rousseau, J. Meunier, "Video surveillance for fall detection." InTech, 2011, ISBN 978-953-307-436-8.

[4] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection." Journal of Telemedicine and Telecare, vol. 11, no. 4, pp. 194198, 2005, 18 ROYAL SOC MEDICINE PRESS LTD LONDON 937FL.

[5] Laila Alhimale, Hussein Zedan, Ali Al-Bayatti, "The Implementation of an Intelligent and Video-based Fall Detection System using a Neural Network." Department of Software Technology Research Laboratory, De Monfort University, Leicester, LE1 9BH, Applied Soft Computing, January 2014.

[6] Toreyin, B.U., Bilkent Univ., Ankara, Dedeoglu, Y., Cetin, A.E. "HMM Based Falling Person Detection Using Both Audio and Video." Signal Processing and Communications Applications, 2006 IEEE 14th.

[7] Xiaodan Zhuang, Jing Huang , Potamianos, G. ,Hasegawa Johnson, M." Acoustic fall detection using Gaussian mixture models and GMM supervectors." Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.

[8]M. N. Nyan, Francis E. H. Tay, and E. Murugasu. "A wearable system for pre-impact fall detection." Journal of Biomechanics, 41(16):3475–3481, 2008.

[9] Jiangpeng Dai, Xiaole Bai, Zhimin Yang, Zhaohui Shen, and Dong Xuan. "PerFallD: A Pervasive Fall Detection System Using Mobile Phones." Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference.

[10] Gokhan Remzi Yavuz, Mustafa Eray Kocak, Gokberk Ergun, Hande Alemdar, Hulya Yalcin, Ozlem Durmaz Incel, Lale Akarun, Cem Ersoy. "A Smartphone Based Fall Detector with Online Location Support." Journal Pervasive and Mobile Computing, Volume 8 Issue 6, December 2012.

[11] Jay Chen, Karric Kwong, Dennis Chang, Jerry Luk, Ruzena Bajcsy. "Wearable Sensors for Reliable Fall Detection." Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference. Shanghai, China, September 2005.

[12] Bilney, B., Morris, M., & Webster, K. (2003). Concurrent related validity of the GAITRite® walkway system for quantification of the spatial and temporal parameters of gait. Gait & Posture, 17(1), 68-74.

[13] Besser, M. P., Selby-Silverstein, L., & Prickett, N. (2001). Predicting Fall Risk in the Elderly Using Temporal-Spatial Parameters of Gait. Symposium of the International Society for Postural and Gait Research, 70-73.

[14] Craig Larman, Victor R.Basili. "Iterative and Incremental Development: A Brief History." Published by the IEEE Computer Society, June 2003.

[15] Kai Petersena, Claes Wohlina."A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case." Journal of Systems and Software Volume 82, Issue 9, September 2009, Pages 1479–1490.

[16] Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.

[17] "Support Vector Machine." *Wikipedia*. Wikimedia Foundation, 05 May 2014. Web. 09 May 2014. http://en.wikipedia.org/wiki/Support_vector_machine

[18]Bakos, Yong Joseph. "Data Mining Portfolio: K-Nearest Neighbor." Colorado School of Mines, CSCI 568: Data Mining, Fall 2011.

[19]Yamato, J, Jun Ohya, Ishii, K."Recognizing human action in time-sequential images using hidden Markov model." Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference.

[20] Harry Zhang, Jiang Su: Naive Bayes for optimal ranking. J. Exp. Theor. Artif. Intell. 20(2): 79-93 (2008)

[21] Yang, Xiuxin. "A wearable real-time system for physical activity recognition and fall detection." University of Saskatchewan, September 2010.

[22] S. Preece et al., "Activity identification using body-mounted sensors—a review of classification techniques," Physiological Measurement, vol. 30, no. 4, pp. R1-R33, April 2009.

[23] Caruana, R. and Niculescu-Mizil, A.: "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international conference on Machine learning, 2006.

[24] Oppenheim, Alan V.; Schafer, Ronald W. (1975). Digital Signal Processing. Prentice Hall. p. 5. ISBN 0-13-214635-5.

[25] "High-pass Filter." *Wikipedia*. Wikimedia Foundation, 05 July 2014. Web. 09 May 2014. http://en.wikipedia.org/wiki/High-pass_filter

[26] A.K. Bourke, J.V. O'Brien, G.M. Lyons. "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm." Science Direct, Gait & Posture 26 (2007) 194–199.
.
[27] Zhang, Harry. "The Optimality of Naive Bayes." Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference FLAIRS 2004, AAAI Press, (2004).

[28] "Naive Bayes Classifier." *Wikipedia*. Wikimedia Foundation, 05 Feb. 2014. Web. 09 May 2014. http://en.wikipedia.org/wiki/Naive_Bayes_classifier

[29] Elgin, Ben (August 17, 2005). "Google Buys Android for Its Mobile Arsenal". Bloomberg Businessweek. Bloomberg. Archived from the original on February 24, 2011.

[30]"Industry Leaders Announce Open Platform for Mobile Devices" (Press release). Open Handset Alliance. November 5, 2007. Retrieved 2012-02-17.

[31] "Activities." *Android Developers*. N.p., n.d. Web. 09 May 2014. http://developer.android.com/guide/components/activities.html

[32] Stephen Brown,Joseph Timoney,Thomas LysaghtandDeshi Ye (2012) 'Software Testing: Principles and Practice'. Beijing, China.

[33] Albert, Mark V., Konrad Kording, Megan Herrmann, and Arun Jayaraman. "Fall Classification by Machine Learning Using Mobile Phones." Ed. Christian Lovis. PLoS ONE 7.5 (2012): E36556. Print.

[34] Ojetola, O., Gaura, E.I.,Brusey, J. "Fall Detection with Wearable Sensors--Safe (Smart Fall Detection)." Intelligent Environments (IE), 2011 7th International Conference.

[35] John P. Chin, Virginia A. Diehl, and Kent L. Norman. "Development of an instrument measuring user satisfaction of the human-computer interface." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 213-218.

[36] Mitchell Yuwono, Steven W. Su, Bruce Moulton. "Fall Detection using a Gaussian Distribution of Clustered Knowledge, Augmented Radial Basis Neural-Network, and Multilayer Perceptron." Broadband and Biomedical Communications (IB2Com), 2011 6th International Conference.

[37] Witten, Ian H., Eibe Frank, and Mark A. Hall. "Data Mining: Practical Machine Learning Tools and Techniques." Burlington, MA: Morgan Kaufmann, 2013. Print.

[38] Martinez, Wendy L., and Angel R. Martinez. Computational Statistics Handbook with MATLAB. Boca Raton: Chapman & Hall/CRC, 2002. Print.

[39] Herzog, Stephan. "Efficient DSP Implementation of Median Filtering for Real-time Audio Noise Reduction." Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13), Maynooth, Ireland, September 2-5, 2013.

[40] Rogelio Reyna, Edgard Palomera, Rogelio Gonzalez, Sergio GarcÌa de Alba, and Michelle Clifford. "Human Fall Detection Using 3-Axis Accelerometer." Reference Manual, Freescale Semiconductor, Inc., 2005.

[41] "NewYork-Presbyterian. The University Hospital of Columbia and Cornell." *Fall Prevention*. N.p., n.d. Web. 28 May 2014. http://nyp.org/health/nontrauma-falls.html

[42] Laybourne, A. H., S. Biggs, and F. C. Martin. "Falls Exercise Interventions and Reduced Falls Rate: Always in the Patient's Interest?" *Age and Ageing* 37.1 (2007): 10-13. Web. http://ageing.oxfordjournals.org/content/37/1/10.full.pdf

[43] Sharkey, Paul. *Seizures Occurring at Night Time Are Not Qualitatively Different from Seizures, Which May Be Present in the Daytime* (n.d.): n. pag. Web. http://www.epilepsy.ie/assets/30/D2E3014F-F928-9571-2D57E3F9D53DD1F0_document/Nocturnal_Seizures.pdf

[44] *WHO Global Report on Falls Prevention in Older Age*. Geneva, Switzerland: World Health Organization, 2008. Web. http://www.who.int/ageing/publications/Falls_prevention7March.pdf

[45] Narayanan, M. R., Lord, S. R., Budge, M. M., Branko G. Celler, & Lovell, N. H. (2007). “Falls Management: Detection and Prevention, using a Waistmounted Triaxial Accelerometer”. Paper presented at the 29th Annual International Conference of the IEEE EMBS, Lyon, France.

[46] Bilney, B., Morris, M., & Webster, K. (2003). Concurrent related validity of the GAITRite® walkway system for quantification of the spatial and temporal parameters of gait. Gait & Posture, 17(1), 68-74.

[47] "What Is Incremental Model- Advantages, Disadvantages and When to Use It? - ISTQB Exam Certification." *ISTQB Exam Certification*. N.p., n.d. Web. 29 May 2014. http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/

[48] Iskander, Magued, Vikram Kapila, and Mohammad A. Karim. *Technological Developments in Education and Automation*. Dordrecht: Springer, 2010. Print.

[49] "Signal Processing." *Wikipedia*. Wikimedia Foundation, 25 May 2014. Web. 01 June 2014. http://en.wikipedia.org/wiki/Signal_processing

[50] Lyons, Richard G. *Understanding Digital Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 2011. Print.

[51] "Using Averages." — *University of Leicester*. N.p., n.d. Web. 01 June 2014. http://www2.le.ac.uk/offices/ld/resources/numeracy/averages

[52] Kuo, Sen M., Bob H. Lee, and Wenshun Tian. *Real-time Digital Signal Processing: Implementations and Applications*. Chichester, England: John Wiley, 2006. Print.

[53] "Android App Development Tutorial." *Android Architecture For System Application Software Stack*. N.p., n.d. Web. 01 June 2014. http://android-app-tutorial.blogspot.ie/2012/08/architecture-system-application-stack.html

[54] "Introducing Android Wear." *Android Developers*. N.p., n.d. Web. 29 May 2014. http://developer.android.com/develop/index.html

[55] "Location Strategies." *Android Developers*. N.p., n.d. Web. 01 June 2014. http://developer.android.com/guide/topics/location/strategies.html

[56] Behutiye, Woubshet. “Android ECG Application Development ”. Oulu University of Applied Sciences.2012.

[57] "Accuracy and Precision." *Wikipedia*. Wikimedia Foundation, 25 May 2014. Web. 02 June 2014. http://en.wikipedia.org/wiki/Accuracy_and_precision

[58] "HCC Education Digital Library." *: Web Lectures*. N.p., n.d. Web. 02 June 2014. http://hccedl.cc.gatech.edu/taxonomy/webLectures.php

[59] "Arduino - ArduinoBoardUno." *Arduino - ArduinoBoardUno*. N.p., n.d. Web. 04 June 2014. http://arduino.cc/en/Main/arduinoBoardUno

[60] "Bildr » Stable Orientation – Digital IMU 6DOF + Arduino." *Bildr RSS*. N.p., n.d. Web. 04 June 2014. http://bildr.org/2012/03/stable-orientation-digital-imu-6dof-arduino/

[61] Chris Soraghan, Postdoctoral Researcher in Technologies for Older People at Mercer's Institute for Successful Ageing (St.James'sHospital), private communication.

[62] "Acer Liquid Z3." - *Full Phone Specifications*. N.p., n.d. Web. 04 June 2014. http://www.gsmarena.com/acer_liquid_z3-5624.php

[63] "UML Tool for Software Design." *UML Tool for Software Design*. N.p., n.d. Web. 12 June 2014. http://www.visual-paradigm.com/features/uml-and-sysml-modeling/

[64] Wang, Xin. "Two-dimensional Bayesian Estimator for Image Filtering." *IEEE Transactions on Image Processing* 8.7 (1999): 993-96. Web. https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Image%20Filtering_2up.pdf

[65] Ehringer, David, and 2010 March. *T H E D A LV I K V I RT U A L M A C H I N E A R C H I T E C T U R E Introduction* (n.d.): n. pag. Web. http://davidehringer.com/software/android/The_Dalvik_Virtual_Machine.pdf

[66] "6 Degrees of Freedom IMU Digital Combo Board - ITG3200/ADXL345." - *SEN-10121*. N.p., n.d. Web. 12 June 2014. https://www.sparkfun.com/products/10121

[67] "SensorEvent." *Android Developers*. N.p., n.d. Web. 13 June 2014. http://developer.android.com/reference/android/hardware/SensorEvent.html