uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Mohammed Alhamid
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Hamon: An Activity Recognition Framework for Health Monitoring Support in Home Environments

TITRE DE LA THÈSE / TITLE OF THESIS

A. El Saddik
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Peter Liu

Jiying Zhao

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Hamon: An Activity Recognition Framework for Health Monitoring Support in Home Environments

Mohammed Alhamid

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Computer Science

Under the auspices of the Ottawa-Carleton Institute for Computer Science

University of Ottawa
Ottawa, Ontario, Canada
July 2010

Your file Votre référence
ISBN: 978-0-494-73784-2
Our file Notre référence
ISBN: 978-0-494-73784-2

Abstract

Nowadays, technology advances accelerate the quality and type of services provided for health care and especially for monitoring health conditions. Sensors have become more powerful to sense different physiological signs and have the ability to be worn on the human body using wireless communication modules. A variety of software tools have been developed to help in processing a variance list of vital signs by analyzing and visualizing data generated by multiple sensors.

In this thesis, we introduced a Health signs and Activity recognition MONitoring framework (Hamon). Hamon, of German origin meaning a home protector, is designed to be an enabling prototype for health monitoring applications. Using off-the-shelf sensors, we implemented an activity detection framework for detecting five types of activity: falling, lying down, sitting, standing, and walking. The framework collects and analyzes sensory data in real-time, and provides different feedback to the users. In addition, it can generate alerts based on the detected events and store the data collected to a medical sever. Context information such as the weather condition, the type of activity, and physiological data collected such as the heart rate, is also integrated in the framework.

A number of challenges have been addressed in this thesis including: the design and the implementation of the framework, the selection and the implementation of the activity recognition algorithm, and the classification method used for detection.

Acknowledgment

My sincere gratitude goes to my academic supervisor Prof. Abdulmotaleb El Saddik, who helped, encouraged, and guided me through the past two years towards finishing this thesis. He always advises me not only in my academic work but also in my personal success.

I am also thankful to Dr. Jamal Saboune, Dr. Heung-Nam Kim, and Dr. Atif Alamri for their efforts in guiding me with their valuable technical feedback and suggestions during the progress of this work.

A special thanks for my beloved parents who were supporting me throughout my graduate studies. They are always the source of strength and cheerfulness throughout my life.

I also don't want to forget my best friends and family members who helped me during hard times and cheered me up for keeping up the good work. I would also like to thank the volunteers who came to the lab for giving me their time so I could perform the experiments. Finally, I am thankful to all the Multimedia Research Laboratory (MCRlab) members for the good times spent together working in different research projects.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
API	Application Programming Interface
AR	Activity Recognition
ADL	Activity of Daily Living
BSN	Body Sensor Network
C++	Referred to the name of a programming language
CPT	Conditional Probability Distribution
ECG	Electrocardiogram
EEG	Electroencephalography
EMG	Electromyography
ERD	Entity-relationship Diagram
GUI	Graphical User Interface
HMM	Hidden Markov Model
IT	Information Technology
PD	Parkinson's Disease
PDA	Personal Digital Assistant
RAM	Random Access Memory
RFID	Radio-Frequency Identification
nesC	Network embedded Systems C
WBSN	Wireless Body Sensor Network

Chapter 1. Introduction

With the rising number of elderly people in many countries around the world, the expectancy of receiving better health care services increases the pressure on health care providers. Therefore, different list of technological solutions become of interest for both researchers and scientists to push the advancements in Information Technology (IT) toward improving the quality of health care. One of the solutions towards such improvements is considering a long-term monitoring of the patient's health situation by building a health monitoring system.

A long-term monitoring service becomes a necessary requirement for detecting any abnormality in the patient's health situation and particularly during their time spent at home. Different technologies have been adapted and integrated to support the monitoring of patients while they perform their regular activities at home. Some of the technologies include: video conferencing, sensor network attached to the patient's body, and virtual reality for simulating physical feedback in a virtual environment [1] [2]. These technologies are used to build up computer-based monitoring application, an optional Body Sensor Network (BSN), and a telemedicine option.

Wearable sensors are considered one of the important solutions for such monitoring requirements. Using wearable sensors, we can detect any emergency that appears in the patient's health status as early as possible. Such sensors can help in continuous monitoring of patients with different chronic diseases. For instance, patients who suffer from cognitive heart failure, hypertension, trial fibrillation, Parkinson's Disease, or Alzheimer's Disease can benefit from many existing wearable systems [3] [4] [5].

BSN refers to the integration of wearable sensors and other computing devices attached to collect information from a human body or a surrounding environment for different applications. The sensors, along with the communication protocols used in connecting them, form a network for collecting sensory information called BSN. Letting patients wear a BSN or installing such a network within their living environment enables the caregiver to report and collect health-relevant information. A Wireless Body Sensor Network (WBSN) incorporates tiny wireless sensors that can be mounted on people and thus allow them to roam freely in their location while they are being monitored.

From the IT point of view, there are two main approaches to improve the quality of health care services: prevention and detection. Designing a monitoring system to be used during daily life activities or within the clinical boundaries can be very useful in detecting a variety of physiological signs or related risks. Multiple physiological signs can provide important facts during the diagnosis of many health risks that appear in the patient's health status such as Electrocardiography (ECG), body temperature, Glucose level, oxygen levels, and blood levels. On the other hand, prevention can be considered an alternative approach by building a system to prevent dangerous consequences of abnormal conditions detected on the patient's physiological signs [6]. Some examples include monitoring the patient after falling down which is known as falling prevention, early warning of heart rate variability which prevents any cognitive heart failure, abnormal execution of movement such as abnormal walking which prevents any loss of balance, and many others that serve in preventing health risks.

The technological advancements in the wearable sensors increased the viability of the required health prevention and detection needs under many monitoring scenarios. The sensors play an important role in quantifying the patient's movement along with monitoring a variety of physiological parameters. Sensors such as accelerometer, gyroscope, Electrocardiography (ECG), Electromyography (EMG), and others provide detailed information, all of which useful in measuring the patient's movements including health factors [5]. Using an accelerometer and gyroscope, we can measure the acceleration of multi-axis movement along with the angle of such movement against gravity. The sensors can also represent the magnitude and the direction of the acceleration, as well as the vibration sense of such movements [7]. In addition, different sensors were helpful in capturing different assessment patterns of body movements and motor disabilities, and clinically used to assess and measure the slowness of movement [8] [9].

Due to the fact that patients prefer staying at home versus going to the hospital, home monitoring and telemedicine become a very important research field to facilitate patient requirements [10]. In fact, not all patients are able to have someone helping them to drive back to the hospital either for regular checkups or even for attending a rehabilitation session. Moreover, some patients such as post-stroke patients have a lower functional level when exercising their daily needs normally. Accordingly, we realize how important it is to consider many monitoring options that overcome many limitations such as the monitoring duration and location settings. In other words, we need a framework to give patients the full freedom to exercise and practice their lives normally at home and, at the same time, allow a health care provider to monitor their progress remotely by receiving performance feedback reports. In addition, the health care provider can guide the patients based on their treatment progress by adjusting the duration or the type of exercises in the long-run plan. The collected measurement values can be stored remotely at the clinic site to be analyzed or stored in an archive database.

To overcome our needs, we developed a framework that can be applied for different health monitoring applications serving a variety of needs which include detection, risk prevention, and a convenient method for monitoring physical rehabilitation. In Hamon, we used different types of sensors including an accelerometer, gyroscope, and ECG, all of which are wireless and wearable so patients can wear and roam freely in their home environment.

1.1. Motivation

The main motivation of the presented research is a reflection of the high number of elderly people who need long-term health monitoring. The number of elderly persons has increased in recent years. Globally, there are more than 600 million persons over the age of 60 around the world, and that number is expected to approach two billion persons in 2025 [11]. The World Health Organization (WHO) estimated that 80 per cent of the elderly population is from developed countries, where 20 per cent of those people would have limitations in living independently [12]. In particular, statistics from 2001 showed that 17 per cent of the population in Europe was 65 or older, and by 2035, an estimated 33 per cent will be 65 or older [13]. In Canada, the number of elderly people aged 65 years and older was 10.6 per cent of the population in 1991. This number is expected to increase rapidly, reaching 14.5 per cent by 2011 and 21.8 per cent by 2031 [14].

On the economical side, monitoring patients in their homes is recently considered cost efficient to maintain good care services. The yearly nursing cost of an Alzheimer's patient is on average $64,000 [15]. Using telemedicine, the cost can be reduced to $20,000 by considering a home care option [15]. W.D. Bradford et al. [16] studied the willingness of the patients to pay for telemedicine. Patients involved in this study suffered from either chronic heart failure or hypertension. Between 30 per cent and 50 per cent of the patients showed an interest to pay $20 per month for having a telemonitoring option. S.J. Dijkstra et al. [17], studied the business model for providing a telemonitoing care services. Patients with chronic heart failure had the highest willingness to have a home monitoring service compared with other types of patients.

As a result, there is enormous pressure on public and private health resources to build and develop alternative IT applications to reduce costs while improving the quality of life provided for elderly people. Accordingly, the demand of having telemonitoring systems to support health care is increasing in North America and Europe particularly. By having home monitoring features, the ability for elderly people to live independently increases in safe manner. Family physicians can easily check and refer back to the status of the patient's health situation with different mobility features. Hospitals then would avoid an unnecessary number of patients hospitalized and maintain good service for those who need urgent care.

1.2. The Research Problem

The recent advances in microelectronic development have accelerated the use of different sensors for measuring different vital signs. Integrating them together could lead to a framework that can help monitoring elderly people's activities and detect any abnormality in their collected physiological data. Moreover, monitoring patients at home improves their treatment progress by providing an effective means of collecting physiological data during patients' Activities of Daily Living (ADL) and prevents any risks that may affect their health status.

One main objective of our work is to develop an open software framework for activity recognition that can be applied in rehabilitation, physiological monitoring, or behavioural research. The other main objective is the development of a convenient method of analyzing and presenting the physiological data collected from a WBSN. Concurrently, a caregiver can extract medical relevant information together with the patient's condition and his/her progress during each physical exercise. This provides the ability to monitor the ADL which includes fall detection, walking, sitting, standing, and lying down as well as physiological monitoring such as heart rate.

1.3. Scenarios

In this section, we illustrate a few scenarios where we can apply the proposed framework. For example, the framework can be used in monitoring of an elderly patient at home. Suppose a patient in her 70s suffering from continuous high blood pressure and a heart disorder. The patient's family physician recommended constant monitoring at home. For this reason, the patient starts wearing an ECG sensor, blood pressure, and motion sensors. A computer station is installed in her home to collect and analyze the sensory data collected from the BSN and send back the results to the medical server. In this case, the patient's physical and context information is monitored for detecting any abnormality in her health situation.

Another scenario for detecting the patient's condition is when the patient performs physical exercises where the monitoring framework can combine the ECG signals along with the acceleration information. For activity monitoring, many different sensors can be used within the patient's environment such as a pressure sensor under the carpet. It can be put under the carpet around the bed to detect whether the patient is getting sleep. The accelerometer along with the gyroscope can give valuable signals used for postural and dynamic activity recognition. Galvanic skin response sensors and body temperature sensors can also be integrated for full context-awareness monitoring.

Hamon can also be applied in monitoring the progress of practicing prescribed physical exercises for patients in cardiac rehabilitation programs and help in measuring the functionality of patients with Parkinson's Disease (PD). Specifically, the framework can monitor the duration of the exercise as well as the number of sensors attached to the patient. At the end, this enables the patient to reach the highest possible level of function by comparing the patient's movement accelerations throughout the treatment plan. To build a prototype with a real scenario, we applied the framework to test the tremor in the hand-reach movement. Tremor is one of the signs of PD where PD patients experience both types of tremor: resting and postural tremor. Both types are considered neurological symptoms which disturb the patient's daily living activities and can be noticed in their hands and fingers.

Rigidity is another symptom that is common in PD patients. It is defined as slowness in executing movement resulted from an increase or stuffiness in the muscle tone, and can impact movement [19]. The last symptom is postural instability where patients may fall down accidentally or experience multiple repeated falls with some difficulties in performing forward or backward movements. In addition, patients with PD also experience problems with walking, arm swinging, or initiating movement tasks. During the experiment, the subject's hand and forearm are held in the same position and the data collected is subjected to Fourier transformation to obtain a spectral analysis. The details of how we measured the level of tremor and the obtained experimental results are available in one of our published works [20].

1.4. Thesis Contributions

In brief, the main contributions of this thesis can be highlighted in the following two points:

1. The design and the development of a flexible real-time physiological monitoring framework using wireless sensors including presentation and alert mechanism in a home environment.
2. The design and the development of an activity detection algorithm based on the standard K-nearest neighbour (KNN) with an applied Bayesian Network for improving the detection rate. In addition, a study of the resulted accuracy and reliability of the proposed algorithm coupled with the detection performance of each accelerometer sensor used has been included.

1.5. Resulting Publications

Three papers have been published:

1. Md. Abdur Rahman, Mohammed F. Alhamid, Wail Gueaieb, Abdulmotaleb El Saddik. A Body Sensor Network Toward Ambient Intelligence: an e-Health Perspective. Medical Measurements and Analysis (MeMeA) 2009, Cetraro, Italy, May 29-30 2009.

2. Md. Abdur Rahman, Mohammed F. Alhamid, Wail Gueaieb, Abdulmotaleb El Saddik. A Framework to Bridge Social Network and Body Sensor Network: an e-Health Perspective. International Conference on Multimedia and Expo (ICME) 2009, Cancun, Mexico, June 28--July 3 2009.
3. Mohammed F. Alhamid, Atif A. Alamri, Abdulmotaleb El Saddik. Measuring Hand-Arm Steadiness for Post-Stroke and Parkinson's Disease Patients Using SIERRA Framework. Medical Measurements and Analysis (MeMeA) 2010, Ottawa, Canada, Apr 30-May 1 2010.

1.6. Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 surveys existing work on Activity Recognition (AR), Body Sensor Network (BSN), and health monitoring application. Chapter 3 describes the framework design and its composed components. Chapter 4 describes the activity recognition algorithm used for detection along with the details of the classification method. Chapter 5 describes the technical details of implementing each component presented in the framework. Chapter 6 presents the experimental results including results discussion. Chapter 7 concludes the thesis with future vision.

Chapter 2. Background and Related Work

2.1. Literature Background

This chapter is organized in a way to start discussing the history usage of accelerometers along with other physiological sensors in health care. Then this chapter shows their suitability to be applied for health care monitoring and diagnoses. Afterward, it brings the attention to different existing health monitoring application and frameworks with a brief comparison between the existing and the proposed framework. Finally, this chapter discusses the background of different activity recognition algorithm used in the literature.

In 1920, McCollum and Peters developed an accelerometer for measuring the resistance of a bridge. In 1935, the accelerometer was widely used commercially in different applications such as aircrafts and bridges. The two-axis version of the accelerometer was founded in 1932, and meant to be used in measuring the accelerations in the airplane catapult, the vibration of steam turbines, and some underground pipes [21]. Similarly, the accelerometer was first used in analyzing human movements in the early 1950s, yet the accuracy and the sensitivity were limited. Now, they are much smaller in size and highly efficient in continuous data recording with many enhanced features. In addition, they are widely integrated with most of our daily used electronics such as smart phones and video game controllers. It is expected of three phones, one with built-in accelerometer feature by the end of this year [22].

Accelerometers are one of the low-cost solutions to quantify and detect physical activity level. They measure the acceleration of multi-axis movement and can present the magnitude and the direction of the acceleration. The data collected from such sensors represent the change of accelerations against both body movement and gravity. Such features enable the accelerometer to capture postural orientation along with body movements [7]. Different factors can be calculated from the collected acceleration data, for instance, static and dynamic acceleration against gravity, the tilt of an object, velocity, vibration, and the object orientation. A single axis accelerometer measures the acceleration of a single axis against gravity. On the other hand, the multi-axis accelerometer measures the change of acceleration of the three axes X, Y, and Z in the 3D environment.

Accelerometers are widely used in different monitoring scenarios. They have been used in monitoring and treating different diseases such as hand tremors in Parkinson's Disease and in stroke and cardiac rehabilitation. Especially, they are helpful in capturing different assessment patterns of hand movements and the level of motor disabilities such as the slowness of executing movement [8], [9]. Real-time monitoring systems also consider accelerometers for measuring the level of functional ability. Measurement of functional abilities is achieved by analyzing different physical activities such as walking, cycling, ascending or descending stairs. In addition, fall detection, and energy spent in executing each activity is also analyzed using pre-set pattern recognition and fixed threshold.

The data collected from the accelerometer sensors used for health monitoring purposes is fundamentally biomechanical signals collected in clinical assessments. During the patient experiments, the therapist or the doctor defines the type of tasks to be performed which vary from quite sitting, finger tapping, walking, or a simple hand movement. Afterward, the data is analyzed by segmenting the data based on time frame. A thirty or fifty second window can be used to analyze the task being studied depending on the selected activity detection algorithm. The communication between the body area network and the analyzer application can be established using wireless or wired microcontrollers structure. The wireless option enables the data to be transferred suing either Bluetooth or ZigBee to the end point application. The analysis can be carried on either using a terminal machine or a Personal Digital Assistance (PDA).

T. Tamura et al. [23], built a mechanism for showing the coloration between motion accelerations and the heart rate. The study focuses on capturing the motion patterns along with the heart rate while subjects are performing daily activities. It also shows the type of activities where heart rate was higher than during other activities. Marie-Laure Wetzler et al. [55] studied the two-axis acceleration signal while monitoring the subject's blood pressure and heart ECG signal. The goal of their work is to evaluate the accuracy and reliability of monitoring the subject's activities based on a dual-axis accelerometer attached to the thigh. The analysis of the acceleration signals collected using their system was based on the wavelet transform.

Telehomecare is an example of telemonitoring by saving time and cost in monitoring body movements in continuous care. Since most elderly patients prefer to stay home, avoiding the option of being monitored within any medical institute, the need of having telehomecare is increasing. The first system proposed was in 1994 by Celler et al. [24]. Celler's system aimed to detect when the person enters a room by analyzing movements where infrared switches are used to define the room's area of interest. The main objective of that work was studying the interaction between the elderly patients and their living environments by monitoring and identifying changes in their health statuses. Aavon Quigley et al. [25], proposed a home-based system for collecting rich context data aiming to help elderly people in fall prevention while improving their cognitive functionality.

Grioux et al. [26] tried to fully monitor patients in their homes by equipping the patient's location with different sensors, such as drawers equipped with electromagnetic contact for detecting when the door was open or closed, and integrating contexts such as whether there was a meal still cooking in the kitchen by measuring the smoke level. Such sensors can save the patient's location and life in case the patient, for example, forgot to switch off the oven. The advances of the Radio-Frequency identification (RFID) technology can enable us to equip the entire location by different RFID tags to control different electronic appliances. Accordingly, Grioux and others helped to detect when the patient forgot to switch off the oven by developing a monitoring system that can detect such a scenario. Using data retrieved from positioning RFID reader and tags, accelerations from the accelerometer and other physiology sensors, Grioux et al. developed an activity tracking system for elderly people at home.

Oliver and Flores-Mangas [27], proposed a wireless real-time system for monitoring different physiological signals. They used a wearable sensor connected via Bluetooth to a cell phone. The cell phone stores, transmits, and analyzes the physiological data. The main purpose of their system is to automatically detect sleep apnea. Mercury [40], is another wearable sensor network used for high-fidelity motion analysis. It was designed to support applications that are data-intensive. SHIMMER sensors [28] have been used in the Mercury system for measuring neuromotor disorders, which is the same type of sensor we used in our proposed framework. The objectives of the Mercury framework are different from what we are trying to achieve here since Mercury focuses more on the hardware and node computational load including resources consumption and performance.

DexterNet [29] is an open body sensor platform for building a real-time body sensor network that enables indoor and outdoor monitoring. It is open source since it relies on the SPINE framework [30], which is another open source framework for building wireless body sensor networks. DexterNet supports real-time human monitoring and was designed to support different application for elderly living support. The framework has been tested with built-in motion sensors with one application called Avatar for body action monitoring. Since DexterNet is essentially built on the top of the SPINE framework, it hasn't been considered to meet our requirements. The details of SPINE limitations to our requirements will be discussed in Framework Comparison 2.2.

Lombriser et al. [38] used TinyOS 2.0 to program some of the accelerometer sensors to collect body motion accelerations in order to detect several types of activities. The activities include moving a computer mouse, writing on a white board, opening a drawer, opening a cupboard, typing on a keyboard, and writing using a pen. The sample rate used is 32 Hz with 2.5 seconds as a window size with 70 per cent overlapping. The collected data is analyzed offline using WEKA tool [57]. WEKA has been used to train the K-NN classifier with Manhattan distance compared to the performance of J48/C4.5 method. The K-NN classifier was successful in scoring 91 per cent for the recognition rate where the k=5. SATIRE [31] is another body network that classifies daily living activities by analyzing the accelerations resulted from body movement. The activity classification used in SATIRE is based on Hidden Markov Model applied on the signal power.

Nisham Ravi et al. [18], attempted to use a single triaxial accelerometer to detect eight set of activities: standing, sitting, walking, running, climbing up stairs, climbing down stairs, vacuuming, and brushing teeth. During their experiments an accelerometer sensor was attached to the pelvic region of the patients and configured to sample at 50 Hz. An HP iPAQ was used to collect the data from the sensor using Bluetooth. The features extracted from the accelerometer signals included mean, standard deviation, energy, and correlation. Five classifiers were implemented using WEKA toolkit to distinguish between the different activities. As a result, the paper shows a performance evaluation of using decision tables, decision trees (C4.5), K-nearest neighbours, SVM, and Naïve Bayes. The main goal of the work presented was to answer a question of whether combining different classifiers would increase the recognition rate.

2.2. Framework Comparison

In this section, we highlight some related frameworks used for healthcare monitoring including software and hardware architectures for building the BSN and structuring the real-time monitoring system. The first paragraphs describe each framework briefly and a comparison table ends this section by highlighting the main features of each candidate.

SPINE framework [30] is an open source platform for building a WBSN. Salmeri et al. [32] built architecture to combine context awareness and a BSN that can be applied in a health care application based on the SPINE platform. The main problem of SPINE to satisfy our specific needs is the inability of SPINE in supporting Bluetooth on the SHIMMER platform. The documentation of the SPINE framework didn't mention in detail such an ability. We consulted the spine-dev mailing list and the reply was as follows: "SPINE doesn't have native support for the Bluetooth on the SHIMMER platform, but we'll certainly appreciate any efforts in this direction..." (May 22, 2009). As a result, we tried to find a solution. After a few unsuccessful attempts, we decided to consider other platforms to help us build our own framework.

CodeBlue [33], is another wireless sensor framework to support health care. Unill the time of writing this thesis, CodeBlue is still under development. Its system source code, according to their website, is bought out soon. The main objective of their work is to integrate different medical sensors with a low-power wireless option. CodeBlue uses TinyOS software along with Graphical User Interface (GUI) application to query and display vital signs.

Karantonis et al. [34] built a real-time monitoring system to monitor human movement classifier using a triaxial accelerometer. This work is not a framework, but rather an implementation of real-time classification system.

Otto et al. [35] proposed a system for monitoring body motion with heart activity. They used two 3-axis accelerometers for upper body tilt for motion detection to determine the orientation and the type of activities. A PDA plays the role of personal server as a medium between the wireless body area network and the medical server. However, the paper didn't mention in detail the detection algorithm used or the classification method. Instead, the focus was more on the software architecture of distributing the sensory data between the personal and the medical servers.

Jovanov et al. [6] used motion sensor for assisting physical rehabilitation. The data collected from different motion sensors are stored in database for offline processing. They used variety of sensors such as ECG, EMG, and EEG. The algorithm used for activity detection is based on wavelet transform to find the correlation between the heel-strike and toe-off which represents the step detection. The angular velocity is measured using a gyroscope attached to the accelerometer. For classification, they used power spectrum analysis and a Gaussian model.

Wood et al. [36], proposed a context-aware wireless sensor network for assisting elderly people. The system enables an access to data in real-time acquiring different sensory information. It provides context-awareness by integrating environment and physiological data by implementing a two-way protocol of data flow.

Rajasekaran et al. [37] proposed a system composed of wearable sensors for monitoring elderly people under drug therapy. The computation of alert detection is carried on the sensor boards while communicating to a medical informatics services with the data collected.

Lombriser et al. [38] proposed an online activity recognition system based on SensorButton architecture for recognizing body motion. The SensorButton used to collect context information for classification. The platform installed on the sensor node is TinyOS 2.0 to mote an accelerometer sensor.

In Table 1, we show the main features of the previously listed frameworks and monitoring systems available in literature. In this table, we arranged the columns to have an alphabetical shortcut of the studied candidates.

The evaluated candidates are listed as following:

A. Architecture to combine context awareness and body sensor networks for health care applications, presented in [32].

B. CodeBlue, a sensor network framework for medical care [39].

C. An implemented system for real-time human movement classifier [34].

D. Proposed system architecture of a wireless body area sensor network for ubiquitous health monitoring, presented in [35].

E. A proposed wireless body area network of intelligent motion sensors for assisting physical rehabilitation, presented in [6].

F. AlarmNet, context-aware wireless sensor networks for assisting-living and residential monitoring, presented in [36].

G. A proposed system for elderly patient monitoring using a wireless sensor network, presented in [37].

H. On-body activity recognition system in a dynamic sensor network, presented in [38].

I. Hamon, our proposed framework.

We divided each evaluated criteria into three levels depending on the satisfaction level of the listed requirement into: satisfied, partially satisfied, and not satisfied.

S: Satisfied.

P: Partially satisfied.

N: Not satisfied.

Table 1: Some related monitoring frameworks and implemented systems compared with Hamon

	Monitoring Framework and Systems								
Features	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
Real-time data acquisition	S	S	S	S	S	S	S	S	S
Provide context-aware information	S	P	N	N	N	P	P	P	S
Combines sensor and context data	S	P	N	N	N	S	P	P	S
Patient's location	S	S	N	N	S	S	N	N	N
Broad signal processing extensions	S	N	S	N	S	P	S	P	S
Physiological parameters monitoring	S	S	N	P	S	S	S	N	S
Activity Monitoring	S	N	S	S	S	P	S	S	S
Logical alert mechanism	S	P	N	N	S	S	S	N	S
Flexibility adding different services	P	S	N	N	N	S	S	N	S
Transmit sensor result wirelessly	S	S	S	S	S	S	S	S	S
Real-time analysis (online activity recognition)	S	P	S	P	S	S	S	S	S
Flexible sensor configuration	P	S	N	S	S	P	P	N	S
Client Side analysis PDA	S	N	N	S	N	S	N	N	N
Collect Data 24 hours	S	P	S	S	S	S	S	P	S
Ability to be deployed on a large scale	P	S	P	S	S	S	S	P	S
Two-way data acquisition	P	S	N	P	N	S	N	N	S
Security-aware architecture	N	S	S	N	N	S	N	N	N

Chapter 3. Hamon Design

3.1. Overview

In this chapter we will discuss the design of Hamon framework. The first section describes in detail the framework architecture including the software and the hardware specification used in the framework. The second section discusses the alert broadcasting technique along with the graphical representation of the physiological data collected from the BSN. We conclude this chapter by highlighting the user roles in the system with some discussion about the data privacy.

3.2. Hamon Architecture

The proposed system architecture consists of two main parts: the structure of the sensor network and the structure of the monitoring application implemented on the server side. The BSN consists of wearable sensors that collect motion data as well as health vital signs such as the heart rate. Each sensor has a wireless Bluetooth module built in to communicate with the user's server side. The user's server side accommodates the monitoring application and can be installed either in the patient's home or in the health institute for collecting and analyzing the data as illustrated in Figure 1. Since the communication range of Bluetooth module used supports between 10 and 15 meters, a Bluetooth signal expander can be considered to increase the area coverage. However, the framework was tested while the subjects were roaming in less than 10 meters away from the server. The server should have the Internet connectivity to send and retrieve data from the medical server or be connected to a local network if both servers are on the same site. The monitoring interface resides on the user side's controls and configures the sensor network with the ability to plug-in or plug-out sensors during the real-time execution. In addition, it provides a Graphical User Interface (GUI) to present the collected signal to the healthcare provider. The collected data can be accessed offline as well.

The information about the subject status along with the triggered alerts are handled by the user's side and manipulated before being stored on the medical server. If an alarm has been triggered like falling down, then a service procedure on the user's server process the alert by sending and activating a group of notification lists. The details of the alarm and alert handling are described in details in Section 5.4. The medical server keeps track of the patient's health records along with all monitoring data recorded by the client side.

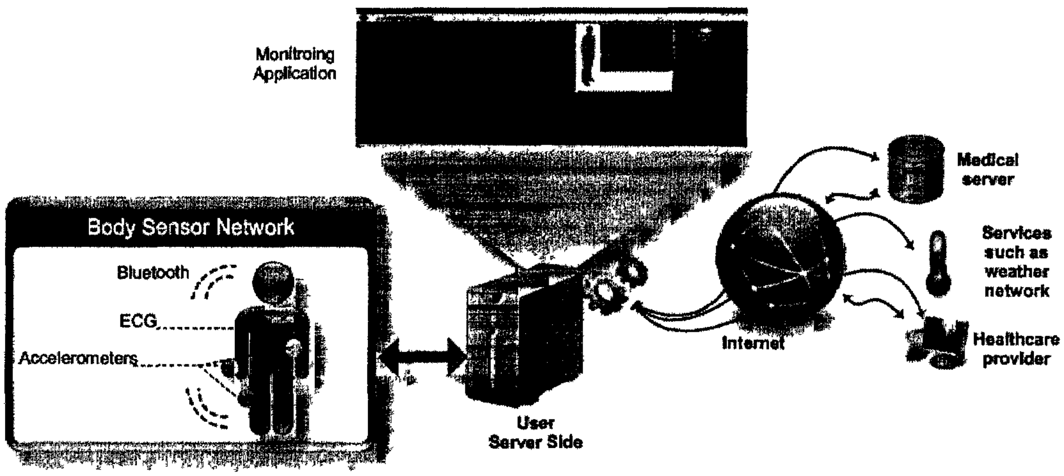

Figure 1: Data flow diagram of the proposed framework.

The healthcare providers have their own privileges for configuring the type of exercise, duration, monitoring mode, number of sensors to be used and other monitoring parameters. The patient's therapist has specific interfaces for controlling the mentioned parameters coupled with the interfaces for viewing and manipulating the sensory data. A technical administration panel is implemented to handle the technical configuration and management of the framework connectivity. The role can take care of setting the connection between the user server side, the medical server, and the Bluetooth COM ports to establish the streaming channel between the three main parts of the framework (BSN, User Server Side, and Healthcare provides).

3.2.1 Hardware Components

During the implementation of the prototype version of the proposed framework, we included a list of hardware devices to incorporate the wanted functionality. The framework hardware components composed of motion sensors, biosensor, and Radio-frequency identification (RFID) reader and tags. The type of motion sensors used are 3-axis accelerometers and 2-axis gyroscope for rotation. The Biosensor used is a 3-leads Electrodiagram (ECG).

Sensors attached to the human body for different monitoring purposes are always preferred to have minimal weight, low-power consumption, the ability to be integrated with other wireless body sensor networks, and easily customizable. In fact, there are some of technical side limitations to many existing wireless sensors used for motion detection, or measuring different physiological signs other than computational resources and power. To name two, wireless communication range and the right body segment for placing the sensor are issues to be addressed to obtain the highest accuracy of data recording.

In order to find the right sensor to use within our framework, we set a list of requirements for the sensor board:

- The sensors should not have the option to turn them on/off independently. They should be always ON unless the power went out in order to provide the monitoring application the full control to switch them ON/OFF.
- Equipped with a wireless interface.
- Small and lightweight with acceptable battery life.
- Wearable and able to be used for long-term monitoring.

The framework acquires the acceleration signals against both body motion and gravity. For this reason, we picked SHIMMER [28] accelerometers after reviewing different related and existing work [40][41][38]. SHIMMER is considered one of the smallest wireless sensors available on market. It has a lithium-polymer battery, and a TI MSPP430 microprocessor with Bluetooth radio for streaming high frequency rates. The size of the SHIMMER is suitable for roaming subjects, since it is only 1.75" x 0.8" x 0.5" and weighs 10g only (see Figure 2).

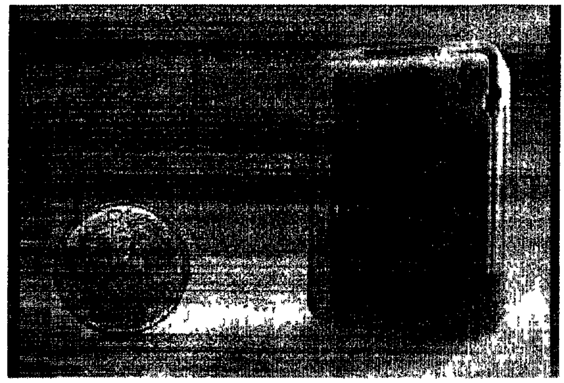

Figure 2: The SHIMMER wireless accelerometer sensor. Source: [28].

Our framework uses the SHIMMER accelerometer and ECG sensors to implement the prototype. However, Hamon is not limited to the SHIMMER platform structure. Figure 3 shows the hardware components also used in the framework. The sensors appearing in the figure are small in size and weigh much less than many other commercial sensors. Each sensor has a 280 mAh lithium-ion rechargeable battery which on average last 15 hours.

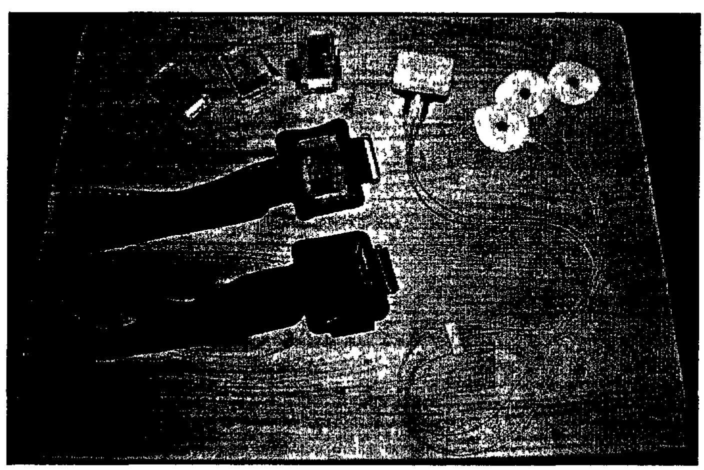

Figure 3: The picture shows the type of sensors used in the implemented prototype: accelerometers and an ECG sensor.

3.2.2 Software Architecture

The software architecture is composed of some components developed during this work and other open source ones. As it appears in Figure 4, the framework consists of three main software levels. Starting from the lowest level, the first layer is TinyOS [42]. TinyOS in an open source operating system written in a similar language to C called Network embedded Systems nesC. TinyOS provides different components to manipulate and work a tiny sensor. A couple of mote sensors can run TinyOS, one of which is Intel SHIMMER mote which has been selected as a hardware component in the implemented prototype. However, if the framework needs to integrate different sensors that use other mote operating system, the layer could be only changed according to the operated mote program. TinyOS has a list of compatible motes and a number of open source applications that can be installed on a mote. An application needs to be compiled using files called "make" before flushing the application to the mote. The details of the TinyOS files and their code are discussed in Chapter 5.

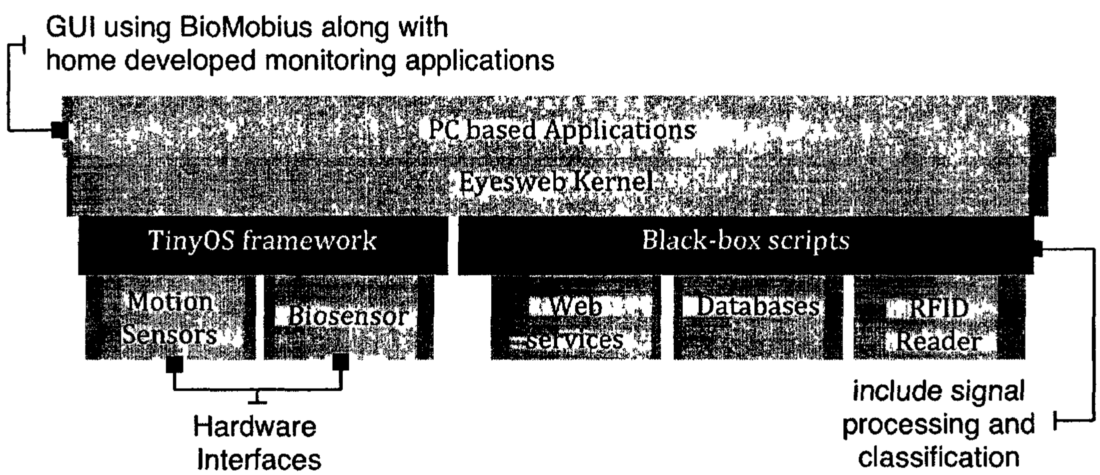

Figure 4: The software architecture.

The bottom layer on the right side of the framework architecture in Figure 4 consists of the web services interfaces for calling web-enabled Application Programming Interfaces (API) such as Google weather web service. This layer contains the database interfaces for querying and manipulating the database. The last element is an interface that communicates with the RFID reader used in the framework. In order for the framework to communicate with the three types of interfaces (web services, database, RFID reader), we need to introduce the layer above them called Black-box Scripts.

We named the layer "Black-box scripts" to show the similarity between the implemented components in the framework and the software concept of Black-box testing. Both parts apply the idea of covering the knowledge of the internal structure of a software object by only pushing an input and receiving an output. Likewise, black-box scripts have been implemented to receive a set of inputs from the upper layers, then manipulating them to return back the processed results. In this way, Hamon would be able to accommodate new services and deal with new interfaces each time a new interface was introduced within the framework. For example, a black-box script is created to receive a city name and call on its behalf Google weather network to retrieve the weather condition. If other weather network needed to be used instead of Google, then we need only to introduce a new black-box script to call the new API while keeping the same type of inputs and respecting the same outputs format. Basically, the main driver for distributing the functional scripts into black boxes is the concept of Block and Catalog of EyesWeb.

EyesWeb [43] is a research project developed in the InfoMus Lab. Hamon is driven by EyesWeb which enables us to build the bases of many software component in order to facilitate the presentation and the flow of data between them. EyesWeb is an open platform for developing a real-time multimodal applications. It consists of multiple elements called patches, where a patch is made of blocks which represent a functional element in the framework. Therefore, our framework can adopt new needs and services since the EyesWeb gives us the ability to build a variety of blocks and adopt them in the framework. Such flexibilities allow the framework to change over time according to the required monitoring or the need to add new biosensors.

EyesWeb patches can be used by BioMOBIUS platform, an open research platform developed by TRIL Centre researchers and developers [44]. BioMobius has been used in a variety of applications, which include physiological monitoring, physical rehabilitation, activities of daily living, fall detection, and cognitive function. The BioMobius enables the users to extend their application functionality by creating EyesWeb patches from either existing components come by default with the application library or by developing new ones and fixing the lower level of functionality. After developing new components, patches can be registered within a median platform layer called EyesWeb kernel which make the new blocks ready to be dragged and dropped in an EyesWeb patch. The EyesWeb Kernel is the next layer in our framework software architecture.

Eyesweb Kernel is a part of the Eyesweb project as mentioned earlier. The Kernel allows the lower layers to use its resources to capture and manipulate the sensory data at the runtime. The Kernel allows different blocks from the lower level to communicate using a predefined EyesWeb data type. However, the Kernel allows the adoption of the new developed data type to server new extensions. The details of the patches and block communication are discussed in Chapter 5.

The top level is where the monitoring application resides and all logics builder come in. Different applications run and control all the lower levels and present data collected along with the extracted logics to the user. We used BioMOBIUS platform for building a number of user graphical interfaces and in-home developed platforms to present the sensory data and apply the monitoring logic.

3.2.3 Use Case Diagram

Figure 5 shows the main use case diagram for monitoring the patient using the proposed framework. Four primary actors have been included in the use case shown on the left side of the diagram. The use cases in the middle represent the goals of monitoring the patient in benefit for each of the primary actors presented. The use cases interact with human actors immediately and with other supportive actors which point out some sub-systems in the framework. However, there are other secondary goals such as viewing sensory data offline, logging in the monitoring application, registering a sensor, establishing a connection to the BSN, and other technical and secondary goals.

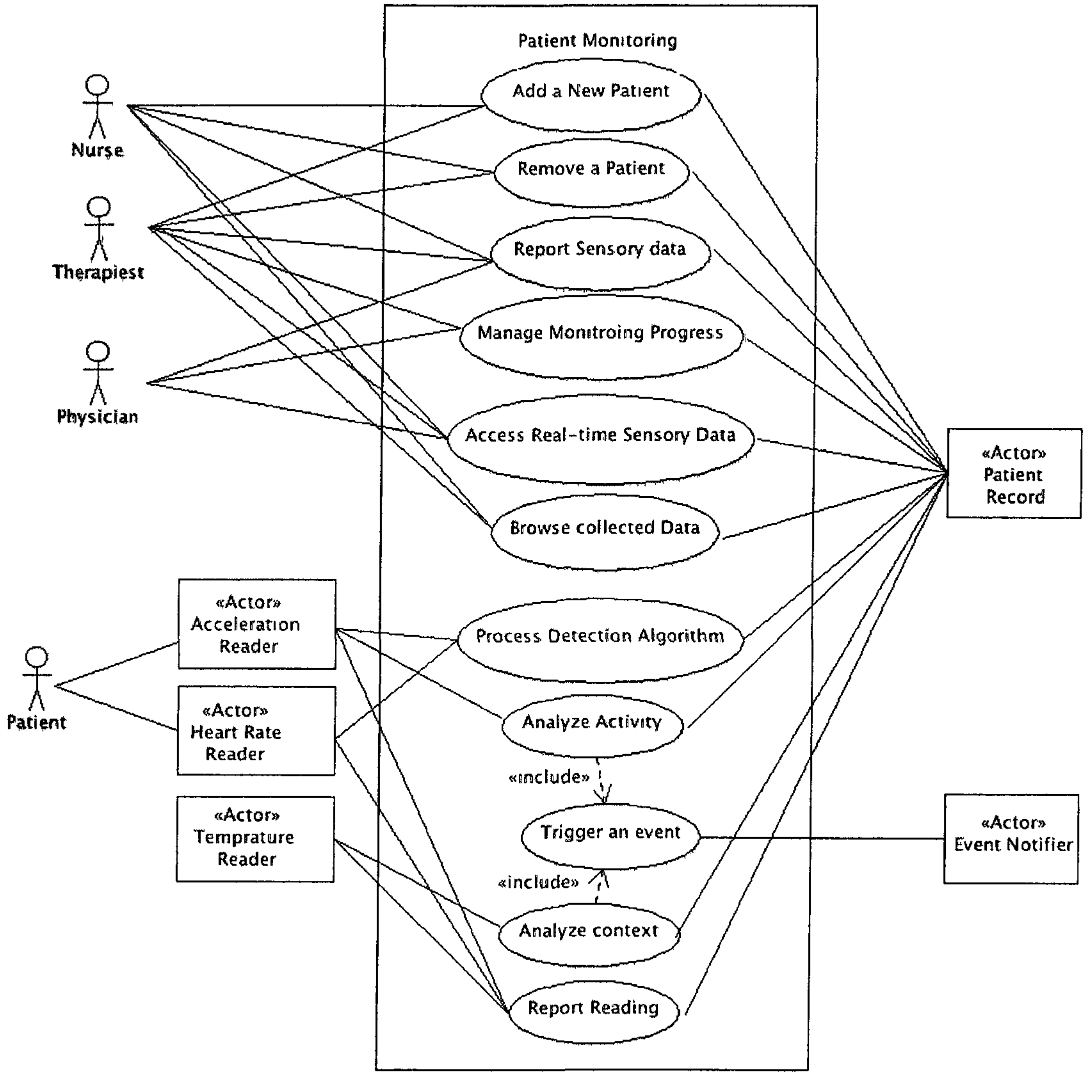

Figure 5: Use case diagram of patient monitoring in Hamon framework

3.2.4 Database Structure

The database is strucutred in a way to organize the recorded sensory data coming from all types of sensors connected to the framework. In order to reach this goal, we need to divide the database into different segments. The first segment takes care of saving all the sensory data related to a subject. The design of this segment is represented as an Entity Relationship Diagram ERD and appears in Figure 6. The second segment deals with the users' privileges and the data that manages the framework configurations. They are secondary tables to help in running the framework.

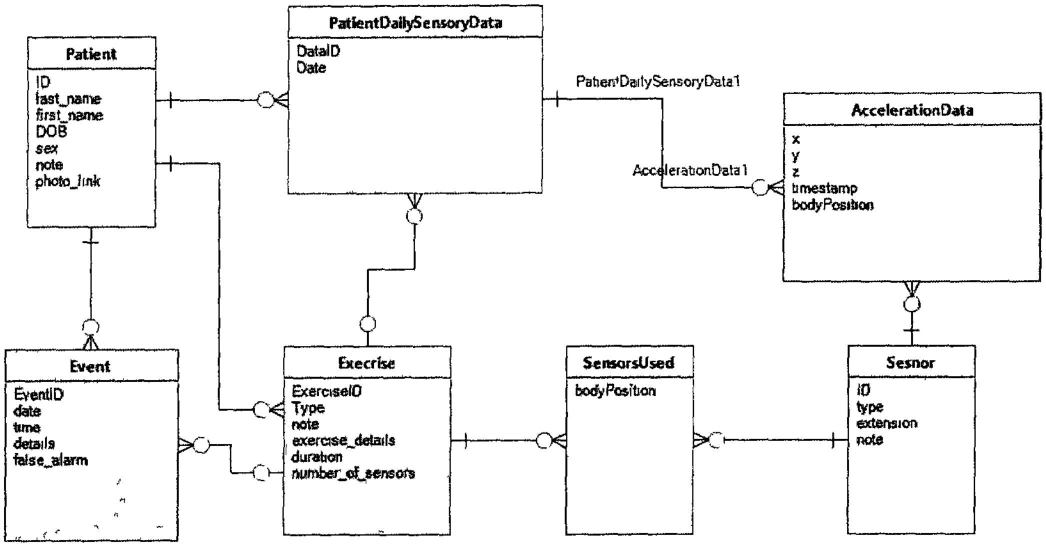

Figure 6: The Entity Relationship diagram of the proposed framework

3.2.5 Data Collection

The types of data collected in the framework can be divided into two main categories: the sensory data coming from the BSN, and the context-aware collected data. The sensory data consists of motion accelerations and physiological data such as heart rate, and the context-aware data represents the environment conditions such as the weather temperature and smoke level. In Hamon, a number of software blocks have been included to capture the sensory data from the SHIMMER sensors and transfer the data between software blocks to apply the logical part such as activity detection. The software block residing on the monitoring server has a buffer to hold the sensory data before flushing them to the database. It supports up to 500 Hz for the sample rate.

The BSN plays the main part of collecting the sensory data before sending the data back to the monitoring application. This collection resides on the user's server side. It consists of wearable sensors where each one detects a single physical phenomenon such as motion accelerations and heart rate. The user server side acts as a gateway between the BSN and the medical server where the server at the user site is the main processor of the sensory data collected. Hence, the user server side collects, stores the sensory data locally and operates the logical algorithms to detect any abnormality like falling down. It then pushes all the data to the medical server. Therefore, once an event is detected, the user server side sends the captured data together with contextual information and event description to the medical server. In addition, an event notification is triggered to notify the members of interest from the health care provider by either email or software alert appear on his/her software interface.

The sensor nodes collect the data from the embedded sensors and send it back to the serial forwarder which represent the gateway to the monitoring application (the upper layer in the software architecture). In other words, the sensors in Hamon capture the sensory data and send it to server gateway without any detection algorithm or data analysis applied on the sensor side.

To ensure the continuous of streaming the data from the BSN to the user server side, the SHIMMER sensor has internal memory which supports up to 2 Gigabytes of data preventing any loss of streaming while the user is roaming. This enables the raw sensory data to be buffered until it is flushed back to the serial forwarder in case the connection is lost or the sensor runs out of battery power.

3.2.6 Network Setup

The communication between the BSN and the user server side is carried out using a wireless Bluetooth connection. Bluetooth is a low-power consumer and low-cost option, but has short transmission range. In the implemented prototype, the SHIMMER sensor used supports a Bluetooth range of more than ten meters and supports 79 channels. Therefore, the built-in Bluetooth radio is suitable for streaming high-frequency rates.

3.3. User Roles

The framework divides the users into groups depending on their roles played in the implemented prototype scenario. As it has been represented in Figure 7, a health care professional has dedicated window interfaces for configuring the type of monitoring: either a time-limited period or a long-term period. In addition, the duration of monitoring along with the number of sensory to track the subject can be adjusted. Physicians and nurses along with other health care providers have access to track all the abnormal events that happened to the subject with details describing the alerts. Particularly the health care provider is responsible for occupying all the logic to monitor the subject and interpret the sensory data coming from the BSN.

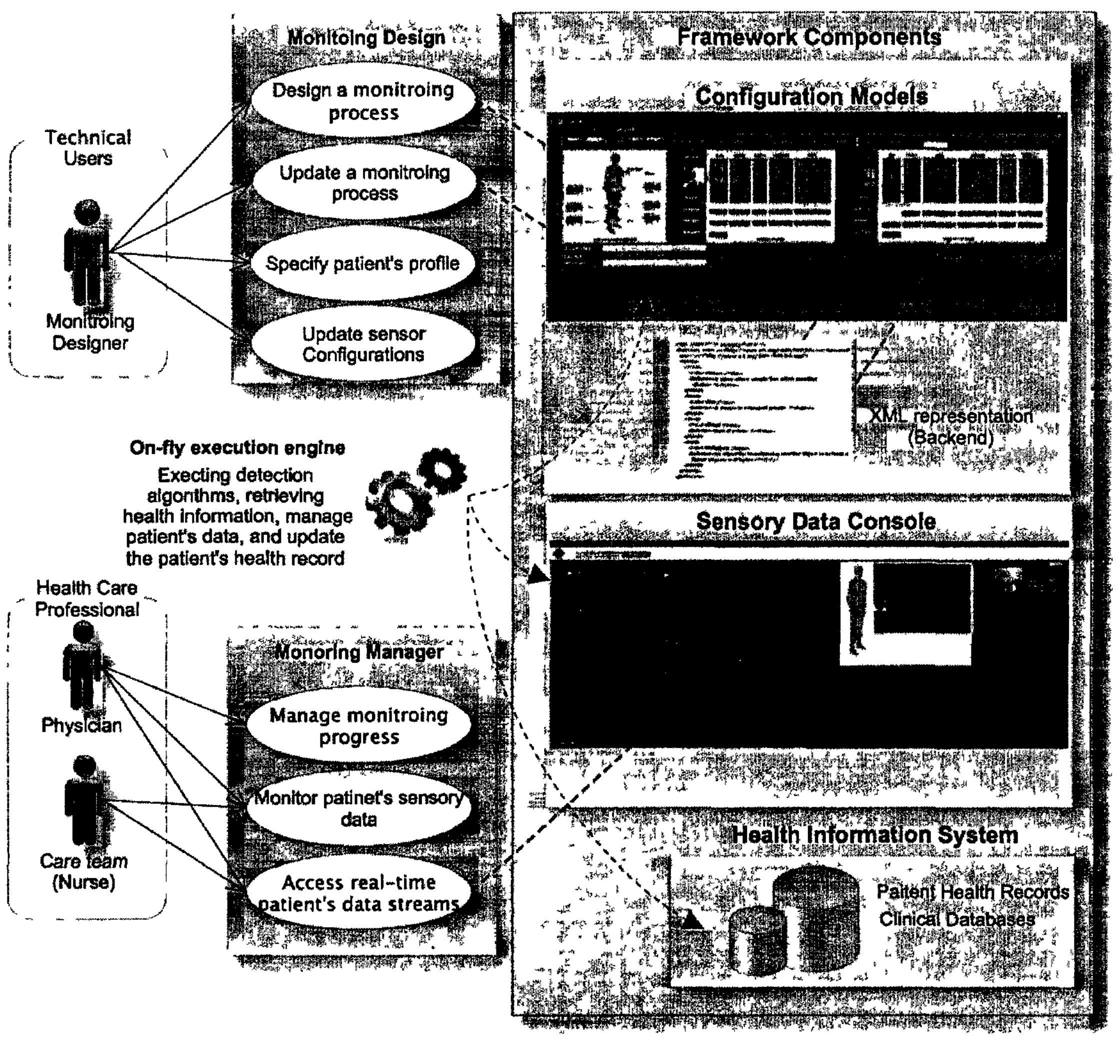

Figure 7: The user roles played in Hamon framework.

On the other hand, a technical user would have the responsibility to take care of the sensor configurations and ensure the connectivity between the BSN, the user server and the medical server. He/she also has the access to the collected sensory data that appears on the monitoring window interface in real-time. Both groups, the health care provider, and the technical administrator have access to an interface presenting an offline tool. The tool analyzes the sensory data including the representation and manipulation of features, such as mean and standard deviation of the coming signal. These roles can be distributed among users or eliminated to accommodate the user's needs.

3.4. Privacy

The user privacy and especially patient data is one of the important issues in designing any health monitoring system. The transfer of data from the user server side to the medical side should be carried in encrypted formats. Likewise, many sensor microcontrollers provide low power data encryption to secure the data transfer between the BSN and the user side. However, privacy and data security is beyond the scope of this thesis.

Chapter 4. Activity Recognition

Activity recognition is another main part of this thesis and an essential aspect for context-aware framework. Physical activity recognition during activity of daily life (ADL) is an important aspect to evaluate functional ability and risk prevention to improve the quality of life. The main challenge in any activity recognition framework is the design of the detection algorithm which can achieve an acceptable recognition rate. In contrast, recognition rate is limited with the available computation resources such as processing time and memory space.

Motion acceleration is widely used to classify the type of executed activity by analyzing the change of accelerations in respect to the time. Therefore, the accelerations are used to assess postures and motions by calculating a vector of features such as mean and standard deviation of the acceleration signal. Attaching an accelerometer to different body segments help in identifying the type of activity and distinguishing one activity from another by analyzing the sensitivity direction [45].

In our framework, we are interested in detecting five types of activities: falling, lying down, sitting, standing, and walking. In order to detect these four types of activities we used 3-axis accelerometers attached to different segment of the patient's body. After reviewing some related work [46], [47], [48], [45], [18], [49], [6] to select the best body segments to reach the highest recognition rate possible, we decided to attach the sensors to four parts of the body: head, lower back, forearm, and on thigh.

4.1. Method

4.1.1 Placement of Sensors

As previously stated, four accelerometers were attached to four body segments: head, lower back, forearm, and thigh. The placement of the sensors and direction of the three axes are maintained the same for all the subjects participated in the experiments. In this case, we tried to have the same placements and directions to obtain acceptable accuracy when we tried classifying the executed activity. Figure 8 shows the four accelerometers placement on one of the volunteers in the lab during the experiments.

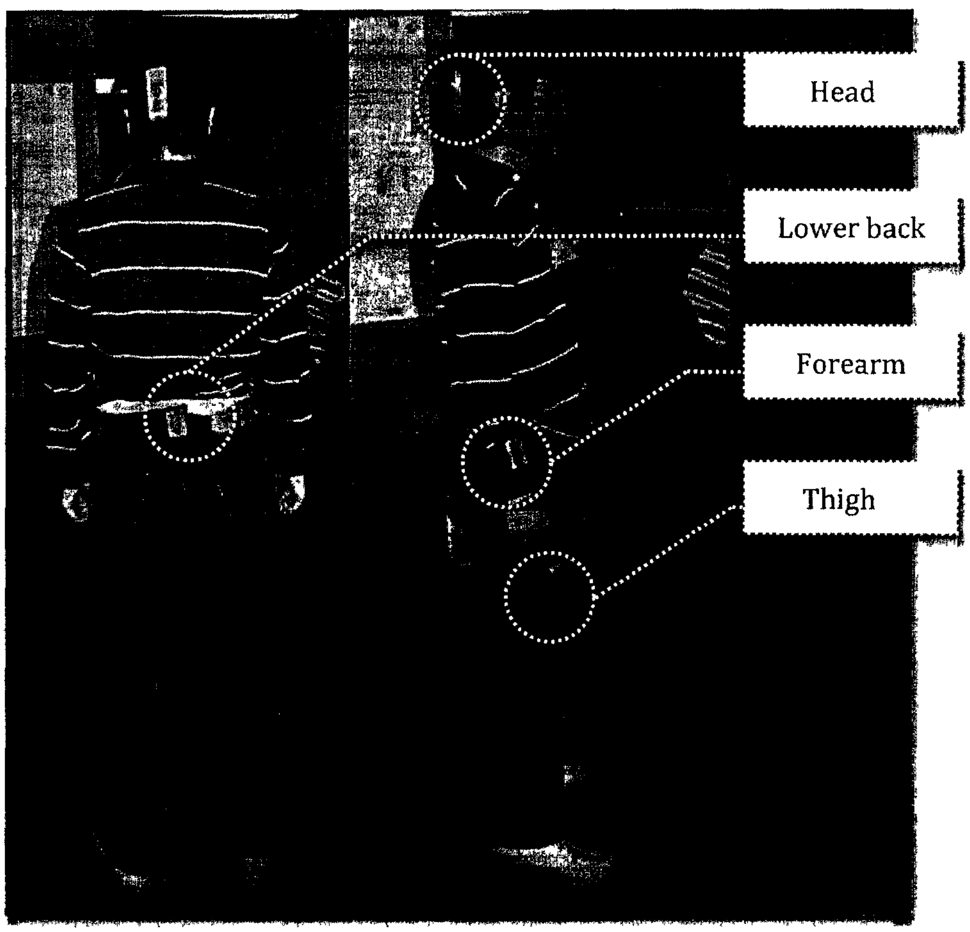

Figure 8: A volunteer wearing four accelerometer sensors in one of the experiments in the lab.

Figure 9 gives a close look at the direction of each axis against gravity for each of the four body segments.

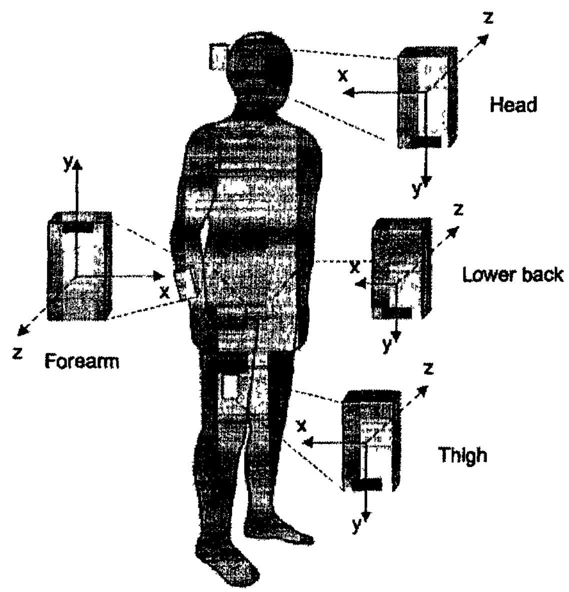

Figure 9: The three axes directions of the four used accelerometers placed on four body segments.

4.1.2 Analysis of accelerometer signals

The acceleration signals received from the BSN are segmented into window frames where each window holds five seconds of data for the three axes. Signal features are then extracted from each window to characterize the signal being received. Thereafter, the resulting computation is fed to the classifier in order to detect the activity.

The loop is repeated again by considering overlapping between the previous and next window to cover the probability that an activity is divided between one or more windows. As we have discussed earlier, for motion detection we used the SHIMMER accelerometer. The sample rate configured for capturing the data is 50 Hz. The acceleration values recorded contain positive or negative accelerations as well as null in case there was no change of acceleration recorded at the same unit of time. Table 2 lists the main configuration and the type of features extracted from the accelerometer.

Table 2: Description of the accelerometer configurations and the types of feature extracted from the acceleration signals used for motion detection.

Sensor	Accelerometer
Segmentation	Sample frequency 50 Hz
Window Size	5 seconds
Window Overlap	70 %
Features	Mean, standard deviation, signal power
Classification	Standard K-NN, K-NN with Bayesian Network

The features computation of the motion signals start from calculating the mean and standard deviation for each axis. Since each window represents five seconds of the data, and the sample rate used is equal to 50 Hz, then each window consists of 50 x 5 = 250 rows. The mean is calculated for X, Y, and the Z signal of the 250 readings according to Equation 1.

$$\bar{x} = \frac{1}{n} \sum_{k=1}^{n} x_k$$

Equation 1: Calculating the mean value of the acceleration readings for a single axis.

Likewise, the standard deviation is calculated for each axis according to Equation 2.

$$\sigma = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (x_k - \bar{x})^2}$$

Equation 2: Calculating the standard deviation value of the acceleration readings for a single axis.

4.1.3 Classification

We need an algorithm to be able to classify each activity by recognizing the signal patterns and matching a vector of features with pre-learned ones. For the activities we are trying to detect, Figure 10 through Figure 20 show what the acceleration signals look like after a subject performed the five activities. From these figures, the shape of each signal can clearly draw a special curve representing each motion movement. In Figure 10, a subject was asked to fall down in any direction and in any form and remain still. The graph represents the accelerations of x, y, and z axis collected from a sensor attached to the head The blue curve represents the x-axis, the red curve represents the y-axis, and the green curve represents the z-axis The horizontal axis of the graph represents the time unit where each 50 unit represents one second The vertical axis of the graph represents the acceleration reading in g-unit. It clearly shows a large peak when the fall happened for all the axes and it is clearer in the z-axis than the others due to the placement direction showed earlier in Figure 9

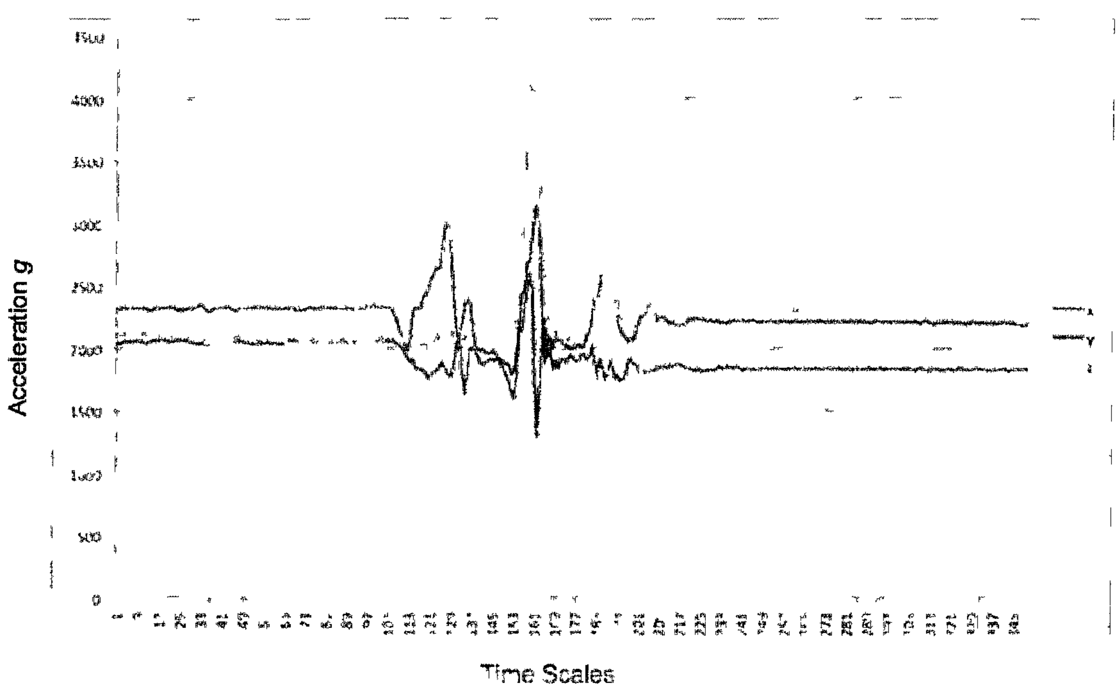

Figure 10: The 3-axis readings for a fall down activity recorded using an accelerometer attached to a subject's head.

Figure 11 shows the reading of the same activity performed, but was recorded using a sensor attached to the lower back. Similarly, as it appears in Figure 10, the blue curve represents the x-axis, the red curve represents the y-axis, and the green curve represents the z-axis. The curves clearly show large peaks compared from the sensor attached to the head when the fall happened for all the axes. The falling curve is clearer in the z-axis and y-axis compared to the same reading from head sensor. This may tell us when presenting all the experimental results how accurate the lower back sensor is to detect such activity compared to sensors attached to the other three body segments.

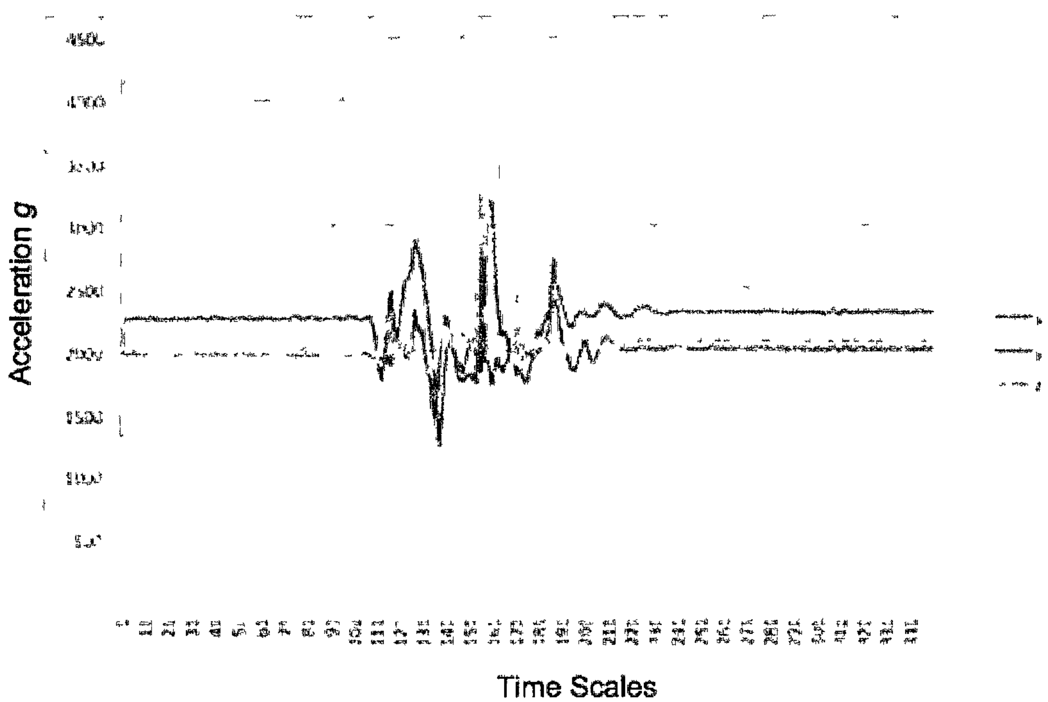

Figure 11: The 3-axis readings for a fall down activity recorded using an accelerometer attached to a subject's lower back.

Figure 12 shows the 3-axis readings of the fall activity recorded using a sensor attached to the forearm. From the blue curve, which represents the x-axis, we can predict clearly the hand movement while the subject was falling towards the ground. The change of accelerations is significant on the y-axis since the arm was moving higher to lower. This is due to the placement of the three directions appears in Figure 9. The green curve, representing the z-axis, shows less changes in its accelerations compared to the previous two. In contrast to the previous reading from the head and lower back sensors, the readings from the forearm shows a larger number of peaks in the signal and thus make another distinct pattern to classify.

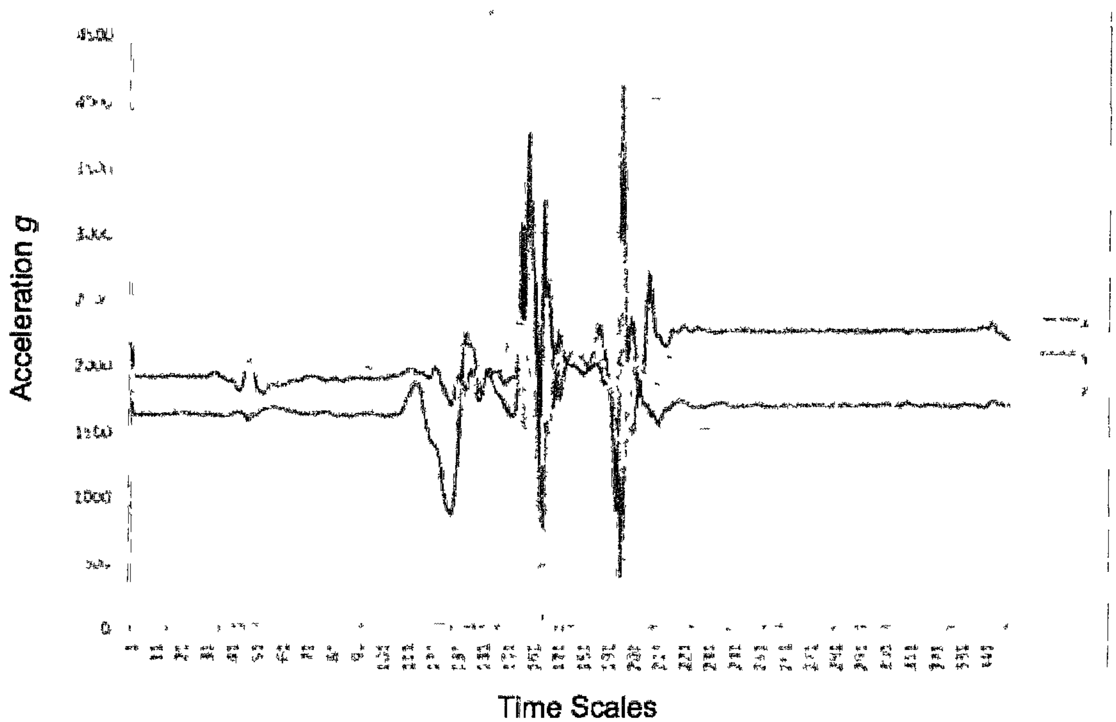

Figure 12: The 3-axis readings for a fall down activity recorded using an accelerometer attached to a subject's forearm.

Figure 13 presents the 3-axis readings of the fall activity recorded using a sensor attached to the subject's thigh. The x-axis represented in blue, shows high change of accelerations of the leg movement while falling was taking place. The change of acceleration is also significant on the z-axis and forms an opposite curve of the x-axis. The y-axis follows the curves of x with lower acceleration values, thus the x and z readings clearly drive the falling readings from thigh position, forming another distinct pattern to classify.

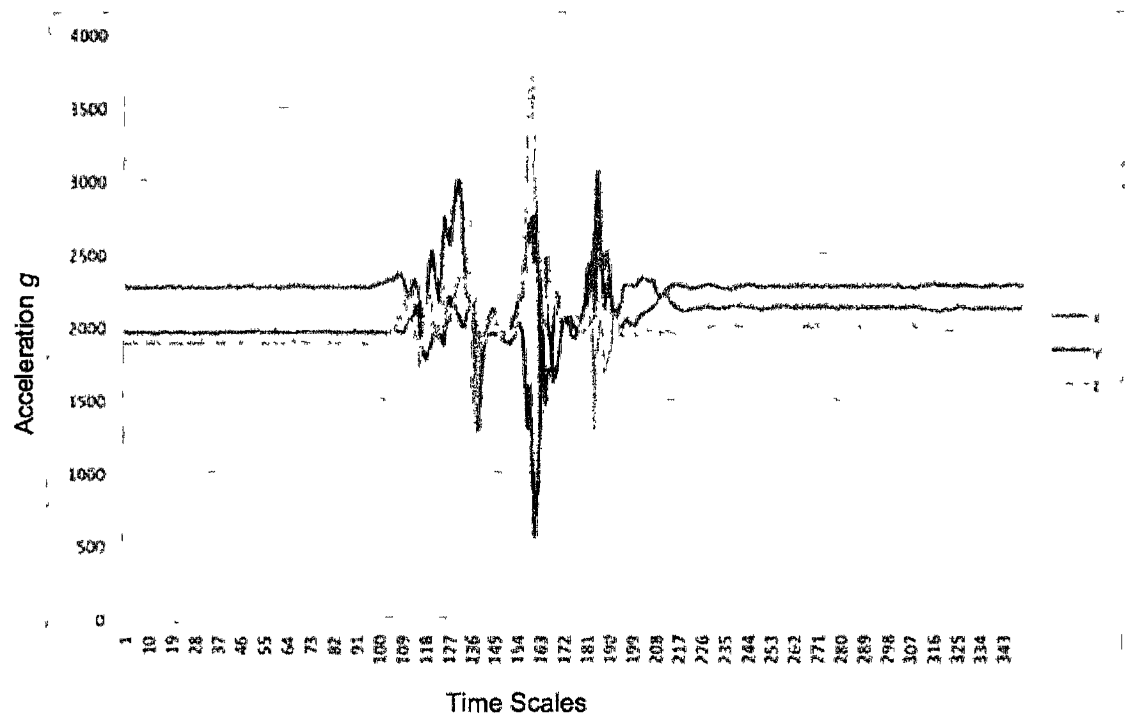

Figure 13: The 3-axis readings for a fall down activity recorded using an accelerometer attached to a subject's thigh.

Figure 14 presents the acceleration signals recorded using the four sensors together but only for one axis of the falling activity. The black curve shows the x-axis readings recorded using a sensor attached to the subject's head. The dark brown curve represents the x-axis readings recorded using a sensor attached to the subject's lower back. The light blue curve shows the x-axis readings recorded using a sensor attached to the subject's forearm. Finally, the orange curve shows the x-axis readings recorded using a sensor attached to the subject's thigh. The readings show the similarity at the start and the end of each peak of the same activity recorded at the same unit of time. Thus we can show clearly what the shape of the signal looks like for the fallıng activity recorded using four sensors attached to different parts of the subject body.

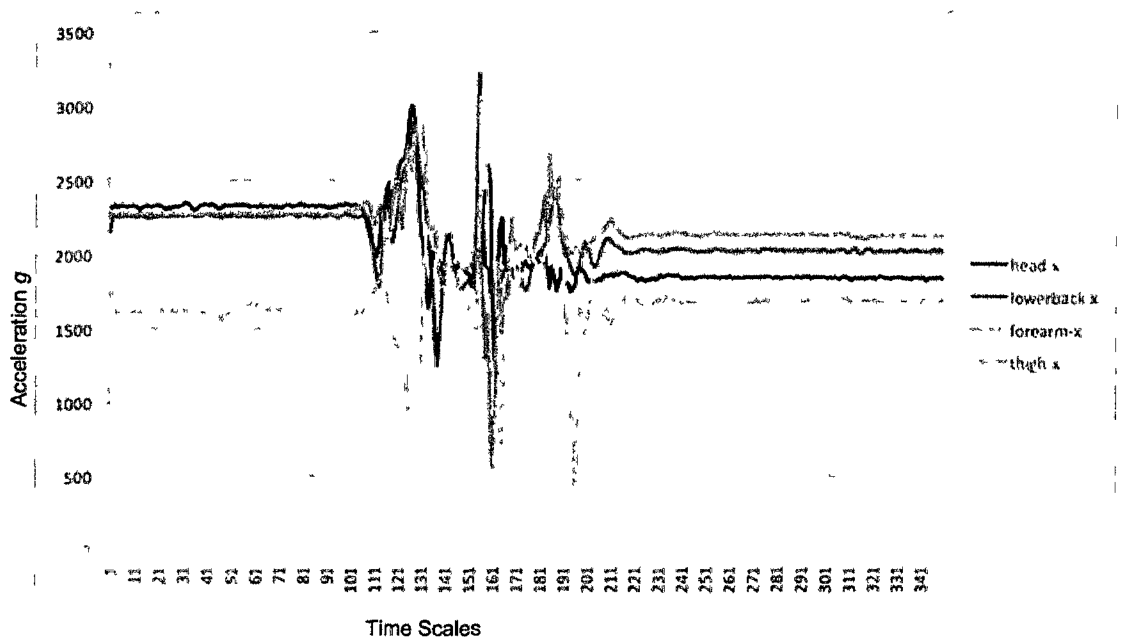

Figure 14: The x-axis readings for a fall down activity recorded using four accelerometers attached to a subject's head, lower back, forearm, and thigh.

The last few pages, we discussed the shape of the recorded signal for only the falling activity using 3-axis accelerometers. Falling down is just one of the five activities we tried to detect using the four attached sensors. Hence, we need to study the shape of the other four activities before demonstrating the classification algorithm. Figure 15 shows the 3-axis readings of the accelerations recorded using a sensor attached to the subject's head. The activities represented in the graph from the left to the right place the readings of falling, lying down, sitting, standing, and walking. The horizontal bar of the graph represents the time of each acceleration reading. However, the activities took place in different time periods and there was no continuous of performing the five activities together, meaning there is no transition between activities. The five activities are placed next to each other in sequence to simplify the visualization of how the shape of each activity looks in a time line. The vertical bar of the graph represents the acceleration values in g-unit.

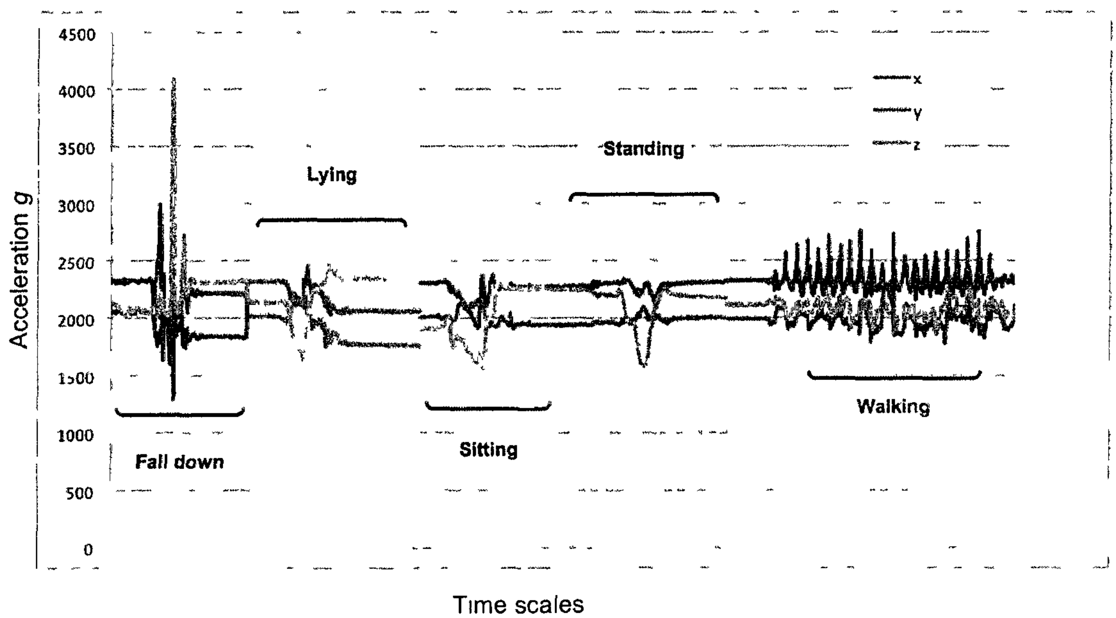

Figure 15: 3-axis acceleration readings from an accelerometer attached to the head while the subject was asked to perform five types of activities.

Similarly, Figure 16 shows the 3-axis readings of the accelerations recorded using an accelerometer attached to the subject's lower back. The activities represented in the graph from the left to the right place the readings of falling, lying down, sitting, standing, and walking. We noticed during the recording of the accelerations that the data collected from the lower back sensor had the fewest number of rows compared to the other sensors. As a result, the data illustrates to us that the change of acceleration happens less in that body segment compared to the others. Chapter 6 discusses whether such findings affect the accuracy of the activity detection or gives a positive impact to overall accuracy rate.

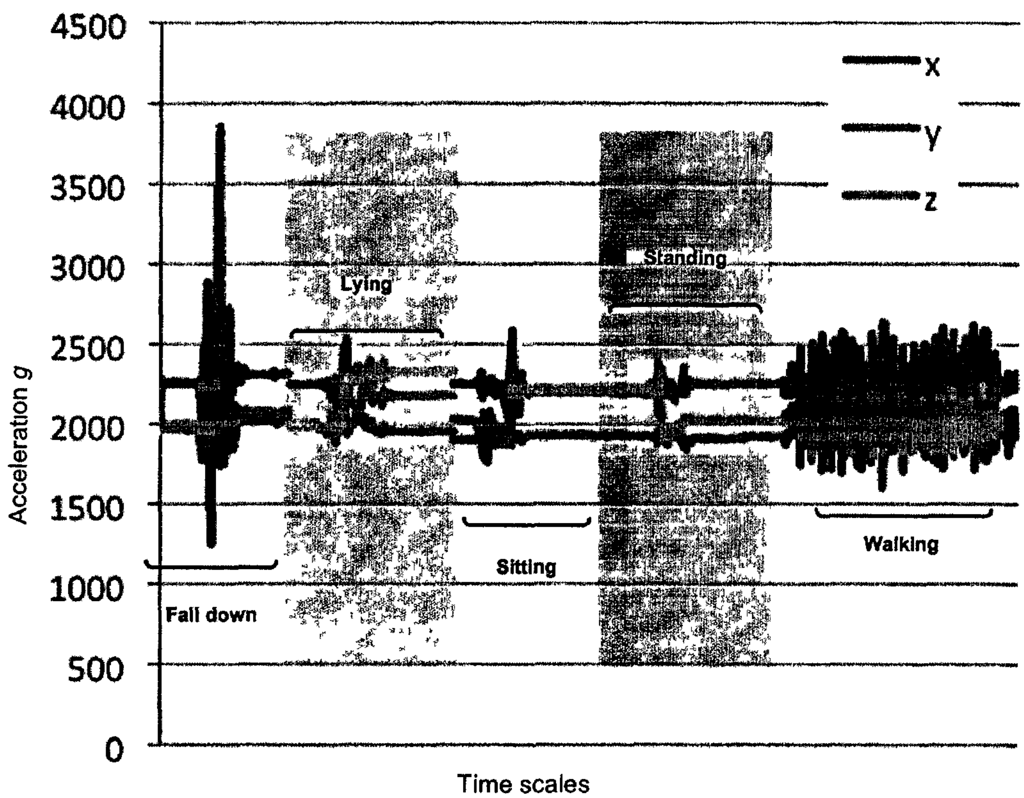

Figure 16: 3-axis acceleration readings from an accelerometer attached to the lower back while the subject was asked to perform five types of activities.

Figure 17 shows the acceleration signals recorded for a subject performing five activities: falling, lying down, sitting, standing, and walking with a sensor attached to the forearm. We noticed from the graph that the accelerations were higher for the lying activity compared to the other three sensors. Thus, the forearm sensor may significantly detect such activity with a higher recognition rate compared to other body sensors. However, hand motion gives lower acceleration values while the subject was executing the walking activity and hence may not be accurate in designing a distinct pattern to detect such activity. For the falling activity, we cannot prove that the forearm sensor can show a unique pattern helping us in classifying activities. This may be due to the fact that subjects fall in different direction and on different sides of their bodies. Therefore, it may be clearer in one subject than in others and may not give us a higher recognition rate compared with other body sensors. Sensors attached to the head and/or lower back are less affected by the style of falling. The resulted accuracy for each sensor is shown in Chapter 6.

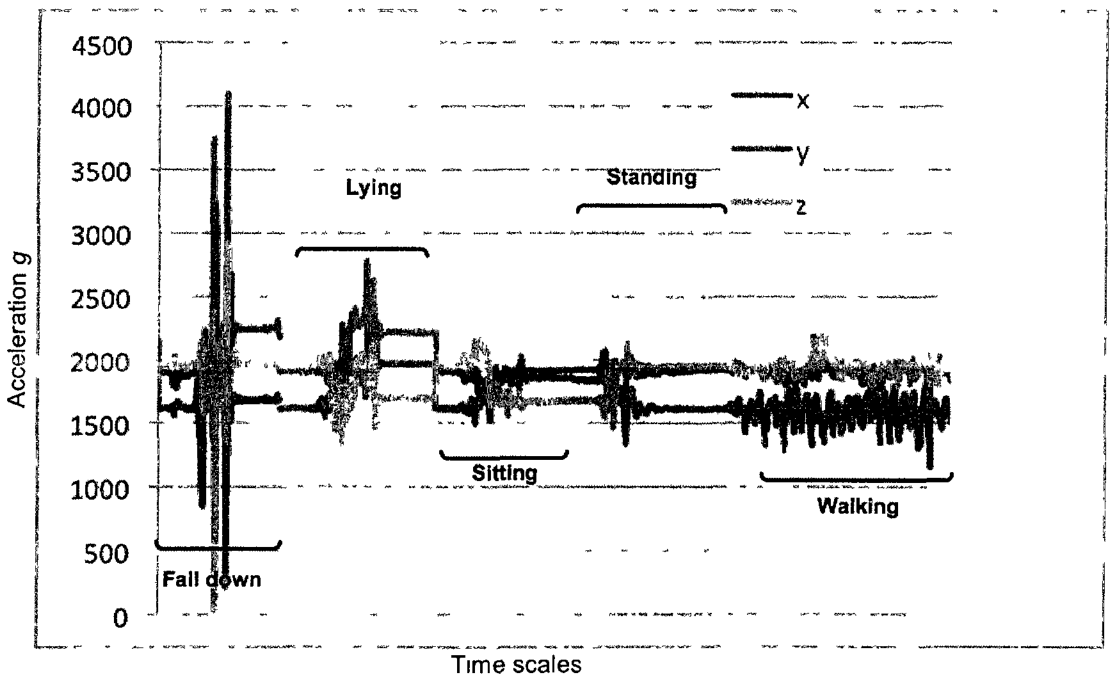

Figure 17: 3-axis acceleration readings from an accelerometer attached to the forearm while the subject was asked to perform five types of activities.

Finally, the shape of the acceleration signals collected for the five recorded activities using an accelerometer sensor attached to the thigh is represented in Figure 18. We notice from the graph that the accelerations were also higher for the lying and walking activities compared to the other two sensors. By comparing the accelerations of the same activities collected from the forearm and the thigh, we can distinguish clearly that the sensor attached to the thigh gives higher accelerations than the forearm for the walking activity. Our first analysis was lately proofed by analyzing the data collected from all the subjected participated in the experiments. Thus, it significantly detects the walking in specific and gives a good accuracy in detecting the other activities but not with the same recognition rate as with recognizing the walking signals. The details of such accuracy are described in Chapter 6. In addition, leg movement from standing position to sitting and the reverse can be clearly seen by tracking the z-axis in both cases.

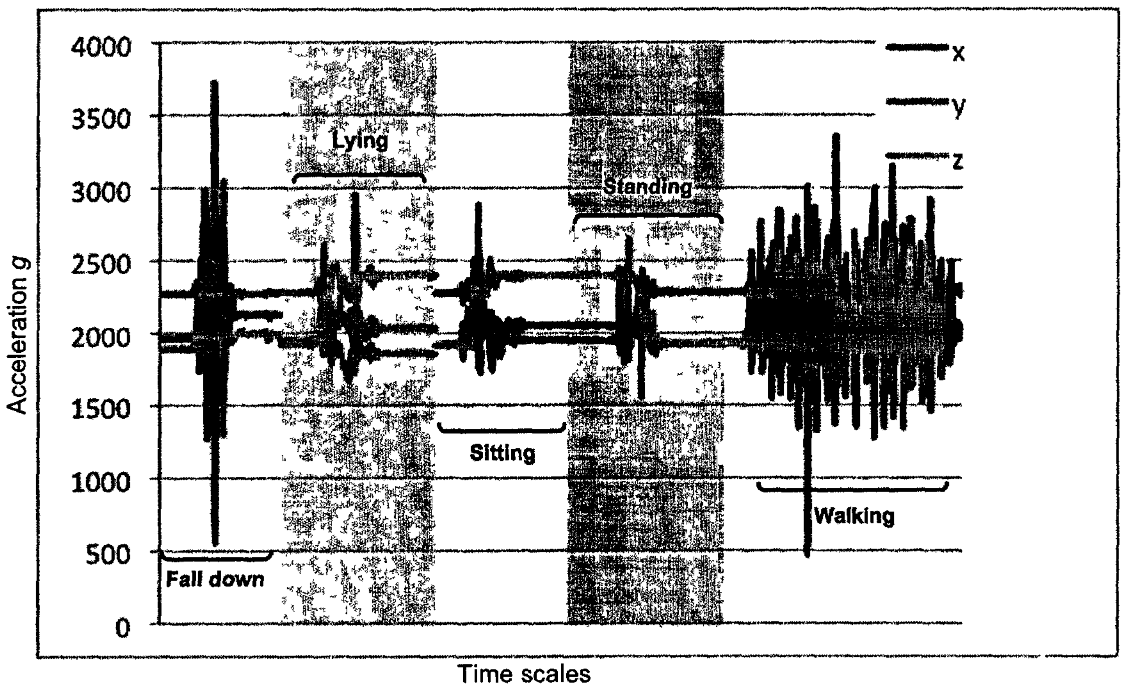

Figure 18: 3-axis acceleration readings from an accelerometer attached to the thigh while the subject was asked to perform five types of activities.

K-Nearest Neighbour (K-NN)

The K-Nearest Neighbour (K-NN) is one of the widely used algorithm for classifying different statistical patterns [50]. K-NN classification algorithms are essentially used to associate new data represented in a vector as a member to one of the distinct classes. For example, Q represents a statistical set of data where $Q = \{v_i\}$, $i=1,...,n$. Then K-NN tries to find the nearest point which represent v_{i+1} by calculating the distance between v_{i+1} and each vector in the set Q. Figure 19 shows briefly the idea of classifying a new data vector by finding the nearest distance to a set of pre-learned set of vectors.

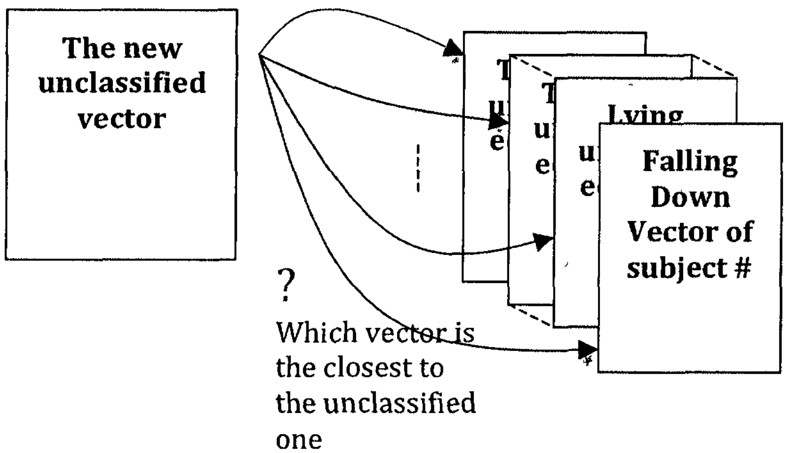

Figure 19: An example shows the classification method of K-NN.

The main issue of the K-NN method is the selection of the right value for k which gives the closest vector to the unclassified value. However, finding this value is not a trivial task. In this thesis, we repeated the resulting classification of all the subjects who participated in the experiments by running the algorithm multiple times where k =1, 3, 4 and 5. The resulting recognition rate and discussion of running the algorithm with each value of k is presented in Chapter 6.

In our activity detection problem, we used K-NN to classify the five distinct classes representing falling, lying down, sitting, standing, and walking of the 23 subjects participating in the experiments. The idea was to compare each vector with distinct classes of all the subject available in the learning database. We analyzed the performance of the K-NN using two approaches. First, we applied K-NN algorithm on the data collected from the four sensors all together, which is referred to as "all sensors classification" in the remainder of this thesis. Second, we applied the K-NN algorithm on the data collected from each of the four sensors. Thus in this case, the classification is based on the data collected from a single sensor at a time, which is referred to as "sensor-based classification" in the remainder of this thesis.

A new, unclassified vector in the all sensors classification methods consists of 24 attributes representing the signal features mentioned in 4.1.2. The 24 attributes consist of three mean and three standard deviation values for the four sensors. We used Equation 3 to find the distance between each two vectors in order to find the minimum distance between the new and the pre-learned ones:

$$\text{distance} = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2+(z_1-z_2)^2}$$

Equation 3: An Euclidean function to calculate the distance between two vectors.

In Equation 3, x_1 places the mean of x-axis of the unclassified class, while x_2 places mean of x-axis of the vector *i* and so on. Similarly, y_1 and y_2 represents the mean of y-axis for the unclassified class and vector *i* respectively. As a result, we compute the square root of the summation of all the differences for the 24 attributes and that shows the final distance.

A new unclassified vector in the sensor-based classification method consists of six instead of 24 attributes representing the signal features mentioned applied on the data collected from only a single sensor at a time. The same method is still applied for calculating the distance between the new unclassified class and all vectors in the learning database. At the end, the K-NN classifier obtains a single predicted activity for each of the four available sensors.

K-Nearest Neighbour (K-NN) with Bayesian Network

Bayesian Network is widely used to present a probability distribution for classifying a set of training data. It helps in inspecting the training data to find the best possible candidate vector by applying a heuristic search and obtaining a scoring metrics [56]. To show in an example the idea of using Bayesian Network over our activity classification problem, we denote ***U*** as a set of random variables representing the five types of activities.

Let G be a directed acyclic graph where vertices represent the falling, lying, sitting, standing, and walking activities. The edges in the acyclic graph G serve as direct independencies from the vertices to the leave nodes representing the sensors used in the detection. Each sensor in the graph is independent from each other where their parents are the activity being detected. The resulting structure of the Bayesian Network in shown in Figure 20.

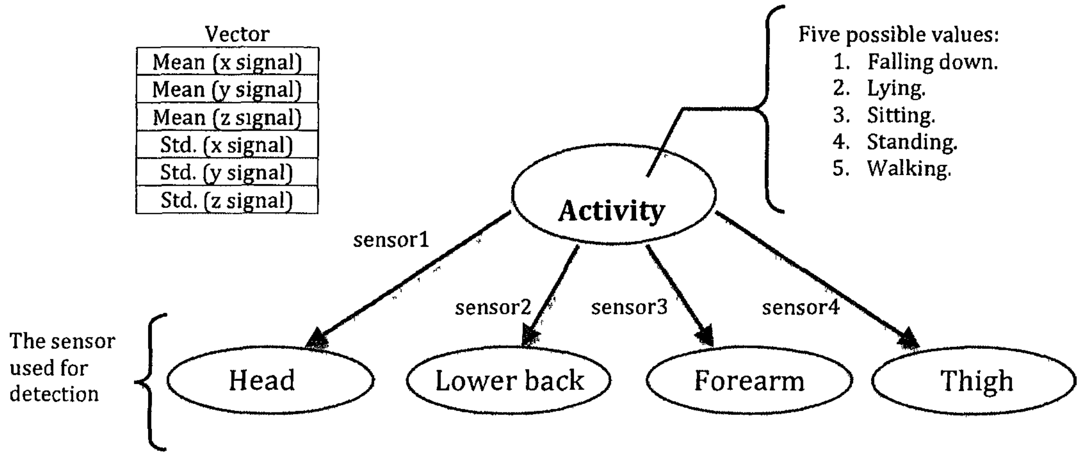

Figure 20: A structure graph of Bayesian Network used for finding the conditional probability for each used sensor in detecting one of the five activities.

The graph presented in Figure 20 shows the random variables identified as nodes and the arcs represent the conditional independence of detecting each activity using one of the available sensors: head, lower back, forearm, or thigh. More precisely, this graphical model structures the representation of joint probability distribution of our classification problem. The graph structure is then converted to a Conditional Probability Distribution (CPT) for each node shown in the graph since all the variables are discrete.

All the leaves nodes in our Bayesian Network graph have five possible values, which are symbolized by: falling (1), lying (2), sitting (3), standing (4), and walking (5). The four leaves are denoted by H: head sensor, L: lower back, F: forearm, and T: thigh. The probability of detecting an activity in the U set of variable in our discussed example is equal to 0.2, since each activity is assumed to be likely and has an equal probability to be detected = 1/5. The arcs in the graph join the probability of each sensor to detect an activity U_i calculated as a conditional relationship probability. For instance, the probability that the sensor attached to the subject's head detecting a fall given that the real activity was falling down is calculated according to the following equation:

$$Pr(H=1 | R=1) = Pr(H=1,R=1) / Pr(R=1)$$

Where H: head node, and R: the real activity.

At this point after structuring the Bayesian network, we used it to solve the probability inference based on Bayes's rule [51]. As a result, we can answer a variety of logical questions such as, "if a head sensor detects a falling activity, what is the percentage that the real activity is also falling? In other ways, if the activity is falling, which sensor is more likely to detect such activity?"

Kevin P. Murphy details on his personal website [51] the use of Bayesian networks in different statistical problems. He provides a Bayes Net Toolbox for Matlab, which has been configured and adapted to solve our classification problem. In Appendix A of this thesis, a code example is given of the Matlab class that has been used to obtain the conditional probability table for each sensor we used for detection.

Chapter 5. Implementation

This chapter discusses in details the implementation of each component mentioned in Chapter 3. Chapter 5 starts with building the Wireless Body Sensor Network (WBSN) including the software framework used to collect the sensory data. Next is the context-aware acquisition, types of user feedback including the alert mechanism and the presentation of the alert history. In addition, the chapter addresses the database query and data manipulations including the offline data analysis and a summarization of the user role and authentication mechanism used in the framework. The chapter concludes with a demonstration of the implemented monitoring interfaces.

5.1. Building the Wireless Body Sensor Network

In our implemented framework prototype, we used the SHIMMER [28] wearable wireless sensor. We picked SHIMMER due to its low-power consumption, lightweight, and its ability to be integrated with our framework software structure. The SHIMMER sensor holds a lithium-polymer battery, and a TI MSPP430 microprocessor with 10 KB of RAM and a flash memory of 48 KB of capacity. It supports two types of wireless communications by having Bluetooth and 802.15.4 wireless modules. The battery life of the SHIMMER is mainly depends on the type of application installed on the tiny node platform and the type of selected mode of power consumption. If the sensor is configured to sample 100 sample per second, then battery life can last between 8-10 days while the data is not being sent in real-time. Therefore, the battery consumes a lot of energy if the sensor is streaming data in real-time to the gateway port using a built-in wireless module.

The embedded software installed on the tiny framework of SHIMMER is TinyOS, which is an open source library used for Telos Mote platform. TinyOS is a platform written in necC, a programming language very similar to the C language. The TinyOS version used is 1.1.15 which has been recommended by the SHIMMER designers. Although TinyOS v2.x is already available for developers, we faced couple of issues in depending on such platform due to its lack of support in dealing with Bluetooth communication. The TinyOS architecture consists of three main components: the TinyOS application, the platform software, and the platform hardware. The TinyOS application is the main controller that initiates an operation and receives back signal events. We used a customized BioMobius 3-axis accelerometer and ECG applications to control both the hardware and the software components. The sensors are configured to sample the data at 50 Hz. We used a Linux OS to build the SHIMMER development environment and a USB port to burn the SHIMMER image.

The captured sensory data is continuously sent back to a serial forwarder using Bluetooth to be analyzed and stored in the database. For example, the accelerometer sensor captures the change of accelerations over the x, y, and z-axis along with the timestamps and send them back to the monitoring application.

5.2. Context-aware acquisition

We built different Eyesweb Catalogs based on some of the black-box scripts presented in our framework to enquire about relevant information for drawing and presenting a context-aware picture for both the patient and the healthcare provider. For instance, we built a C++ Eyesweb catalog using Microsoft Visual Studio, which can call a java listener to inquire about the weather condition. Through that listener, the catalog calls the Google API web service available on the web, which returns an XML file containing the weather information. The block applies the black-box concept of accepting one input variable in a text format indicating city name, and return back four types of information: the date (ex: "04-06-2010"), temperature (ex: 16°), condition (ex: "Cloudy"), humidity (ex: 72%), and wind (ex: "NE at 1 mph"). Figure 21 shows a screenshot of the implemented component integrated in the framework.

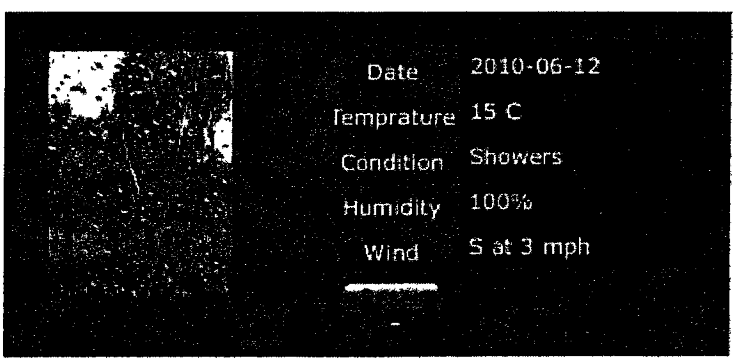

Figure 21: A screenshot of an implemented component to capture the weather condition.

5.3. Database Query and Data Manipulation

For data manipulation, we built a few of Eyesweb catalog using C++ for retrieving and querying an SQLite [52] database based on a given input string. SQLite is considered one of the most widely-developed SQL database engine as it is simple to install and configure. The C++ blocks for data query and manipulation receive a "string" text and generate a query to an SQLite database for retrieving back a full database record. The block is able to prepare the needed SQL select command according to a preconfigured database path and SQL string configurations.

The database plays an important role in different stages of the framework. Besides the usefulness of having such storage medium for offline research analysis, the database maintains different parameters showing the performance of the subject throughout the monitoring period or after taking specific treatment. The structure of the database is designed to store the sensory data in a way to maintain the following requirements:

- Showing the output of each recorded sensory data ordered by time and date.
- Providing the emergency events and error warnings.
- Storing the patient's personal and health-related information.
- Listing the patient's detected activities.
- Displaying the context related information.

5.4. Audio and Visual Feedback

The framework interacts with the user of the system by providing different audio and visual feedback to send warnings or specific instructions required by the healthcare provider. To give an example, suppose a patient in her 70s falls down in the washroom. The monitoring application would detect this risk and send a warning message through speakers attached in different rooms of the woman's home. As a result, the patient's daughter would hear this message and be able to find her mom and give her the needed help. Simultaneously, the health provider would see the same warning by either receiving an email or a software warning popping on the health monitoring panel. He/she can then check the heart rate and other physiological factors.

5.4.1 Alerts and Instructions Distribution Mechanism

We built a C++ Eyesweb catalog to apply the narrated scenario using Microsoft Visual Studio, which can receive a "string" text and generates an audio file containing the voice output of the given text. For the text-speech conversion, we used Microsoft Speech API available on the web. The input to this block is a single string value, for example, "Alert. Your heart rate reading is below 50 beats per minute." Then, the block would generate a single audio file in a .wav format.

Whenever the heart rate goes above or below certain levels the system can generate voice alarm through speakers that are setup. In addition, the other usage of the implemented voice alarm is to alert about falling down. Technical errors triggered by the framework or an important message from the nurse or the therapist needs to be relayed to the patient at the start of a monitoring session or at specific time of the day. It is also useful to give instructions to the patient before any engagement in a rehabilitation exercise at home.

Another way for alert notification is email. We built a C++ Eyesweb catalog using Microsoft Visual Studio, which can receive a "string" text and send an email notification to a recipient. The block prepares the needed SQL select command to find who should be notified from the database using a table named Alert. Then, it prepares the email header, footer, and attaches a file if needed.

5.4.2 Visualizing the history of events in a timeline

The aim of this task is to visualize all events that happened to a patient in a web-based timeline format. In our database structure we added a table called Event. This table has four attributes: event ID, date, time, details, and a Boolean variable showing if the triggered alarm is actually false. These attributes are presented in a timeline format using an open source data visualization tool to give the user the ability to skim through different events triggered over time. The implementation of this component is built based on an open-source library called SIMILE Widgets [53]. Figure 22 presents a screenshot of the implemented component showing a falling alram detected on June 6, 2010 at 1:16 pm.

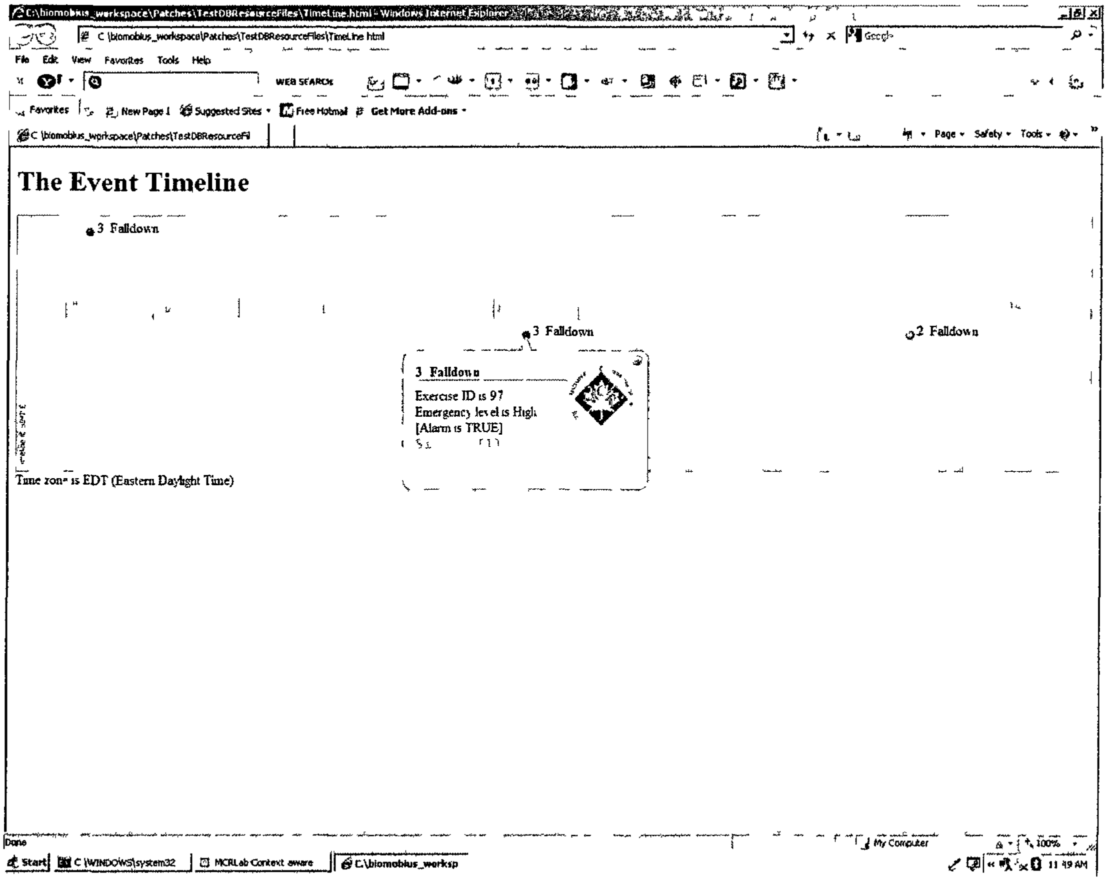

Figure 22: A screenshot showing a timeline history of the entire events triggered for a specific patient.

5.5. Offline Data Analysis Interface

The healthcare provider may need to see in detail how the recorded signal looks offline for further analysis of the collected data. For instance, if a patient is performing a specific prescribed exercise, the related recorded accelerations can be displayed offline for such a need. We developed a GUI tool that can be called from our framework using C++. This tool can query the database to retrieve the required sensory data according to different query inputs and present the data in a 2-dimensional graph.

Within this tool, the system's user has the ability to see two lists. The first lists all the Data IDs identified by the date and time of recording belonging to a specific patient. Clicking on one of the Data IDs should query the database and present the sensory required data. The second list has all the exercises that were previously performed by the patient over a set date and time.

The framework's user has the ability to obtain a list of all the sensors used along with the body segments labelled with during the monitoring such as forearm, head, etc. Furthermore, the list can present all the sensory signals identified by specific sensors engaged at the time of recording. For example, if four motion sensors are used, then the application shows four graphs representing the accelerations for each sensor, where each graph has three data curves for the x, y, and z axes.

Figure 23 shows a screenshot of the implemented tool where the vertical axis of the graph presents the time domain. The horizontal axis represents the acceleration frequencies (the accelerometer reading). Under each graph, the signal features are presented, such as the mean and standard deviation.

The tool also gives the user the ability to save the data presented in one of the two dimensional graphs into a Microsoft Excel format file. For all graphs that are related to one Data ID, the user has the ability to save the raw data collected from all the engaged sensors into one Excel file. Clicking on the "save" button gives the user the ability to select the location to save the file. If the sensory data collected is from the accelerometer, the content of the file has (*n*) number of sheets, where (*n*) represents the number of sensors used in the recording. Each sheet lists only three columns x, y, and z holding the accelerations of each axis. Saving the data in such a format is to broadens the user's ability to easily analyze the data with a more powerful tool such as Matlab.

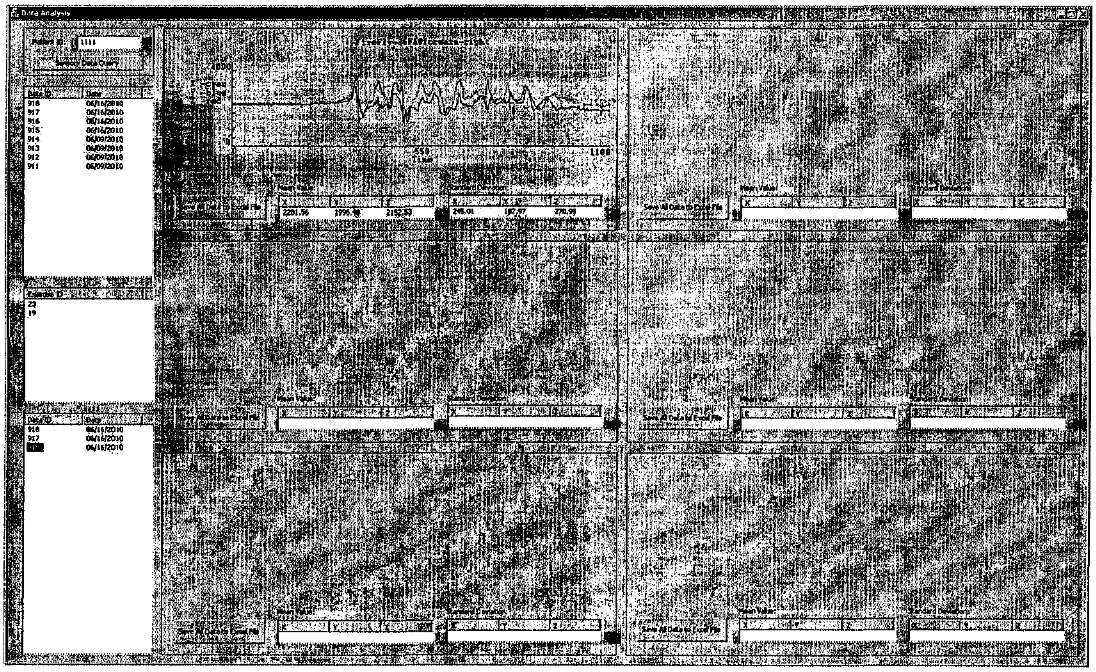

Figure 23: A screenshot of the implemented offline data analysis interface.

5.6. Authentication and Users' Role

The medical server holds the information of the system's user for authentication and access services. Two ways exist to authenticate to the system: traditional username and password or using a Radio-Frequency IDentification (RFID) tag. RFID tags can be useful in identifying users, especially elderly ones, for reasons discussed in the following patagraph. The RFID can also be useful in retrieving patient data easily.

The framework's user can use RFID tags instead of the traditional method to login in the system to reduce the challenges of using keyboard or remembering the login password. A block has been developed to initiate the connection through Bluetooth between the framework and an RFID reader. If the RFID reader has not been found, a message appears to the user using a text field or a voice alert, identifying a failed connection. If the log in was successful, the username is passed to the framework after extracting other relevant information from the database using the RFID tag ID.

Another advantage of using these tags is the ability to accommodate other important information which may help the patient's through their physical exercises For example, specific RFID tags hold the type of exercise the patient must execute at specific day or at specific time without relying on manual configuration or the memory of the user However, the user still has the option of using the traditional method of logging in using a user name and a password Figure 24 shows our built-in microcontroller for reading an RFID tag and communicating with the Hamon framework through Bluetooth

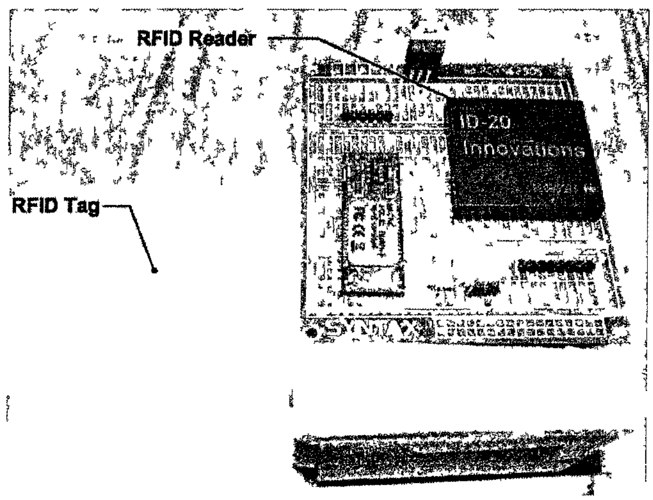

Figure 24: An RFID reader for the Hamon framework The reader connects to the framework using Bluetooth. The left side of the figure shows an RFID tag.

5.7. The Monitoring Interfaces

In this section, we demonstrate the implemented monitoring interfaces including the presentation of each Graphical User Interface (GUI) and overview of what the monitoring application looks like. The main interface of the framework is the monitoring interface where all the sensory data travels through and is presented to the healthcare provider (see Figure 25).

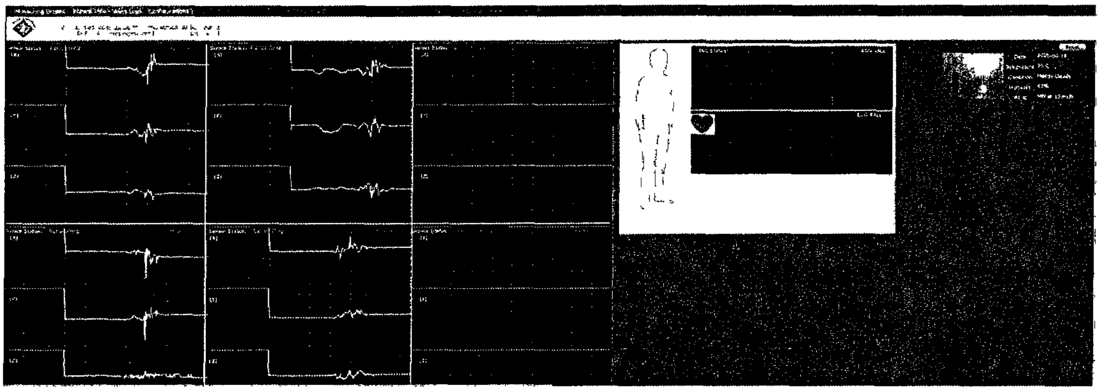

Figure 25: A screenshot shows the acceleration signals coming to the monitoring panel.

The next interface is the configuration panel which has been added to adjust the communication port numbers associated to the RFID reader, accelerometers, ECG, and any other sensors connected through serial ports. This interface is accessible to a technical administrator in order to customize all technical needs (see Figure 26 and Figure 27). In addition, the healthcare provider has a radio-control group with two modes for running the framework. The first mode is long-term monitoring, and the other mode is an exercise mode. In the first option, all the sensory data is stored in the database while the sensors are collecting data continuously. Thus, the framework knows it is long-term monitoring and some turns configurations to active mode, such as event trigger and event notifier.

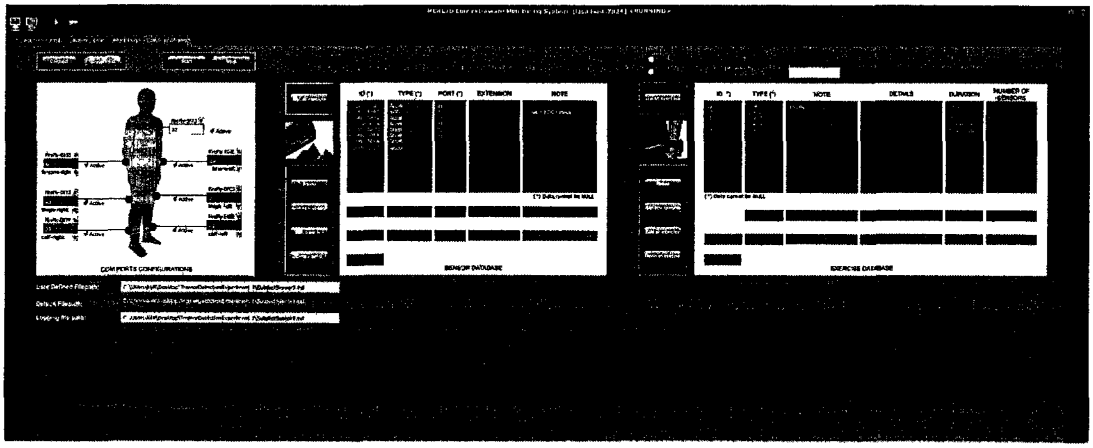

Figure 26: A screenshot of the configuration panel to control the monitoring mode and adjust the communication ports.

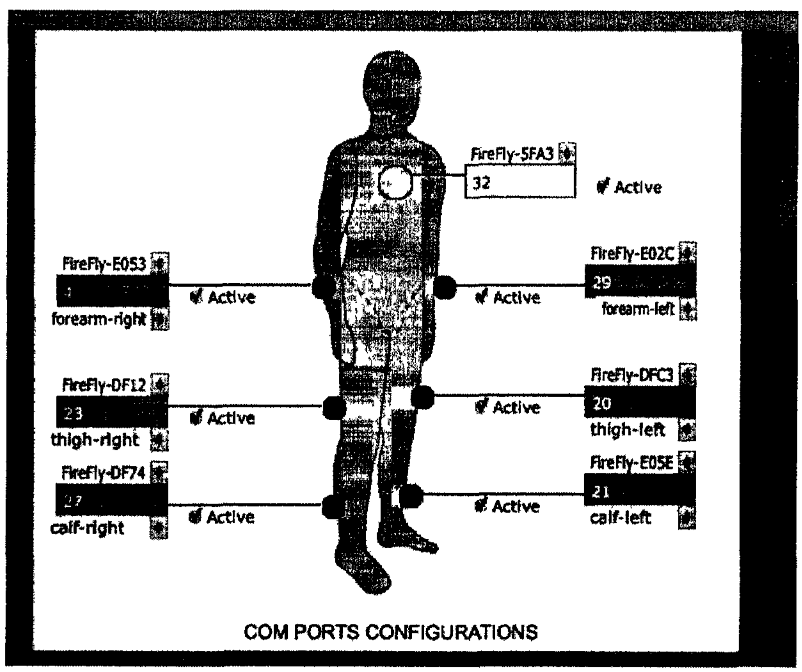

Figure 27: A close screenshot shows the sensor configuration to be used during the monitoring.

The next option is the exercise mode, where the therapist or healthcare provider has the option to either select an existing configuration of the exercise or create a new one. A form appears if the provider wants to create a new exercise, requiring related attributes as shown in Figure 28 and Figure 29. The attributes are used to describe an exercise, composed of the exercise type, date, number and the type of sensors used, duration, and a text-field to describe the exercise in details.

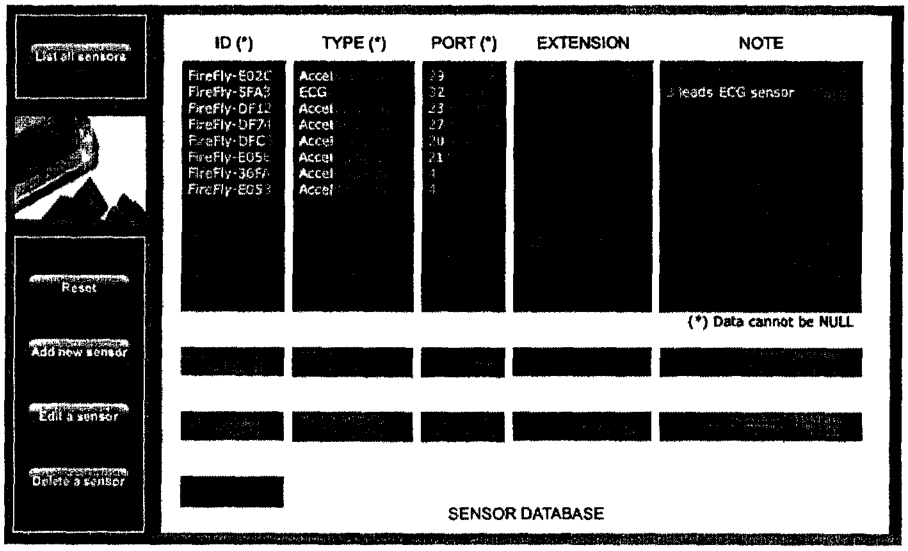

Figure 28: A screenshot presents a list of all the sensors that can be used in the monitoring application.

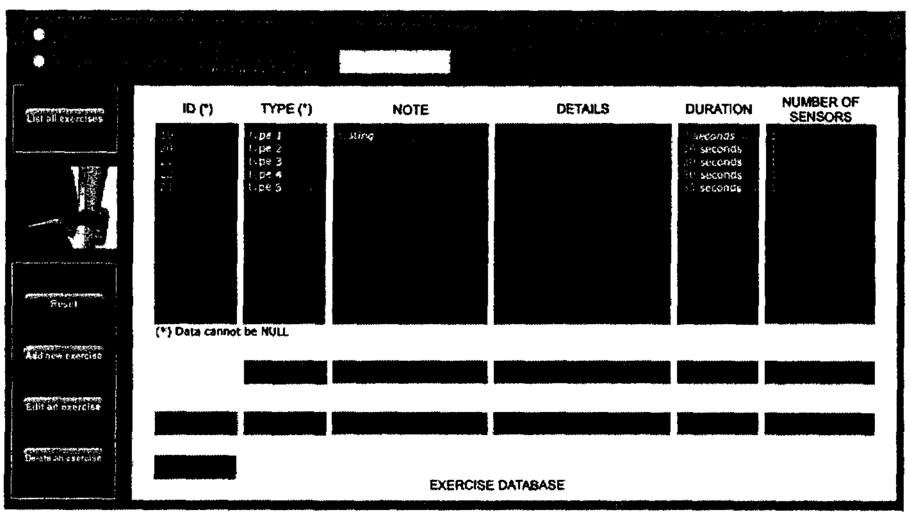

Figure 29: A screen capture shows the ability given to the user to choose the duration and the number of sensors for monitoring the patient and the ability to label the recorded data used in the exercise mode.

The last window to mentioned in this section is the patient profile which holds his/her health record. In the implemented prototype, we assumed that we had a health record database where the patients' information and their health files were available. Thus, they can be integrated easily in the framework to obtain any information that may be of interest of the health care provider. A screen capture with a patient's information retrieved from the medical database is shown in Figure 30.

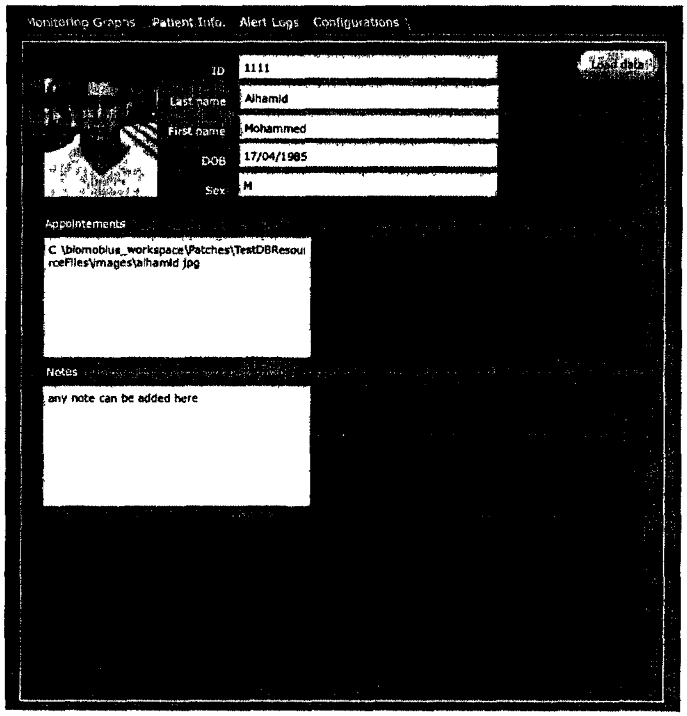

Figure 30: A screen capture shows a patient's information retrieved from the medical database.

Chapter 6. Experimental Results and Discussion

We point out in this chapter the performance of our proposed framework by analyzing five different activities: falling, lying down, sitting, standing, and walking of 23 healthy volunteers. The subjects were from different age, ethnic, sex, and weight demographics (five female, 18 male, with ages ranging from 20 to 55). During the experiments, none of the volunteers experienced any technical problems and the required data was collected successfully. The description of activities recorded for each subject is presented in Table 3.

Table 3: Description of the type of activities performed by volunteer subjects in the experiment

Task	Description	Duration Recording (Avg.)
Falling down	Initially, the subject is in the standing position and then falls down on a sponge mattress. The subject can fall in any direction and in any form and remains in this position after falling.	10 seconds
Lying	Initially, the subject is in the standing position and then he/she lies down on a sofa on their back and remains in this position.	12 seconds
Sitting	Initially, the subject is in the standing position and then sits down on a sofa and remains in this position.	7 seconds
Standing	Initially, the subject is in the sitting position, and then rises to an upright position on his/her feet. The subjects remains in this position thereafter.	7 seconds
Walking	Initially, the subject is in the standing position and then walks normally between 10 to 15 meters.	15 seconds

To evaluate the activity detection algorithm, we separated our work into different steps. The first step was analyzing the performance of the standard K-NN performance based on finding the nearest *k* vector. In order to achieve such findings, we needed to create a learning algorithm to train the K-NN classifier. The learning algorithm extracted the signal features discussed in 4.1.2 from the accelerations collected from the four sensors all together. Each feature computations were applied on the accelerations collected from the x, y, and z-axes. Thereafter, a single subject activity was pushed to the K-NN classifier to obtain the resulting classification label representing either falling, lying, sitting, standing, or walking. The loop was repeated for the rest of subject recorded activities and the results were recorded in a matrix to extract the overall recognition rate.

After analyzing the performance of the standard K-NN based on using the data collected from the four sensors all together, the second step was analyzing the performance of each sensor alone to detect an activity. Therefore, an algorithm was created to extract the signal features on the signal collected from only one sensor at a time, training the K-NN classifier with the resulting vector. Once the training database was formed, the third step was using K-NN classifier to categorize the activity executed from the data collected from one sensor. As a result, we ended up with four detection decisions for each sensor per activity, keeping in mind that we used four sensors during the experiments. The results were recorded to obtain the detection rate for each sensor independently.

The fourth step in evaluating the activity detection algorithm was feeding the resulted detection rates for the results collected to a Bayesian Network learning algorithm. The results came from classifying the activity based on a single sensor collected in step two. The learning algorithm was used to calculate the maximum likelihood in parameter learning to compute the Bayesian Network.

Before presenting the experimental results, we want to point out that finding a minimum "one" which is the closet vector in distance during the classification is not always accurate. Therefore, our classification algorithm is designed to be flexible to set the value of *k*. The K-NN algorithm and the parameter learning of Bayesian Network was repeated a few times to influence *k*'s value on the detection rate.

6.1. Results

6.1.1 Standard K-NN

An evaluation of the standard K-NN algorithm involved comparing the algorithm performance over different values for *k* is presented in this section. Table 4 compares the activity classified by the K-NN algorithm with the actual movements labelled during the time of doing the experiments, where the value of *k*=1. If the actual activity matched the classified activity, then this was counted as correct detection; if the classifier generated a detection that mismatched the labelled activity, then this was counted an incorrect detection. Afterward, the overall accuracy of detection was calculated measuring the performance of the standard K-NN in detecting each singular activity as well as the total detection rate.

Table 4: Experimental Results for activity detection using standard K-NN algorithm, where *k*=1.

Task	No. of subjects performed this task	Overall Correct Detection	Overall Incorrect Detection	Detection Accuracy (%)
Falling	23	16	7	69.57%
Lying	23	15	8	65.22%
Sitting	23	14	9	60.87%
Standing	23	17	6	73.91%
Walking	23	19	4	82.60%
Total	115	81	34	70.43%

In Table 5, we show the activity classified by the K-NN algorithm with the actual movements labelled during the time of the experiments, where the value of k=3. The calculations method used for counting the correct and incorrect detections along with the accuracy rate remains the same as when k =1.

Table 5: Experimental Results for activity detection using standard K-NN algorithm, where k=3.

Task	No. of subjects performed this task	Overall Correct Detection	Overall Incorrect Detection	Detection Accuracy (%)
Falling	23	17	6	73.91%
Lying	23	14	9	60.87%
Sitting	23	14	9	60.87%
Standing	23	17	6	73.91%
Walking	23	21	2	91.30%
Total	115	83	32	72.17%

Table 6 shows the performance of the K-NN algorithm where the value of k=4. The calculations method used for counting the correct and incorrect detections along with the accuracy rate remains the same as in the previous tables.

Table 6: Experimental Results for activity detection using standard K-NN algorithm, where k=4.

Task	No. of subjects performed this task	Overall Correct Detection	Overall Incorrect Detection	Detection Accuracy (%)
Falling	23	18	5	78.26%
Lying	23	14	9	60.87%
Sitting	23	14	9	60.87%
Standing	23	13	10	56.52%
Walking	23	18	5	78.26%
Total	115	77	38	66.95%

The last case is measuring the performance of standard K-NN where the value of k=5 (see Table 7). The discussion of all the results presented so far is mentioned in Section 6.2.

Table 7: Experimental Results for activity detection using standard K-NN algorithm, where k=5.

Task	No. of subjects performed this task	Overall Correct Detection	Overall Incorrect Detection	Detection Accuracy (%)
Falling	23	17	6	73.91%
Lying	23	14	9	60.87%
Sitting	23	15	8	65.22%
Standing	23	16	7	69.56%
Walking	23	19	4	82.60%
Total	115	81	34	70.43%

6.1.2 K-NN and Bayesian Network

In the previous section, we showed the performance of the standard K-NN over different values chosen for k. In contrast, this section highlights the results of Bayesian networks over different values chosen for k, where the K-NN algorithm is applied on each sensor's set of data. Table 8 through Table 23 display the accuracies of each sensor used in the experiments in classifying the five measured activities. The columns marked Falling, Lying, Sitting, Standing, and Walking illustrate the real activity, and the rows marked Falling, Lying, Sitting, Standing, and Walking illustrate the classification label resulting from applying the K-NN algorithm.

Each cell in the tables describes the probabilities of each sensor in detecting the activity labelled with the row index value over the real actual activity labelled with the column index value. For instance, the first row in Table 8 can be interpreted as the following: the first cell from left to the right represents the probability of the head sensor in classifying the activity as falling while the actual activity is also falling. The second cell represents the probability of the head sensor in classifying the activity as falling while the real activity is lying. The third cell represents the probability of the head sensor in classifying the activity as falling while the real activity is sitting. The fourth cell represents the probability of the head sensor in classifying the activity as falling while the actual activity is standing. Similarly, the fifth cell represents the probability of the head sensor in classifying the activity as falling while the actual activity is walking.

These tables show that learning Bayesian networks can measure the accuracy of each sensor in detecting each of the five activities along with showing the false detection relations between the classified and the real activity. On the other hand, we can predict different enhancement way to detect each activity by either combining the decision of two sensors or three in classifying the activities with least detection accuracy based on a single sensor classification. More discussion of the results is presented in section 6.2.

K = 1

Table 8 through Table 11 display the probability of the classified activities over the actual movements labelled during the time of doing the experiments by applying the K-NN algorithm with value of 1 for the variable *k*.

Table 8: The conditional probability table for detecting activities based on the accelerations data collected from the head sensor only, where *k*=1.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.55	0.4	0.05	0	0
Lying	0.25	0.7	0	0.05	0
Sitting	0.1	0	0.3	0.4	0.2
Standing	0	0	0.15	0.6	0.25
Walking	0	0	0.1	0.2	0.7

Table 9: The conditional probability table for detecting activities based on the accelerations data collected from the lower back sensor only, where *k*=1.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.55	0.35	0	0.1	0
Lying	0.35	0.35	0.05	0.05	0.2
Sitting	0.05	0	0.4	0.45	0.1
Standing	0	0	0.35	0.5	0.15
Walking	0	0	0.05	0.05	0.9

Table 10: The conditional probability table for detecting activities based on the accelerations data collected from the forearm sensor only, where *k*=1.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.6	0.25	0.1	0.05	0
Lying	0.2	0.5	0.15	0.1	0.05
Sitting	0.1	0.05	0.4	0.45	0
Standing	0.05	0.2	0.4	0.25	0.1
Walking	0.05	0.1	0	0	0.85

Table 11: The conditional probability table for detecting activities based on the accelerations data collected from the thigh sensor only, where *k*=1.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.6	0.35	0	0.05	0
Lying	0.3	0.4	0.1	0.2	0
Sitting	0.1	0.2	0.4	0.3	0
Standing	0.2	0.15	0.2	0.45	0
Walking	0	0	0	0	1

K = 3

Table 12 through Table 15 display the probability of the classified activities over the actual movements labelled during the time of doing the experiments by applying the K-NN algorithm with value of 3 for the variable ***k***.

Table 12: The conditional probability table for detecting activities based on the accelerations data collected from the head sensor only, where *k*=3.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.5	0.4	0.05	0.05	0
Lying	0.25	0.7	0	0.05	0
Sitting	0.05	0	0.3	0.45	0.2
Standing	0	0	0.1	0.7	0.2
Walking	0	0	0.05	0.1	0.85

Table 13: The conditional probability table for detecting activities based on the accelerations data collected from the lower back sensor only, where *k*=3.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.55	0.35	0	0.1	0
Lying	0.45	0.2	0.1	0.05	0.2
Sitting	0.05	0	0.4	0.45	0.1
Standing	0	0	0.3	0.55	0.15
Walking	0	0	0.05	0.05	0.9

Table 14: The conditional probability table for detecting activities based on the accelerations data collected from the forearm sensor only, where *k*=3.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.5	0.35	0.05	0.1	0
Lying	0.25	0.45	0.15	0.1	0.05
Sitting	0.1	0.05	0.35	0.5	0
Standing	0	0.15	0.35	0.35	0.15
Walking	0.1	0.1	0	0	0.8

Table 15: The conditional probability table for detecting activities based on the accelerations data collected from the thigh sensor only, where *k*=3.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.75	0.2	0	0.05	0
Lying	0.3	0.45	0.1	0.15	0
Sitting	0.1	0.15	0.4	0.35	0
Standing	0.1	0.15	0.25	0.5	0
Walking	0	0	0	0	1

K = 4

Table 16 through Table 19 display the probability of the classified activities over the actual movements labelled during the time of doing the experiments by applying the K-NN algorithm with value of 4 for the variable *k*.

Table 16: The conditional probability table for detecting activities based on the accelerations data collected from the head sensor only, where *k*=4.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.7	0.2	0.05	0.05	0
Lying	0.35	0.6	0	0.05	0
Sitting	0.05	0	0.3	0.5	0.15
Standing	0	0	0.4	0.35	0.25
Walking	0	0	0.05	0.2	0.75

Table 17: The conditional probability table for detecting activities based on the accelerations data collected from the lower back sensor only, where *k*=4.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.7	0.2	0	0.1	0
Lying	0.55	0.1	0.1	0.05	0.2
Sitting	0.05	0	0.4	0.4	0.15
Standing	0	0	0.35	0.5	0.15
Walking	0	0	0.05	0.15	0.8

Table 18: The conditional probability table for detecting activities based on the accelerations data collected from the forearm sensor only, where *k*=4.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.5	0.25	0.15	0.1	0
Lying	0.1	0.45	0.25	0.1	0.1
Sitting	0.1	0.1	0.45	0.35	0
Standing	0	0.1	0.4	0.4	0.1
Walking	0.1	0.1	0	0.05	0.75

Table 19: The conditional probability table for detecting activities based on the accelerations data collected from the thigh sensor only, where *k*=4.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.75	0.2	0	0.05	0
Lying	0.3	0.55	0.1	0.05	0
Sitting	0.1	0.25	0.45	0.2	0
Standing	0.1	0.15	0.3	0.45	0
Walking	0	0	0	0	1

K = 5

Table 20 through Table 23 display the probability of the classified activities over the actual movements labelled during the time of doing the experiments by applying the K-NN algorithm with value of 5 for the variable *k*.

Table 20: The conditional probability table for detecting activities based on the accelerations data collected from the head sensor only, where *k*=5.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.7	0.2	0.05	0.05	0
Lying	0.35	0.6	0	0.05	0
Sitting	0.05	0	0.3	0.55	0.1
Standing	0.05	0	0.45	0.15	0.35
Walking	0	0	0.1	0.15	0.75

Table 21: The conditional probability table for detecting activities based on the accelerations data collected from the lower back sensor only, where *k*=5.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.75	0.15	0	0.1	0
Lying	0.45	0.15	0.15	0.05	0.2
Sitting	0.05	0	0.35	0.4	0.2
Standing	0	0	0.3	0.55	0.15
Walking	0	0	0	0.1	0.9

Table 22: The conditional probability table for detecting activities based on the accelerations data collected from the forearm sensor only, where *k*=5.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.7	0.15	0.15	0	0
Lying	0.2	0.45	0.2	0.15	0
Sitting	0.2	0.2	0.35	0.2	0.05
Standing	0.05	0.05	0.4	0.4	0.1
Walking	0.05	0.15	0	0.15	0.65

Table 23: The conditional probability table for detecting activities based on the accelerations data collected from the thigh sensor only, where *k*=5.

	Falling	Lying	Sitting	Standing	Walking
Falling	0.6	0.3	0	0.1	0
Lying	0.35	0.5	0.15	0	0
Sitting	0.1	0.2	0.35	0.3	0.05
Standing	0.1	0.15	0.15	0.6	0
Walking	0	0	0	0	1

6.2. Discussion

Testing the framework showed its ability to collect sensory data from the BSN successfully and revealed its ability to support different health monitoring scenarios. The data collected has been successfully stored in an archive database and can be retrieved and manipulated in both online and offline modes. In addition, the implemented prototype showed the ability to distinguish between five types of activities and to recognize falling, lying down, sitting, standing, and walking using four sensors attached to the patient's body with a varying degree of accuracy.

The classification algorithms employed showed some acceptable accuracy in detecting some activities and lower accuracy on others. Hence, the contributions of this thesis come to point out the limitations of the standard K-NN algorithm employed in detecting the five types of activities. Contribution also includes the attempt to enhance the detection rate by applying a Bayesian Network for building a probability distribution for classifying a set of training data. In the next paragraph, we point out the limitations of the resulted K-NN applied to classify the five set of activities.

The performance of the K-NN algorithm used for classifying the training data showed a higher recognition rate in detecting the walking activity with 91.30 per cent accuracy and acceptable rates on the rest of the activities. Figure 31 shows the accuracy of detecting the five activities by applying the standard K-NN classifier. The horizontal axis of the graph represents the five activities to be detected: falling, lying down, sitting, standing, and walking, where the vertical axis represents the accuracy rate in percentage. For each activity presented in the graph, four columns show the detection rate resulting from applying the K-NN classifier with four sets of values for k containing k=1, k=3, k=4, and k=5. The K-NN classifier gives a higher accuracy in overall detection when the value of k is set to 3, where the walking activity was detected with accuracy rate equal to 91.30 per cent. The falling activity was detected with an accuracy rate equal to 73.91 per cent, while sitting and lying down were detected with 60.87 per cent accuracy. The standing activity was detected with 73.91 per cent. The overall detection rate of the standard K-NN classifier of the detected activities was 72.17 per cent.

The behaviours of the K-NN classifier in classifying activities when the *k* value is changed has shown an agreeable expected performance to a mathematical model for selecting the right value of *k*, proposed in one of the related works.

Enas et al. [54] suggested a *k* scaling method after performing a simulation study to investigate *k*-nearest neighbour algorithm for classification. Enas et al. showed the importance of picking the right value of *k* in solving a classification problem. Enas suggested scaling the *k* value as $n^{2/8}$ or $n^{3/8}$ based on the size of the training set. In our activity classification problem, the value of n=23, therefore, the k=3.24 which shows the same accuracy obtained in our experimental results.

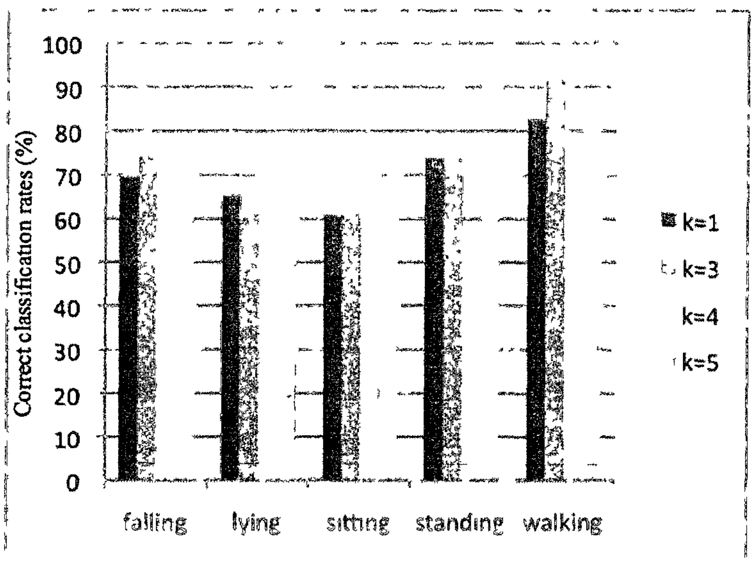

Figure 31: Correct classification rates (%) obtained from applying the standard K-NN classifier in classifying the five types of motion activities.

Now we point out the classification rates of each sensor used in classifying the five sets of activities independently. The graph presented in Figure 32 shows the resulted correct classification rates obtained from applying the K-NN classifier on the data collected from the head sensor. In the graph, there are four *k* values represented in columns for each attributed containing: *k*=1, *k*=3, *k*=4, and *k*=5 used to detect the five activities: falling, lying down, sitting, standing, and walking. From the same graph, we can observe the ability of an accelerometer attached to the head to detect the walking activity with 90 per cent accuracy when *k*=3. However, the head sensor was not able to distinguish between the sitting and the standing activities. Hence, 35 per cent of the sitting cases were classified as standing. Similarly, the head sensor was not able to distinguish between the falling and lying activities, so 30 percent of the lying cases were classified as falling.

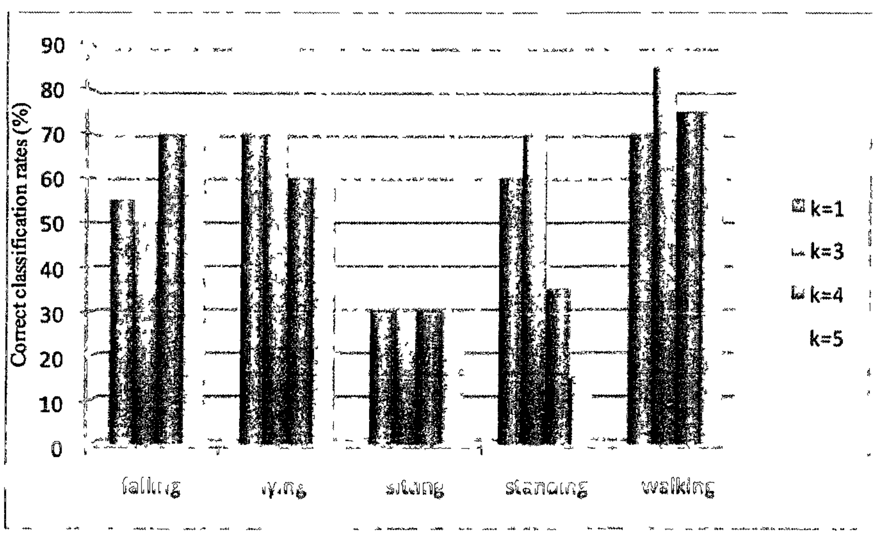

Figure 32: Correct classification rates (%) obtained from applying the K-NN classifier on the accelerations collected from a sensor attached to the subjects' head.

Figure 33 shows the correct classification rates resulting from classifying the training set of data collected using an accelerometer attached to the lower back. Similar to the results obtained from the head sensor, the lower back attached sensor showed the ability to recognize the walking activity with 90 per cent accuracy. The accuracy of detecting the falling motions matches the same ranges of the one recorded using the head sensor. However, the lower back sensor results constrained compared with other obtained results. Thus, it is not highly recommended to depend on the lower back sensor to detect the lying down activity.

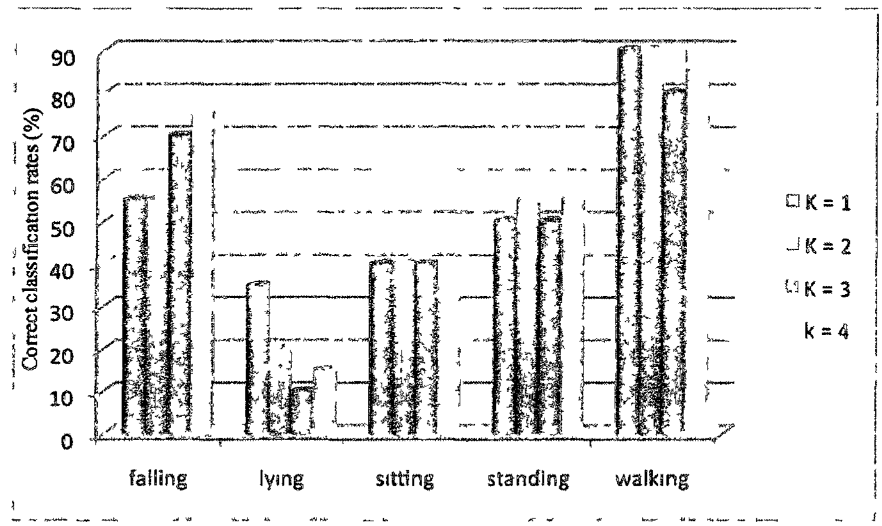

Figure 33: Correct classification rates (%) obtained from applying the K-NN classifier on the accelerations collected from a sensor attached to the subjects' lower back.

Figure 34 presents the correct classification rates from classifying the training data set by depending on the motion accelerations collected from an accelerometer attached to the forearm. The ability of the forearm sensor in detecting the walking activity varies from 65 per cent to 85 per cent depending on the value of k. Therefore, it has the lowest accuracy rate obtained in detecting such activity compared with the other results, especially the thigh sensor results presented in Figure 35.

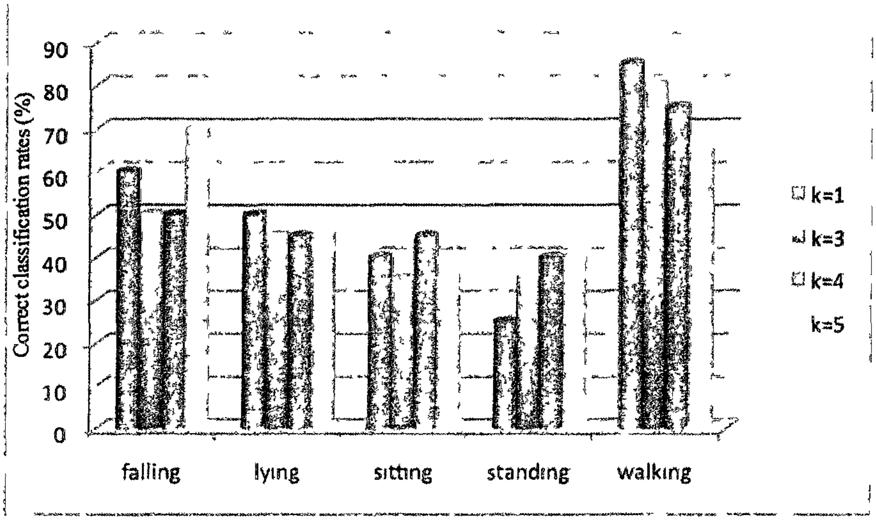

Figure 34: Correct classification rates (%) obtained from applying the K-NN classifier on the accelerations collected from a sensor attached to the subjects' forearm.

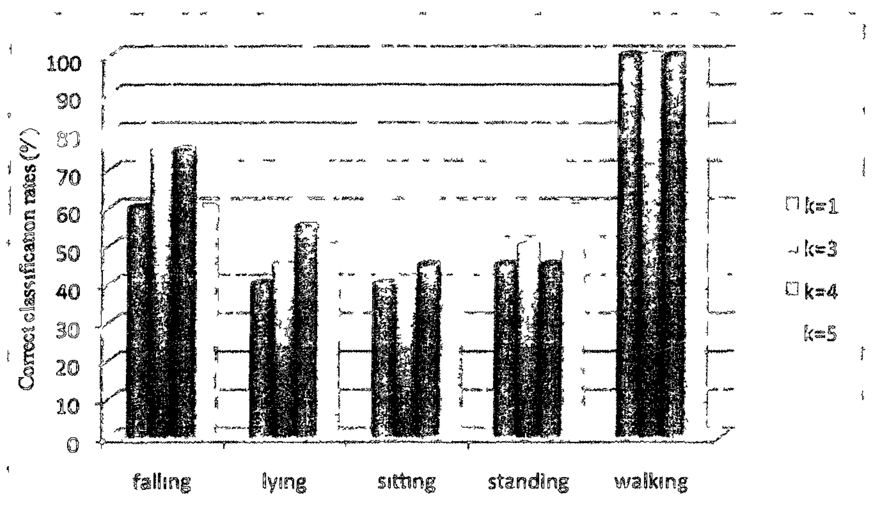

Figure 35: Correct classification rates (%) obtained from applying the K-NN classifier on the accelerations collected from a sensor attached to the subjects' thigh.

The time performance of the K-NN algorithm was acceptable when the value of k changed from one to another. For instance, the time spent to classify an activity was on average 140, 171, 203, and 218 milliseconds respectively for k=1, k=3, k=4, and k=5.

The ability to detect the five activities overall using one of the sensors attached to one of the four body segments is combined in Figure 36. The k value used during the classification is k=3. In Figure 35, we can clearly determine the ability of the thigh sensor in achieving 100 per cent accuracy of classifying walking from other activities. Moreover, the thigh sensor achieved an acceptable accuracy of 75 per cent in detecting the falling activity. Lying and standing activities were highly detected by the sensor attached to the head at a rate of 70 per cent when compared with the other three sensors. The standing activity is the hardest to be detect using only one sensor. None of the sensors used produced acceptable results; all accuracy rates were below the 40 per cent.

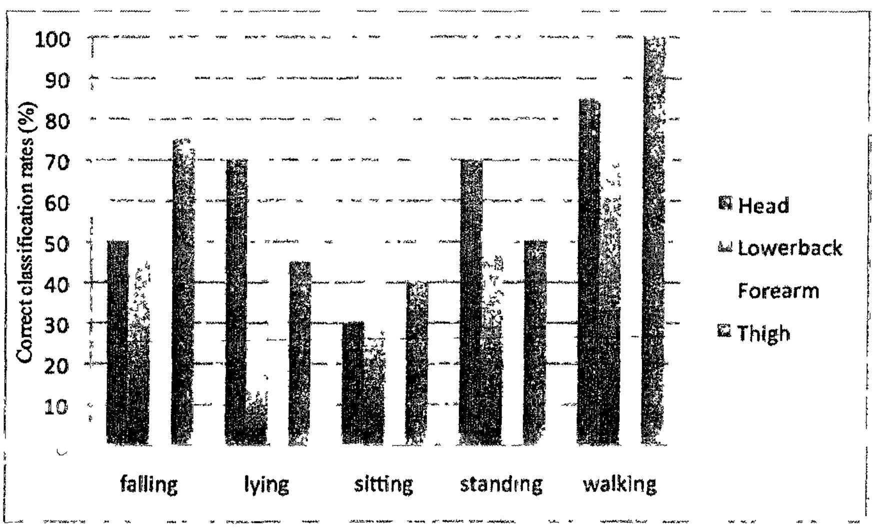

Figure 36: Correct classification rates (%) obtained from applying the K-NN classifier on the accelerations collected from a sensor attached to the subjects' head, lower back, forearm, and thigh.

In order to improve the accuracy rate of the K-NN algorithm, we used Bayesian networks to acquire the probabilistic inference. We fed the algorithm presented in Appendix B with some facts, such as the head sensor detecting the activity as falling. There are five possible cases for the detected activity: falling, lying down, sitting, standing, or walking. Therefore, the question we tried to solve using Bayesian inference probabilities is: "Which activity is most likely occurring when an activity classified as falling?" Bayes' rule is used to calculate each probability for the five classified activities for each sensor using Kevin Murphy Matlab tool [51].

Table 24 through Table 27 show the resulting probability inference of using each sensor in detecting an activity. Table 24 presents the probabilities divided in five groups: falling, sitting, standing, and walking activities. For each group, a five posterior probability is presented to explain how each activity being classified. For example, "P(R=fall down | H=fall down)=0.625" shows the probability of the activity to be falling by observing that the head sensor classified the activity as falling. The table cells highlighted in bold represent the correct classification; hence, our current example shows 62.5 percent of all the activity classified as falling is actually a falling activity in real.

Table 24: The probabilistic inference resulted from the Bayesian Networks for the sensor attached to the subject's head during the experiments.

<table>
<tr><th>Falling Activity</th><th>Sitting Activity</th></tr>
<tr><td>P(R=fall down | H=fall down) = 0.625</td><td>P(R=fall down | H=sitting) = 0.100</td></tr>
<tr><td>P(R=lying | H=fall down) = 0.313</td><td>P(R=lying | H=sitting) = 0.000</td></tr>
<tr><td>P(R=sitting | H=fall down) = 0.063</td><td>P(R=sitting | H=sitting) = 0.600</td></tr>
<tr><td>P(R=standing | H=fall down) = 0.000</td><td>P(R=standing | H=sitting) = 0.200</td></tr>
<tr><td>P(R=walking | H=fall down) = 0.000</td><td>P(R=walking | H=sitting) = 0.100</td></tr>
<tr><td>Lying Activity</td><td>Standing Activity</td></tr>
<tr><td>P(R=fall down | H=lying) = 0.364</td><td>P(R=fall down | H=standing) = 0.037</td></tr>
<tr><td>P(R=lying | H=lying) = 0.636</td><td>P(R=lying | H=standing) = 0.037</td></tr>
<tr><td>P(R=sitting | H=lying) = 0.000</td><td>P(R=sitting | H=standing) = 0.333</td></tr>
<tr><td>P(R=standing | H=lying) = 0.000</td><td>P(R=standing | H=standing) = 0.519</td></tr>
<tr><td>P(R=walking | H=lying) = 0.000</td><td>P(R=walking | H=standing) = 0.074</td></tr>
<tr><td>Walking Activity</td><td></td></tr>
<tr><td>P(R=fall down | H=walking) = 0.000</td><td></td></tr>
<tr><td>P(R=lying | H=walking) = 0.000</td><td></td></tr>
<tr><td>P(R=sitting | H=walking) = 0.160</td><td></td></tr>
<tr><td>P(R=standing | H=walking) = 0.160</td><td></td></tr>
<tr><td>P(R=walking | H=walking) = 0.680</td><td></td></tr>
</table>

Table 25 shows the inference probability resulted from using the lower back sensor in classifying the training data set. The table shows the weaknesses of the lower back in detecting the lying activity as it always misclassified as falling which proofs the first findings from Figure 16. As a result, the lower back sensor is excluded in classifying such activity.

Table 25: The probabilistic inference resulted from the Bayesian Networks for the sensor attached to the subject's lower back during the experiments.

Falling Activity	Sitting Activity
P(R=fall down | L=fall down) = 0.524	P(R=fall down | L=sitting) = 0.000
P(R=lying | L=fall down) = 0.429	P(R=lying | L=sitting) = 0.118
P(R=sitting | L=fall down) = 0.048	**P(R=sitting | L=sitting) = 0.471**
P(R=standing | L=fall down) = 0.000	P(R=standing | L=sitting) = 0.353
P(R=walking | L=fall down) = 0.000	P(R=walking | L=sitting) = 0.059
Lying Activity	Standing Activity
P(R=fall down | L=lying) = 0.636	P(R=fall down | L=standing) = 0.083
P(R=lying | L=lying) = 0.364	P(R=lying | L=standing) = 0.042
P(R=sitting | L=lying) = 0.000	P(R=sitting | L=standing) = 0.375
P(R=standing | L=lying) = 0.000	**P(R=standing | L=standing) = 0.458**
P(R=walking | L=lying) = 0.000	P(R=walking | L=standing) = 0.042
Walking Activity	
P(R=fall down | L=walking) = 0.000	
P(R=lying | L=walking) = 0.148	
P(R=sitting | L=walking) = 0.074	
P(R=standing | L=walking) = 0.111	
P(R=walking | L=walking) = 0.667	

Table 26 shows the inference probability resulting from using the forearm sensor in classifying the training data set. The table shows the weaknesses of the forearm sensor in detecting the sitting and standing activities; these activities were often misclassified as the other which proves the first findings from Figure 17. As a result, the forearm sensor is excluded in classifying the sitting and standing activities.

Table 26: The probabilistic inference resulted from the Bayesian Networks for the sensor attached to the subject's forearm during the experiments.

<table>
<tr><td>Falling Activity</td><td>Sitting Activity</td></tr>
<tr><td>P(R=fall down | F=fall down) = 0.526</td><td>P(R=fall down | F=sitting) = 0.056</td></tr>
<tr><td>P(R=lying | F=fall down) = 0.263</td><td>P(R=lying | F=sitting) = 0.167</td></tr>
<tr><td>P(R=sitting | F=fall down) = 0.105</td><td>P(R=sitting | F=sitting) = 0.389</td></tr>
<tr><td>P(R=standing | F=fall down) = 0.000</td><td>P(R=standing | F=sitting) = 0.389</td></tr>
<tr><td>P(R=walking | F=fall down) = 0.105</td><td>P(R=walking | F=sitting) = 0.000</td></tr>
<tr><td>Lying Activity</td><td>Standing Activity</td></tr>
<tr><td>P(R=fall down | F=lying) = 0.318</td><td>P(R=fall down | F=standing) = 0.095</td></tr>
<tr><td>P(R=lying | F=lying) = 0.409</td><td>P(R=lying | F=standing) = 0.095</td></tr>
<tr><td>P(R=sitting | F=lying) = 0.045</td><td>P(R=sitting | F=standing) = 0.476</td></tr>
<tr><td>P(R=standing | F=lying) = 0.136</td><td>P(R=standing | F=standing) = 0.333</td></tr>
<tr><td>P(R=walking | F=lying) = 0.091</td><td>P(R=walking | F=standing) = 0.000</td></tr>
<tr><td>Walking Activity</td><td></td></tr>
<tr><td>P(R=fall down | F=walking) = 0.000</td><td></td></tr>
<tr><td>P(R=lying | F=walking) = 0.050</td><td></td></tr>
<tr><td>P(R=sitting | F=walking) = 0.000</td><td></td></tr>
<tr><td>P(R=standing | F=walking) = 0.150</td><td></td></tr>
<tr><td>P(R=walking | F=walking) = 0.800</td><td></td></tr>
</table>

Table 27 shows the inference probabilistic resulted from using the thigh sensor in classifying the training data set. The table shows a high recognition rate in detecting the walking activity which also proves the first findings from Figure 18. As a result, the thigh sensor is always used while ignoring the other sensors for classifying the walking activity.

Table 27: The probabilistic inference resulted from the Bayesian Networks for the sensor attached to the subject's thigh during the experiments.

Falling Activity	Sitting Activity
P(R=fall down | T=fall down) = 0.600	P(R=fall down | T=sitting) = 0.000
P(R=lying | T=fall down) = 0.240	P(R=lying | T=sitting) = 0.133
P(R=sitting |T=fall down) = 0.080	**P(R=sitting | T=sitting) = 0.533**
P(R=standing | T=fall down) = 0.080	P(R=standing | T=sitting) = 0.333
P(R=walking | T=fall down) = 0.000	P(R=walking | T=sitting) = 0.000
Lying Activity	Standing Activity
P(R=fall down | T=lying) = 0.211	P(R=fall down | T=standing) = 0.048
P(R=lying | T=lying) = 0.474	P(R=lying | T=standing) = 0.143
P(R=sitting | T=lying) = 0.158	P(R=sitting | T=standing) = 0.333
P(R=standing | T=lying) = 0.158	**P(R=standing | T=standing) = 0.476**
P(R=walking | T=lying) = 0.000	P(R=walking | T=standing) = 0.000
Walking Activity	
P(R=fall down | T=walking) = 0.000	
P(R=lying | T=walking) = 0.000	
P(R=sitting | T=walking) = 0.000	
P(R=standing | T=walking) = 0.000	
P(R=walking | T=walking) = 1.000	

After observing the inference probabilities using the Bayesian networks as presented in the previous tables, we ran the new algorithm again on our recorded experiments. The resulted performance of the proposed K-NN and Bayesian networks algorithm showed its feasibility to improve the accuracy rate of the K-NN algorithm. We fed the algorithm presented in Appendix B with some facts, such as the head sensor detected the activity as falling, lower back detected the same activity as falling and so on for the other sensors. Using the Bayesian Network inference algorithm, we classify the activity based on the four sensor decisions. An evaluation of the new proposed algorithm is presented in Table 28.

The data presented in Table 28 compares the activity classified by the K-NN and Bayesian Network algorithm with the actual movements labelled during the time of doing the experiments, where the value of k=3. If the actual activity matched the classified activity, then this was counted as correct detection; if the classifier generated a detection that mismatched the labelled activity, then this was counted an incorrect detection. Afterward, the overall accuracy of detection was calculated measuring the performance of the K-NN and Bayesian Network algorithm in detecting each singular activity as well as the total detection rate.

Table 28: Experimental Results for activity detection using the proposed K-NN and Bayesian Network algorithm, where k=3.

Task	No. of subjects performed this task	Overall Correct Detection	Overall Incorrect Detection	Detection Accuracy (%)
Falling	23	19	4	82 61%
Lying	23	19	4	82.61%
Sitting	23	14	9	60.86%
Standing	23	18	5	78.26%
Walking	23	23	0	100%
Total	115	93	22	80.86%

Chapter 7. Conclusions

In this thesis, we developed and studied a framework called Hamon to help in real-time monitoring with processing a list of vital signs for health care support. Hamon is aimed at monitoring elderly people with wearable sensors in their homes by analyzing and visualizing data generated by sensor nodes. Hamon has been proposed to be applied for different health monitoring applications serving a variety of needs which include detection, risk prevention, and a convenient method for monitoring physical rehabilitation.

The focus of our research is subdivided into two main parts: building the activity recognition framework for health monitoring support in the home environment together with the Wireless Body Sensor Network (WBSN), and building the activity recognition algorithm used for classifying five activities: falling, lying down, sitting, standing, and walking.

The Hamon framework included a sensors network, multiple monitoring interfaces, and a number of methods for classifying and displaying the collected data. These components helped in determining the type of activity, and collecting physiological data such as heart rate. The structure of the framework was divided into main parts: the sensory network and the monitoring software architecture on the server side. The hardware components of the framework including the sensor nodes, and the supportive software components including the implementation details have been described.

Hamon supports a variety of data collection from the WBSN, along with the context-aware data such as weather temperature. Hamon also supports the dynamic addition of different sensor nodes providing an open model for future services extension. We tested the implemented prototype of Hamon with various sensors containing accelerometers, gyroscopes, and an ECG, all of which are wireless wearable sensors enabling the patients to wear and roam freely in the their home environment. The experimental results show the technical feasibility of Hamon to be used in a long-term monitoring at home including a method of acquiring the patient's health condition.

Activity recognition is another main part of this thesis where an implemented algorithm is proposed to recognize falling, lying down, sitting, standing, and walking. The classification in Hamon is based on the acceleration data acquired from four 3-axis accelerometers mounted on the subject's head, lower back, forearm, and thigh. We have presented an approach for activity recognition using the basis of Bayesian networks to improve the standard K-NN algorithm.

Bayesian networks are used as an alternative to depending only on the signal features by calculating maximum likelihood estimation for each sensor in detecting each activity. We studied the detection rate of the algorithm proposed by running an empirical study to test the performance of the four accelerometer sensors attached to the four body segments. We found that the falling, lying, standing, and walking activities can be detected with fairly high accuracy. Sitting was the hardiest activity to detect, where all four sensors did not achieve accuracy rates higher than 60 per cent. The other benefit of using our proposed algorithm is its ability to tell what activity misled the classification and what the probability is of each sensor detecting each of the five activities independently.

A number of challenges have been addressed in this thesis including: the design and the implementation of the framework, the selection and the implementation of the activity recognition, and the difficulties of recognizing each activity. In addition, we showed the accuracy of each body segment used in detecting falling, sitting, standing, walking, and lying down.

7.1. Future work

Directions for future work are divided into two parts. First, adding some additional sensors such as blood pressure, oxygen level, and body temperature to be integrated in the Hamon framework to strengthen our risk prevention mechanism. In addition, the ECG signals can be studied intensively to help in identifying the type of activity, especially the heart-rate ranges while the patient is executing an activity. Thus, we may find some unique patterns that can be added to the feature vectors during the classification of an activity. The second direction is refining our detection algorithm; extending the features calculation by adding signal correlation, signal power and others; and comparing the algorithm with the Hidden Markov Model (HMM), J48/C4.5, it to our classification problem.

Studying the gender differences is also an alternative direction for future work. Facts about the patient being monitored such as gender and age differences may affect the overall recognition performance. Therefore, our classifier has to deal with additional input parameters including weight, gender, age, and whether the patient suffers from any movement difficulties. In addition, the learning step has to accommodate and differentiate between patient activities to place into groups for later classifications. These aspects can be covered by extending the recorded experiments and increasing the number of training sets.

An interesting extension would also be testing other body segments to attach the accelerometer sensor. Finding a body segment that is effective in producing a higher overall detection rate or classifying hard-to-recognize activities such as sitting would be beneficial. Increasing the number of subjects to perform the experiments, would enable us to reach an accurate set of data to train the data set. Thus, the classification method would have more data to compare with and be more accurate as a result.

References

[1] P.M. Kelly-Hayes, J.T. Robertson, J.P. Broderick, P.W. Duncan, L.A. Hershey, E.J. Roth, et. al., "The American heart association stroke outcome classification," Stroke, vol. 29, pp. 1274-1280, 1998.

[2] G. Burdea, V. Popescu, V. Hentz, and K. Colbert, "Virtual reality-based orthopedic telerehabilitation," IEEE Trans. Rehab Eng. vol. 8, pp. 430-432, 2000.

[3] W. Dafoe, and P. Huston "Currents trends in cardiac rehabilitation," Canadian Medical Association Journal, vol. 156, pp. 527-532, 1997.

[4] P. Bonato, "Applications of wearable technology in rehabilitation," pHealth, 2008.

[5] T.G. Russell, "Physical rehabilitation using telemedicine," Journal of Telemedicine and Telecare, vol. 13(5), pp. 217-220, 2007.

[6] E. Jovanov, A. Milenkovic, C. Otto and P.C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," Journal of NeuroEngineering and Rehabilitation, vol. 2(6), 2005.

[7] M.J. Mathie, B.G. Celler, N.H. Lovell, and A.C.F. Coster, "Classification of basic daily movements using a triaxial accelerometer," Med. Biol. Eng Comput, vol. 42, pp. 679-687, 2004.

[8] T. Hester, R. Hughes, D.M. Sherril, B. Knorr, M. Akay, J. Stein, and P. Bonato, "Using wearable sensors to measure motor abilities following stroke," Proceeding of the International Workshop on Wearable and Implantable Body Sensor Network, 2006.

[9] R.J.W. Dunnewold, C.E. Jacobi, and J.J. van Hilten, "Quantitative assessment of bradykinesia in patients with Parkinson's disease," J. of Neuroscience Method, vol. 74, pp. 107-112, 1997.

[10] M.K. Holden, T.A. Dyar, L. Schwamm, and E. Bizzi, "Virtual environment-based telerehabilitation in patients with stoke," Presence: Teleoperators Virtual Environment, vol. 14(2), pp. 214-233, 2005.

[11] A.J. Jara, M.A. Zamora-Izquierdo, and A.F. Gomez-Skarmeta, "An ambient assisted living system for telemedicine with detection of symptoms," International Work-conference on the Interplay between Natural and Artificial Computing, vol.2, pp. 75-84, 2009.

[12] D. Hutton, "Older people in emergencies: considerations for action and policy development," World Health Organization WHO, 2008.

[13] C.N. Scanaill, S. Carew, P. Barralon, N. Noury, D. Lyons, and G.M. Lyons, "A review of approaches to mobility telemonitoring of the elderly in their living environment," Annals of Biomedical Engineering, vol. 34, pp. 547-563, 2006.

[14] H. Pigot, A. Mayers, and S. Giroux, "The intelligent habitat and everyday life activity support," International conference on Simulations in Biomedicine, pp. 507-516, 2003.

[15] J. Bravo, D. Lopez-de-Ipina, C. Fuentes, R. Hervas, R. Pena, M. Vergara, and G. Casero, "Enabling NFC technology for supporting chronic diseases: a proposal for Alzheimer caregivers," Ambient Intelligent, vol. 5355, pp. 109-125, 2008.

[16] W.D. Bradford, A. Kleit, M.A. Krousel-Wood, and R.M. RE, "Comparing willingness to pay for telemedicine across a chronic heart failure and hypertension population," Telemedicine and e-Helath, vol. 11, pp. 430-438, 2005.

[17] S.J. Dijkstra, J.A. Jurrlis, and R.D. van der Mei, "A business model for telemonitoring services," in Proceedings High Technology Small Firms Workshop, University of Twente, 2006.

[18] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman, "Activity Recognition from accelerometer data," Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference, pp. 11-18, 2005.

[19] M. Sekine, M. Akay, T. Tamura, Y. Higashi, and T. Fujimoto, "Fractal dynamics of body motion in patients with Parkinson's disease," J. of Neural Engineering, vol. 1, pp. 8-15, 2004.

[20] M.F. Alhamid, A.A. Alamri, A. El Saddik. Measuring Hand-Arm Steadiness for Post-Stroke and Parkinson's Disease Patients Using SIERRA Framework. Medical Measurements and Analysis (MeMeA), Canada, 2010.

[21] P.L. Walter, "The history of the accelerometer," Sound and vibration, pp. 16-22, 1997.

[22] A. Godfrey, R. Conway, D. Meagher, and G. OLaighin, "Direct measurement of human movement by accelerometer," Medical engineering and physics, vol. 30, pp. 1364-86, 2008.

[23] T. Tamura, T. Fujimoto, H. Sakaki, Y. Higashi, T. Yoshida, and T. Togawa, "A solid-state ambulatory physical activity monitoring and its application to measuring daily activity of the elderly," Journal of Medical Engineering & Technology, vol. 21, pp. 96-105, 1997.

[24] B.G. Celler, T. Hesketh, W. Earnshaw, and E. Ilsar, "An instrumentation system for the remote monitoring of changes in functional health status of the elderly at home," International Conference of the IEEE EMBS, vol. 2, pp. 908-909, 1994.

[25] A. Quigley, M. McGrath, P. Nixon, and T. Dishongh, "Home deployments for independent living," Pervasive Computing @ Home Workshop, 2008.

[26] S. Giroux, J. Bauchet, H. Pigot, D. Lussier-Desrochers, and Y. Lachappelle, "Pervasive behavior tracking for cognitive assistance," International Conference on Pervasive Technologies Related to Assistive Environments, PERTA, 2008.

[27] N. Oliver and F. Flores-Mangas, “HealthGear: A real-time wearable system for monitoring and analyzing physiological signals,” Micorsoft Corporation, Redmond, Wass, USA, 2005.

[28] SHIMMER, http://shimmer-research.com. Accessed October 2009.

[29] P. Kuryloski, A. Giani, R. Giannantonio, K. Gilani, V. Seppa, E. Seto, R. Gravina, V. Shia, C. Wang, P. Yan, A.Y. Yang, J. Hyttinen, S. Sastry, S. Wicker, and R. Bajcsy, “DexterNet: An open platform for heterogeneous body sensor networks and its applications,” Tech Report UCB/EECS-2008-174, EECS Department, University of California, Berkeley, 2008.

[30] SPINE, http://spine.tilab.com. Accessed February 2010.

[31] R.K. Ganti, P. Jayachandran, T.F. Abdelzaher, and J.A. Stankovic, “SATIRE: a software architecture for smart AtTIRE,” MobiSys, Sweden, 2006.

[32] A. Salmeri, C.A. Licciardi, L. Lamorte, et al., "An Architecture to combine context awareness and body sensor networks for health care applications," ICOST 2009, LNCS 5597, pp. 90-97, 2009.

[33] CodeBlue, http://fiji.eecs.harvard.edu/CodeBlue . Accessed January 2010.

[34] D.M. karantonis, M.R. Narayanan, M.Mathie, N.H. Lovell, and B.G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," IEEE Transaction on Information Technology in Biomedicine, vol. 10(1), 2006.

[35] C. Otto, A. Milenkovic, C. Sanders, E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," Journal of Mobile Multimedia, vol. 1(4), pp. 307-326, 2006.

[36] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, R. Stoleru,“Context-aware wireless sensor network for assisted living and residential monitoring,” IEEE Network, 2008.

[37] M.P. Rajasekaran, S. Radhakrishnan, and P. Subbaraj, "Elderly patient monitoring system using a wireless sensor network," Telemedicine and e-Health, vol. 15, pp. 73-79, 2009.

[38] C. Lombriser, N.B. Bharatula, and D. Roggen, "On-body recognition in Dynamic sensor network," Proceedings of the ICST 2nd international conference on Body area networks, 2007.

[39] V. Shnayder, B. Chen, K. Lorincz, T.R.F. Fulford-Jones, M. Welsh, “Sensor Networks for Medical Care,” Technical Report TR-08-05, Division of Engineering and Applied Sciences, Harvard University, 2005.

[40] K. Lorincz, B. Chen, G.W. Challen, A.R. Chowhury, S. Patel, P. Bonato, and M. Welsh, “Mercury: a wearable sensor network platform for high-fidelity motion analysis,” SenSys, 2009.

[41] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J.H. Growdon, M. Welsh, P. Bonato, "Analysis of feature space for monitoring persons with Parkinson's disease with

application to a wireless wearable sensor system," Precceding of the 29th Annual International Conference of the IEEE EMBS, France, 2007.

[42] TinyOS, http://www.tinyos.net. Accessed April 2010.

[43] EyesWeb Project, http://www.infomus.org/EywIndex.html . Accessed January 2010.

[44] BioMOBIUS, http://www.biomobius.org . Accessed February 2010.

[45] J.B.J. Bussmann, W.L.J. Martens, J.H.M. Tulen, and F.C. Schasfoort et al. , "Measuring daily behavior using ambulatory accelerometry: The Activity Monitoring," Behavior Research Methods, Instruments, & computers, vol. 33(3), pp. 349-356, 2001.

[46] F. Foerster and J. Fahrenberg, "Motion Pattern and posture: correctly assessed by calibrated accelerometers," Behavior Research Methods, Instruments, & Computers, vol. 32(3), pp. 450-7, 2000.

[47] M. Makikawa, S. Kurata, Y. Higa, Y. Araki, R. Toku, "Ambulatory Monitoring of behavior in daily life by accelerometer set at both-near-sides of the joint," In Proceedings of Medinfo, vol. 10(1), pp. 840-3, 2001.

[48] J.H.M. Tulen, D. L. Stronks, J.B.J. Bussman, L. Pepplinkhuizen, J. Passchier, "Towards an objective quantitative assessment of daily functioning in migraine: a feasibility study," Pain, vol. 86, pp. 139-149, 2000.

[49] N.P. Bidiargaddi and A. Sarela, "Activity and heart rate based measures for outpatient cardiac rehabilitation," Methods of Information in Medicine, vol. 47(3), pp. 208-216, 2008.

[50] C.C. Holmes and N.M. Adams, "A probabilistic nearest neighbour method for statistical pattern recognition," Journal Royal Statistical Society B, vol. 64, pp. 1-12, 2002.

[51] K.P. Murphy, http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html . Accessed May 2010.

[52] SQLite, an SQL database engine, http://www.sqlite.org . Accessed March 2010.

[53] SIMILE Widgets, http://www.simile-widgets.org . Accessed March 2010.

[54] G.G. Enas and S.C. Choi, "Choice of the smoothing parameter and efficiency of k-nearest neighbour classification," Comput. Math. Applic. A, vol. 12, pp. 235-244, 1986.

[55] M. Wetzler, J.R. Borderies, O. Bigaignon, P. Guillo, and P. Gosse, "Validation of a two-axis accelerometer for monitoring patient activity during blood pressure or ECG hotler monitoring," Clinical and Pathological Studies, 2003.

[56] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," Machine Learning, vol. 29, pp. 131-163, 1997.

[57] Weka: a Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka. Accessed May 2010.

Appendix A: A Matlab code to train our Bayesian Network

```
% Creating a Bayesian Network for 5 types of activities

N = 5;
% we have 5 nodes: the Real Activity, SensorHead, SensorLower-
back, Sensor Forearm, Sensor Thigh
dag = zeros(N,N);
R = 1; H = 2; L = 3; F = 4; T = 5;
% R: RealActivity, H: SensorHead, L: Lowerback, F: Forearm, T:
Thigh
% only one level of association
dag(R,[H L F T]) = 1;
% creating the association between the Real activity and the
four sensors

% labeling false and true values
false = 1; true = 2;

ns = 5*ones(1,N); % 5 possibility for the activity detection

bnet = mk_bnet(dag, ns);
% creating the Bayesian Net

bnet.CPD{R} = tabular_CPD(bnet, R, [0.2 0.2 0.2 0.2 0.2]);
% the probability of each activity = 1/5 = 0.2
bnet.CPD{H} = tabular_CPD(bnet, H, [1 0 0 0 0 0 1 0 0 0 0 0 1
0 0 0 0 0 1 0 0 0 0 0 1]);
bnet.CPD{L} = tabular_CPD(bnet, L, [1 0 0 0 0 0 1 0 0 0 0 0 1
0 0 0 0 0 1 0 0 0 0 0 1]);
bnet.CPD{F} = tabular_CPD(bnet, F, [1 0 0 0 0 0 1 0 0 0 0 0 1
0 0 0 0 0 1 0 0 0 0 0 1]);
bnet.CPD{T} = tabular_CPD(bnet, T, [1 0 0 0 0 0 1 0 0 0 0 0 1
0 0 0 0 0 1 0 0 0 0 0 1]);

CPT = cell(1,N);
for i=1:N
  s=struct(bnet.CPD{i});
  CPT{i}=s.CPT;
end
```

```
% Generate training data
data = load('C:\datasamples.txt');
% note: we need to flip-flop the column and the rows later own
nsamples = size(data, 1);
% getting the number of cases available in the data set
samples = num2cell(data');
data = data';

bnet4 = learn_params(bnet, samples);

CPT4 = cell(1,N);
for i=1:N
  s=struct(bnet4.CPD{i});
  CPT4{i}=s.CPT;
end
```

Appendix B: A Matlab code to compute the posterior probability of our Bayesian Network

```
% INFERENCE
engine = jtree_inf_engine(bnet4);
ev = cell(1,N);
ev{T} = 1;
engine = enter_evidence(engine, ev);
m = marginal_nodes(engine, R);
fprintf('P(R=faldown|H=falldown) = %5.3f\n', m.T(1)); % the
probabiklity that it's a falldonw giving that head classified
it as falldown
fprintf('P(R=lying|H=falldown) = %5.3f\n', m.T(2));
fprintf('P(R=sitting|H=falldown) = %5.3f\n', m.T(3));
fprintf('P(R=standing|H=falldown) = %5.3f\n', m.T(4));
fprintf('P(R=walking|H=falldown) = %5.3f\n', m.T(5));

% SIMILARIY TO THE OTHER 4 ACTIVITIES
```