Context-aware Real-time Assistant Architecture For Pervasive Healthcare

Bingchuan Yuan

Department of Computer Science

University College Cork, Ireland

A thesis submitted for the degree of

Doctor of Philosophy

26th June 2014

Declaration

This dissertation is submitted to University College Cork, in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Science. I declare that this thesis entitled *Context-aware Real-time Assistant Architecture For Pervasive Healthcare* is the result of my own research except as cited in references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Name: Bingchuan Yuan
Date: June 2014

Acknowledgements

I would like to acknowledge the genuine support and motivation provided by my supervisor Dr. John Herbert. His continual guidance throughout my research studies in UCC has been of great assistance. I really appreciate it.

I would like to acknowledge the hardware/firmware support provided by Ivan Stancl (Electrical Engineer, Czech Republic) and Philip Angove (Tyndall National Institute). This enabled the design and development of the Wireless Sensor Network, which played a major part in the evaluation of the CARA architecture.

I would also like to thank my colleagues Dapeng Dong and Mohammad Hashemi for their helpfulness in providing brilliant advices and technical supports throughout the development of the CARA architecture.

A special thank you to Guanghui Li, Shuo Tian, Chao Li, Nathaniel Simcox and Yalda Emamian (Taught M.Sc. students, Department of Computer Science, UCC) who conducted research projects that related to aspects of the CARA project as part of their studies.

I am extremely grateful for the financial support provided by the Department of Computer Science, UCC and Science Foundation Ireland (SFI).

Finally, I appreciate the continuous support and encouragement of my family.

Dedication

I would like to dedicate this thesis to my loving parents and my wife Hui, for their love, continuous support and encouragement.

Abstract

The aging population profile in many countries brings into focus rising healthcare costs and pressure on conventional healthcare services. Pervasive healthcare has emerged as a viable solution capable of providing a technology-driven approach to alleviate such problems by allowing healthcare to move from the hospital-centred care to self-care, mobile care, and at-home care. The state-of-the-art studies in this field, however, lack a systematic approach for providing comprehensive pervasive healthcare solutions from data collection to data interpretation and from data analysis to data delivery.

In this thesis we introduce a Context-aware Real-time Assistant (CARA) architecture that integrates novel approaches with state-of-the-art technology solutions to provide a full-scale pervasive healthcare solution with the emphasis on context awareness to help maintaining the well-being of elderly people. CARA collects information about and around the individual in a home environment, and enables accurately recognition and continuously monitoring activities of daily living. It employs an innovative reasoning engine to provide accurate real-time interpretation of the context and current situation assessment. Being mindful of the use of the system for sensitive personal applications, CARA includes several mechanisms to make the sophisticated intelligent components as transparent and accountable as possible, it also includes a novel cloud-based component for more effective data analysis. To deliver the automated real-time services, CARA supports interactive video and medical sensor based remote consultation.

Our proposal has been validated in three application domains that are rich in pervasive contexts and real-time scenarios: (i) Mobile-based Activity Recognition, (ii) Intelligent Healthcare Decision Support Systems and (iii) Home-based Remote Monitoring Systems.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

The current ageing of the population in most European countries is exerting pressure on the healthcare systems in various ways: more chronic medical conditions, increasing health costs, and the need for long-term care and assistance of elderly people [Codagnone, 2009]. Driven by the quality of life and cost issues, there is a need to change the current healthcare system from a hospital focused setting to a home centred setting and from managing illness to maintaining wellness. Innovative pervasive technology [Varshney, 2003] has been identified as a promising solution for facilitating this transformation. Research on pervasive computing technologies for healthcare does not aim to replace traditional healthcare, but rather is directed towards paving the way for a user-centred preventive healthcare model [Arnrich et al., 2010].

Pervasive healthcare can be viewed from two perspectives: i) as the application of pervasive computing technologies for healthcare, and ii) as making healthcare available anywhere, anytime and to anyone [Fortino et al., 2014]. Pervasive healthcare has evolved based on biomedical engineering (BE), medical informatics (MI), and pervasive computing (PerCom). BE combines engineering principles with medical and biological sciences to advance healthcare treatment. MI deals with large medical resources to optimize the acquisition, storage, retrieval, and use of information in healthcare. PerCom designs, develops and evaluates the use of the new paradigm of healthcare systems deployed on a large scale throughout everyday life. While each one of these technologies can be used to improve the existing health delivery model, pervasive healthcare in contrast tries to change the healthcare delivery model: "**from doctor-centric to patient-centric, from acute reactive to continuous preventive, from sampling to monitoring**" [Bardram, 2008]. Overall, pervasive healthcare have been widely recognized as promising solutions for providing in-home healthcare services to the elderly, in particular those suffering from chronic illness, as well as for reducing long-term healthcare costs and improving the quality of care [Alwan and Nobel, 2008]. These advantages have motivated us to research and develop an innovative software architecture for pervasive healthcare.

In designing an intelligent software infrastructure for pervasive healthcare, three specific application domains are targeted: *Activity of Daily Living Analysis*, *Context-aware Reasoning* and *Remote Monitoring*. Focusing attention on real-world life-critical problems results in a number of important technical goals such as the use of various contextual information, the need for real-time intelligent analysis of time-critical data and the delivery of pervasive healthcare services. The overall objective of producing an effective solution will be evaluated by performing two in-depth case studies for activity recognition and anomaly detection within a pervasive home environment. One of these will focus on tracking activities of patients using smartphones, and the second will be concerned with independent in-home monitoring of patients through a Wireless Sensor Network (WSN).

In this dissertation, the Context-aware Real-time Assistant (CARA) system is presented which is designed to provide personalized healthcare services for the elderly in a timely and appropriate manner by adapting the healthcare technology to seamlessly integrate in normal activities of the elderly and working practices of the caregivers. It addresses the need for a new generation of pervasive healthcare systems that allows early detection of anomalies and health problems for elderly people by identifying behaviour and physiological changes over time. The incorporation of small, low-cost, low-intrusion sensors, including Body Area Networks (portable electronic devices and smartphone sensors capable of monitoring and communicating patient vital signs) and Wireless Sensor

Networks (such as smart home sensors capable of monitoring the home environment and detecting behavioural patterns of the patient), with the intelligent reasoning framework provides a real-time monitoring and context-aware analysis capability that should lead to better medical diagnosis and better patient quality of life. Moreover, comprehensive and efficient real-time remote monitoring will contribute to autonomous healthcare and less hospitalization.

1.2 Challenges and Solutions

While pervasive healthcare offers new possibilities, there are many technical challenges to be overcome before the vision of pervasive healthcare can be realized. First, most existing solutions focus on developing individual techniques that lead to segmented solutions and poor interoperability [Alwan and Nobel, 2008]. A comprehensive pervasive healthcare system requires an appropriate infrastructure that integrates all enabling technologies for information acquisition, management, analysis and delivery. Second, it is essential for environments that aim at providing pervasive healthcare services to have some sort of Ambient intelligence (AmI). AmI refers to electronic environments that are capable of recognizing and responding to the presence of individuals in a seamless and unobtrusive way [Ducatel et al., 2010]. AmI as a technological paradigm has the potential to make a significant impact upon everyday human life by building an environment where devices work in concert to support people in carrying out their everyday life activities, tasks and routines in an easy, natural way. Third, contextual information is the key to enable pervasive healthcare systems to behave and adapt intelligently [Couto et al., 2012]. Interpreting and managing such context in a reasoning infrastructure is essential for providing an efficient healthcare solution for the elderly, their family members, and caregivers. Last but not least, a pervasive healthcare system should have a wide usability [Pung et al., 2009]. It must provide interactive user interfaces and easy to use mechanisms for elderly patients and caregivers, and also provide appropriate data analysis models for the efficient management of healthcare data by service providers.

This dissertation addresses these and other problems: it aims to develop the infrastructure to provide a previously unavailable solution [Chiang et al., 2013;

Sain et al., 2010] that can improve healthcare and the patient quality of life through the combination of intelligent software and a smart home environment. The following characteristics of pervasive healthcare have been considered during the design and development of the CARA architecture:

- **Real-time Data Processing:** This is required because the real-time sensor data for healthcare sometimes contains time-critical information. Besides, the context is dynamic and its elements will vary with the situation and the availability of sensor readings. Intelligent data fusion to produce the real-time context is needed. The system should be able to merge the various real-time sensor readings, along with other pertinent data such as patient profile and time, deal with all available data sources, and provide, in real-time, a current context.

- **Context Awareness:** Context awareness is an important notion in pervasive computing. The context of a patient's current needs can be derived from the patient's medical history, current time, location and activity, patient's current vital signs, among other pieces of information [Varshney, 2009]. The use of context is imperative because, in an environment with many sources of data (some of which may be unreliable), the context can be used to disambiguate the real critical conditions from false alarms.

- **Healthcare Decision Support:** An essential goal is to incorporate enough intelligence into the system to provide accurate real-time interpretation of the sensor readings. This requires a general reasoning engine suitable for real-time execution of the set of application-specific rules incorporating domain knowledge. The reasoning engine should be flexible and customizable with different rules for distinct scenarios, and be able to analyse the full context of the sensor readings in order to distinguish critical from non-critical situations. Rules representing the domain knowledge should be examinable by stakeholders to ensure the accountability of the reasoning engine. Furthermore, it is desirable for the system to be as sophisticated and adaptable as possible, while also being as transparent as possible for both subject and caregivers by presenting the provenance of the reasoning output.

- **Remote Monitoring:** Remote monitoring is a technology to enable monitoring of patients outside of conventional clinical settings (e.g. in the home), which allows the pervasive healthcare system to remotely and continuously monitor the patient through a wireless sensor network and deliver health related information to a remote caregiver in real-time. Incorporating remote monitoring in chronic disease management can significantly improve an individual's quality of life. It allows patients to maintain independence, reduce hospital visits, and minimize healthcare costs [Liang et al., 2012].

- **Healthcare Data Analysis:** Effectively analyzing various forms of healthcare data over a period of time can predict impending healthcare problems [Jensen et al., 2012]. However, analyzing large amounts of data from complex heterogeneous patient sources is a computationally intensive task. By utilizing a cloud infrastructure, the system is able to infer knowledge from the massive amounts of data and provide personalized care to the patient. A cloud-based solution can also improve the scalability and availability of a pervasive healthcare system.

On the other hand, due to the critical nature of health related work, this thesis does not cover the entire aspect of disease management and prevention. [Mihailidis and Bardram, 2010] defined usage models of pervasive healthcare outside of hospital conditions. These usage models have different requirements in regard of criticalness, user's participation, privacy, usability, etc. This dissertation primarily covers the usage models of Fitness, and partially Risk management, and marginally Chronic disease management.

1.3 Contributions

The advances in technology allow the emergence of pervasive healthcare systems, which can seamlessly integrate to the daily lives of people. Thus, it is now possible to bring healthcare systems outside of traditional medical environments to our everyday lives in the form of wearable, non-invasive sensors to track external and internal aspects of our bodies [Smarr, 2012]. The state-of-the-art studies in this field, however, lack a systematic approach for providing comprehensive pervasive healthcare solutions from data collection to data interpretation and from data analysis to data delivery.

The primary contribution of this research is the design and development of the Context-aware Real-time Assistant architecture for pervasive healthcare (CARA) which involves data collection, data processing, data analysis and data delivery. This is achieved by pervasive sensing, real-time activity recognition, contextual data modelling, context-aware reasoning, interactive remote monitoring and cloud-based data analysis.

The CARA architecture consists of a number of distributed components which are designed and developed to provide an integrated technology-driven solution for pervasive at-home healthcare. Notable strengths of the CARA architecture include:

- the integration of wireless wearable body sensors and smart home sensors for data collection and context acquisition;
- the incorporation of a smartphone-based activity recognition approach that involves identifying a user's activity through the combined use of a threshold-based mechanism and multiple machine learning algorithms;
- the design of a context-aware reasoning framework which makes most use of contextual information about and around a person in the pervasive home environment to achieve early detection and prevention of health problems;
- the implementation of a remote monitoring approach that enables the person within the home environment to be monitored by a remote caregiver in an interactive and efficient manner;
- the development of a cloud-based data analysis solution which provides an efficient means for data sharing and data mining.

In collaboration with the Tyndall National Institute and an external electrical engineer, we developed the **wireless sensor network** (WSN) consisting of wearable medical sensors and smart home sensors for continuously monitoring the patient vital signs and the home environment. The Zephyr BioHarness sensor [Zephyr Inc., 2013] was integrated into the WSN (as a replacement for the

Tyndall medical sensors) to provide more accurate sensing and a more reliable solution. An Android smartphone was introduced as an additional sensing platform for **activity recognition** using a hierarchical classification method and an adaptive classification model. Data fusion was applied to sensor data gathered from the patient and environment, and this provides comprehensive context to a hybrid **reasoning engine** for anomaly detection and home automation based on expert domain knowledge as well as user behavioural patterns. A semantic-based approach is used to examine the rules for inconsistency or possible conflict, and indicate this to the user, and a provenance mechanism is used to explain the outcome of the reasoning system. These support both transparency and accountability in the sophisticated reasoning system. We integrated a **remote monitoring** service into the system which allows the assessment of patient state along with other relevant context to be shared with remote caregivers in real-time through the internet. The functionality of video-conferencing enables real-time interaction between the patient and the remote caregiver while the patient vital signs are being monitored. Finally, a **cloud-based data analytics** framework was developed to exploit the cloud infrastructure to improve the effectiveness of data mining through machine learning techniques, and to overcome the limitation of hardware resources in a thin client. Various aspects of the research have been published, a list of all the included published work can be found in Appendix A: List of Publications.

1.3.1 Possible examples of use of CARA

The use of pervasive computing within a healthcare environment has been shown to increase productivity in assisting medical practitioner during their daily tasks [Bali et al., 2013]. The CARA system is capable of interpreting sensor data into multiple context elements and reasoning with all available knowledge and experience which may assist the medical practitioner in providing an effective continuous service. It can benefit patients and caregivers in different ways.

A possible scenario is to monitor a patient under supervisory circumstances, allowing critical information about existing medical conditions to be checked by the on-site caregiver or a remote medical consultant. This provides an incremental introduction of CARA as a pervasive healthcare system when the wireless sensors are initially introduced to the patient. For example:

- Scenario of On-site Care

  Jacob, a patient at his local clinic, has been complaining of a slight chest pain. After preliminary tests, the doctor can not diagnose anything in particular, but decides to keep him under observation using CARA and also allow remote diagnosis by a specialist. Wireless wearable sensors were introduced and attached to Jacob. This enabled him to be monitored on-site under supervision while his real-time vital signs can be examined by remote medical practitioners for consultancy. After a period of monitoring, the system did not detect any abnormalities of Jacob's current medical condition. So the doctor advised the patient to set up the CARA system in his home and to carry on the monitoring session at home. The system can send notification to relevant practitioners if he needs instant attention.

Another scenario involves remote at-home monitoring and checkup. A patient with chronic disease is monitored remotely and continuously within a smart home environment, the intelligent reasoning engine could detect anomalies based on the daily routine and perceived medical information of the patient,and transmit alert messages to healthcare providers for getting emergency services. The patient could also use the system to communicate with a doctor or GP while his/her physiological indices are measured by wearable sensors at home. Thus reducing the amount of efforts and inconsistency in coming to a hospital. For example:

- Scenario of At-home Care

  Michael lived independently by himself in the suburb area but suffered from chronic heart disease. It was inconvenient for him to see the doctor in the hospital weekly for a routine heart function check. So the medical support team decided to have him monitored in his home environment. Smart home sensors were deployed in Michael's house and wireless medical sensors were worn on his body. Vital signs and ambient information were collected by the system to learn the everyday routine of Michael. The current condition of Michael was assessed and shared with remote caregivers. Every

Tuesday morning, Michael was supposed to meet the doctor through the video-conferencing so that he could be checked up remotely at home.

1.4 Thesis Outline

The following chapters are included in this dissertation:

- Chapter 2 presents background research and related work on pervasive healthcare. The changing demographic of the worldwide aging population and its impact on healthcare systems around the world is reviewed. The role of technology in supporting independent living in the home is then examined. Technologies that support developing pervasive healthcare systems is presented, and the state-of-the-art of existing pervasive healthcare approaches is reviewed.

- Chapter 3 introduces CARA in a general way, including an overview of the architecture and the description of each component.

- Chapter 4 is a description of the development of the wireless sensor network (WSN) in the CARA architecture. The WSN supports CARA with physiological contexts, environmental contexts and activity contexts. It is used to sense a user's vital signs, surroundings, and activities. To achieve this, a Body Area Network (BAN) is used to measure the vital signs; smart home sensors are deployed to monitor the surroundings in home environment; and a smartphone is used to identify the activities associated with the BAN.

- Chapter 5 introduces the methodology of smartphone-based real-time activity recognition. The ability to accurately recognize and continuously monitor activities of daily living (ADLs) is one of the key features that CARA is expected to provide. The solution involves identifying a user's activity through the combined use of inertial sensors(accelerometer and gyroscopes) built into the smartphone along with the wearable wireless sensors (e.g. Zephyr BioHarness). A hybrid classifier is developed by combining threshold-based methods and machine learning mechanisms. The adaptive machine learning mechanism makes the model customizable and adaptable.

Incorporating a cloud infrastructure overcomes the limitations of computing power and storage of a smartphone. The activity recognition application is used as a case study for evaluating the cloud-based data analytics framework.

- Chapter 6 discusses the design and development of a personalized and extensible context-aware reasoning framework in the CARA architecture. This provides a novel approach that combines context awareness, general domain knowledge, and automated intelligence for pervasive healthcare. It plays a crucial role in CARA by interpreting sensor data within a wide context, reasoning with all available knowledge for situation assessment, and reacting according to the reasoning outputs. The incorporation of rule-based and case-based reasoning mechanisms enables the system to become more robust and adaptive to a changing environment. Furthermore, a semantic-based analysis is included for detecting any conflict in rules, and a provenance recording mechanism is applied to case-based reasoning for the explanation of the reasoning conclusion to the user.

- Chapter 7 describes scenarios of using CARA in a visualized interactive manner where continuous monitoring of the patient and home environment is carried out in a non-intrusive way via the wireless sensor network. In addition, a telecare function provides interaction between the patient and remote caregiver through real-time video communication. The remote monitoring solution is integrated into the system and can make use of CARA's intelligent analysis as well as its recording and playback facilities. This interactive user friendly approach provides an introduction to the technology for an elderly person in advance of deploying a more automated solution.

- Chapter 8 gives conclusions about the presented CARA architecture and provides an overview of future development of pervasive healthcare using the CARA architecture.

Chapter 2

Background Research

2.1 Introduction

In recent decades, developed countries have experienced an increase of average life-span with a consequent impact of chronic conditions on the population. Pervasive and context-aware healthcare applications have been widely recognized as promising solutions for improving quality of life of both patients suffering from chronic conditions and of their caregivers, as well as for reducing long-term healthcare costs and improving quality of healthcare services.

In this chapter, the changing demographic of the worldwide aging population and its impact on the healthcare systems of countries around the world is reviewed. The role of technology in supporting independent living in the home is then examined, with an emphasis on the potential value of moving routine monitoring and care from hospitals into the home. This is followed by introducing the pervasive computing paradigm for healthcare and reviewing the state-of-the-art of existing approaches.

2.2 Pervasive Healthcare Evolution

2.2.1 Aging Society

The increasing elderly population worldwide brings a need for more healthcare options. In general, senior citizens are more vulnerable to chronic diseases such as heart disease, cancer and Alzheimer's requiring medical care, than the rest of the population. This places enormous demands on healthcare systems, not only in terms of acute hospital care but also for routine monitoring and health maintenance.

174 million people in Europe and North America are aged 65 years or older. This is about 40 million people more than 20 years ago and a further increase of about 93 million people is expected within the next 20 years, which clearly illustrates that population ageing is accelerating. Figure 2.1 shows the most significant demographic changes within the EU candidate and potential candidate countries. Although this group of countries will still be among those with the lowest share of people aged 80 and older in the total population, the number of people aged 80 and older will almost double within the next 20 years [Rodrigues et al., 2012]. This increase will have a direct impact on healthcare institutions, specifically considering the fact that the healthcare costs per capita for persons over 65 years are three to five times greater in comparison with the healthcare costs of people under 65.

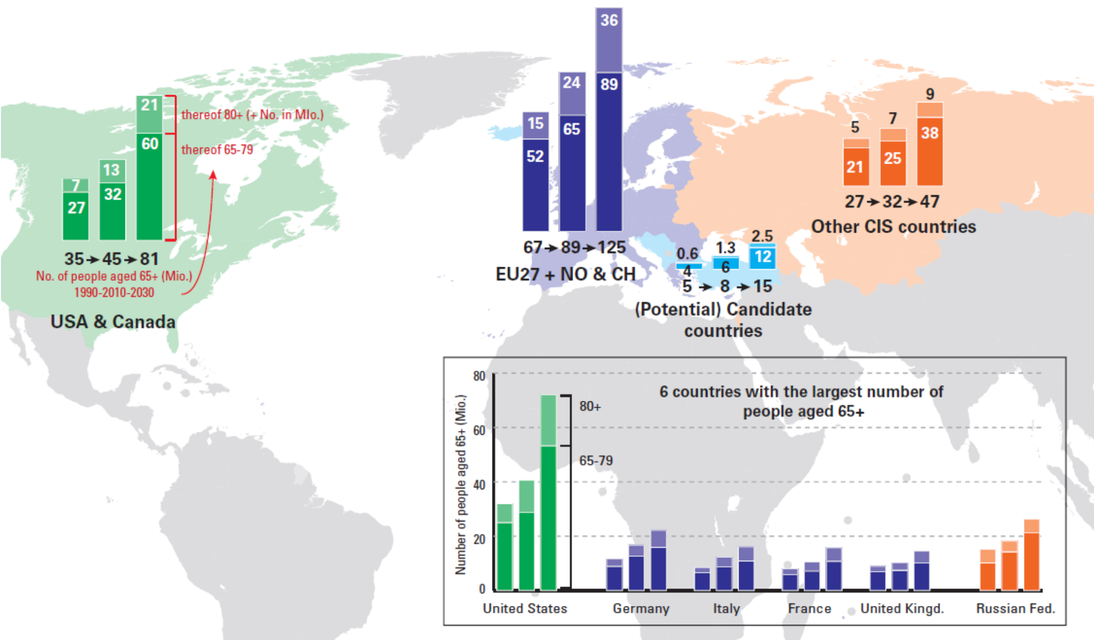


Figure 2.1: Evolution of the population in the older age groups 1990, 2010 and 2030 (Source: UNPP 2011: World Population Prospects)

Some 861 million people worldwide with chronic diseases incur up to 85% of healthcare expenditure. This includes a large amount of care for the growing elderly population. Figure 2.2 provides information on the public resources devoted to long-term care in a comparable way. It indicates a large share of public expenditure is spent on institutional care in most countries. Clearly, traditional healthcare systems will not work with the increasing demands. There is a pressing need to explore a different way of caring for a rapidly growing population of elderly while reducing healthcare costs. Andy Grove, former CEO of Intel, in an interview in Fortune magazine described the healthcare situation as follows: "Given the high cost of institutional care, helping older people to live independent lives in their own home must be a priority for healthcare systems" [Schlender, 2003].

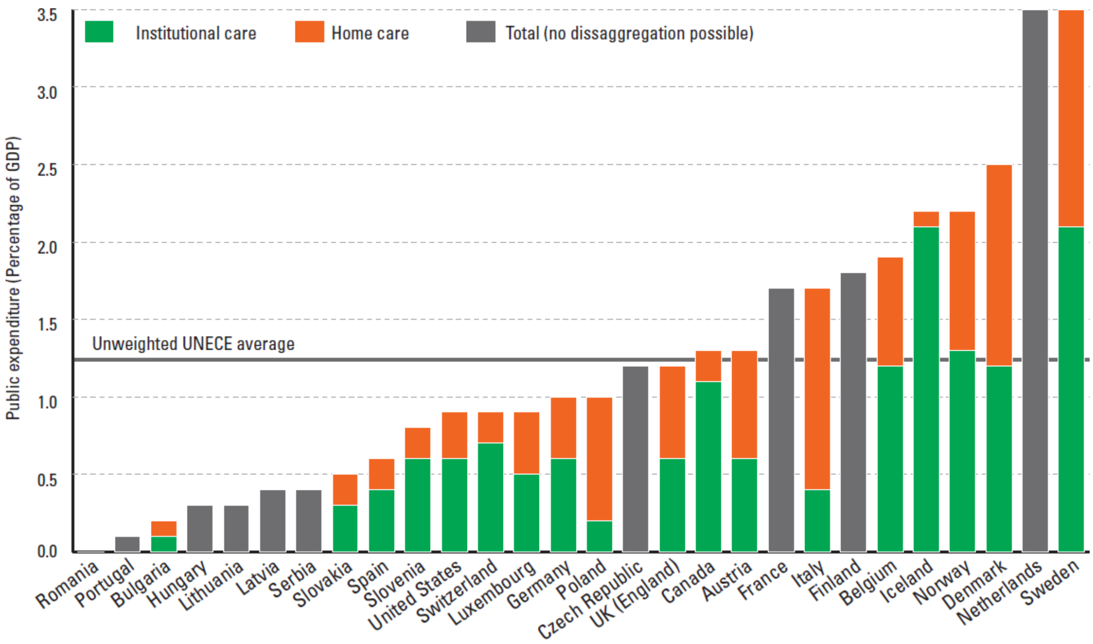


Figure 2.2: Public expenditure on long-term care, by care setting 2009 or latest available year (Source: OECD Health Data 2011).

2.2.2 Potential of Technology

Fortunately advances in technology can assist people in maintaining and monitoring their own health. Advanced sensing and communication technologies have started to converge, which provides the opportunity for healthcare solutions that may become both ambient and pervasive [Black et al., 2011].

At-home healthcare

Pervasive computing has the potential to revolutionize healthcare in the home. At-home healthcare can help address the social and financial burdens of an ageing population. At the same time, technology can support the network of caregivers such as family members, and clinicians with new and innovative ways to monitor the wellbeing of older people, and to enable rapid response to emergency situations [Dishongh and McGrath, 2010]. The contemporary social reality is that many family members will be geographically located away from an elderly relative, yet there is a compelling need for them to play an active part in the care giving duties to an elderly parent or relative. Technological advances will result in time savings and reduce travel overheads for caregivers and relatives.

At-home healthcare systems will enable people to monitor themselves with devices that give proactive warnings of illness so that they can turn to their doctors earlier, when intervention can be the most effective [McCullagh and Augusto, 2011]. As devices are enabled with processing capacities and the ability to communicate autonomously, there is the potential for body wearable and environmental sensors to work collaboratively with healthcare systems by collecting, processing and exchanging data and information. Appropriate information or alerts can be fed back directly to a person using a home PC or an intelligent mobile device. In addition, data and information can be stored in a central facility (a home computer, or possibly a remote server or a cloud server) to provide a longer term view, or possibly to contribute to the compilation of population-based statistics.

William Herman, director of the division of physical sciences in the Food and Drug Administration's Center for Devices and Radiological Health (CDRH), which regulates medical devices in the United States, calls home-care systems "**the fastest growing segment of the medical device industry**" [Lewis, 2009]. The at-home healthcare model is not designed to replace the traditional acute care role of hospitals and clinicians; instead, it enables the elderly person to be an active participant in their own healthcare, particularly in the daily routine maintenance and monitoring of health. Eventually, the home can become an equally important location for healthcare as the hospital.

Challenge

Although deployment of technology in support of at-home care has the potential to radically reduce the pressure on hospital resources, it remains a significant challenge since many of the required technology solutions do not yet exist or are still in early prototyping stages. It is important to note that the use of such technologies on a regular basis is required so that relevant trends or deviations can be identified. However, the major challenge is the scale of deployment of such systems [Intille, 2013]. The initiatives are run on a small scale by innovation players and researchers, making their wider deployment difficult. While there are some successful institutional programmes incorporating ICTs (information and communication technologies)[Corporation, 2013; McKinstry, 2009], and a few larger scale commercial players [MayoClinic, 2014; OPNET, 2014], many services are still developed and provided by rather small scale organisations, often start ups or research laboratories. The fragmentation of care services is a barrier to the entry of these new organisations to the market [Carretero et al., 2012].

2.3 State of the Art and Related Work

2.3.1 Wireless Sensor Networks in Healthcare

In Wireless Sensor Networks (WSNs), a number of tiny, battery-powered computing devices are scattered throughout a physical environment. Each device is capable of sensing, and transmitting information. It is a packaged data collection and transmission component, which consists of a sensor module, an embedded processor, a transceiver module, and a power delivery mechanism. Components are held within an enclosure.

A sensor board is the part that actually interacts with the environment and sends an appropriate signal to the embedded processor (microcontroller unit). The Shimmer ECG board is an example [Shimmer, 2013]. The microcontroller unit may decide to forward the sensed signal to a base station, or to do some processing locally. When ready, the processor sends the signal to the transceiver board which contains the radio stack and antenna; the transceiver then uses a communications protocol (e.g. IEEE 802.15.4, Bluetooth, ZigBee) to pass the information to the base station.

WSNs have been used in commercial, industrial, and academic applications to monitor data that would be difficult or expensive to be captured using wired sensors. A variety of applications have been presented in the literature for wireless sensor networks. These include environmental monitoring [Oliveira and Rodrigues, 2011], building monitoring [Yoon et al., 2011], natural disaster prevention [Chen et al., 2013], and structural health monitoring [Hu et al., 2013]. Recently, researchers have investigated the potential of developing WSNs for healthcare systems [Aminian and Naji, 2013; Caldeira et al., 2012; Peiris, 2013]. What differentiates healthcare from other WSN technology applications is the criticality of the application and the human centred aspect. To design a WSN for the solution of healthcare problems, the following key principles should be kept in mind throughout the process:

- This is a healthcare problem, not a technology problem. The patient plays the decisive role, not the technology.
- There is often more than one approach to achieve a clinical or care objective.
- The simpler the technology, the better the solution.
- It has to work in the home, not just in the lab.

Concerning the ability of the WSN to function properly and the impact on the safety of patients and caregivers, a range of new factors must be considered, not only technological but also user-centred and healthcare associated, at all stages of designing a WSN architecture for healthcare solutions. The most essential requirements include: *Cost Efficiency, Fault Tolerance, Stability and Scalability,* *Low Power Consumption, Long-term Usability, Privacy and Security.* However, the interpretation and implementation of these guidelines totally depends on the particular design of each healthcare application. In some cases, not all the points above will be relevant.

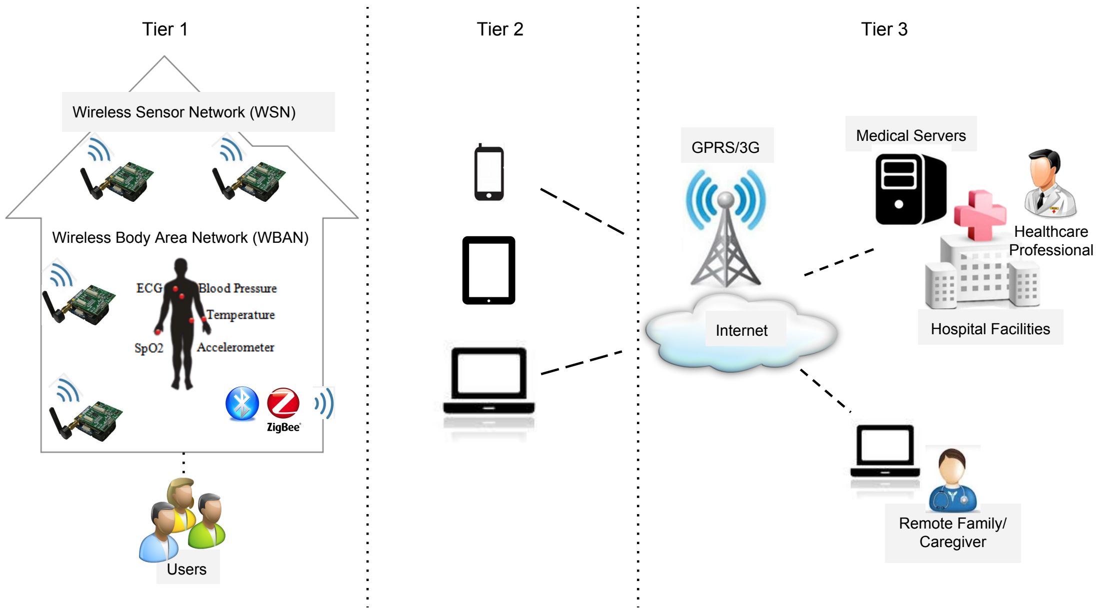


Figure 2.3: Generalized WSN architecture for pervasive healthcare

Figure 2.3 shows a generalized WSN architecture for pervasive healthcare. While this does not illustrate any specific system, it shows the components of a pervasive healthcare system and their relationships with one another. At the core of the system is the user, also referred to as the "subject" (in a research environment) and as the "patient" (in a clinical or therapeutic environment). The user is monitored by wireless sensor networks. This is referred to as Tier 1. The information gathered by the components of the WSN is sent to a base station, or home gateway (often a PC or a smart phone) for data processing and analysis. This is referred to as Tier 2. The communication links used between the WSN and the home gateway will vary according to circumstances (e.g. ZigBee, Bluetooth, WiFi). The home gateway connects over the internet and/or other long range communications protocols to various Tier 3 services. These may include a medical server, a healthcare provider, a family member, a caregiver and emergency services etc. Again, a range of communications protocols are possible here, depending on the requirements of the particular problem domain [Dishongh and McGrath, 2010].

More and more prototype and commercial systems for pervasive healthcare monitoring have been developed for the elderly, and chronically ill people, using the generalized WSN architecture. After exploring these systems, it is observed that the main application areas include:

- Activities of daily living monitoring [Benzo et al., 2014; Charlon et al., 2013; Lu and Fu, 2009].
- Fall and movement detection [Gannot et al., 2013; Schwickert et al., 2013; Wang et al., 2008].
- Location tracking [Lee et al., 2013; Marco et al., 2008; Thomas et al., 2013].
- Medication adherence prompting [Chen et al., 2014; Chu, 2013; Pang et al., 2009].
- Medical status monitoring [Kailanto et al., 2008; Tharion et al., 2013; Triantafyllidis et al., 2013].

In the first category, applications try to identify and differentiate everyday activities of the patients and the elderly such as watching television, sleeping, preparing meals, and be able to detect abnormal conditions. Fall and movement detection applications are focused on physiological conditions such as posture and fall detection for people that need special care like the elderly people who are susceptible to falls. Location tracking, medication intake reminders and monitoring systems can help cognitively impaired people to survive independently. Medical monitoring applications make use of medical and environmental sensors in order to obtain comprehensive health status information of the patients, including ECG, heart rate, blood pressure, skin temperature, and oxygen saturation.

In this dissertation, a comprehensive WSN is developed, incorporating body area network and smart home sensors, to achieve monitoring of a person and surroundings in a home environment. Distinguished from existing works, which mostly focus on a single application area as listed above, the WSN is dedicated to gather as much as possible information about and around a person to cover multiple tasks of a healthcare application.

2.3.2 Activity Monitoring and At-home Care

Activity of Daily Living Recognition

Advances in ubiquitous and pervasive computing have resulted in the development of a number of sensing technologies for capturing information related to human daily activities. Different approaches for lifestyle monitoring in terms of activity recognition have been studied using different underlying sensing mechanisms. The environmental sensor-based approach has received significant attention in recent years. It is a promising approach for recognizing activities which can not be simply distinguished by body movement alone. Additional sensors such as motion sensors, door contact sensors, pressure sensors, object RFID tags and video cameras are required for gathering activity related information [Tapia et al., 2004]. However, it requires the installation of a lot of equipment, and in some cases it is only feasible for use in laboratory settings.

On the other hand, wearable sensors have proved to be an effective and reliable method for human activity recognition. They are small in size, lightweight, low cost and non-invasive. Some of the existing work on wearable sensor based activity recognition utilizes multiple accelerometers placed on different parts of the body[Krishnan et al., 2008; Ravi et al., 2005; Tapia et al., 2007]. [Krishnan et al., 2008] collected data using two accelerometers for recognizing locomotion activities in real-time using adaptive decision trees. Other research has explored the use of multiple kinds of wearable sensors for activity recognition. [Maurer et al., 2006] used accelerometers, temperature sensors and microphones to analyze multiple time domain feature sets for activity recognition. Most of the wearable sensors are required to be fixed onto specific locations on the subject's body, such as the chest, ankle, thigh, wrist or waist. Current body-fixed sensors may be considered to be inconvenient to wear and impractical for continuous long term monitoring in a normal daily living environment. Smartphones have the potential to be an alternative platform for activity recognition with the advantages of unobtrusiveness and not requiring any additional equipment for data collecting or processing.

Recent research has focused on using smartphones as mobile sensing and computing platforms to achieve activity recognition [Khan et al., 2010; Kwapisz et al., 2010; Stefan et al., 2012]. [Kwapisz et al., 2010] utilize a smartphone for recognizing very simple activities such as walking, jogging, climbing up and down stairs, sitting and standing with a threshold based method. More comprehensive approaches have been investigated by leveraging sophisticated machine learning algorithms. [Stefan et al., 2012] use the inertial sensor data of a smartphone to build machine learning models for recognizing both simple and complex activities of daily living at home. A single smartphone was mounted onto the pelvic region of subjects to collect data on different activities, and conventional classifiers were trained using supervised machine learning mechanism. However, the performance of their approach is less effective due to the ambiguity of upper body movement tracking.

In this dissertation, we build on these approaches and extend them to improve the ability of real-time identification of everyday activities. Our approach, using two sets of sensors including one set in a smartphone is, in effect, a combination of previous techniques, and is able to distinguish static and dynamic activities and identify activities utilizing both threshold based and machine learning methods. The performance of activity recognition is significantly improved by this combination.

Telecare Supported At-home Care

The desire to overcome the challenges facing all healthcare systems, including the need to increase the quality of care while decreasing overall healthcare costs, has led to a growing interest in the application of telecare solutions. In particular, telecare supported at-home care, also known as remote patient monitoring, is gaining more and more attraction. The global market for telecare is set to exceed \$1 billion by 2016 and could jump to \$6 billion in 2020, according to a new report by [InMedica, 2011], the leading independent provider of medical electronics market research group.

Telecare is the use of a digital network to provide automated monitoring and treatment delivery to a patient who is in a different physical location than the medical expert providing treatment. It can refer to simple communications like messages sent between patients and providers to extremely complex procedures like remote robotic surgery [Sudan et al., 2011]. It offers the potential for a home based healthcare solution, in which the patient and healthcare professional collaborate on the care plan. The patient collects health related information in their own home, using home monitoring devices. The doctor can then remotely view the data and provide appropriate advice, either at a subsequent appointment or by video communication. Thus, technology can act as a filter, enabling the doctor to attend to urgent cases.

Today, common forms of telecare include home monitoring services and video-conferencing. Home monitoring is becoming increasingly relevant in the treatment of chronic diseases. For instance, home monitoring of blood-pressure allows sufferers of hypertension to manage their condition better and monitor their progress, and diabetics would be able to transmit their blood sugar levels to their caregiver for review, via the internet from the comfort of their home [Wakefield et al., 2012]. This kind of monitoring can improve the quality of life of patients and reduce the number of visits patients need to make to the hospital. Moreover, telecare supported at-home care can extend this paradigm with the use of smart home sensors (such as Passive Infra Red (PIR) devices, bed occupancy sensors, door and window switches, and enabled devices such as cookers, heaters etc.), to build up patterns of behaviour and provide information to the person for self-management, or to a remote caregivers for consultation [Wang et al., 2010]. As for video-conferencing facilities, the prevalence of broadband communication offers tele-consultation with a healthcare professional, using well-known communication utilities such as Skype. Systems with this type of functionality are available from companies such as Doc@Home [Docobo, 2013], Health Buddy [Bosch, 2013], and Intel PHS6000 [Corporation, 2013]. However, such systems are often designed specifically for a particular healthcare purpose. They usually provide standalone functionality with no adaptability to other monitoring systems or devices.

In the dissertation, we propose a general solution supporting an at-home telecare system. The Context-aware Real-time Assistant (CARA) architecture supports multifunctional remote monitoring, which is compatible not only with medical devices but also with smart home sensors and mobile devices, and is more adaptable to the dynamic environment by utilizing the intelligence of a reasoning engine. A live video communication component is also integrated in the system with access from any web browser.

2.3.3 Context Awareness and Context Reasoning

Context-aware Healthcare in a Smart Home Environment

To positively alter the relationship between humans and technology, it is essential for systems that aim at helping people in their daily life to be context-aware [Makris et al., 2013]. Context-aware computing has the potential to make a significant impact upon everyday human life by building an environment that is capable of recognizing and responding to the presence of individuals in a seamless and unobtrusive way [Ducatel et al., 2010]. As one form of many possible realizations of context awareness, smart homes with the deployment of WSNs, BANs and smart devices have been extensively explored over the last decade.

In a smart home environment, hardware, software and networks have to cooperate in an efficient and effective way to provide a suitable result to users. This area has attracted a significant amount of research, and some prototype systems have already been developed and deployed. For example, the Aging in Place project at the University of Missouri aims to provide a long-term care model for seniors in terms of supportive health [Rantz et al., 2011]. Elite care is an assisted living facility equipped with sensors to monitor indicators such as time in bed, bodyweight, and sleep restlessness using various sensors [Adami et al., 2010]. Taking into account the complexity of smart home systems, each research project has focused upon different aspects of such complex architectures. In this dissertation, we explore the integration of the smart home technology with the healthcare system, which can assist the healthcare support system:

- to understand the usual behaviour of the user so that caregivers can supervise and assess the pattern of behaviour and detect anomalies [Pecht and Jaai, 2010] (e.g., by knowing how long and when the subject usually sleeps, the system allows the caregiver to assess their rest).

- to automate activation/deactivation of some devices depending on the needs of the user [Zhang et al., 2011] (e.g. by turning the heater on when the subject goes into the bathroom when the temperature is low).

- to increase safety and to save energy [Zeifman and Roth, 2011; Ziekow et al., 2013] (e.g. by either switching the cooker off or issuing an alarm when detecting that the subject has left it on; by switching the lights off when the subject has gone out).

Essentially, the smart home environment should have intelligence and learning ability to support healthcare systems, and this should be achieved in an unobtrusive and transparent way. Learning means that the environment has to gain knowledge about the preferences, needs and habits of the user in order to better assist the user [Aztiria et al., 2010]. Furthermore, it is insufficient to learn user patterns only once because preferences and routines can change with time. Hence, it is necessary for a smart home environment to adapt itself to these new patterns continuously. Learning and adapting to user patterns is a key feature of the successful development of a healthcare system in a smart home environment. In the CARA architecture, this is achieved by developing a hybrid reasoning framework with the capacity of learning and adaptation.

Context Reasoning

Using reasoning mechanisms to achieve context awareness has been addressed previously [Lum, 2002; Ranganathan and Campbell, 2003; Wallace and Stamou, 2002]. The reasoning component is usually developed to understand behaviours of the user, or combined with other types of knowledge, to make high-level decisions. There are a large number of different context reasoning decision models. Most of the models originated and are employed in the fields of artificial intelligence and machine learning. Therefore, these models are not specific to context-reasoning but commonly used across many different fields in computing and engineering. After studying systems developed by different research groups, we realized that current approaches have very specific applications with focused goals. Therefore, they use the most suitable machine learning technique and only consider the needs of a specific domain. [Bettini et al., 2010; Rashidi and Mihailidis, 2013] summarize the state-of-the-art of context reasoning technique for elderly based on ambient intelligence paradigm in their survey. Our intention here is not to survey context reasoning techniques but to briefly introduce them so it will help to understand

and appreciate the role of context reasoning in pervasive healthcare. Furthermore, we investigate the combination of multiple machine learning techniques to discover patterns of user behaviour for the purpose of activity recognition and anomaly detection.

- Artificial Neural Networks

[Chan et al., 1995] was among the first reports of an application using Artificial Neural Networks (ANNs) for smart home environments in which user behaviour patterns were considered. The aim of their system was to design an adaptive house that considers the lifestyle of the inhabitants to achieve home automation. For that, they used a feed-forward neural network to assess if a given situation was normal or abnormal. After validating this application in an institution for elderly and disabled people, they claimed that 90% of predictions by the system were correct. Other researchers have followed on using ANNs to provide intelligent services in smart home environments. [Badlani and Bhanot, 2011] proposes an adaptive smart home system for optimal utilization of power, through ANN. The system comprises of a recurrent neural network to capture Human behavior patterns and a feed forward architecture in ANN for security applications in the smart homes.

However, there are some limitations of using ANNs regarding their black box nature, in that their internal structure is not human-readable. If we want to examine user patterns in order to understand common behaviour of the user, the system would not be able to represent the learned patterns in a comprehensible and transparent way. In other words, it would not be able to explain how it inferred a particular result. In pervasive healthcare systems where the user plays a fundamental role, the ability to represent patterns and explain the actions carried out by the system is essential.

- Statistical Classification

In machine learning, statistical classification is another approach of identifying to which set of categories a new observation belongs. It is considered as an instance of supervised learning, where a training set of correctly identified observations is available. An algorithm that implements classification, is known as a classifier. It uses a training set to build a statistical model according to a mathematical function that is capable of mapping input data to a category. Typical classifiers for supporting machine learning include: Bayesian Networks, Hidden Markov Models, Decision Trees and Support Vector Machines. They have been widely used for mobile phone sensing [Lane et al., 2010] and activity recognition [Riboni and Bettini, 2009].

According to [Brdiczka et al., 2005], "**a user is only willing to accept an intelligent environment offering services implicitly if he understands and foresees its decisions**". [Stankovski and Tmkoczy, 2006] pointed out that the model generated by decision trees is easy to understand and suitable for human inspection. In their work, they proposed to use a decision tree classifier in order to detect abnormal situations. For that, they trained a decision tree with the collected data that described the normal state of the environment. Thus, any situation external to the tree would be considered as abnormal. However, a system constructed under such a hypothesis can generate some false alarms, because new situations inherent to the environments are not always abnormal.

- Fuzzy Logic

Fuzzy Logic is considered as a collection of methods, tools, and techniques for modelling and reasoning about vague concepts [Biacino, 2002]. In contrast to a "black box" model like neural networks, the fuzzy system works as a "white box" model that is more comprehensible and transparent. It is a rule-based approach that allows approximate reasoning instead of fixed and crisp reasoning. Fuzzy logic is similar to probabilistic reasoning but confidence values represent degrees of membership rather than probability. In traditional logic theory, acceptable truth values are 0 or 1. In fuzzy logic partial truth values are acceptable. It allows real world scenarios to be represented more naturally; as most real world facts are not crisp. It further allows the use of natural language (e.g.temperature: slightly warm, fairly cold) definitions rather than exact numerical values (e.g. temperature: 10 degrees Celsius). In other words it provides a simple way to reach a definite conclusion based upon ambiguous, imprecise, noisy, or missing input information, which is critical in context information processing.

In a fuzzy system, knowledge is represented in terms of fuzzy sets and fuzzy rules, which are considered more robust when dealing with data of a continuous nature (e.g., temperature, humidity and time). Rules are another knowledge representation mechanism often used in reasoning systems. In addition to their human-readable representation, they have the advantage that they are easy to add, modify or delete. As a rule-based approach, fuzzy logic has been successfully applied in knowledge-based reasoning systems. The work of [Vainio et al., 2008] was focussed on developing an application that generated a set of fuzzy rules representing habits of the user in the smart home environment. They manually constructed the membership functions that mapped data into fuzzy sets, and made inference on fuzzy sets according to fuzzy rules. Several examples of applying fuzzy logic to represent context information and to handle uncertainty are presented in [Padovitz et al., 2008].

However, the fact is that the "knowledge acquisition bottleneck" still remains one of the key problems in the design of intelligent and knowledge-based systems. Indeed, experience has shown that a purely knowledge-driven approach, which aims at representing problem related human expert knowledge, is often difficult and tedious.

- Case-based Reasoning

Research conducted under the UT-AGENT project [Kushwaha et al., 2004] used Case-Based Reasoning (CBR), which can be defined as an instance-based learning technique, in order to learn the preferences of the user. The system recorded the set of tasks that the user usually performed, and tried to provide relevant information to the user through a CBR module that matched cases using a K-nearest neighbor algorithm. The principle of CBR is to retrieve former, already solved problems similar to the current one and to attempt to modify their solutions to fit the current problem [Aamodt, 1994]. It has proved an effective machine learning mechanism in various practical domains. For instance, a CBR-based expert system was developed by [Chattopadhyay et al., 2013] with the design of a flexible auto-set tolerance (T), which serves as a threshold to extract cases for which similarities are greater than the assigned value of T, to help doctors diagnose complex diseases particularly those that involve multiple domains in medicine.

However, use of instance-based learning techniques has some limitations. As this process infers a solution for each specific instance, it does not create a model that represents patterns. Therefore, it would not be possible to extract a general pattern indicating the behaviour of users. Furthermore, as each instance can be represented by means of a large number of parameters, the matching process can be very complicated because there are no clues regarding the importance of each parameter in each case.

Table 2.1: Strengths and weaknesses of machine learning techniques

Techniques	**Strength**	**Weakness**
Neural Network	Capacity to generalize complex situations; Possibility of introducing neurons dynamically	Not human readable; Need of restructuring of the whole network for adaptation
Classifier	Human readable; Discovering of conditions; Various algorithms available	Event-Situation relations only; Need to re-build models for adaptation
Fuzzy Logic	Human readable; Robust to uncertainties; Easy to add, delete or change rules	Knowledge acquisition bottleneck; Need of restructuring to avoid conflicts
Instance-based	No need of training, learning from experience; Possible source of information for other techniques	No creation of general patterns; High computational and temporal cost

As discussed above, each technique has pros and cons. To sum up, Table 2.1 highlights the strengths and weaknesses of each technique regarding machine learning characteristics. Analysis of the different techniques reveals that there is still no single approach that provides a global solution to fulfil all requirements of intelligent services. Nevertheless, a combined use of techniques may result in effective problem solving in comparison with each technology used individually and exclusively. Some research projects e.g. Gaia [Román et al., 2002], CDMS [Xue et al., 2008], and HCoM [Ejigu et al., 2007] highlight the importance of employing multiple reasoning techniques such as Bayesian networks, probabilistic and fuzzy logic, where each technique performs well in different situations. Incorporation of multiple modelling and reasoning techniques can mitigate individual weaknesses using each others strengths. COSAR [Riboni and Bettini, 2009] combines statistical reasoning and ontological reasoning techniques to achieve more accurate results.

Some existing reasoning approaches, such as semantics based ontology mechanisms and rule based expert systems, have the potential to support pervasive healthcare systems. However, such systems require as much domain knowledge as possible in advance to produce better reasoning results. Other machine learning approaches like neural networks and Bayesian networks, although they are capable of gaining understanding about a problem domain and providing reasoning functions after learning, take great effort to train. We are not dismissing these existing methods and their capabilities in this dissertation. We are trying to improve them by adapting the rule-based mechanism with learning ability. This makes our reasoning system more adaptive and more intelligent.

In this dissertation, we propose a promising approach to achieve pervasive healthcare by combining a classification model, fuzzy rule-based reasoning and case-based reasoning mechanisms into a reasoning framework that allows the CARA system to learn and understand patterns of an inhabitant's daily living, reason with knowledge and experience to make decisions based upon observations, adapt to a changing environment while recording data for further analysis. In addition, being aware of CROCO [Pietschmann et al., 2008], in which validation (e.g. consistency), conflict resolution, and clarity concerns are given attention where they are rarely addressed by many other solutions, we develop a semantic-based approach to detect conflicts between rules and a provenance-based mechanism to enables the user to have a better understanding of the reasoning result.

2.3.4 Cloud Computing

Cloud Assisted Pervasive Healthcare

Like many other fields, the healthcare industry is looking to cloud computing as a means to improve the quality of service and the efficiency of operations while reducing costs. With cloud computing, the limitations of traditional healthcare systems could be minimized(e.g. small physical storage, less processing power, and high cost of distributed computing [Kopec et al., 2003]). Also through the process of cloud-based data mining, the system can do fast analysis of information taken from the patient. Therefore, the user can have a more efficient service.

As presented by [Ahuja et al., 2012], various cloud assisted healthcare systems have been designed and deployed for the pervasive healthcare system:

- Comprehensive health monitoring services enable patients to be monitored at any time and anywhere through broadband wireless communications.
- An intelligent emergency management system can manage and coordinate a fleet of emergency teams effectively and quickly when receiving calls from accidents or incidents.
- Health-aware mobile devices detect pulse-rate, blood pressure, and level of alcohol to alert the healthcare emergency system.
- Pervasive access to healthcare information allows patients or healthcare providers to access current and past medical information.

As a practical example, a telemedicine home-care management system [Tang et al., 2010] is implemented in Taiwan to monitor participants, especially for patients with hypertension and diabetes. The system monitors 300 participants and stores more than 4736 records of blood pressure and sugar measurement data on the cloud. When a participant performs blood glucose and blood pressure measurement via specialized equipment, the equipment can send the measured parameters to the cloud system automatically. After that, the cloud system will gather and analyze the information about the participant and return results. The cloud-based system provides a useful healthcare service for participants. However, the information to be collected and managed relating to personal health is sensitive. Therefore, [Hoang and Chen, 2011; Nkosi and Mekuria, 2011] propose solutions to protect participants' health information, thereby increasing the privacy of the services.

Cloud-based Data Analysis

Traditional machine learning applications have limitations in terms of the high cost of servers and network, limited local computational resources and storage

[Chen et al., 2010; Gao and Zhai, 2010]. Cloud-based data analysis is introduced to solve these limitations, especially for mobile applications. For example, cloud computing with large storage capacity and powerful processing ability can provide users with much richer services in terms of data size, faster processing speed, and bigger storage.

A few mobile applications in machine learning have already been developed utilizing the capability of cloud computing [Ferzli and Khalife, 2011; Yin et al., 2009; Zhao et al., 2010]. [Ferzli and Khalife, 2011] presents the benefits of combining machine learning techniques and cloud computing to enhance the image/video processing. Through mobile clients, users can understand and compare different algorithms processed in the cloud environment (e.g. de-blurring, de-noising, face detection, and image enhancement). In this dissertation, we took advantage of cloud computing in order to achieve the analysis of the healthcare data in a more efficient and reliable way. As a case study of using the CARA system for pervasive healthcare, we deployed the machine learning module for activity recognition in the cloud infrastructure whereby the data processing and storage are moved from the local mobile device to the powerful and centralized computing platform located in the cloud. The results of data analysis and machine learning classification models are then accessed over the internet connection through a thin native client.

Chapter 3

Context-aware Real-time Assistant Architecture

This chapter describes the CARA system in a general way, including an overview of the system architecture and description of each component. Details on the implementation and evaluation of these components are discussed in later chapters.

3.1 System Design Overview

Advancements in wireless sensor networks and mobile technologies have made possible innovative methods for the delivery of healthcare. To fulfil the vision of pervasive healthcare, a comprehensive Context-aware Real-time Assistant (CARA) architecture is proposed which requires the utilization and integration of a significant number of data acquisition, processing, analysis and presentation components. The design overview of CARA is presented in Figure 3.1.

As a pervasive healthcare system, CARA makes use of a mixture of different data sensing, processing, analysis and delivery mechanisms. Consequently, the CARA system consists of four main components:

- Wireless Sensor Network (WSN): patient specific monitoring of vital signs, activity recognition as well as sensing the ambient context in a smart home environment.

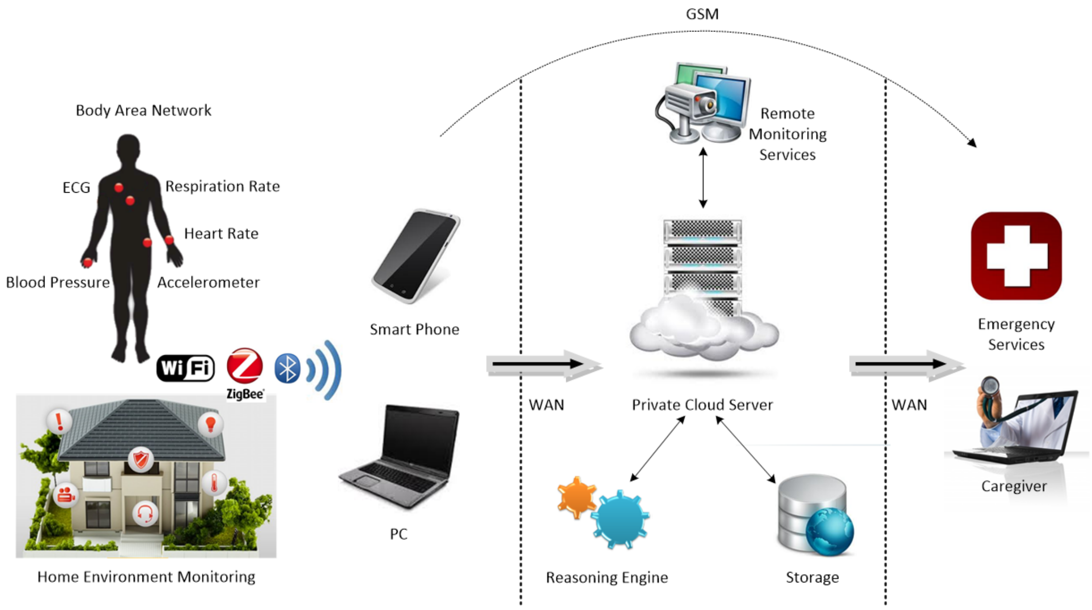


Figure 3.1: CARA pervasive healthcare system design overview

- Reasoning and decision support: interpreting sensor data within a wider context, reasoning with all available knowledge and previous experience for situation assessment, and performing actions according to the decision of the reasoning output.

- Remote monitoring service: remotely and continuously measuring physiological indices of the patient, monitoring changes in the home environment and transmitting data to the remote server in real-time.

- Cloud server: utilizing the computational power of a cloud infrastructure to provide services for sensor data storage, data analysis through machine learning mechanisms.

Figure 3.2 illustrates the architecture of the CARA system. In the pervasive healthcare scenario, a BAN(Body Area Network) and various environmental sensors are deployed in a home environment to gather as much information about and around the person as possible. The system listens to all available sensor data via wireless communication protocols (i.e. Bluetooth, MiWi). On the other hand, the mobile application collects raw accelerometer and gyroscope readings

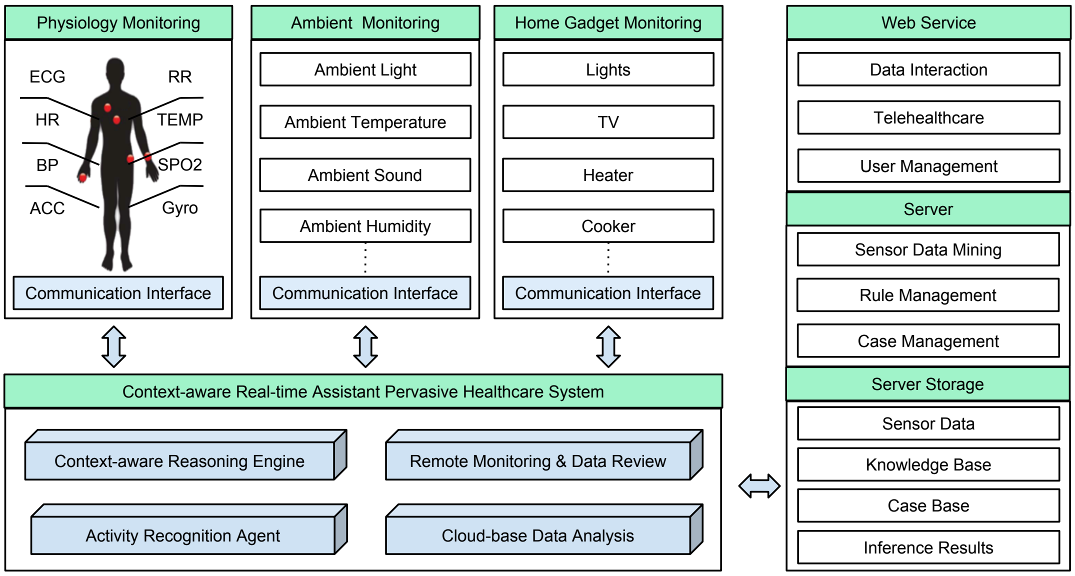


Figure 3.2: Architecture of the CARA system

from wearable body sensors and the smartphone itself to produce low level activity contexts(i.e. sitting, lying, standing, walking, rolling). These contexts along with other environmental and physiological sensor readings are interpreted into contextual information for the monitored individual and environment. They can then be used by the context-aware reasoning engine, which works as the core of the system, to provide medical condition assessment and anomaly detection at home. The real-time reasoning task is carried out in parallel with remote monitoring services which enable the pervasive context, as well as the patient's current condition, to be shared with a remote caregiver in real-time. The local client connects over the internet to the remote server which provides data analysis and reasoning function management services. A client integrated with a webcam also publishes real-time sensor data along with live video streams to the server so that the remote caregiver can communicate with the elderly throughout the monitoring session.

3.2 Pervasive Sensing in Smart Homes

In computing, ambient intelligence (AmI) refers to electronic environments that are sensitive and responsive to the presence of people [Aarts and Marzano, 2003]. Ambient intelligence is a vision of future consumer electronics, telecommunications and computing that aim at helping people in their everyday life. The AmI paradigm is characterized by systems and technologies that is:

- Embedded: many networked devices are integrated into the environment.
- Context aware: these devices can recognize you and your ambient context.
- Personalized: they can be tailored to your needs.
- Adaptive: they can change in response to you.

As a typical realization of AmI, smart homes have been extensively explored to reduce long-term healthcare cost and to improve healthcare services over the last decade. In a smart home environment, hardware, software and networks need to cooperate in an efficient and effective way to provide sensing, reasoning and acting functions. Pervasive sensing in the smart home allows the state of the user and the environment to be perceived by means of wireless sensors. The reasoning engine uses these contexts to make decisions based on the environment to achieve certain goals, and finally, the system reacts based on these decisions.

Figure 3.3 shows the infrastructure of the smart home Wireless Sensor Network (WSN) developed for the CARA system. The WSN provides significant amounts of contextual information about the individuals and home environment using sensors such as environmental sensors (e.g. temperature, light, sound, and humidity), room occupancy sensors (e.g. location), device and gadget sensors (e.g. status of TV, cooker, windows, lights and heater), smartphone (e.g. kinematics) and Body Area Networks (e.g. physiology readings). Such infrastructure has the potential to make a significant impact upon everyday life by providing a smart home environment that is capable of recognizing and responding to the needs of individuals. Furthermore, the proposed WSN would also be extensible and cater for many other devices and other healthcare or smart home applications.

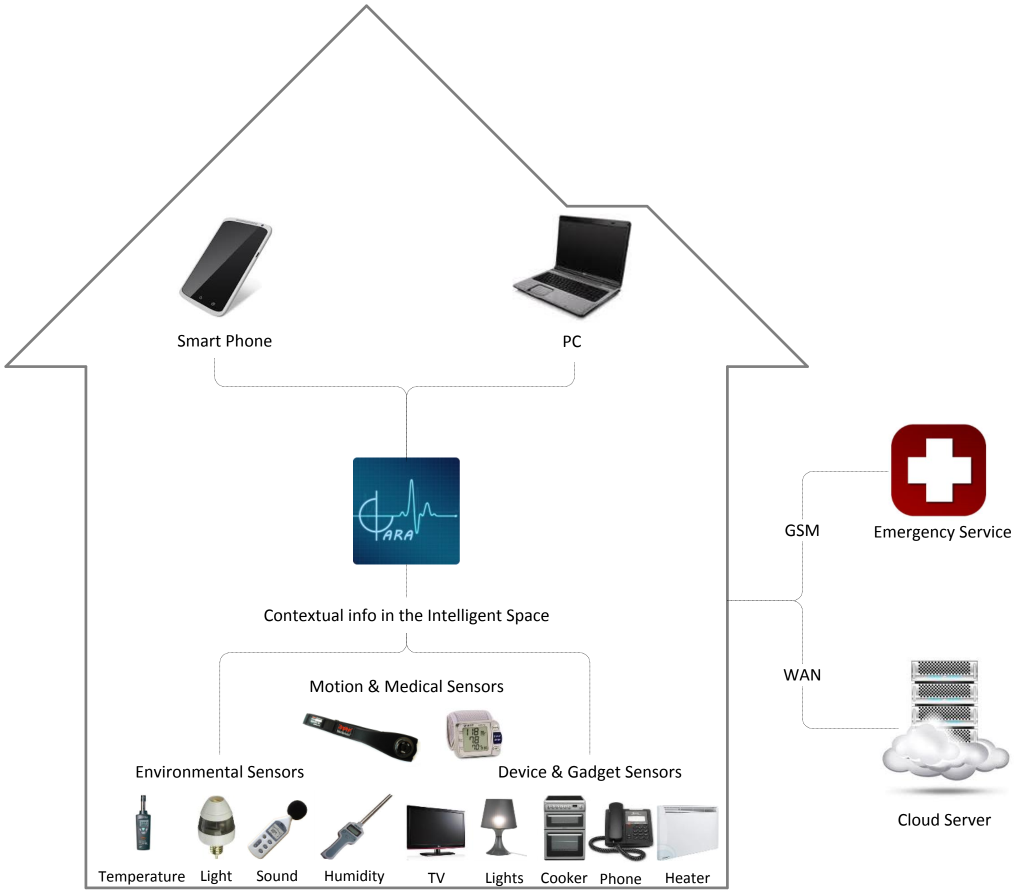


Figure 3.3: Overview of the Wireless Sensor Network in smart home

The most noteworthy use of the WSN is the development of an activity recognition agent for pervasive sensing. Learning and recognizing the activities of daily living (ADLs) of an individual is vital when providing an individual with context-aware at-home healthcare. In this dissertation, we assess the unobtrusive detection of inhabitants' activities in the smart home environment through the use of a smartphone and a wearable body sensor. The solution involves a combination of a threshold-based technique for identifying simpler static activities, and a sophisticated machine learning technique for identifying more complex dynamic activities. This can be used to detect changes in a subject's routine and to assist the automated reasoning with activity contexts.

3.3 Context-aware Healthcare Decision Support

A system is considered context aware if it uses context to provide relevant information and services to the user, where relevance depends on the user's task [Dey and Abowd, 1999]. This could include both "passive" context awareness where the system becomes aware of, but does not adapt to the changing contexts, and "active" context awareness, where the system adapts to the changing contexts. The CARA system is designed to achieve active context awareness, where its adaptive learning aids in better decision making on the patient's medical condition and healthcare needs.

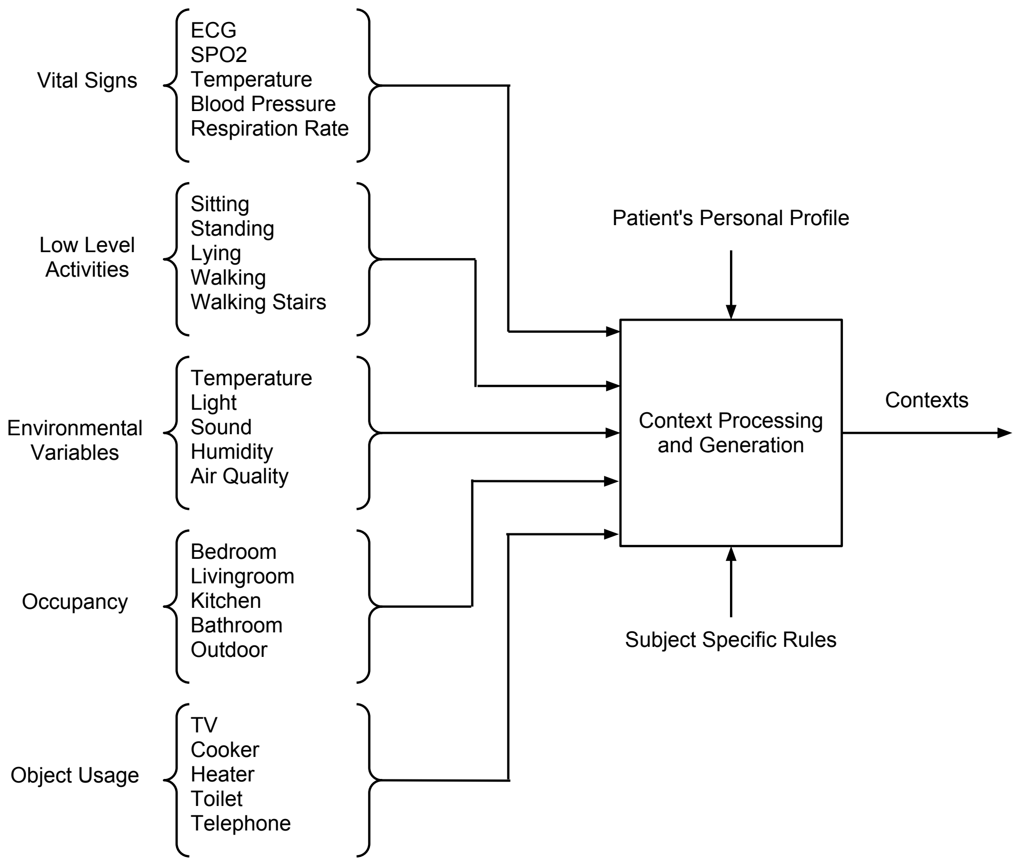


Figure 3.4: Context-awareness in the CARA system

The workflow of context processing in the CARA system is illustrated in Figure 3.4. Raw sensor data are gathered and interpreted into contexts to support the reasoning function. A hierarchical structure is designed for context modelling. For instance, low level activity context (e.g. sitting, lying, walking) is produced from accelerometer and gyroscope sensor data using activity recognition mechanisms. These contexts are then interpreted into high level activity contexts (watching TV, sleeping, exercising, etc.) by incorporating other related contexts (e.g. occupancy and object usage) according to specific rules and personal profile. The profile and rules are adaptive to individuals and changing contexts.

Context-aware healthcare decision support provides caregivers and patients with a reasoning mechanism and built-in intelligence to enhance healthcare services. As a novel approach to achieve intelligent reasoning in the CARA system, a context-aware hybrid reasoning framework is developed (consisting of case-based reasoning mechanisms incorporating fuzzy-based domain knowledge) to compensate for the deficiencies of a single reasoning model. Although the straightforward rule-based reasoning engine is a competent approach, it still has some unsatisfactory limitations. Especially in the medical domain, the knowledge of experts does not only consist of rules, but of a mixture of explicit knowledge and experience. In order to improve the performance of the inference mechanism, a case-based reasoning approach is adopted, which makes use of the accumulation of previously solved cases to accomplish the reasoning task. Figure 3.5 presents the workflow of the reasoning system in CARA.

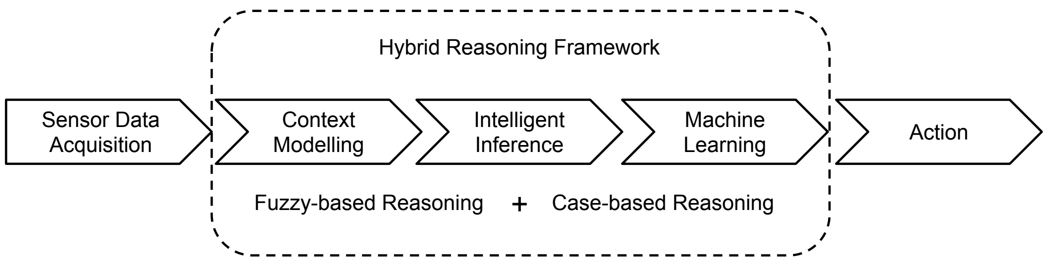


Figure 3.5: Workflow of the CARA reasoning framework

The reasoning engine can be tailored with different knowledge based rules for different applications (such as for in-clinic assessment or at-home monitoring).

Such rules are customizable and must be verified to ensure the correctness and consistency of the reasoning system. The reasoning engine also executes in real-time and offers immediate notification of abnormal conditions. Such anomalies can be identified from correlating different contexts and trends in behavioural patterns accumulated over time. Moreover, provenance of the reasoning output is recorded to give an explanation of the decision to the user, which makes the sophisticated reasoning engine more transparent and accountable. Overall, the CARA reasoning component is capable of performing the following reasoning tasks:

- Continuous contextualization of the ambient condition of a patient.
- Configuration of custom rules and verification of rule conflicts.
- Prediction of possibly abnormal situations with reasoning provenance.
- Notification of emergency situations indicating a health risk.
- Home automation or user prompting within a smart home environment.

3.4 Remote Monitoring Service

For people with chronic diseases like hypertension and heart conditions, it requires a lifelong collaboration with caregivers to manage their health condition. A key goal in managing chronic disease is to catch evolving conditions early so that the caregiver and patient can make medical or lifestyle changes before the condition worsens and requires more complex and costly treatments.

The CARA remote monitoring system is designed to provide long-term healthcare services in an appropriate timely manner, which allows patients with certain medical devices to be monitored or examined remotely in real-time. It achieves interactive at-home care by integrating a video-conferencing function and transferring healthcare data captured by sensors via the internet to a remote caregiver in real-time. This creates a seamless link between the clinical view and home healthcare which can lead to greater cost-efficiency of healthcare services.

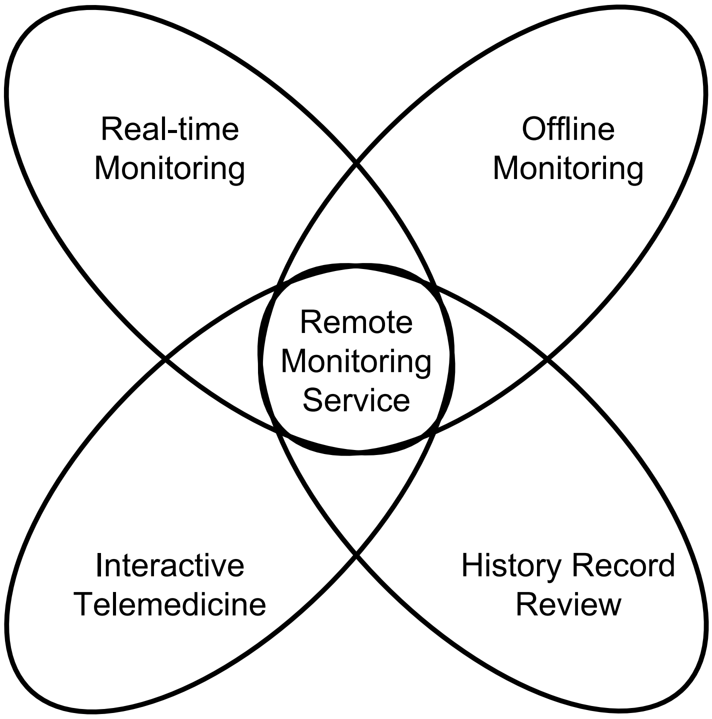


Figure 3.6: CARA remote monitoring conceptual model

Figure 3.6 shows the conceptual model of the remote monitoring service in the CARA system. It is fulfilled with four functionalities and deals with diverse stakeholders. Real-time remote monitoring enables caregivers to monitor a patient remotely through medical devices and smart home sensors. The relevant information, such as patient vital signs, activity status, medical conditions, and home environment readings, are updated in real-time and shared with remote medical professionals through an internet connection, so they can make an early diagnosis based on the current situation of the patient. On the other hand, patients can get more efficient medical consultation and treatment with fewer visits to the hospital/clinic. Real-time remote monitoring is expected to provide comparable health outcomes to traditional in-person patient encounters, and supply greater satisfaction to patients.

Remote monitoring can also be carried out autonomously in an offline model.

Distinct from the real-time monitoring, it does not require the presence of a medical specialist and patient at the same time. The patient with wearable medical devices is continuously monitored within a smart home environment. Sensor data are processed and transmitted to the remote server. They are stored in the server as healthcare data in an electronic form, instead of actual physical examination and history, for the medical specialist to make an assessment at a convenient time. Additionally, the remote monitoring service can make smart, timely healthcare decisions with collaboration of the CARA reasoning framework so that it can send notifications or alerts in case any anomaly or emergency occurs.

In addition to the traditional remote monitoring services, we developed an interactive telecare component which provides real-time interactions between the patient and healthcare provider using a web-camera integrated device (e.g. Tablet or PC). A video link is established along with the real-time sensor data transmission that allows a patient to communicate with a remote caregiver throughout the monitoring session. It enables various activities, such as medical consultation, physical examination, and psychological counselling, comparable to those done in traditional face-to-face visits. The audio and video sessions are recorded as healthcare data and stored on the server as well.

Last but not least, a user interface is developed to allow the medical consultant to review the historical data in a graphical view. It consists of sensor data review and video review functions. The continuous sensor data can be segmented by timestamps and viewed in web-based graphical charts with scrolling, zooming and tagging functions, and the recorded video can be replayed with synchronized sensor data for the better understanding of the patient's condition at that moment. Furthermore, the recorded data review component can be used not only for remote medical assessment but also for teaching purposes.

3.5 Cloud-based Data Analysis for Healthcare

The growth of sensors and intelligent systems has created a whole new warehouse of valuable data for healthcare. The problem in healthcare is never the lack of data, but the lack of information that can be used to support decision-making, planning and strategy. By applying effective data analysis technology to the data gathered from sensors and intelligent systems, it can make decision support for healthcare more efficient and ultimately more accurate.

Data analysis technology is designed to work with large volumes of heterogeneous data. It uses sophisticated statistical methods such as machine learning, computational mathematics, and artificial intelligence to explore the data and to discover interrelationships and patterns. Deeper insights are possible when more data are available. Typical data analysis systems require clusters of servers to support the tools that process the large volumes and varied formats of data. However, traditional methodologies such as distributed systems and database management systems are often not suitable to handle very large data sets in terms of scalability and availability. Therefore, innovative solutions need to be explored, which should be able to offer a large number of machines exposing huge storage and computing power.

Lately, cloud computing has received a substantial amount of attention from both industry and academia. Cloud computing can offer scalable data analysis solutions while enabling greater efficiencies and reducing costs. As the cloud infrastructure is distributed and fault tolerant, a cloud-based data analysis framework can deploy multiple instances on pools of server, storage, and networking resources that can be scaled up or down as needed. These instances can be moved around depending on the need to make the best use of the hardware without compromising performance. Indeed, cloud computing offers a cost-effective way to support data analysis technologies and intelligent healthcare applications with high flexibility, scalability and availability for accessing healthcare data, discovering patterns, and processing machine learning tasks.

In the CARA system, a comprehensive cloud-based data analysis framework is developed incorporating machine learning and cloud computing technologies. This provides analysis as a service - from data delivery and data analysis to data storage, in order to optimize the total value of healthcare data. A general structure of the cloud-based data analysis framework is presented in Figure. 3.7.

In the case of CARA activity recognition, different choices of machine learning algorithms are investigated to build classification models (i.e. Decision Tree, Neural Network, Nearest Neighbour and Bayesian Network), each of which can be viewed as a blackbox that is capable of inferring current activity based on the

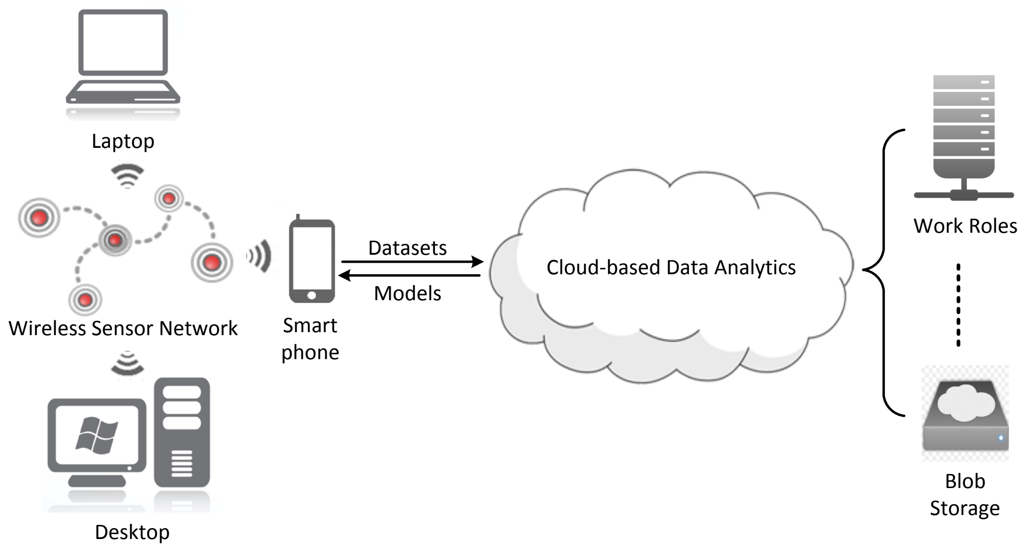


Figure 3.7: CARA cloud-based data analysis structure

input dataset. However, each model comes from a unique algorithm approach and will perform differently for the same dataset; also the same model may not fit all the users since the behavioural pattern of each individual is slightly different. The best approach is to use cross-validation to determine which model performs best for a given dataset. In addition, continuous gathering of activity data will result in huge amounts of data, especially if many users are involved. Ideally, one might want to keep a large amount of this raw data for future (and maybe different) analysis, and also analyse the data to produce a compact model which can be used in the smartphone for real-time analysis of new data. This motivates a cloud computing solution for data analysis where data from many users can be stored and analysed efficiently, and then the compact results of the analysis can be downloaded and used in the smartphone.

In this dissertation, the activity monitoring system is used as a case study for evaluating the cloud-based data analysis framework. Several machine learning nodes are deployed on a private cloud server sharing blob storage. Each node holds an instance of one type of machine learning method for dealing with the input dataset. Once the new dataset is uploaded to the cloud by the user, every machine learning node starts analysing data in parallel, producing machine learning models as well as evaluating them. After examining the evaluation result, the best model is selected to support activity recognition in the CARA system. New training datasets can be generated locally in a home environment in an unsupervised manner through using the best classification model to classify and label the new input data. As more data are gathered and continually uploaded to the cloud, the model is adapted through an unsupervised scheme to produce enhanced models which are then downloaded onto the smartphone for improved real-time activity analysis. Consequentially the system becomes more accurate and reliable. Moreover, models and datasets are stored in the blob as resources to be shared with other applications or healthcare systems.

Chapter 4

Wireless Sensor Networks for Pervasive Healthcare

4.1 Introduction

This chapter is a description of the development of wireless sensor networks (WSNs) in the CARA system. The goal of developing WSNs for the CARA system is to sense a user's vital signs, surroundings, and activities. To achieve this, a body area network (BAN) is used to measure the vital signs; smart home sensors are deployed to monitor the surroundings in a home environment; and an Android smartphone is used to identify the activities associated with the BAN. Thus, the WSN supports the CARA system with physiological contexts ,environmental contexts and activity contexts.

4.2 BANs for Medical Sensing

A BAN is formally defined by IEEE 802.15 as, *"a communication standard optimized for low power devices and operation on, in or around the human body (but not limited to humans) to serve a variety of applications including medical, consumer electronics, personal entertainment and other"* [IEEE-BAN, 2013]. In more general terms, a BAN is a wireless network of wearable computing devices that cooperate for the benefit of the user. As an emerging technology, BANs are mostly used in the medical field for the purpose of pervasive healthcare. They are quite flexible and scalable in terms of integration of different wearable medical sensors. Some of the conventional medical sensors for physiology measurement are listed in Table 4.1.

Table 4.1: Properties that can be measured from wearable medical sensors

Property	Description
Temperature	Using thermistor to measure the skin temperature of the body.
Respiration Rate	Using plethysmograph to measure breathing. Impedance of fabric changes with stretching.
ECG and Heart Rate	Using wearable electrodes which attach to the skin to detect ECG trace and heart rate.
Weight	Using scales to measure body weight, body fat percentage and body mass index (BMI).
Skin Conductance	Using Galvanic Skin Response (GSR) sensor to detect electrical conductance of the skin.
Pulse Oximetry (SPO_2)	Using a light source to measure changes in pigmentation which reflect oxygen in the blood stream.
Blood Glucose	Piercing the finger to provide a blood test. Particularly important in the care of diabetes.
Blood Pressure	Using a cuff to measure systolic/diastolic pressure of the blood.

These wireless medical sensors are usually non-invasive, replaceable and low cost. However, each of the devices has unique requirements in terms of bandwidth, latency, power usage, and signal distance. IEEE 802.15 established a task group to develop the standards for the use of BAN sensors [IEEE-BAN, 2013]. The task group has outlined the ideal position for a BAN in the power vs data rate spectrum as described in Figure 4.1.

As you can see the range of BAN devices can vary greatly in terms of bandwidth and power consumption. Lots of research prototypes and commercial products have been designed to support the needs of healthcare based on IEEE standards [CodeBlue, 2013] [Shimmer, 2013] [Biosensics, 2013] [Equivital, 2013] [Zephyr Inc., 2013].

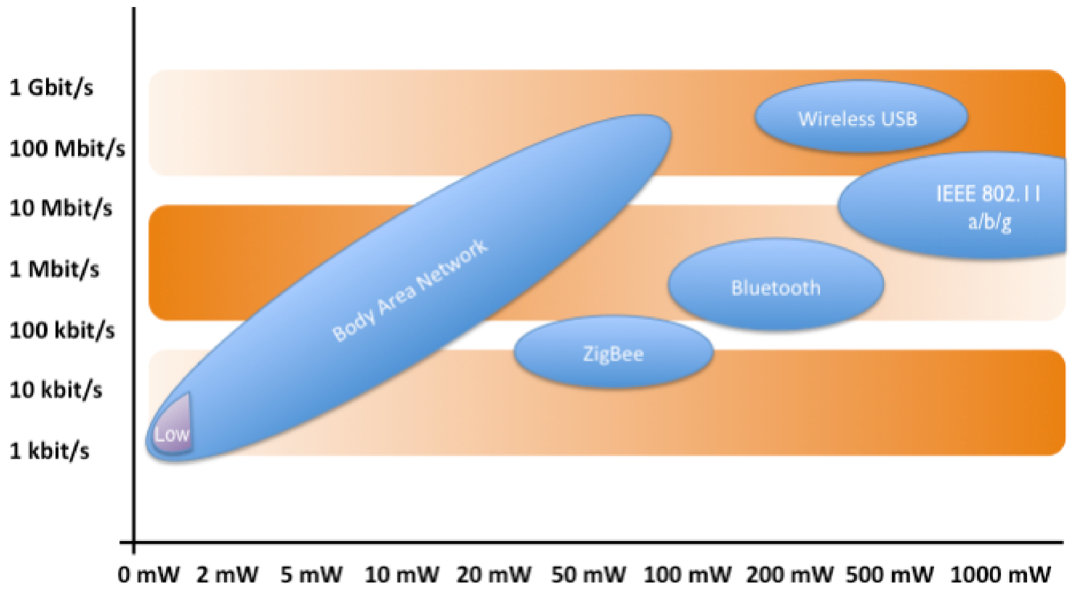


Figure 4.1: Data rate vs Power consumption [IEEE-BAN, 2013]

4.2.1 Tyndall Sensors

At the first stage of developing the CARA system, the BAN is constructed using a range of wireless sensors built upon the Tyndall 25mm hardware platform [O'Flynn et al., 2004]. It is a layered wireless sensor networking solution, consisting of 25mm x 25mm layers which may be combined in a plug and play fashion. The platform configuration illustrated in figure 4.2 consists of a battery layer for energy provision, an 802.15.4 compliant transceiver/microcontroller layer for computation and communication and a sensor layer for sensing.

The Tyndall 25mm 802.15.4 transceiver/microcontroller layer has been developed for wireless sensors incorporating Chipcon's CC2420 transceiver [Chipcon, 2013] and the Atmel Atmega128 microcontroller [Atmel, 2013] into a small versatile package for processing and communication. Two accelerometer boards, a pulse oximeter and a two-lead Electrocardiogram (ECG) have been used as the sensor layer to collect vital sign data. The sensors, when coupled with a transceiver layer, monitor movement and collect heart rate, blood oxygen saturation (SpO_2), and ECG data. The data are relayed in raw form over a short-range (100m) wireless network to a basestation in real-time. The basestation is built

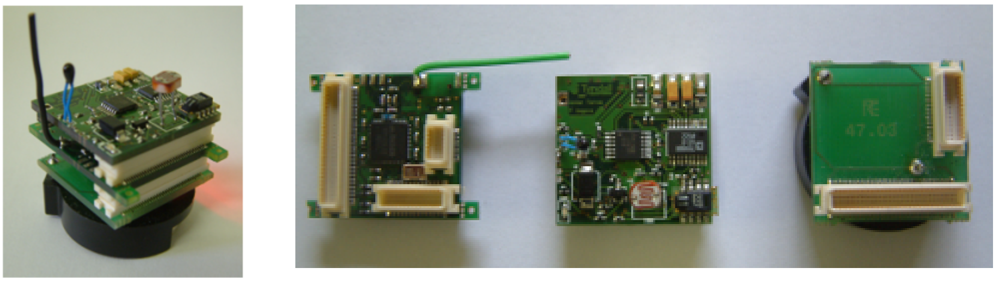


Figure 4.2: The Tyndall 25mm platform (a) Stacked, (b) From left to right Communication and Processing Layer, Sensor Layer and Power Layer.

upon a Tyndall Programming Board which transmits data to a laptop.

Although Tyndall sensors are capable of building the basic BAN for the CARA system, they have some limitations in terms of connectivity and mobility. The USB-serial port connection between basestation and PC makes it non-portable and not compatible with other mobile devices that do not have a serial port build-in. Besides, it turns out unpackaged sensor nodes sometimes have difficulty to be worn in an unobtrusive way from the user point of view. As a result, we started seeking for a more integrated and packaged solution in order to improve the user experience.

4.2.2 Zephyr BioHarness

After investigating several research development kits [CodeBlue, 2013] [Shimmer, 2013] and commercial products [Biosensics, 2013] [Equivital, 2013] [Zephyr Inc., 2013], we decided to use Zephyr BioHarness sensors in an infrastructure incorporating Tyndall sensors to build the BAN for CARA.

The BioHarness is a physiological monitoring wireless device designed for monitoring of patients in the home environment and alternate care settings. It can be used as a general patient monitor to provide physiological information as part of a healthcare monitoring system, for general research purposes. The device consists of a chest strap and an electronics module that attaches to the strap. The device provides a facility to detect and transmit vital signs data including ECG, heart rate, respiration rate, skin temperature, body orientation and 3D-

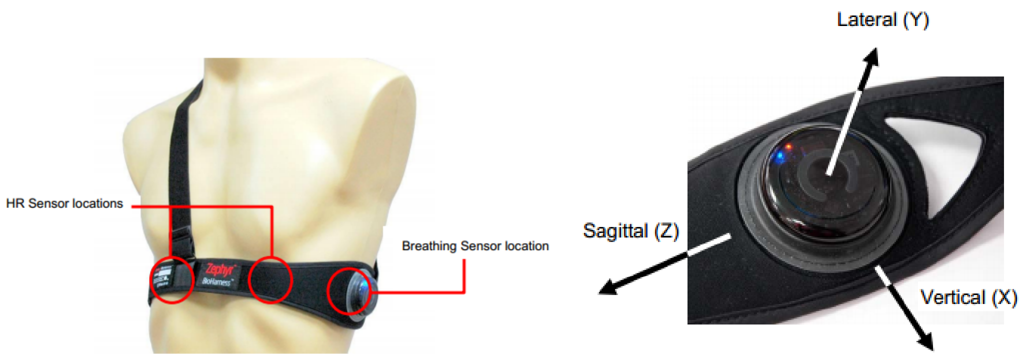


Figure 4.3: The Zephyr BioHarness 3

acceleration as shown in figure 4.3. Part of the specification of sensor data is shown in table 4.2 .

The BioHarness is powered by an internal battery which can be charged in the USB charging cradle. It can last over 20 hours for real-time data transmitting with a fully charged battery. The device can collect measurements captured during both static as well as dynamic activity and transmit the data to any Bluetooth enabled device in real-time. Table 4.3 gives the Radio Frequency characteristics of the Bluetooth protocol used for data communication. Overall, the Zephyr BioHarness is a powerful and portable device with integration of all the features required for a monitoring system. It provides an ideal packaged solution to build the BAN for the CARA system.

4.3 Smart Home Sensors

To achieve pervasive healthcare for independent living, a context-aware system should be able to observe, interpret and reason about dynamic situations in the home environment. However, the system might make quite a different assessment of medical conditions without the information about surrounding contexts. For instance, the inference result might be different if a person is cooking in the kitchen rather than sleeping in the bedroom. For the CARA system to be as useful and unobtrusive as possible, it needs to be aware of environmental conditions as well as the medical condition of individuals by incorporating small, low-cost, low-

Table 4.2: Data output of Zephyr BioHarness [Zephyr Inc., 2013]

Parameter	Freq. (Hz)	Range	Units	Description
General Data Packet				
Heart Rate	1	25 - 240	BPM	Beats per Minute
Breathing Rate	1	3 - 70	BPM	Breaths per Minute
Temperature	1	10 - 60	°C	Skin Temperature
Posture	1	±180	Degrees	Vertical=0°, Inverted=180°
Peak Acceleration	1	±16	g	Peak absolute acceleration, 1/10 g's
X Acceleration Min	1	±16	g	Vertical axis
X Acceleration Peak	1	±16	g	
Y Acceleration Min	1	±16	g	Lateral axis
Y Acceleration Peak	1	±16	g	
Z Acceleration Min	1	±16	g	Sagittal axis
Z Acceleration Peak	1	±16	g	
Battery	1	0 - 100	%	% of full capacity
ECG Packet				
ECG sensor output	250	0 - 1023	bit	1 bit=0.013405 mV

Table 4.3: RF characteristics [Zephyr Inc., 2013]

Bluetooth Compliance	Version 2.1 + EDR
Supported Profile	Serial Port
Frequency	2.4 to 2.835 GHz
Output Power	10 dBm
Operating Range	Up to 300ft/100m
Sensitivity	-91 dBm
Antenna Type	Internal

intrusion smart home sensors with the BAN in a home environment. This makes a further comprehensive level of real-time home monitoring possible and should lead to better healthcare decisions and quality of life for independent living.

4.3.1 Sensor Development

In a WSN, each wireless sensor is a node in a wireless sensor network that is capable of gathering sensory information, processing it in some manner, and

Table 4.4: Smart Home sensors specification

Sensor	Signal Type	Sample Rate	Functionality
Passive Infrared	Thermal	< 1 Hz	Motion, movement in rooms
Electromagnetic	Electrical	< 1 Hz	Home appliances usage
Proximity	Optical	< 40 Hz	Presence of nearby objects
Capacitive	Mechanical	< 1 Hz	Objects occupancy
Light	Optical	< 1 Hz	Ambient light in the room
Temperature	Thermal	< 1 Hz	Environmental temperature
Sound	Acoustic	< 1 Hz	Ambient noise in the room
Humidity	Thermal	< 1 Hz	Environmental humidity

communicating with other nodes in the network. The majority of wireless sensor platforms share a common set of system components:

- Microcontroller: provides the computational capabilities of the platform.
- Radio transceiver: provides low-power wireless communications.
- Sensor board: provides hardware interfaces to external sensors.
- Power Layer: provides power through batteries, capacitors, or solar arrays.

Although lots of off-the-shelf smart home products are available in the market, none of them meet all the requirement of our system. In addition, these commercial products usually provide packaged hardware solutions using a unique firmware or communication protocol, which makes them incompatible with other applications. Therefore, it is difficult to integrate them in our system. Consequently, we decided to build customized devices to construct the WSN from modules that support processing, sensing, radio, or other functions. The custom designed WSN for smart home applications is built upon MICROCHIP modules and adopts the MiWi network protocol for wireless communication. A summary of smart home sensors in the WSN is given in Table 4.4, and an example of a sensor prototype is shown in Figure 4.4.

4.3.2 Wireless Networking Protocol

There is a growing expectation that devices will have built-in abilities to communicate with each other without a hard-wired connection. The benefits of wireless

Figure 4.4: Prototype of the PIR sensor

communication are reduced costs and ease of implementation. It does not require cabling and other hardware, and associated installation costs. To implement the application with the smart home WSN, the challenge is to select the right wireless networking protocol and implement it in a cost-effective manner.

The three primary protocols available on the market today are the IEEE 802.15.4 based ZigBee and MiWi protocols, and the ISO Dash7 standard. A brief comparison of Dash7, ZigBee, and MiWi is shown in Table 4.5. Since for many

Table 4.5: Comparison of protocols for wireless personal area networks (WPANs)

Protocol	**Underlying Standard**	**Frequency**	**Range**	**Footprint**
Zigbee	IEEE 802.15.4	2.4 GHz, 915 MHz, 868 MHz	30-500 m	40-100 K
Miwi	IEEE 802.15.4	2.4 GHz	10-100 m	3-17 K
Dash7	ISO 18000-7	433 MHz	1 km	15-40 K

applications the full ZigBee protocol has become too comprehensive and complex, a large percentage of IEEE 802.15.4-compliant wireless networks use alternative proprietary protocols. The Microchip MiWi Wireless Networking Protocol Stack is a simple protocol designed for low data rate, short distance, low-cost networks.

Based on IEEE 802.15.4 for wireless personal area networks, the MiWi protocol provides an easy-to-use alternative for wireless communication. In particular, it targets smaller applications that have relatively small network sizes, with few hops between [Flowers and Yang, 2010]. Since the scale of our smart home WSN is small and the topology of the WSN is simple, we decided to use the Microchip MiWi P2P wireless protocol which is the simplified wireless protocol that are supported in the MiWi Development Environment. The protocol has a rich feature set that can be compiled in and out of the stack to meet a wide range of design needs. However, as a simpler variation of the MiWi protocol it only supports peer-to-peer and star topologies without a routing mechanism, so the wireless communication coverage is defined by the radio range [Yang, 2010].

The MiWi P2P protocol categorizes devices based on IEEE standards and their role in making the communication connections as shown in Table 4.6 and Table 4.7.

Table 4.6: IEEE 802.15.4 device types - based on functionality

Functional Type	**Power Source**	**Receiver Idle Configuration**	**Data Transmission Method**
Full Function Device (FFD)	Mains	On	Poll from associated devices
Reduced Function Device (RFD)	Battery	Off	Push to the associated device

Table 4.7: IEEE 802.15.4 device types - based on role

Role Type	**Functional Type**	**Role Description**
Personal Area Network (PAN) Coordinator	FFD	The device starts first and waits for a connection.
End Device	RFD	The device starts after the PAN coordinator has begun to establish a connection.

A star topology supported by the MiWi P2P protocol is adopted for the development of our WSN (as shown in Figure 4.5). From a device role perspective, the topology has one PAN (Personal Area Network) coordinator, which in our case is a Raspberry Pi working as the basestation that initiates communications and accepts connections from other devices. It has several end devices that join the communication. These are the various wireless sensors. From the functionality perspective, the PAN coordinator is a FFD (Full Function Device). An end device is a RFD (Reduced Function Device) with its radio off when it is idle. The MiWi P2P protocol supports two ways of transmitting a message: broadcast and unicast. Broadcast packets have all devices in the radio range as their destination. There is no acknowledgement for broadcasting messages. Unicast transmissions have only one destination and use the long address as the destination address. It requires acknowledgement for all unicast messages. In the implementation of the smart home WSN, the unicast mode is adopted, so end devices can establish connections only with the PAN coordinator.

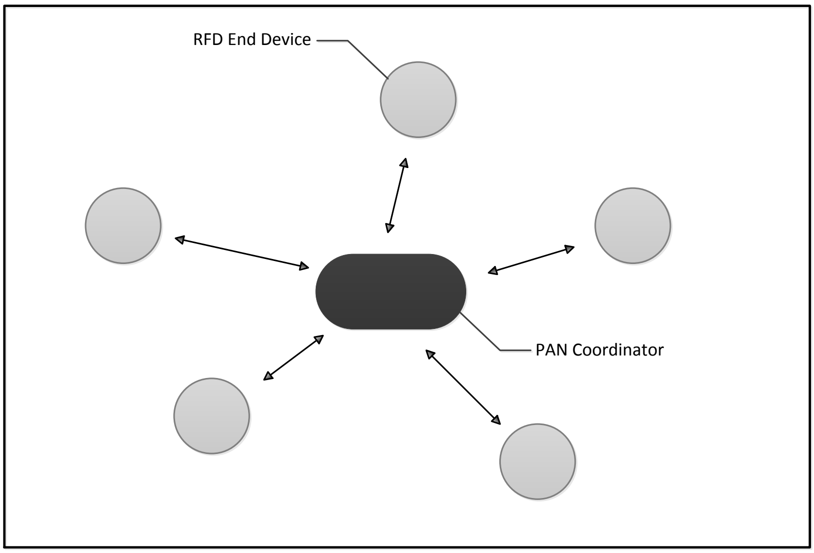


Figure 4.5: Star Topology [Yang, 2010]

A typical MiWi P2P protocol application starts by initializing the hardware and MiWi P2P protocol. Then, it attempts to establish a connection and to enter the normal operation mode of receiving and transmitting data. Figure 4.6 illustrates the typical flow of the MiWi P2P protocol application. The MiWi P2P

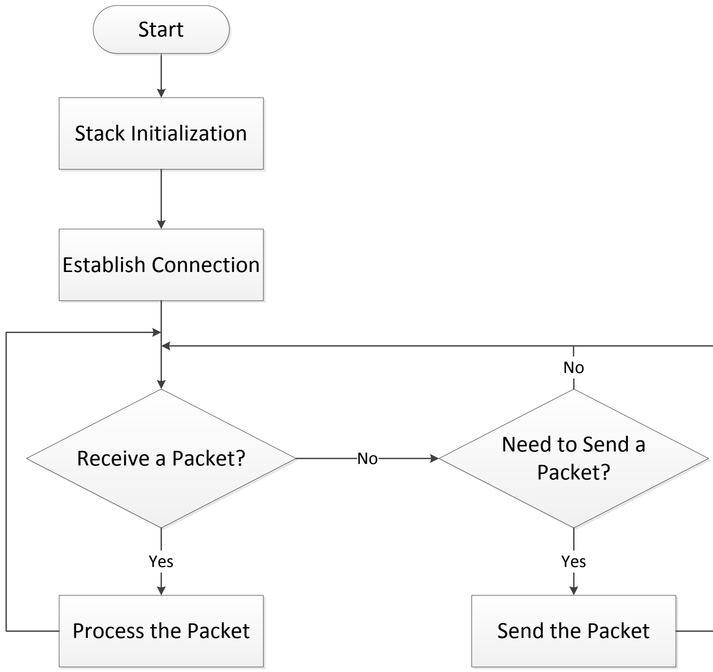


Figure 4.6: Flowchart for the MiWi P2P wireless protocol application

protocol uses MiApp as its application programming interface.

4.3.3 Field Deployment

In order to carry out experiments to evaluate the CARA system, we deploy the WSN in a friendly environment. A "friendly" environment is a real-world (out-of-lab) deployment environment, lived in by a researcher rather than by a real patient. The purpose of friendly environment deployment is to assess the system in a real home environment over a protracted period.

A key issue for deploying is the layout of the home environment and the identification of ideal sensor locations. The ideal sensor locations reflect the range of the radios being used (e.g., MiWi, Bluetooth), the size of the rooms, the sensitivity of sensors, and so forth. Figure 4.7 presents the floor plan of the home trial with sensor locations. Although the sensor locations are customized to

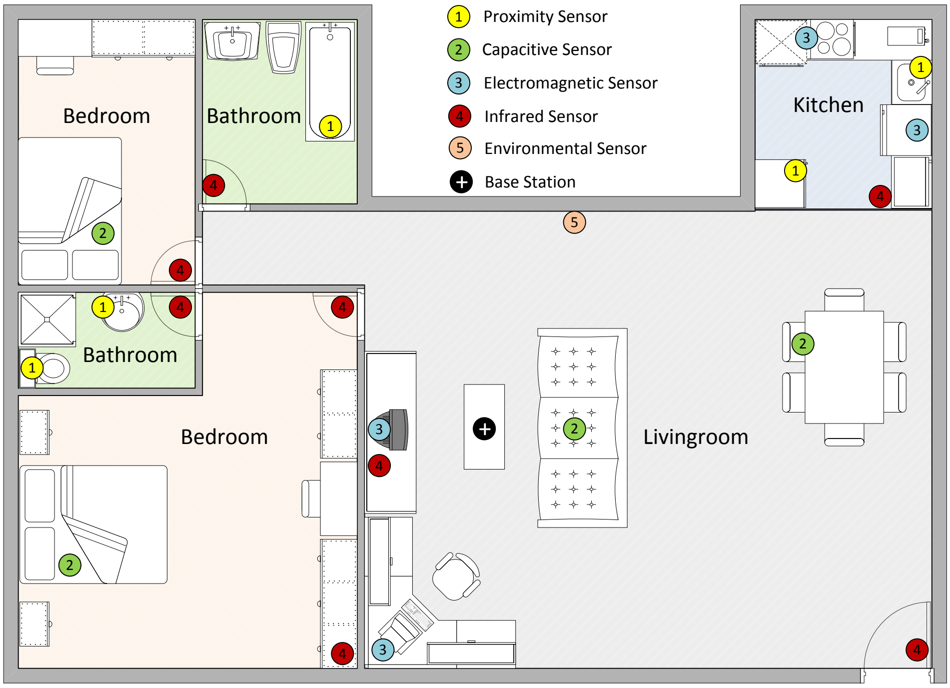


Figure 4.7: Deployment of wireless sensors within a real-world home environment

this specific home layout in this dissertation, it is desirable the system provides integration mechanisms that allow users to design and deploy sensors in their own home. As shown in the floor map, various types of wireless sensors are deployed to monitor the ambient changes within the home environment:

- Proximity sensors in yellow indicate the approach to the sink, toilet, bathtub and fridge.
- Capacitive sensors in green detect the occupancy of the couch, bed and dining chair.
- Electromagnetic sensors in blue detect the use of home appliances, e.g. TV, cooker, washing machine and PC.
- Passive infrared sensors in red detect the motion around.

- Integrated environmental sensors in pink measure the temperature, humidity, noise and light level in the home.

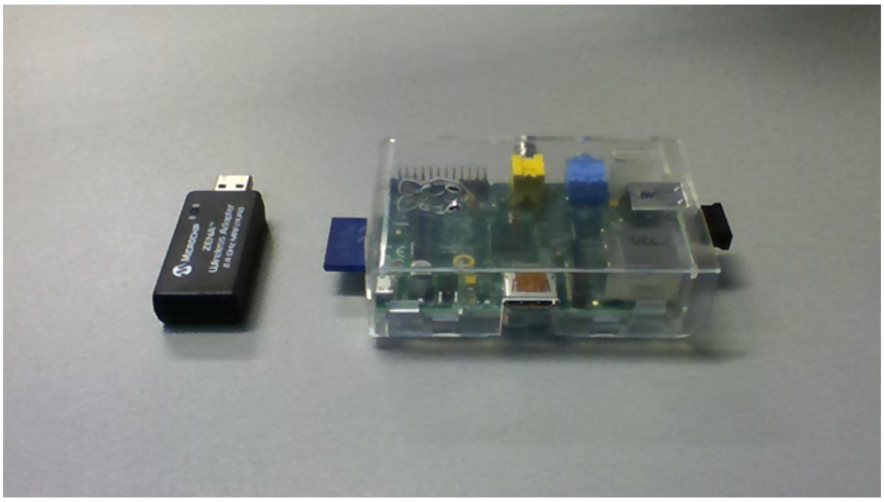


Figure 4.8: Raspberry Pi hosted basestation with the ZENA Wireless Adapter

Real-time signals are transmitted to a basestation which is developed on a Raspberry Pi (See Figure 4.8). To establish the connection between sensor nodes and the base station, the ZENA Wireless Adapter [Microchip Technology Inc., 2012] is used to connect the Raspberry Pi as a wireless node to the sensor network for data collection. The ZENA Wireless Adapter is a multi-function USB wireless adapter connecting USB-equipped devices with Microchip wireless sensors for development of a WSN. It is pre-programmed with a MiWi Wireless Protocol Sniffer application that enables the user to receive MiWi Wireless Protocol packets. The base station also communicates with a home gateway (PC/Laptop) through a Bluetooth connection. In this way, the customized WSN is integrated into the CARA system.

The above MiWi-based WSN was developed in collaboration with an external mechanical engineer who designed and developed the hardware and low level software.

4.4 Smartphone Associated Sensing

Advances in computing capability and connectivity have accelerated the convergence between mobile phones and powerful computers facilitating the development of smartphone technology in recent years. Mobile phones are getting smarter by utilizing embedded sensors and additional resources in the phone. They can be used to understand our life patterns, manage our health and well-being, help us navigate through our day, and intervene on our behalf. This is giving rise to a new area of research called smartphone sensing [Lane et al., 2010].

4.4.1 Smartphone Development

Early mobile phones had only basic phone features, so people with computational needs had to carry a separate dedicated personal digital assistant (PDA) device, running early versions of operating systems such as Palm OS, BlackBerry OS or Windows CE/Pocket PC [Nusca, 2009]. It is not until the late 1990s, IBM introduced the Simon that combined the functions of a (PDA) with a mobile phone, which is referred to as the first smartphone [Sager, 2012]. With the rapid development of smartphone technology, today's top-end smartphones come with 1.4-GHz quad-core processors and a growing set of low cost yet powerful embedded sensors (see Table 4.8). They also have multiple radios for communications, large storage, and a touch screen. Sensor enabled smartphones are becoming more and more central to people's lives.

An early smartphone such as the Nokia N95, had an embedded accelerometer and GPS. However, it lacked an efficient software infrastructure to support mobile sensing applications. Nokia didn't provide competent application programming interfaces (APIs) to access the sensor data, because the accelerometer was only there for video stabilization and photo orientation. There were also limitations for implementing efficient resource management routines, e.g., turning the sensors into sleep mode when not needed. Advanced smartphone operating systems, such as iOS, Android, BlackBerry, and Windows Mobile, are much more supportive platforms for the development of smartphone sensing applications. They provide a more comprehensive set of APIs to access the low level components of the phone while taking advantage of more powerful hardware (CPU and RAM). Each new

Table 4.8: Common smartphone integrated sensors

Type	Measurand
Present	
3-axis Accelerometer	Acceleration in the X, Y, Z axes
3-axis Gyroscope	Rotation in space (Roll, Pitch, Yaw)
3-axis Magnetometer	Location direction (compass)
Microphone	Audio
Camera	Images, video
GPS	Location
Ambient Light	Illuminance
Proximity	Nearby objects
Pressure	Pressure (used to determine altitude)
Future	
6-dimensional accelerometer	Combination of accelerometer and gyroscope
9-axis motion sensor	Combination of accelerometer, compass and gyroscope
Biochemical	Biochemical agents
Physiological	Physiological agents

release of smartphones offers advances in sensing, computation, and communications which make the smartphone a truly ubiquitous mobile computing device. Smartphone associated sensing, combined with pervasive computing techniques, is enabling a new generation of mobile healthcare that can both monitor and enhance physical and emotional well-being.

4.4.2 Smartphone Sensing for Activity Recognition

Enabling context-awareness in smartphones has been the focus of much research recently. A key for developing such applications is the utilization of a variety of sensors in modern smartphones, allowing them to learn about the user's behaviour and the surrounding environment. Among all these applications, activity recognition has been considered as an important issue for the successful realization of context-awareness in pervasive environments. This relates to the fact that activities in a pervasive environment provide important contextual information and the intelligent use of such an environment relies on the activity context. As a consequence, smartphone based activity recognition has attracted increasing attention for real-world pervasive healthcare applications.

Early activity recognition work focused on using a network of wearable sensors to track the movement of specific limbs as well as the body as a whole [Maurer et al., 2006]. These sensors are usually small in size, lightweight, low cost and non-invasive. It has proved to be an effective and reliable method for human activity recognition. Nevertheless, most of the wearable sensors need to be fixed onto specific locations on the subject's body. As a consequence, wearing these specialized sensors on a daily basis becomes a major challenge, making these systems less practical. In contrast, research shows that a variety of daily human activities can be inferred most successfully from embedded sensors on the smartphone (i.e. accelerometer, gyroscope and GPS). By taking advantage of smartphones' computational power and sensing capabilities, and their tight coupling with users' everyday lives, smartphones can become an alternative platform to customized sensors that researchers have previously adopted to recognize activities of daily living [Choudhury et al., 2008]. Using a smartphone as a primary device for data collection and processing increases the likelihood of data coverage and represents a minimal cost and maintenance commitment to the user.

In this dissertation, a novel Activity of Daily Living (ADL) recognition approach for pervasive healthcare is discussed. The solution involves identifying a user's activity through the combined use of inertial sensors(accelerometer and gyroscopes) built-in to the smartphone along with the Zephyr BioHarness sensor. The patient wears the BioHarness sensor on the chest and puts a smartphone in the pants pocket as shown in Figure 4.9. These devices are continuously monitoring the 3D acceleration and rotation of a patient's body and transmitting data to the smartphone via a Bluetooth connection. Raw sensor data are collected, filtered and extracted into distinctive features on the smartphone. These features are then used to build and update statistical models using multiple machine learning algorithms. The trained machine learning classifier working on the smartphone is able to identify the user activities in real-time. Compared with other work using multiple kinds of wearable sensors [Maurer et al., 2006] [Tapia et al., 2008], our approach is more robust in terms of flexibility and scalability. Unlike other approaches [Kwapisz et al., 2010] [Khan et al., 2010] [Stefan et al.,

Figure 4.9: Activity recognition using smartphone associated BAN

2012], using only a single smartphone to detect activity, with undistinguished performance due to the ambiguity of upper body movement tracking, the overall activity recognition rate of our approach achieved over 95% in a real usage environment.

4.4.3 Sensing Paradigms

Researchers discuss how much the user should be actively involved during the sensing activity (e.g. taking the phone out of the pocket to collect an activity sample). The work in this dissertation is founded on two sensing paradigms: participatory sensing [Burke et al., 2006], where the user actively engages in the data collection activity (i.e. the user manually determines how, when, what, and where to sample) and opportunistic sensing [Campbell et al., 2006], where the data collection stage is fully automated with no user involvement. Each of these sensing paradigms presents significant trade-offs.

Participatory sensing places a higher burden or cost on the user. For example, manually selecting data to collect and then sampling it. An advantage is that complex operations can be supported by leveraging the intelligence of the person. The drawback of participatory sensing is that the quality of data is dependent on participant enthusiasm to reliably collect sensing data. In our work, this paradigm is applied in the first stage to collect training data for supervised learning. The volunteers are asked to perform specific activities carrying the smartphone. There is a wireless controller application that enables the user to label the ongoing activity and start/stop recording the activity data. The labelled data are saved in data files for the purpose of machine learning.

Opportunistic sensing, on the other hand, has the benefit of a lower burden placed on the user, allowing large-scale data collection, analysis and sharing. In this case, sensing happens automatically and continuously when the system starts working. It is particularly useful for activity recognition, where each user's behaviour pattern may be hard to quantify and only accrues over a long time. The smartphone-based activity recognition application is designed to be robust and cope with possible changes in user behaviour patterns, and the classification model is required to be adaptive to different users. With opportunistic sensing, the smartphone is able to continuously gather a user's activity data in an unobtrusive way and autonomously re-train the machine learning classifier to optimize the model of activity patterns. One of the foremost challenges of using opportunistic sensing is the phone usage problem. For instance, the application wants to only take samples for activity feature extraction when the phone is in the user's pants pocket. These types of usage issues can be solved by using other embedded smartphone sensors(e.g. the light sensors to determine if the phone is out of the pocket).

4.4.4 Smartphone Sensing Issues

In spite of the increasingly powerful hardware platforms, there are still issues that limit smartphone sensing applications. The battery capacity of smartphones is still a limitation, which makes it impossible to run long-term sensing applications. Another issue is the unpredictable and unexpected use of the smartphones. Smartphones are often used on the go and in ways that are difficult to anticipate in advance. This complicates the use of machine learning models that may fail to generalize under unexpected environments. For example, to identify the user's activity correctly, the smartphone is supposed to be carried in the user's pants pocket. It may fail to infer the physical actions of the person if users carry their smartphones in different locations.

Programmability of the smartphone also remains a challenge. Different vendors offer different APIs, making porting the same sensing application to multi-platforms become an issue. It is useful for the vendors to think about and propose sensing mechanisms and APIs that could be standardized and adopted by different smartphone platforms. Moreover, the stability of these kinetic sensors varies with different devices. For instance, the sensor readings are unreliable with electrostatic interference in some older smartphones such as HTC Nexus One, while the sensor data are of much better quality in some recent released smartphones such as Samsung Galaxy III and HTC One.

Although smartphones continue to provide more computation, memory, storage, sensing, and communication bandwidth, the smartphone is still a resource-limited device if complex signal processing and inference are required. Signal processing and machine learning tasks can stress the resources of the phones in different ways. In particular, for healthcare applications that require continuous sensing, it is more resource demanding in that real-time processing and classification of the incoming stream of sensor data are required. To mitigate this problem, some smartphone sensing systems tend to trade off accuracy for lower resource usage by implementing simple algorithms that require less computation or a decreased amount of sensor data. Another strategy is to leverage client-server architecture where different sensor data processing stages are offloaded to back-end servers when possible [Cuervo et al., 2010]. In our work, the cloud infrastructure is exploited, whereby a thin client is run on the smartphone to handle real-time activity recognition while more resource-intensive tasks, such as training and evaluating a classification model, are offloaded to the cloud server. Being aware of battery consumption and network latency issues, feature extraction and some pre-processing of the raw data are performed locally in the smartphone to minimize the amount of data to be sent to the cloud server.

Last but not least, privacy and security are very sensitive issues for pervasive sensing that infer a user's activity, context, and surrounding conditions. Solutions to protect users' privacy for smartphone sensing applications have been proposed [Cornelius et al., 2008] [Ganti et al., 2008] [Ahmadi et al., 2010]. In the CARA system, the security issue is addressed on the server side by generating a private key for each user and developing a mechanism for user management. However, we still need to take into account the risk of local malware that can access a user's data stored on the device. This could be addressed by encryption of critical data and a secure communication protocol.

4.5 Conclusion

An at-home healthcare solution must detect and respond to the activities and conditions of the patient. A wireless sensor network (WSN) is an ideal technology platform for detecting and responding to health-relevant parameters such as movement, breathing, ECG, and daily activity. The WSN we used consists of: a wireless body area network (BAN) that can monitor various vital signs while providing real-time feedback to the patient and remote caregiver; smart home sensors deployed in a patient's home environment that provide real-time and extended monitoring of activity and wellbeing; as well as a smartphone carried by the user that detect daily activities. The WSN can deliver a long-term home monitoring service to provide assistance in diagnostics and identification of changes in a person's behavioural pattern. As well as offering excellent long-term care benefits, the always-on nature of the WSN means that it can identify and respond to anomalies in a timely manner. In particular, the BAN can provide notice of significant shifts in critical physiological parameters in order to prevent a health crisis.

The potential value of WSNs for pervasive healthcare can be further exploited when coupled with other technologies such as context-aware reasoning, telecare, data mining and cloud computing. By mining the extensive data set collected by the WSN and incorporating intelligent reasoning functions, the WSN can keep patient, family, and caregivers linked, while also establishing trends and detecting variations in patient health condition. Other quality-of-life issues, such as privacy, dignity, and convenience, are supported and enhanced by the ability to unobtrusively provide services in the patient's own home.

Chapter 5

Activity Recognition in Smart Home Environment

5.1 Introduction

The methodology of the activity recognition system is introduced in this chapter. The ability to accurately recognize and continuously monitor activities of daily living (ADLs) is one of the key features that the CARA system is expected to provide. ADL is associated with both physical and mental health and is a primary indicator of quality of life [Skelton and McLaughlin, 1996]. Indeed, some age-related neurological diseases (e.g. cognitive impairments, mild dementia and Parkinson's disease) have a direct impact on the ADL of the elderly [White et al., 2001]. Activity recognition has been studied as part of a pervasive healthcare solution to reduce the necessity for caregiver supervision of patients. With the addition of on-going pattern analysis, activity recognition can assist in identifying changes in a subject's routine.

In this dissertation, a robust ADL recognition approach for pervasive healthcare is developed. The solution involves identifying a user's activity through the combined use of inertial sensors(accelerometer and gyroscopes) built-in to the smartphone along with a wearable wireless sensor (e.g. Zephyr BioHarness), which provides the activity context in the CARA system. Raw sensor data are gathered and segmented. Various features are then extracted from segmented data and are used to build classification models using different machine learning algorithms. For the real-time classification of a person's activity, a hybrid classifier is developed by combining threshold-based methods for simple activity recognition and machine learning classification models for complex activity recognition. An adaptation mechanism is also incorporated by adapting a universal model, which is trained using the dataset of all users, to an individual user through an unsupervised learning scheme.

On the other hand, continuous gathering of data will result in huge amounts of data, especially if many users are involved. Ideally, one might want to keep a large amount of this raw data for future (and maybe different) analysis. It is also desirable to analyse the data to produce a compact model which can be used in the client for real-time analysis of new data. This motivates a cloud computing solution where data from many users can be stored and analysed efficiently, and then the compact results of the analysis can be downloaded and used in the client. As a result, a cloud-based data analysis framework is developed to provide an efficient means for data sharing and data analysis. This cloud-based approach is demonstrated using a case study of the activity monitoring system. It incorporates various machine learning algorithms to enhance and build classification models for each individual in parallel and to select the most suitable model for the user. As more activity data are gathered and continually uploaded to the cloud, the classification model is adapted using an unsupervised learning approach to produce enhanced models which are then downloaded on to the smartphone for improved real-time activity analysis.

The evolving cloud-based machine learning mechanism makes the activity recognition system become more customizable and self-adaptive. It also overcomes the limitation of the computational and storage resources of a smartphone. The evaluation results of the experiments conducted on eight participants indicate that the activity recognition system can robustly identify activities in real-time across multiple individuals: the average recognition rate of ADL is over 95% in a real usage environment.

5.2 ADLs in a Home Environment

As a context-aware system especially designed for pervasive healthcare, the CARA system can help extend independent living for the elderly in a smart home environment by monitoring the person and ambient changes to detect anomaly situations. To achieve that, it is important to detect human body posture as well as movement which can provide the basic activity context to support better healthcare reasoning functions.

The basic activity of daily living in the home environment can be divided into two categories: static posture and dynamic movement. Static posture indicates the state of the body which consists of sitting, standing, lying, bending and leaning back, whereas dynamic movement is the compilation of a series of multiple actions. The dynamic activities within a home environment that we considered in our system include:

- **Walking**: Subject walks from one room to another.
- **Running**: Subject jogs in the yard.
- **Walking Stairs**: Subject climbs up or down stairs.
- **Washing Hands**: Subject washes hands at the sink.
- **Sweeping**: Subject sweeps the living room area.
- **Falling**: Subject falls on the ground and remains lying afterwards.

5.3 Activity Recognition Overview

A comprehensive activity recognition application has been implemented and runs on an Android smartphone for the real-time classification of activities. The overall process of activity recognition is summarised in Figure 5.1 and further details are discussed in the following sections. The work flow is described as follows:

1. The machine learning classification model is loaded from the cloud into a smartphone.

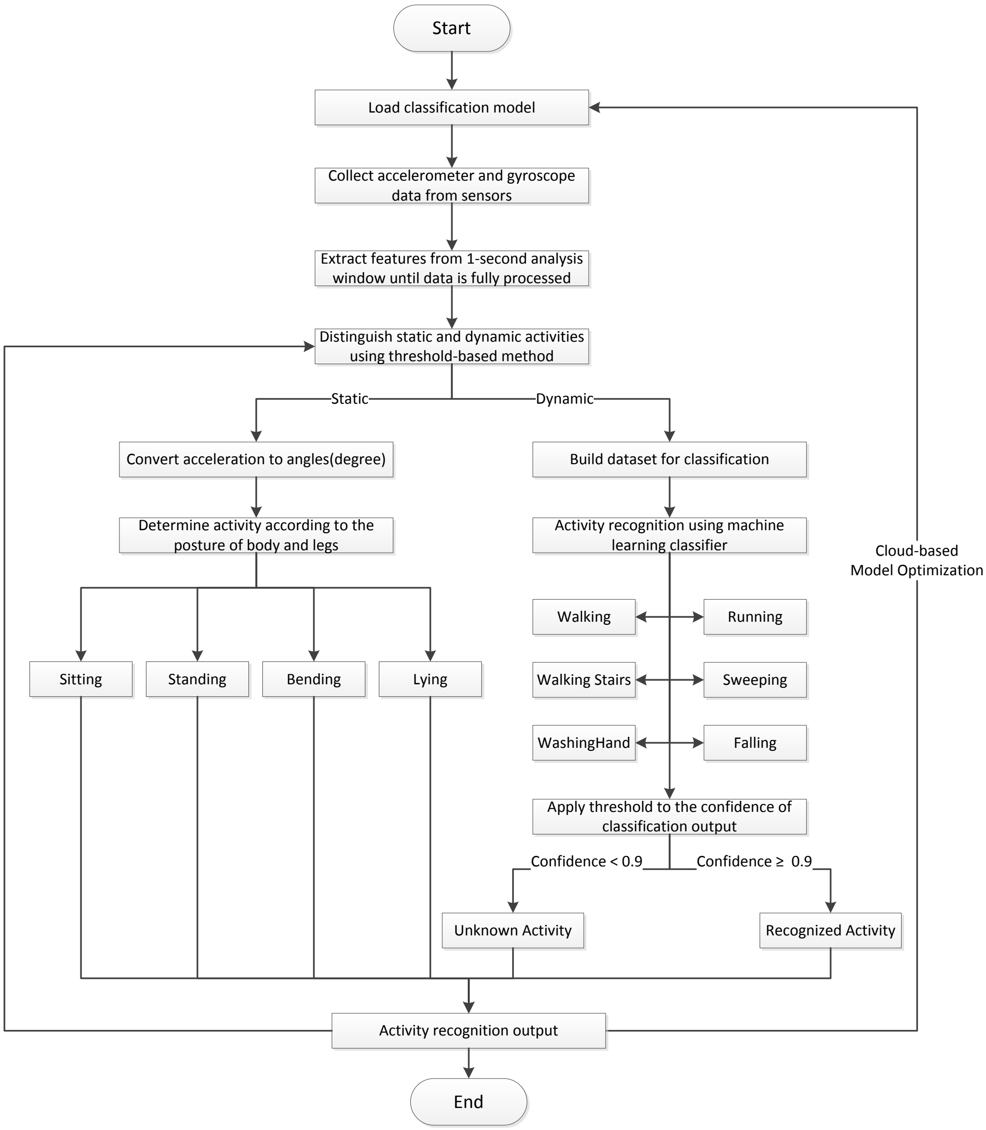


Figure 5.1: Flowchart of the activity recognition in the CARA system

2. Raw data are collected from BioHarness sensors and smartphone built-in accelerometer and gyroscope sensors.

3. The signals are then segmented and a 1-s window is moved over the signal and overlapped every 500ms.

4. Features corresponding to each window are extracted.

5. Static and dynamic activities are distinguished using a threshold-based method where a threshold value of acceleration is applied.

6. Static activities are divided into sitting, standing, bending, lying and leaning back by applying threshold angles for both trunk and thigh.

7. Dynamic activities are classified using machine learning classifiers based on the input dataset of extracted features.

8. Activities are labelled as correctly classified if the output confidence is high enough.

9. Features and the activity label of each detected case are stored in the data file.

10. The recorded data are used to retrain the classifier and optimize the classification model using the cloud-based data analysis framework.

Wearable wireless sensors and smartphone sensors generate huge amounts of raw data while the patient is under monitoring. However, raw data make no sense for identifying activities, they have need to be processed and built into models, which can be constructed via machine learning methods to describe salient features in the dataset. As a result, features of the data are then extracted and built into datasets which are used to build models. In this dissertation, we investigate different machine learning algorithms to build classification models for activity recognition (i.e. Decision Tree, Neural Network, Nearest Neighbour and Bayesian Network), each of which can be viewed as a blackbox that is capable of inferring the current movement based on the input dataset. However, each model comes from a unique algorithm approach and will perform differently for the same dataset. The best approach is to use cross-validation to determine which model performs best for a given dataset. As a consequence, the cloud-based data analysis approach is explored for the model optimization and adaptation.

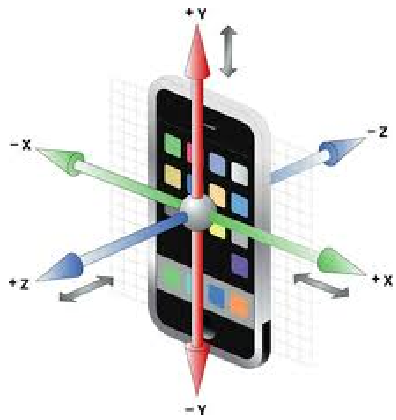
5.4 Data Collection

The first step of processing activity recognition is sensor data collection. Recently, two methods of collecting data have been extensively researched. The first method relies upon environmental sensors (e.g. RFID tags, cameras) to track aspects such as motion, location and object interaction. The second method uses a network of wearable sensors to track the acceleration of specific limbs as well as the body as a whole. Both of these methods have demonstrated impressive results in constrained laboratory settings. A major hurdle in implementing these systems outside of trials is how intrusive these sensors are. Environmental sensors involve a large investment in setting up and maintaining the system. Body Area Networks (BANs) are more unobtrusive and require less effort to set up. However, it is often impractical for the subject to wear sensors all over the body in normal daily life. The ubiquity of smartphones and their capability to support sensor data collection, processing and communications makes them a natural alternative to wearable sensors. As a result, using a smartphone as a primary device for data collection and processing increases the likelihood of data coverage and represents a minimal maintenance commitment and cost to the user.

Recently, some research has focussed on using smartphones as mobile sensing and computing platforms to achieve activity recognition [Kwapisz et al., 2010; Stefan et al., 2012]. A common issue of these approaches is the ambiguity of upper body movement tracking, which leads to low-performing of activity recognition. In this dissertation, we build on these approaches and extend them to enhance the ability of real-time identification of daily activities. Instead of using only a single smartphone, we also integrate a Zephyr BioHarness sensor which can be easily worn on the chest to track the trunk movement. The recognition rate of activities is dramatically improved according to the evaluation results.

The sensor data were collected from a Samsung Galaxy SIII mobile phone [Samsung Inc., 2013] and the wearable Zephyr BioHarness sensor as shown in Figure. 5.2(a) [Zephyr Inc., 2013]. The smartphone embedded triaxial accelerometer and gyroscope sensors measure the 3D-acceleration and orientation of the phone. The three axes of acceleration are dependent upon the orientation of the phone. The x-axis runs parallel to the width of the phone, the y-axis runs the length of the phone, and the z-axis runs perpendicular to the face of the phone, as shown in Figure. 5.2(b). The sensor integrated into the mobile device is easy to use without assistance and the phone can be carried comfortably for long periods of time [Choudhury et al., 2008].

(a) Zephyr BioHarness sensor

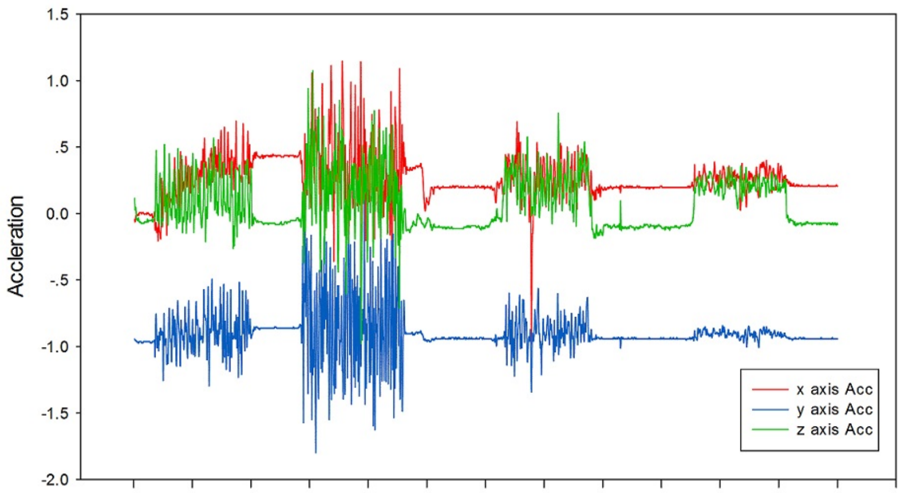


(b) Smartphone 3D-acceleration

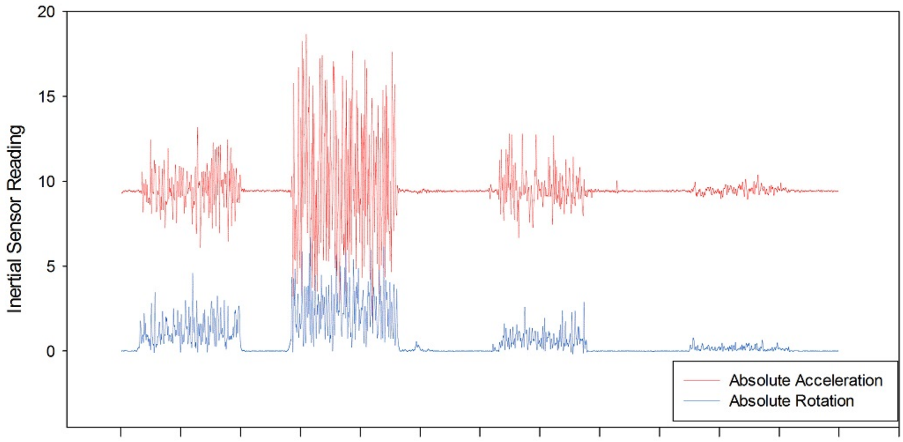
Figure 5.2: Body Area Network for sensor data collection

The data collection for supervised learning was conducted by performing experiments on eight postgraduate students at University College Cork, Ireland. Subjects carried the smartphone in their front pocket of their trousers and wore the BioHarness sensor on the chest. The BioHarness sensor data were transmitted to the smartphone through a Bluetooth connection in real-time. Subjects were asked to perform a series of activities while the real-time sensor readings were recorded for the purpose of training the machine learning classification models. Another android application running on a tablet device communicated with the smartphone through Bluetooth. It worked as a monitor as well as a controller that enables the selection of the on-going activity label and controls the smartphone to start or stop recording sensor data.

Nine channels of sensor readings, 3D-acceleration and orientation of the thigh and 3D-acceleration of the trunk with associated timestamps *(t, Ax, Ay, Az, Gx, Gy, Gz, Tx, Ty, Tz)* were collected as the training dataset and subsequently used for the evaluation of the activity classification algorithms. Figure. 5.3 shows the sample of raw sensor data collected by the smartphone. As to the



(a) 3-axis acceleration of the smartphone



(b) Absolute accelerometer and gyroscope sensor readings

Figure 5.3: Plot of the raw sensor data of the smartphone

sampling frequency, the sensors in the Android phone trigger an event whenever the accelerometer or gyroscope values change. The rate of events can be set to one of four thresholds: fastest, game, normal, and UI, with the fastest being the fastest sampling rate and UI being the slowest. In order to balance the speed of data processing and activity feature extraction, the sampling rate for the experiment was set to normal which is about 15Hz and is fast enough for the daily activity classification.

5.5 Features Extraction

After obtaining sensor readings, the machine learning classifier should be trained to build statistical classification models for real-time activity recognition. However, standard classifiers may not work well on the raw sensor data due to the characteristics of sensor readings, e.g. instability and noise. It is critical to extract features from the raw sensor data in order to improve the quality and reduce the quantity of readings. This is usually performed by breaking the continuous data into windows of a certain duration. In this dissertation, we explored a fixed length of a time window which is overlapped by one half of the window length. Thus, each window is a single instance, but any given data point contributes to two instances. This method has been shown to be effective in earlier work using accelerometer data [Bao and Intille, 2004]. We also experimented with 1-s, 2-s and 10-s time windows respectively for feature extraction. However, these yielded very similar results in terms of classification rate but with different lags for recognition, where a 10-s time window has the largest lag. As a result, we decide to use the 1-s time window in the process of feature extraction.

Table 5.1 summarizes a number of features extracted from each window. We compute both time-domain and frequency-domain features for each axis of accelerometer and gyroscope readings. The time-domain features measure the temporal variation of a signal, and consist of the following four features. The dynamic range is defined as *Min*, *Max* and *Mean*, which represents the minimum, maximum and the mean value in a time interval. The *Standard Deviation* is calculated to characterize the stability of a signal, normalized by the mean value of the readings in the interval. For the frequency domain, we consider two features. The *Zero-Crossing Rate* and *Mean-Crossing Rate* indicate the frequency of sign-changes along a signal in a time interval, which measures the rate of signal changes from positive to negative and from higher than the mean value to lower than the mean value. Additionally, we also investigate a feature concerning the *Angular Degree* of each axis acceleration signal, it is calculated based on the equation proposed by [Lyons et al., 2005]. Thus, in total a 66-dimensional feature vector is generated every second. Processed features were saved in the smartphone in an arff(Attribute-Relation File Format) file for data analysis.

The extracted features actually generate a pattern for a certain activity, and these patterns (consisting of feature vectors) are then used to train the classifiers and build machine learning classification models. Figure 5.4 demonstrates the samples of extracted standard deviation of two accelerometers and one gyroscope readings associated with different activities. As we can see, there is a significant difference between different activity patterns. More details of activity patterns can be found in the Appendix B: Activity Patterns with Accelerometer Signals.

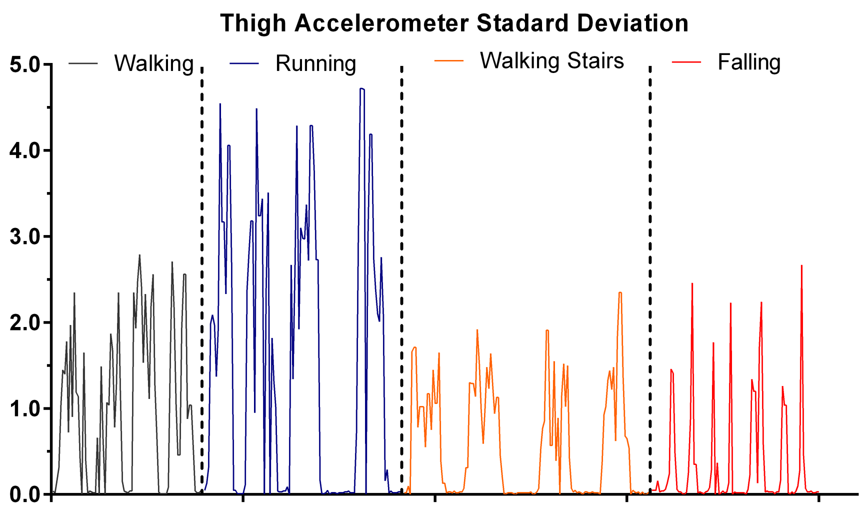
5.6 Hybrid Classifier For Activity Recognition

Activities of daily living in the home can be split into two categories. Simple motionless activities, such as sitting, standing, lying and bending, which correspond to static postures of the human body. These activities may be successfully recognized by using a threshold based method. [Lyons et al., 2005] concluded that using a minimum of two accelerometers, one mounted on the trunk and another mounted on the thigh, was sufficient to distinguish postures from movements. Complex dynamic activities, such as walking on stairs, cooking, sweeping and washing hands, can be represented as single repeated actions or even involve multiple overlapping actions. These activities can be recognized by identifying patterns of people's movement through machine learning classifiers.

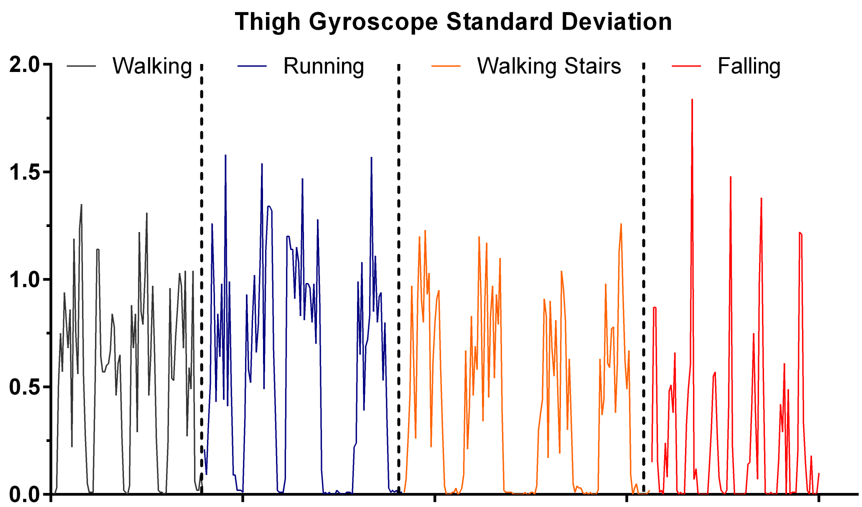
In this dissertation, we propose a multilayer hybrid classifier which is, in effect, a combination of the aforementioned techniques, and is able to distinguish static and dynamic activities and identify activities using different methods. To achieve this, a threshold based method is used to distinguish static and dynamic activities, a rule-based reasoning mechanism is applied to identify simple static activities, and various machine learning classifiers are applied to classify complex dynamic activities. In this case, the cost efficiency of activity recognition using a mobile device can be significantly improved. The smartphone, even with limited computing power, can perform intelligent real-time classification and this provides novel functionality in our solution.

Table 5.1: Description of features extraced from raw sensor data

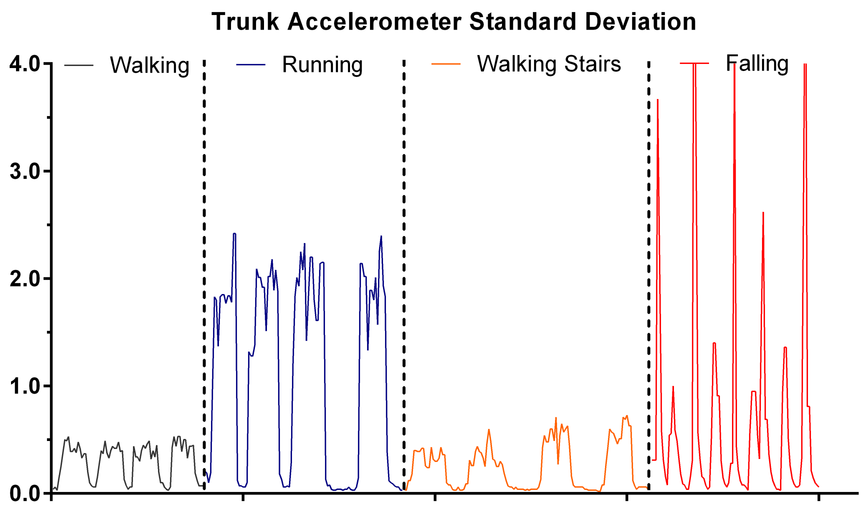
Feature	**Trunk Acceleration**	**Thigh Acceleration**	**Thigh Orientation**
Min	X, Y, Z, Absolute ACC	X, Y, Z, Absolute ACC	Azimuth, Pitch, Roll, Absolute GYRO
Max	X, Y, Z, Absolute ACC	X, Y, Z, Absolute ACC	Azimuth, Pitch, Roll, Absolute GYRO
Mean	X, Y, Z, Absolute ACC	X, Y, Z, Absolute ACC	Azimuth, Pitch, Roll, Absolute GYR
Standard Deviation	X, Y, Z, Peak ACC	X, Y, Z, Absolute ACC	Azimuth, Pitch, Roll, Absolute GYRO
Zero Cross	X, Y, Z	X, Y, Z	Azimuth, Pitch, Roll
Mean Cross	Absolute ACC	Absolute ACC	Absolute GYRO
Angular Degree	X, Y, Z	X, Y, Z	



(a) Standard Deviation of the acceleration of the smartphone



(b) Standard Deviation of the rotation of the smartphone



(c) Standard Deviation of the acceleration of the BioHarness sensor

Figure 5.4: Plot of the extracted Standard Deviation feature

5.6.1 Distinguishing Static and Dynamic Activities

To distinguish the static and dynamic activities, we investigated features extracted from segmented sensor readings and it turns out that the standard deviation is the most useful metric for classifying static and dynamic activities. The standard deviation indicates the variability of the accelerometer signal for each 1-s window of recorded data. High variability would be expected during dynamic activities whereas static activities result in low variability. Thresholds are then applied to the standard deviation of both smartphone and BioHarness accelerometer signals. Thresholds were determined empirically from the gathered data. The threshold for trunk acceleration was set at 0.25 m/s^2 and the threshold for thigh acceleration was set to 0.2 m/s^2 to make sure that all motionless activities are detected. Figure 5.5 shows an example of the standard deviation threshold for BioHarness accelerometer readings.

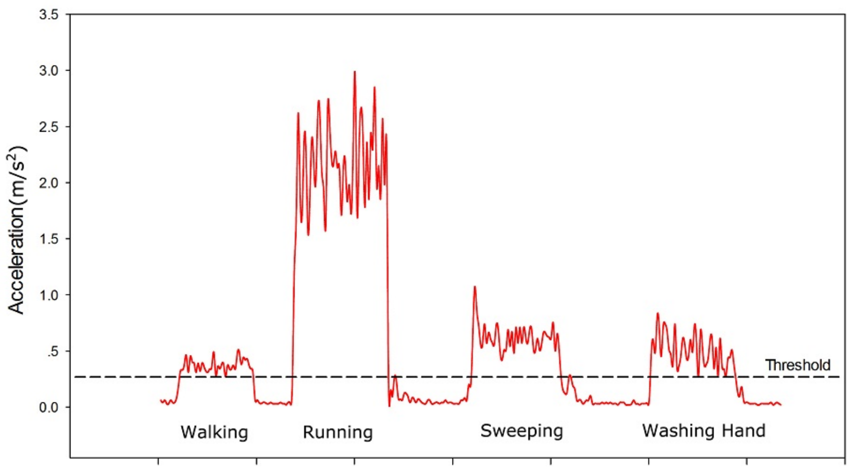


Figure 5.5: Standard Deviation threshold for trunk accelerometer signals

Algorithm 1 is implemented to determine the Activity State using these thresholds. Here, we classify the Activity State into three categories:

- **Stationary**: Subject is in a state of rest, one of various postures (e.g. lying, sitting, standing).

- **Transitional**: The state of the subject's activity transitions from static to dynamic or the other way round.
- **Active**: Subject is performing dynamic activities (e.g. walking, running, sweeping).

The activity state of a subject is determined by the standard deviations of both thigh and trunk absolute accelerations as well as the previous activity state. For example, we assume the subject was resting and the activity state was identified as stationary. The next moment, if both standard deviations of accelerations are above the threshold for that second, the activity state is changed to transitional, and if both of them are below the threshold, the activity state is still deemed stationary.

5.6.2 Real-time Classification

When the activity is deemed static, the mean accelerations over the one second window are converted to a corresponding inclination angle using the arc cosine transformation of Equation 5.1, where a is the mean acceleration of the corresponding axis (e.g. y-axis in our case), g is the gravity of the earth, θ *in degrees* corresponds to the angle of inclination for the trunk or thigh.

$$\theta_{degrees} = \frac{180}{\pi} \arccos(\frac{a}{g}) \tag{5.1}$$

Specific trunk and thigh inclination ranges are set based on the findings of [Lyons et al., 2005]. They proposed the best estimate threshold which accurately reflects real-life trunk and thigh ranges. However, they only considered three static activities, sitting, standing and lying. In this dissertation, we added two more static activities, bending and leaning back for classification. Figure 5.6 outlines the upper and lower threshold values for determining the trunk position. Similarly, thresholds are applied to the thigh acceleration. The overall thresholds applied for the thigh and trunk angles for each posture are listed in Table 5.2. The static activity is classified according to the position of the trunk and thigh through a rule-based approach. For instance, if the thigh position is detected as horizontal while the trunk is in any position except horizontal, the posture of the

```
input  : Extracted Feature Vector, Previous Activity State
output: Activity State
begin
    sdThigh ⟶ StandardDeviationoftheSmartphoneAcceleration;
    sdTrunk ⟶ StandardDeviationoftheBioHarnessAcceleration;
    foreach sdThigh and sdTrunk of the feature vector do
        switch Previous Activity State do
            case Transitional
                if sdThigh ⩽ 0.2 and sdTrunk ⩽ 0.25 then
                    ActivityState ⟵ Stationary;
                else
                    ActivityState ⟵ Active;
                end
            end
            case Active
                if sdThigh ⩽ 0.2 and sdTrunk ⩽ 0.25 then
                    ActivityState ⟵ Transitional;
                else
                    ActivityState ⟵ Active;
                end
            end
            case Stationary
                if sdThigh ⩾ 0.2 and sdTrunk ⩾ 0.25 then
                    ActivityState ⟵ Transitional;
                else
                    ActivityState ⟵ Stationary;
                end
            end
        endsw
    end
end
```

Algorithm 1: Threshold-based algorithm for distinguishing activity state subject is deemed sitting. During the experiment, detection accuracies of 98% and greater were achieved for each of the static activities (See Table 5.5).

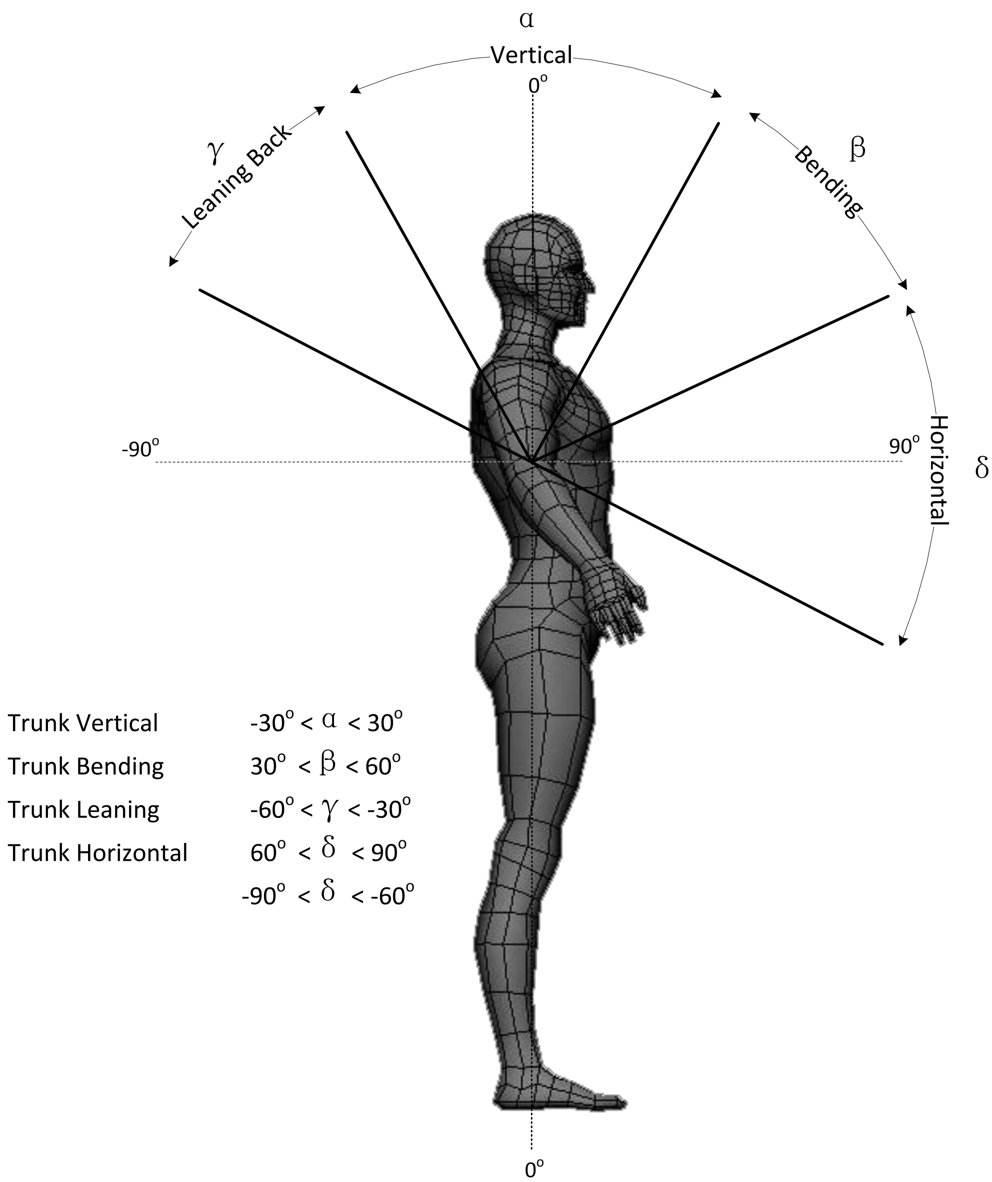


Figure 5.6: The inclination threshold arrangement for various trunk positions

For real-time recognition of dynamic activities, machine learning classifiers are employed to identify patterns of people's movement. The open source *Weka* ma-

Table 5.2: Threshold in degrees for thigh and trunk posture detection

Thigh Posture	**Upper Threshold**	**Lower Threshold**
Vertical	0	45
Horizontal	45	90
Trunk Posture	**Upper Threshold**	**Lower Threshold**
Vertical	-30	30
Bending	30	60
Horizontal	$\lvert \pm 90 \rvert$	$\lvert \pm 60 \rvert$
Leaning Back	-60	-30

chine learning package implemented by [Witten et al., 2011] is used in this study to develop the machine learning mechanism. This package provides a collection of machine learning algorithms for data mining tasks and can be integrated into other applications. In this dissertation, four popular machine learning algorithms are explored and evaluated:

- **Bayesian Network**: It learns from the training data by building a probability distribution table for each attribute. The strength of the Bayesian Network is that it is highly scalable and can learn incrementally because all it does is to count the observed variables and update the probability distribution table.

- **Decision Tree**: It is expressed as a recursive partition of the instance space. Each leaf in the tree is assigned to one class which shows the most appropriate target value. The way that instances are classified is by navigating them from the root of the tree down to a leaf. Some advantages of the Decision Trees are: as a classification it can handle datasets that have errors and missing values, the models it builds are transparent and understandable and their flexibility makes them applicable to a wide range of problems.

- **K-Nearest Neighbour**: It is an instance-based classifier that tries to classify an instance by comparing it to pre-classified examples. Classification is based on the distance function which measures the similarity value of two instances.

- **Neural Network**: It can be considered as a multi-layer perceptron that uses back propagation to classify instances. This multi-layer model enables the Neural Network to learn the non-linear relationship between the input and output.

We use the default parameters associated with each of the classifiers. The classifier obtains a classification model during the training stage. After training, it can predict the class membership for new instances using the classification model. The real-time classification is achieved by feeding the classification model with the input dataset of features extracted from real-time sensor data. Specifically, the machine learning classifier loads the classification model and then does an inference on the extracted features of sensor readings. The classification results are then produced with associated confidence values. The one with the highest confidence value is considered as the predicted activity.

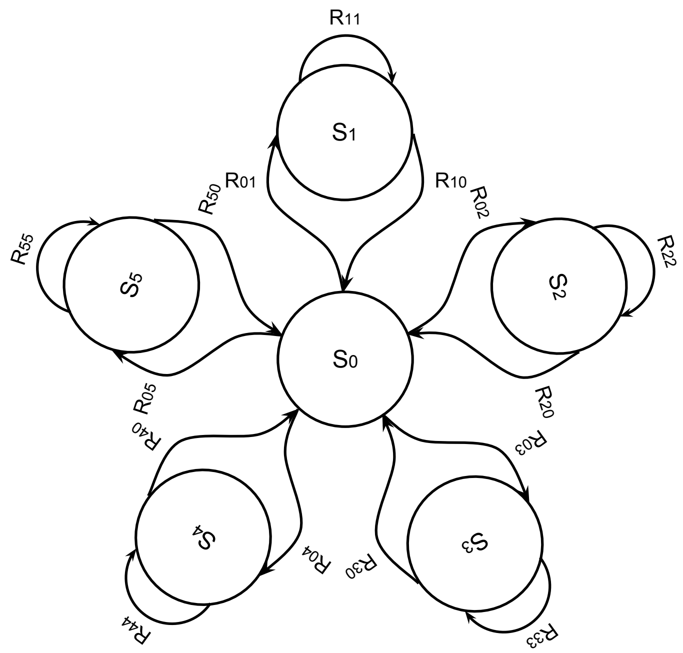


Figure 5.7: Diagram of the transition of activity states

As discussed in the previous section, we distinguish static activities from dynamic activities using a threshold based method, and a hybrid classifier is applied to handle different types of activities. During the experiment, we found that static activities and a single repeated dynamic activity can be relatively easy to identify. However, often the machine learning classifier misclassifies the activity when the movement of a subject is changing (e.g. from walking to running). To solve this problem, we introduce the idea of *Transitional State* to fill the gap between activities. Figure 5.7 illustrates the transition of different activity states, where S_0 is the transitional state and S_1-S_5 represents the state of each activity respectively. Some basic rules shown as R_{ii}, R_{ji} and R_{ij} are applied to the classification result to determine the current state of activity considering the previous state of activity. Specifically, whenever an activity is recognized based on a new dataset and is deemed to be different from the previous one, the activity state will be set to be transitional for a certain time period (e.g. 1 second) until the next activity is the same as the current one and the new dataset will not be recorded in the new training set. This method can improve the recognition rate by minimizing the number of misclassified activities, as well as excluding the disqualified activity from the training dataset. However, not all the daily activities can be recognized by the classification models. The unknown or untrained activity will either be misclassified as another activity or marked as an *Unknown State* according to the confidence value of the classification result. Algorithm 2 implements these rules:

```
input  : Predicted Activity: ACT, Previous Activity State: S_P
output: Current Activity State: S_C
begin
    S_T ⟶ TransitionalState;
    foreach ACT of Classification Result do
        if ACT ≠ S_P then
            S_C ⟵ S_T;
        else
            S_C ⟵ ACT;
        end
    end
end
```

Algorithm 2: Applying the transitional state when distinguishing two activities

5.6.3 Adaptive Classification Models

To start with, data collected from each individual are manually labelled for supervised learning. The individual's data are randomized and divided into training set and testing set. The classification model is built on the training set and evaluated on the testing set for each individual, which is referred to as the *Personalized Model* in this dissertation. Although the personalized model is able to provide a better results for a specific user, it may not be reliable or suitable for other users. In other words, it has the limitation of usability and scalability. In addition, the supervised learning scheme requires a great effort of manually labelling the activity data which is a tedious and burdensome task for the user. As a result, we studied the adaptation of a classification model and developed the model optimization framework which is designed to provide a one-model-fits-all solution.

Since the personalized model only works for a specific user, a generalized model is required to cater for all users. Hence, data collected from all the subjects are pooled together to build a universal classification model referred to as the *Universal Model* in the rest of the dissertation. However, it may not exactly fit each individual regarding a specific activity because of physical differences between individuals. In order to improve the classification performance, we introduced the idea of model adaptation to enhance the classification model.

As shown in Figure 5.8, all users start with a default classification model (the universal model), which in turn gets enhanced and adapted to each individual user for better performance when more activity data are available as users carry the phone. For each user, a *Filtered Model* was built based upon the dataset, where all misclassified instances are removed after evaluating the default model using a 10-fold cross validation method. Then the filtered model is employed in real-time classification. An unsupervised learning scheme is used to build an *Adapted Model* which is based on self-labelled data without any user input. It reuses the predicted label and confidence statistics generated by the filtered model during the inference process to select new training samples. The adaptation method determines whether a data sample is appropriate for adaptation according to the confidence level of the inference result. If the normalized likelihoods of the

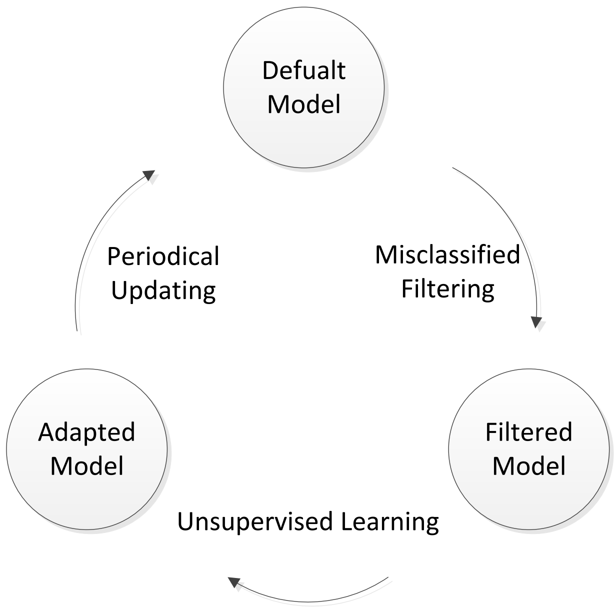


Figure 5.8: The process of model adaptation

two classes are quite close to each other, the classifier will have a low confidence for the prediction. Hence, a confidence threshold is used to filter the doubtful samples and only high confidence samples are added into the dataset to retrain the filtered model. Because unlabelled data are abundant, the system can use a very high threshold to ensure data quality. Finally, the universal model gets updated by the adapted model periodically. The principle of model adaptation is to keep updating the classification model for an individual user while the activity recognition task is carried on.

Table 5.3 presents the detail of the activity instances for each classification model. As we can see, the default model starts with 10218 instances in total and it gets filtered to 9340 instances before the filtered model is used for real-time activity classification. After the trial run, the adapted model is able to be built based on the filtered model of 11277 instances, it then becomes the new default model for the specific user. The classification model keeps being filtered and updated every time the new application cycle starts so that it gets more and

Table 5.3: Number of activity instances for different classification models

Activity	Default Model	Filtered Model	Adapted Model
Walking	1680	1516	1868
Running	1237	1057	1139
Walking Stairs	2169	1801	1995
Sweeping	1316	1243	1403
Washing Hands	833	775	883
Falling	69	52	52
Standing	510	494	822
Sitting	615	609	972
Lying	561	545	650
Bending	586	580	718
Leaning Back	462	458	557
Rolling	180	174	218
Total	10218	9340	11277

more adapted to an individual.

We evaluate the adapted model for each individual using different machine learning classifiers. The accuracy of the classifiers is tested using 10-fold cross validation for each classification model. Based on our experiments we have found that the Neural Network has provided slightly superior results in most of the cases. However, it turns out that the performance of different classifiers varies from person to person which mostly depends on the training dataset collected from each individual. A desirable approach should manipulate multiple machine learning classifiers to yield the best result. As a result, we investigate a cloud-based data analysis approach for machine learning.

5.7 Cloud-based Data Analytics

Although smartphones continue to provide more computation, memory, storage, sensing, and communication bandwidth, it is still a resource-limited device if complex signal processing and inference are required. Signal processing and machine learning algorithms consumes a large amount of time and computational resources, especially for activity recognition that require continuous sensing and real-time processing which is very resource demanding. As a consequence, a new incredibly scalable approach for data analysis and model adaptation has been explored by utilizing a cloud infrastructure with large storage capacity and powerful processing ability. This system can offer a cost-effective way to support data analysis technologies with high flexibility, scalability and availability for accessing data, discovering patterns, and deriving models.

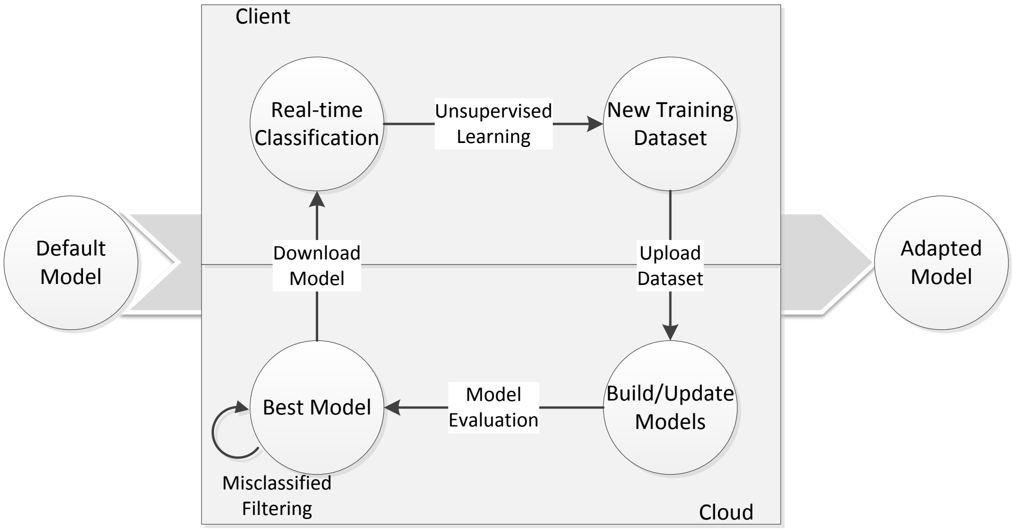


Figure 5.9: The process of model optimization

The cloud-based data analysis framework is designed to optimize a classification model through using multiple machine learning algorithms so that the model becomes customizable and adaptive for different individuals. As a cloud-centred approach, it provides high scalability and availability, with the solution ready to be deployed to hundreds of potential users without any change in the architecture. Since the generation and evaluation of the classification model is offloaded to the cloud, the mobile device can go beyond simple recording of data. In this case, the cloud-based solution provides an efficient means of data sharing and data mining which overcomes limitations of the mobile device.

The workflow of the cloud-based model adaptation approach is presented in Figure. 5.9. The original data collected from each subject are stored as the personal testing set. After supervised off-line training, the first universal model was built based on all subjects' data using a Neural Network classifier. It works as the default for a new user when there is no existing adapted model available for the user. On the mobile client, real-time activity recognition is carried out in the smartphone using the classification model while new training data are generated in an unsupervised manner and stored in the smartphone temporarily. The recorded data can be uploaded to the cloud periodically or on demand to retrain an adapted model and update the universal model.

On the cloud server, blob storage, multiple queues and worker roles are deployed for the purpose of analysing a user's activity data and producing the best classification model for each user. Blob storage in the cloud stores all the available activity data and classification models separately for each user. Each single user is assigned an independent container which contains individual activity data, including existing data and new uploaded training data, and the adapted classification model for this user.

5.7.1 A Basic Synchronous Approach

At first, the cloud-based data analysis framework is designed to work in a synchronized manner which is simple in terms of implementation. This provides us with reasonable evaluation results at the first stage of our experiment, and it lays the foundation for the improved asynchronous approach. Figure. 5.10 illustrates the structure of the synchronous approach. The framework consists of five types of queues:

- **Data Queue**: This queue is a general queue and is used to communicate between the client and the Controller node. When a client uploads a training file into the blob, the URL of the link plus the user id is added into the data queue.

- **Result Queue**: When the most suitable model is selected by the allocated worker role (evaluation node), the URL of the best model is added to that user's result queue. Unlike the data queue, which is unique and all the users have access to it, the result queue is independently created for each individual and doesn't provide public access.

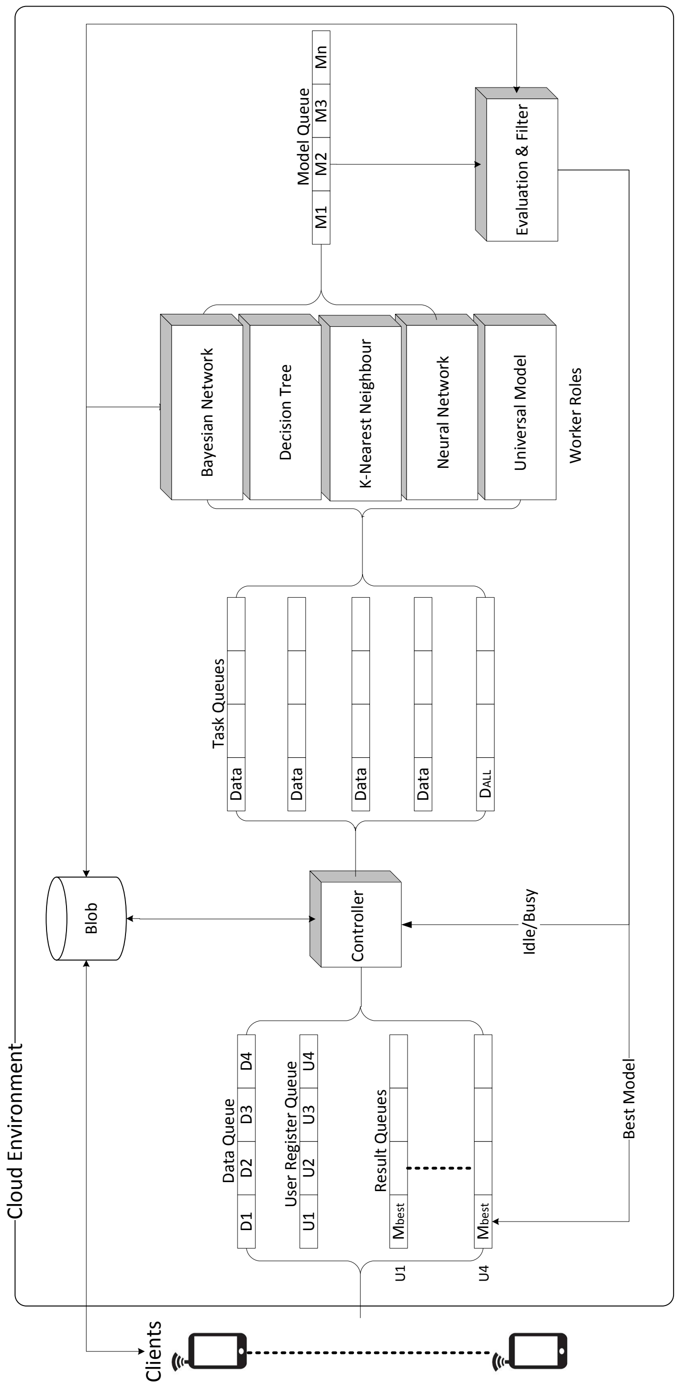


Figure 5.10: Structure of the synchronous cloud-based data analysis framework

- **Register Queue**: It is used to transfer the information of a new user. Once a new user is registered, the system assigns him/her a dedicated result queue and a blob container. The default model will be sent to the new user initially.

- **Task Queue**: The controller reads data from the data queue and assigns it into different task queues. There is a dedicated task queue for the universal node which will update the default model based on the data from all users.

- **Model Queue**: Once the model of each classifier is retrained and evaluated, the evaluation result will be sent to the evaluation node through the model queue.

The definition of each worker role and their responsibilities are listed below:

- **Controller Node**: The main role of the controller is to control and manage the flow of incoming data and assign new tasks to the task queue for further processing.

- **Machine Learning Nodes**: Each of the nodes reads a message from the task queue and starts building a classification model based on a specified machine learning classifier. When the model is retrained, it will be evaluated by using a manually labelled testing set.

- **Universal Node**: Unlike the other machine learning nodes, it is designed to deal with data from all the users to update the universal model using a dedicated classifier (e.g. Neural Network). The model is evaluated using a cross-validation method and misclassified instances are filtered out.

- **Evaluation Node**: The duty of this worker role is to select the most suitable model for a specific user. It can be achieved by comparing the evaluation results (e.g. error rate, precision, recall) of each classification model excluding the universal model. Moreover, a misclassify filter is also applied to filter the user's data once the best model is found.

The process of cloud-based model adaptation and optimization is described here. Once new training data are uploaded by the client, the URL of the training data is received by the controller node and assigned in different tasks to machine learning nodes. The data are then used to either build new classification models (in the case of a new user) or enhance the existing models. Moreover, the data are also fused with the available data from all users to update the default model in the universal node. In the deployment, four machine learning nodes work in parallel to process the data analysis tasks. Each of them is designed to handle one popular machine learning algorithm. The models are generated and evaluated for each machine learning classifier in the corresponding node. After gathering the evaluation results of all the models in the evaluation node, the most suitable model is selected according to the performance of each model. Data (combining new training data and any previously existing data) for that user are then filtered by a misclassify filter and stored in the blob storage replacing the previous data. Once the process is finished, the best model is sent back to the client device automatically. Thus, the new adapted model is available in the client for real-time classification.

The design is synchronous, since the controller node has to wait for the result from the evaluation node and assign the next task after the whole machine learning process is finished. The synchronous approach implies that the data assignment process is halted while machine learning tasks are being executed. Even in this cloud scenario, where machine learning algorithms execute in different worker roles, the controller is blocked until the best model is selected and every machine learning node returns to an idle state. In this case, all machine learning nodes work as a single unit and the actual processing time of a best model depends on the worst case of all machine learning algorithms. It turns out that the slowest algorithm (e.g. Neural Network) requires approximately 100 times of processing time compared to the fastest algorithm (e.g. Decision Tree) based on the experimental results. For instance, user A and user B upload their data to the cloud simultaneously. User A's data arrives first and gets assigned as the new task to machine learning nodes by the controller, and user B has to wait in line until user A's task is fully finished. It only takes 1 second to finish a model in the Decision Tree node while 2 minutes are required to build a model in the Neural Network node. So user A receives the best classification model after approximate 2 minutes, while the user B has to wait about 4 minutes until he gets his result. The blocking of the controller not only causes unnecessary delay but also wastes the resource of the cloud since there are several machine learning nodes remaining idle after they finish the current task. This violates the immediacy of synchronous communication as well as the availability of the cloud infrastructure.

The synchronous approach can be useful for real-time and immediate communication where the delay of data processing is not a problem. However, in our scenario, there is a huge delay in terms of building the classification model using specified machine learning algorithms, and this makes it an inappropriate approach for data analysis tasks. As a result, we design and develop an evolved asynchronous approach in order to improve the performance of data analysis in the cloud.

5.7.2 An Evolved Asynchronous Approach

In the asynchronous approach, the controller uses a send-and-forget approach that enables it to continue to execute after it assigns a new task. Consequently, the assignment procedure continues to run while the machine learning nodes are being invoked. Figure. 5.11 presents the structure of the asynchronous approach.

In the asynchronous structure, the machine learning node is no longer dedicated to a specific machine learning algorithm, rather that each machine learning node is designed to run any available machine learning algorithm so that it can handle any machine learning task on demand. Different from the synchronous approach, the controller now continuously and sequentially assigns a new task to the task queue until there is no more available data in the data queue. Each message in the task queue consists of the data structure of 5.2 which distinguishes the task from others.

$$\{USER\_ID, TASK\_ID, DATA\_URL, ALGORITHM\} \tag{5.2}$$

Any machine learning node that is currently idle is obliged to retrieve the next available message in the task queue and start processing data using the corresponding machine learning algorithm. In this case, the cloud resource is fully

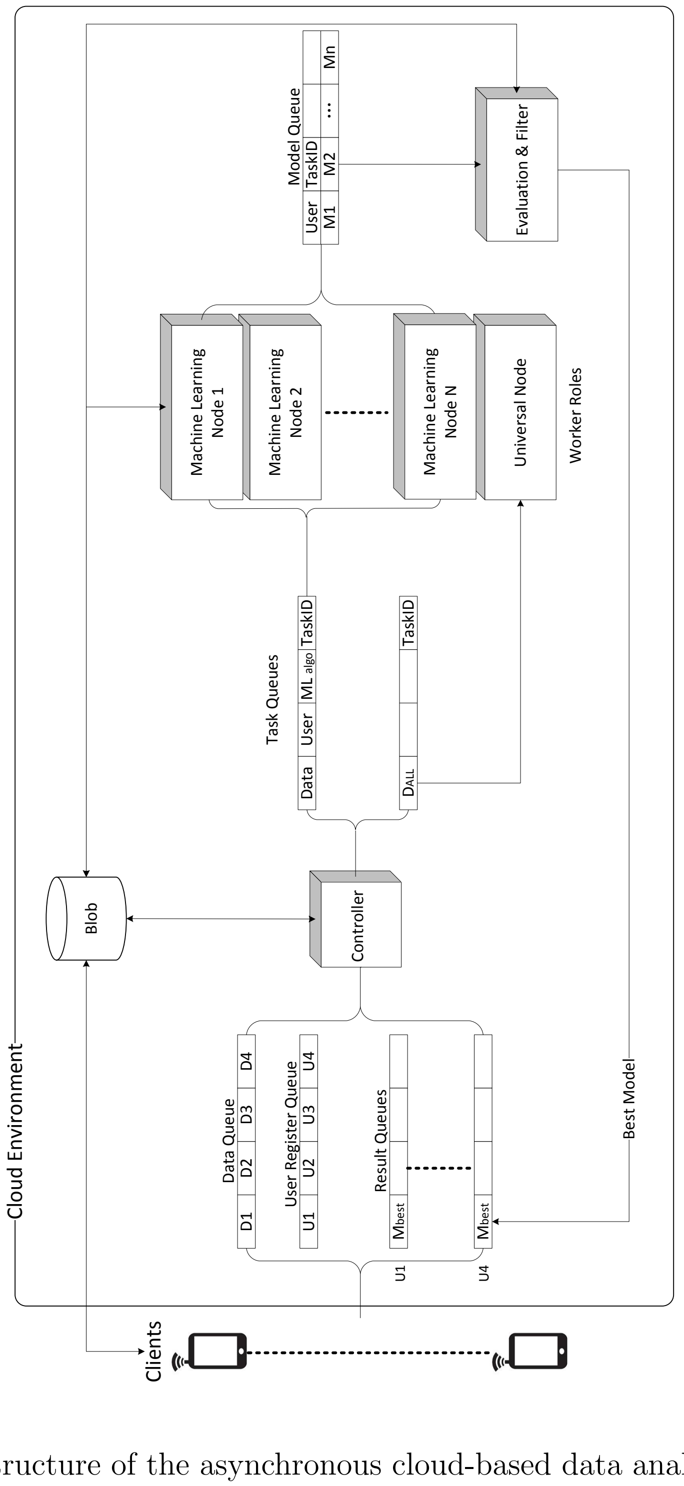


Figure 5.11: Structure of the asynchronous cloud-based data analysis framework

exploited and the efficiency of data analysis is improved. In addition, aware that the processing time of certain machine learning algorithm can be significantly long, it is necessary to develop an alternative mechanism for machine learning result delivery. As a result, we investigate two different approaches for the acquisition of a classification model in terms of priority in speed or priority in quality. In the normal case, the system will run a full-scale analysis on a user's dataset and produce the best classification model through using all available machine learning classifiers. This ensures the best quality model while the whole process is time consuming. Alternatively, in the case that there is an urgent demand for a new classification model, the system will produce the quickest classification model generated by whichever machine learning node finishes data analysis first while ignoring the results of the rest of the machine learning nodes that are still processing. This is an efficient approach when the quality of classification model is not a major concern. Thus, the user has the option to choose whether the classification model that he/she receives is selected based on the first-in-first-out approach or the best-among-all approach. Ideally, the controller node should incorporate an intelligent component that could automatically choose a suitable data analysis approach for the user under different circumstances.

This design of the cloud-based data analysis framework allows a plug-in style for machine learning nodes and classification algorithms. Accordingly, the number of machine learning nodes can be adjusted on demand to increase the throughput of the system. It takes advantage of the cloud infrastructure with a queue-based architecture where worker roles are asynchronously coupled which means that scaling or adding/removing instances does not affect the other worker roles. In this case, there is no hard dependency between cloud nodes. The framework provides horizontal ***scalability*** for the various machine learning tasks. Furthermore, Kmiecik [2013] pointed out that the size of the classifier model is independent of the size of the training data whereby even though the training dataset is very huge the model does not need to be very big. A huge amount of training data can be stored in the cloud while compact models generated from the training data can be downloaded and used in any mobile device. That can be interpreted as another advantage of this framework: improving the ***availability*** of the smartphone-based activity recognition system. In our experiments, training data of 54.4 MB, which contains one hour activity data (more than 10000 samples), only generates a model file of 170 KB (Decision Tree classifier). In the worse case scenario, a model file of 1.8 MB (Neural Network classifier) was built based on a dataset of 78.6 MB, which is acceptable for a smartphone-based application. Last but not least, the framework also minimizes the processing time of the machine learning tasks by leveraging the ***computing power*** of the cloud infrastructure in an asynchronous manner which enables a quicker response in the mobile client.

5.8 Experiments and Results

Eight volunteers (five male, three female) were involved in our experiments to help us evaluate the feasibility of the proposed approach. The activity data were collected in a home setting and recorded in data files for training and testing machine learning classifiers. During the experiment, subjects were asked to perform static and dynamic activities sequentially while carrying the smartphone and wearing the body sensor. There was a remote control application running on another Android tablet that controlled the smartphone through a Bluetooth connection to start or stop recording activity data and to label the recorded data on the fly. The recorded activity data were used to build classification models in a supervised learning manner. They also worked as the personal standard testing set to evaluate the adapted classification model generated for each individual.

Firstly, we evaluated the mechanism of activity recognition locally in a smartphone. We built the personalized model which is a completely user dependent approach. It requires training machine learning classifiers on each individual user's activity data and generates a user dependent model for each user. Clearly, this scheme is superior to others in terms of performance for a specific user, but its lack of general usability and scalability greatly limits its application. In addition, the supervised learning scheme requires a great effort of manually labelling the activity data which is a tedious task for the user. To improve it, we decided to move on to a semi-supervised approach where a universal model (also referred as the default model) is built based on manually labelled data from all users. As we discussed in the previous section, an unsupervised learning scheme was applied to gather additional training data to update the model so that it gets adapted to each individual user in a progressive manner. In this case, an adapted model was gradually generated for each user. Figure 5.12 presents the number of instances of each activity in the training set that is used to build the default model.

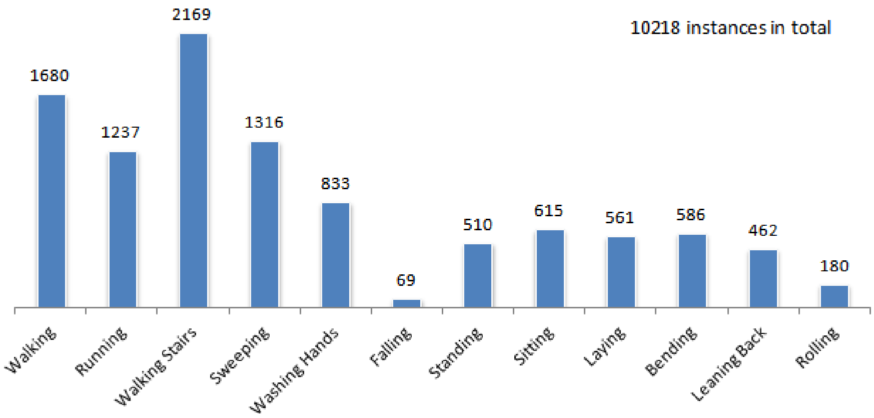
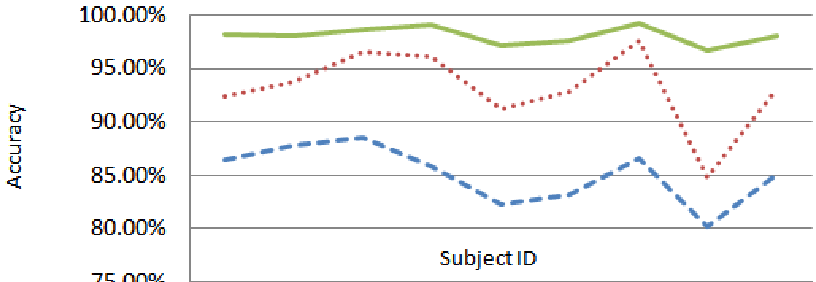


Figure 5.12: Example of overall activity instances of Default Model

Four different machine learning algorithms were investigated in the local trial run: *Bayesian Network*, *Decision Tree*, *K-Nearest Neighbour* and *Neural Network*. Each of them represents a typical approach of supervised learning for classification problems. The overall performance of each classification model obtained from different classifiers is summarized in Table 5.4. It can be observed that the classification accuracy of each adapted model remains consistently above 90%. Due to the transitional state we introduced in our classification mechanism, massive miscellaneous data are filtered during real-time data classification. As a result, high accuracy rate is achieved through the relatively clean data sample.

The best accuracy was obtained with the Decision Tree classifier. In a case study of the Bayesian network classifier, the accuracies of the different classification models for each of the eight subjects are presented in Figure 5.13. As shown in the plots the universal model is penalized for its one-size-fits-all philosophy. The universal model provides the lowest accuracy of 80.17%. The personalized model provides the highest accuracy of 99.19% and the adapted model yields 93.16% average accuracy.

The confusion matrix of considered daily activities is presented in Table 5.5. The result is for the k-nearest neighbour classifier using an adapted model. The



	1	2	3	4	5	6	7	8	Avg
Personalized	98.27	98.06	98.70	99.04	97.24	97.56	99.19	96.78	98.11
Adapted	92.45	93.76	96.54	96.09	91.15	92.87	97.57	84.82	93.16
Universal	86.39	87.77	88.53	85.94	82.26	83.19	86.65	80.17	85.11

Figure 5.13: Accuracy of different models using Bayesian Network classifier

Table 5.4: Weighted average accuracy using different classifiers and models

Classifier	**Precision**	**Recall**	**F-Score**	**MMC**	**Accuracy**
Personal Model					
Bayesian Network	98.3%	98.3%	98.3%	98.0%	98.3%
K-Nearest Neighbour	99.1%	99.1%	99.0%	98.9%	99.0%
Neural Network	98.8%	98.7%	98.7%	98.6%	98.7%
Decision Tree	99.0%	99.0%	99.0%	98.9%	99.0%
Default Model					
Bayesian Network	86.3%	85.1%	85.2%	82.8%	85.1%
K-Nearest Neighbour	98.1%	98.1%	98.1%	98.0%	98.1%
Neural Network	97.1%	96.9%	96.9%	96.6%	96.9%
Decision Tree	97.9%	97.9%	97.9%	97.6%	97.9%
Adapted Model					
Bayesian Network	93.2%	93.2%	93.1%	93.0%	93.2%
K-Nearest Neighbour	98.3%	98.3%	98.2%	98.0%	98.3%
Neural Network	98.3%	98.2%	98.2%	98.1%	98.2%
Decision Tree	98.5%	98.5%	98.5%	98.3%	98.5%

matrix shows that most errors occurred between *Falling*, *Standing* and *Lying*. Nevertheless, the average accuracy is still over 95% since the original default model was built using the supervised learning scheme. The reason for misclassifying falling is that it is a quick action which usually involves a series of static pre-actions and post-actions (e.g. standing and lying). So it is difficult to classify it through a machine learning model using a fixed sliding window method.

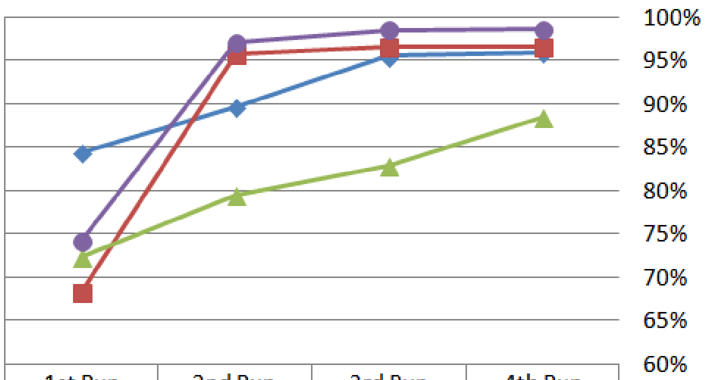
Table 5.5: Confusion matrix by using adapted K-Nearest Neighbour classifier

Activity	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**Accuracy**
WALKING (a)	1852	0	10	0	0	0	6	0	0	0	0	0	99.14%
RUNNING (b)	1	1105	32	1	0	0	0	0	0	0	0	0	97.01%
WALK STAIRS (c)	47	0	1937	4	0	0	1	4	0	0	2	0	97.09%
SWEEPING (d)	6	0	1	1378	14	0	4	0	0	0	0	0	98.22%
WASHING HANDS (e)	0	0	0	0	873	0	7	0	0	2	1	0	98.87%
FALLING (f)	0	0	2	1	5	32	6	1	2	3	0	0	61.54%
STANDING (g)	0	0	0	0	0	0	822	0	0	0	0	0	100%
SITTING (h)	0	0	0	0	0	0	0	972	0	0	0	0	100%
LYING (i)	0	0	0	0	0	0	0	1	649	0	0	0	99.85%
BENDING (j)	0	0	0	0	0	0	0	0	0	718	0	0	100%
LEANING BACK (k)	0	0	0	0	0	0	0	0	0	0	557	0	100%
ROLLING (l)	0	0	4	10	1	1	1	5	8	0	1	187	85.78%

However, a threshold based method would solve this problem because the change of activity readings are dramatic at the moment people fall. We evaluated this method and the recognition rate of falling was boosted from 62% to 91%. More details of the classification results can be found in Appendix C: Details of Activity Recognition Results.

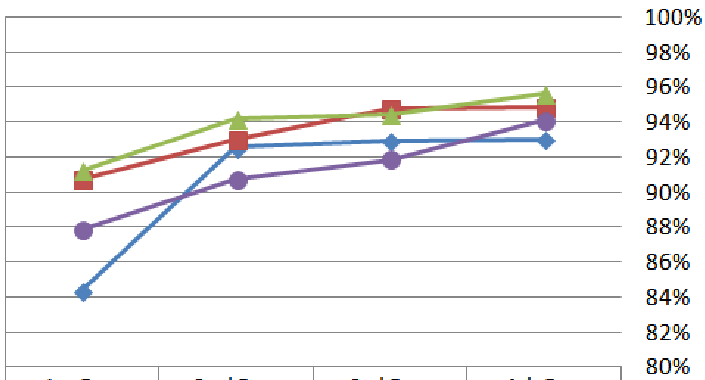
After evaluating the activity recognition approach on the local mobile client, we decided to carry out more experiments to evaluate the cloud-based approach. The cloud-based approach was first tested on two users(one female user A and one male user B) using the synchronous design. The manually labelled data gathered from these two users were used as the testing set to evaluate four classification models. Each user started with the default model and was given his/her own adapted model after the first run. The model was updated and optimized as the experiment proceeded. After four runs for each user, the overall accuracy of the adapted model was boosted and the average performance achieved was over 95% (see Fig. 5.14). We notice the fact that the performance of each machine learning algorithm varies from subject to subject. The detailed results for each classification model obtained from each run of User A are summarized in Table 5.6. It can be observed that the performance of each model in the first run is quite poor, because the default model was actually used for classification and it does not suit the user very well. However, after updating the adapted model and filtering misclassified instances in the next few runs, the accuracy of most classifiers remains consistently above 90% except for K-NN. The best accuracy was obtained from the Neural Network classifier, while the most efficient model was the Decision Tree. Note that the execution time for building and evaluating the model using the Decision Tree algorithm was around 1.5s on average while the processing time of the Neural Network linearly increases with the growth of the instances. In this case, the Decision Tree classifier was considered as the best classifier for building the adapted model for this user. The adapted model represents a middle ground between the default model and the personalized model. It got enhanced and eventually yielded 98% accuracy after four runs through model optimization and adaptation, and it is this model that is used for real-time activity recognition in the smartphone.

To evaluate the efficiency of the cloud-based data analysis framework, we con-



	1st Run	2nd Run	3rd Run	4th Run
Bayes Network	84.45%	89.67%	95.48%	95.97%
Decision Tree	68.37%	95.80%	96.61%	96.54%
K-NN	72.35%	79.51%	82.84%	88.43%
Neural Network	74.23%	97.04%	98.49%	98.55%

(a) Overall recognition accuracy for the female user A



	1st Run	2nd Run	3rd Run	4th Run
Bayes Network	84.38%	92.57%	92.93%	93.04%
Decision Tree	90.74%	93.02%	94.77%	94.85%
K-NN	91.23%	94.14%	94.41%	95.57%
Neural Network	87.83%	90.69%	91.86%	94.05%

(b) Overall recognition accuracy for the male user B

Figure 5.14: Cloud-based classification model optimization for different classifiers

Table 5.6: Performance of different classification models using the cloud-based data analysis framework

Classifier	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Score**	**Time (ms)**	**Accuracy**
First Run (1980 instances before filtering ⟶ 1874 instances after filtering)							
Decision Tree	0.684	0.034	0.839	0.684	0.657	1838	68.37%
Bayesian Network	0.845	0.011	0.931	0.845	0.860	2725	84.45%
K-Nearest Neighbour	0.724	0.044	0.811	0.724	0.684	4849	72.35%
Neural Network	0.742	0.040	0.811	0.742	0.720	84682	74.23%
Second Run (3493 instances before filtering ⟶ 3392 instances after filtering)							
Decision Tree	0.958	0.006	0.960	0.958	0.957	1248	95.80%
Bayesian Network	0.897	0.010	0.943	0.897	0.899	1482	89.67%
K-Nearest Neighbour	0.795	0.036	0.860	0.795	0.763	5504	79.51%
Neural Network	0.970	0.004	0.972	0.970	0.970	145111	97.04%
Third Run (5482 instances before filtering ⟶ 5403 instances after filtering)							
Decision Tree	0.966	0.006	0.967	0.966	0.966	1358	96.61%
Bayesian Network	0.955	0.005	0.967	0.955	0.958	1727	95.48%
K-Nearest Neighbour	0.828	0.031	0.873	0.828	0.810	7019	82.84%
Neural Network	0.985	0.002	0.985	0.985	0.985	227774	98.49%
Fourth Run (6790 instances before filtering ⟶ 6714 instances after filtering)							
Decision Tree	0.955	0.007	0.958	0.965	0.965	1692	96.54%
Bayesian Network	0.960	0.005	0.966	0.960	0.962	1357	95.97%
K-Nearest Neighbour	0.884	0.020	0.899	0.884	0.881	9041	88.43%
Neural Network	0.985	0.002	0.986	0.985	0.985	281971	98.55%

ducted another cloud experiment with the same dataset using the asynchronous approach, and compared the results to the previous synchronous approach. In this experiment, we asked User A and User B to upload their datasets simultaneously and measured the elapsed time for both tasks. Table 5.7 lists the actual processing time for the two users' best classification models and the waiting time for both users to receive the best classification model. It is clear that the asynchronous approach significantly reduced waiting time for the second user. As in the synchronous approach, User B's task can only be started after User A's task has been finished. So we conclude that the asynchronous messaging is a better approach when dealing with concurrent tasks for the cloud-based data analysis framework.

Table 5.7: Efficiency test: Synchronous vs Asynchronous approach

Synchronous Approach	**1st Run**	**2nd Run**	**3rd Run**	**4th Run**
User A Processing Time (ms)	85276	152932	232983	291587
User A Waiting Time (ms)	85276	152932	232983	291587
User B Processing Time (ms)	59680	101760	143565	210758
User B Waiting Time (ms)	144956	254692	376548	502345
Asynchronous Approach				
User A Processing Time (ms)	88149	148095	270207	302478
User A Waiting Time (ms)	88149	148095	270207	302478
User B Processing Time (ms)	62473	108415	186713	227724
User B Waiting Time (ms)	62473	108415	186713	227724

Table 5.8: Feature ranking evaluated by two different methods

Rank	**Correlation**	**Subset**
1	AbsGyoMax	AbsAccMax
2	AbsGyoMean	AbsAccMin
3	GyoX Standard Deviation	AbsAccMean
4	AbsGyo Standard Deviation	AbsAcc Standard Deviation
5	GyoXMax	AccXMean
6	GyoY Standard Deviation	AccX Standard Deviation
7	AccY Standard Deviation	AccXZeroCross
8	AccZ Standard Deviation	AccYMin
9	AbsAcc Standard Deviation	AccY Standard Deviation
10	GyoXMin	AccZMin

To study the importance of each extracted feature for the purpose of machine learning classification, we measure the correlation of the different features using information gain based on the attribute ranker search method [Witten et al., 2011], which evaluates the worth of an attribute by measuring the correlation between it and the class. We also evaluate the worth of a subset of features using the greedy stepwise method [Witten et al., 2011] by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. Table 5.8 presents the top 10 features evaluated on all user data using both methods. It is clear that the smartphone's accelerometer and gyroscope reading are ranked as the most important signals, and standard deviations of signals are selected as the most predictive features.

5.9 Conclusion

A robust system of daily activity recognition is designed to provide activity context for the CARA system using a hierarchical classification method by combining rule-based reasoning and multi-classifier machine learning algorithms. An Android smartphone and an additional motion sensor are placed on the thigh and chest, respectively, to provide the basic sensor readings. Human daily activities can be naturally represented through hierarchies, such as motional (dynamic) and motionless (static). Firstly, a threshold-based mechanism was used to separate the sensing data into two groups: static and dynamic. Static activities are identified based on the posture of the body which is calculated from the acceleration, and dynamic activities are classified by using adapted machine learning classification models. By utilizing the cloud infrastructure, the system provides high scalability and availability for data analysis and machine learning model optimization. Data processing and classification algorithms are implemented in the smartphone for real-time activity monitoring while the data analysis and evaluation are offloaded to the cloud. The experimental results compare favourably with other work using body sensors [Cook, 2012]. Moreover, the performance of our approach shows a significant improvement in comparison to an approach using a single smartphone [Zhang et al., 2010]. This shows a lot of promise for using smartphones as an alternative to dedicated accelerometers as well as using the cloud-based data analysis framework to process machine learning tasks.

Chapter 6

Context-aware Reasoning Framework

6.1 Introduction

To provide pervasive healthcare services, the CARA system should be able to observe, interpret and reason about dynamic situations (both temporal and spatial) in a home environment [Mileo, 2010]. This is achieved through an intelligent reasoning component. There are several approaches to build a reasoning system [Riesbeck and Schank, 2013]. Among them, rule-based reasoning (RBR) is one of the most popular reasoning paradigms for implementing healthcare decision support systems, where it can be used to store and manipulate knowledge to interpret information in a useful way [Bassiliades et al., 2011]. A classic example of a RBR system is a domain-specific expert system that uses rules to make decisions or choices. For example, a RBR system can help a doctor choose the correct diagnosis based on a number of symptoms.

Although the straightforward RBR is a competent approach, it still has some unsatisfactory limitations. For example, RBR requires eliciting an explicit model from a domain where domain knowledge may be hard to be structured. In addition, very specific rules may be easy to apply and are reliable, but only apply to a narrow range of adaptation problems; whereas more abstract rules may span a broad range of potential adaptations but not provide domain-specific guidance [Fan et al., 2011]. Case-based reasoning (CBR) [Ma, 2005] is another approach targeting problem resolution which can be used to bypass the knowledge-acquisition bottleneck. Instead of relying solely on general knowledge of a problem domain, CBR is able to utilize the specific knowledge of previously experienced problems to solve a new problem by finding a similar past case, and reusing it in the new problem solution. Moreover, CBR also is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems. However, it requires an accumulation of sufficient previous cases to accomplish the reasoning task.

In this dissertation, we introduce a personalized, flexible and extensible hybrid reasoning framework for the CARA system which provides a comprehensive solution that combines context awareness, general domain knowledge, and automated intelligence for pervasive healthcare. As a part of the CARA system, the reasoning framework plays a crucial role by interpreting sensor data within a wide context, reasoning with all available knowledge for situation assessment, and reacting according to the reasoning output. We studied how the incorporation of fuzzy rule-based reasoning (FRBR) and CBR mechanisms enable the CARA system to become more robust and adaptive to a changing environment. In theory, case-based reasoning is capable of making an inference based on previous experience by solving new problems based on the solutions of similar past problems, and fuzzy rule-based reasoning makes use of domain knowledge represented in terms of fuzzy sets and rules to interpret useful information. The combination of these two approaches is achieved by adopting the fuzzy adaptation model in the CBR cycle to retrieve similar case and to revise the solution depending on the circumstance.

The context-aware hybrid reasoning framework is applicable to various domains. Especially in the medical field, the knowledge of experts does not only consist of rules, but a mixture of explicit knowledge and experience. Therefore, most medical knowledge based systems should consider two types of knowledge [Schmidt and Gierl, 2001]:

- **Objective knowledge** is fact-based, measurable and observable, and is usually defined in encyclopedias or textbooks and can be updated by experts.

- **Subjective knowledge** is interpreted based on personal opinions, beliefs and experiences, and is limited in space and time and varies from individuals.

Both sorts of knowledge can clearly be identified in the CARA system, where objective knowledge can be represented in the form of fuzzy membership functions and fuzzy rules, and subjective knowledge is contained in cases that identify the significant features. The limitations of objective knowledge can be solved by incrementally updating the cases, whereas the limitations of subjective knowledge can be overcome by adapting objective knowledge in the process of reusing cases. Moreover, we introduce the idea of query-sensitive similarity measures in the process of case retrieving where weights of contexts are dynamically adjusted based on the output of the fuzzy rule-based reasoning engine.

The CARA hybrid reasoning framework provides the basis for a pervasive healthcare solution in a smart home environment. While there is a need for the system to be as sophisticated and adaptable as possible, there is a danger that the system may become too complex and difficult to understand for both subject and caregiver. Transparency and accountability are therefore important concerns that must be addressed. These concerns are especially important as the system aims to provide intelligent assistance for (possibly vulnerable) individuals in possibly critical situations. The use of a structured natural language and incorporation of fuzzy logic which supports the subject or caregiver inspecting and maybe modifying the rules aims to satisfy this goal. Transparency and clarity in interpreting the reasoning outcomes are also important, and this concern leads to more critical analysis of the reasoning output and indicates the need for appropriate metrics to be used. This is achieved through a provenance-based result tracing approach which enables the user to have a better understanding of the reasoning result.

System accountability also relies on the correctness of the reasoning engine. For the rule-based reasoning, an important goal is that the rules used in the smart-home analysis should be consistent enough to prevent possible conflicts in the inference. A semantic-based approach is exploited for detecting the indication of possible conflicts between the rules to ensure the coherence of the rule-based reasoning. In this case, the hybrid reasoning framework incorporating both FRBR and CBR enables CARA to be more robust and sophisticated.

6.2 Hybrid Reasoning Framework

A pervasive healthcare system is an ambient intelligent system that is able to (i) reason about gathered data providing a context-aware interpretation of their meaning, (ii) support understanding and decision making and (iii) provide corresponding healthcare services. During the first stage of developing a reasoning engine for the CARA system, a fuzzy logic based context model and a related context-aware reasoning approach are investigated. This reasoning engine provides context-aware data modelling and representation as well as inference mechanisms that support remote patient monitoring and caregiver notification. Noteworthy about the work is the use of a fuzzy context model to deal with the imperfections of the data, and the use of both structure and hierarchy to control the application of rules in the context reasoning system. Although the model and rules can be specified and modified manually by medical experts, it is difficult to comprehensively cover all domain knowledge in terms of rules and fuzzy sets. On the other hand, the rules are limited to a specific domain and not automatically adaptable to the changing environment. In other words, it lacks a self-updating function and learning abilities. To improve that, we introduce a context-aware hybrid reasoning framework that combines FRBR and CBR. The key features and most notable benefits of our hybrid reasoning framework include:

- Hierarchical Context Modelling
- Context-aware Case Retrieving
- Fuzzy Validated Case Adaptation
- Provenance-based Result Tracing
- Semantic-based Rule Validation

The high-level interactions in the hybrid reasoning engine are presented in Figure 6.1. Raw data collected from sensors are processed and combined with

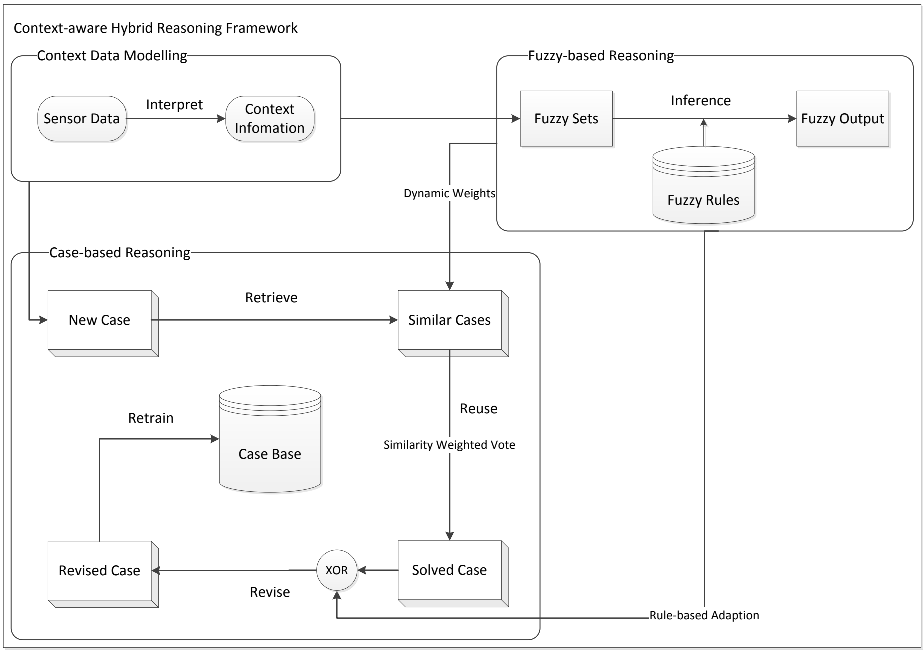


Figure 6.1: The structure of the context-aware hybrid reasoning framework

context knowledge through context modelling, producing contexts for the reasoning functions. After that, the reasoning system starts running a standard CBR cycle (Retrieve, Reuse, Revise and Retain) to perform anomaly detection and home automation. Simultaneously, the FRBR component loads fuzzy rules, which have been validated using a semantic-based approach, from the inference rule database and applies these rules to generate higher level contexts (e.g. medical conditions, and accident events) and further to identify the current situation of the patient (normal, abnormal or emergency). Unlike the standard CBR approach, we integrate the fuzzy rule-based reasoning mechanism into both the process of case retrieving and case adaptation. Specifically, the result of the fuzzy reasoning output is used to dynamically adjust weights of features or groups for case retrieving, and it also affects the adaptation of the retrieved solution to the new case. The intuition behind the heuristic weighting of features in the CBR is that the importance of features is indicated by their presence in any fuzzy rule relevant to the situation. In this case, the solution of a new case is revised according to solutions of the retrieved similar cases and the fuzzy reasoning outputs. The new case features and revised case solution are stored for enhancing the case base and subsequent additional analysis. Finally, if the detected situation is abnormal or an emergency, a notification or alarm is automatically sent to the remote monitoring server and an emergency service call can be triggered. The details of designing context models for both FRBR and CBR approaches, the principle of each reasoning mechanism and the integration of the two reasoning engines are discussed in the following sections.

6.3 Context Modelling

Context is any information that can be used to characterize the situation of an entity. In a context-aware system, the key feature is using context to provide relevant information and intelligent services to the user, where relevance depends on the particular task of the user. The study of [Mileo, 2010] indicates that there are certain entities in contexts that, in practice, are more important than others for home monitoring. These are location, identity, activity and time. We divide the basic context required in a pervasive healthcare environment into the following entities:

- *Person* entity to model the person, their clinical profile and movement.
- *Physiology* entity to model the vital signs of a person.
- *Area* entity to model rooms and surroundings in the home environment.
- *Object* entity to model objects or resources the person can interact with.

These entities can be observed and measured by pervasive sensing through the wireless sensor network (WSN) deployed in a smart home environment and the body area network (BAN) worn by a patient. However, in a real world deployment, it is sometimes difficult to obtain an accurate and well-defined context which could be classified as 'unambiguous' since the interpretation of sensed data as context is in general imperfect and incomplete. We contend that the hybrid reasoning framework we propose has the potential to minimize this problem. To deal with vagueness and uncertainty of the context, the FRBR method introduces an infinite number of truth-values for constructing fuzzy sets along a spectrum between perfect truth and perfect falsity (e.g. perfect truth may be represented by "1", and perfect falsity by "0"), and also allows us to represent knowledge in linguistic terms. On the other hand, the CBR approach addresses problems of incomplete data using feature weights and global similarity metrics. The objective of our context modelling approach is to provide an extensible and flexible infrastructure for the delivery and management of the information around a patient in the home environment.

6.3.1 Fuzzy Context Model

The main problem that we consider for context modelling is the following: given the current raw data, how can we structure the context, (i.e. the current values of relevant context parameters), and deal with data coming from multiple sources where part of the data may be erroneous or missing. For the FRBR approach, we adopt the fuzzy set to represent the relevant variables and to build low level and high level context models. An overview of the fuzzy context model is shown in Figure 6.2 where we structure the low level contexts into *Personal Context* and *Environmental Context*, and generate the high level contexts consisting of *Activity Event* and *Medical Condition*. The final reasoning output is inferred from the high level contexts which represent the current situation of a person. These are the contexts required in a ubiquitous context-aware environment.

All pieces of information gathered by sensors can be indexed as the attributes of context entities. In this dissertation, we map these attributes into discrete fuzzy sets in the FRBR engine. The principle of building fuzzy sets is to design appropriate membership functions. A membership function represents the magnitude of participation of each input. It defines functional overlap between inputs, and ultimately determines the output response [Kosko, 1992]. Some of the attributes associated with entities in our context model and their membership functions are listed in Table 6.1. These fuzzy sets can be used for the high level context interpretation and further for reasoning inference.

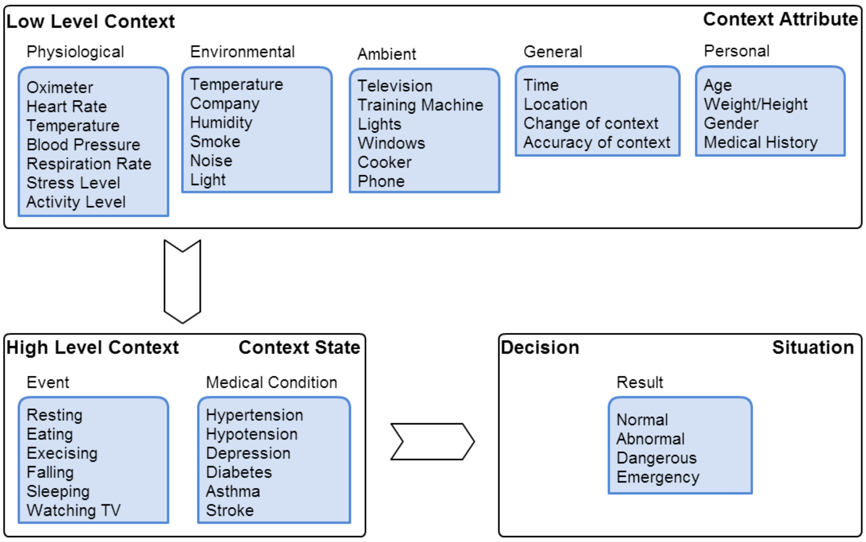


Figure 6.2: The fuzzy context model

Table 6.1: Examples of fuzzy sets representing Person and Area entities

Fuzzy Set	**Attributes**	**Description**
Age	{young, middle-aged, old}	Age of the person
Gender	{male, female}	Gender of the person
Time	{morning...late night}	Time of the day
Medical Condition	{hypertension...diabetes}	State of health
TV	{on, off}	Status of TV
Window	{open, close}	Status of windows
Temperature	{cold, warm, hot}	Room temperature
Light	{dark, regular, bright}	Brightness
Sound	{mute, regular, noisy}	Noise level
Humidity	{dry, normal, wet}	Humidity level
Location	{bedroom...living room}	Current location

6.3.2 Contexts in Case-based Reasoning

Research has proven that case based reasoning is a reliable approach to provide smart home solutions [Nguyen et al., 2009]. In a smart home scenario, contexts include the home environment and physical information about the subject that can be sensed. Thus, the context supplies the dynamic part of the environment and physiological information that is used to identify the current situation. These contexts can represent the location where the situation is occurring, the activity of a person is performing, and the time of a day, etc.

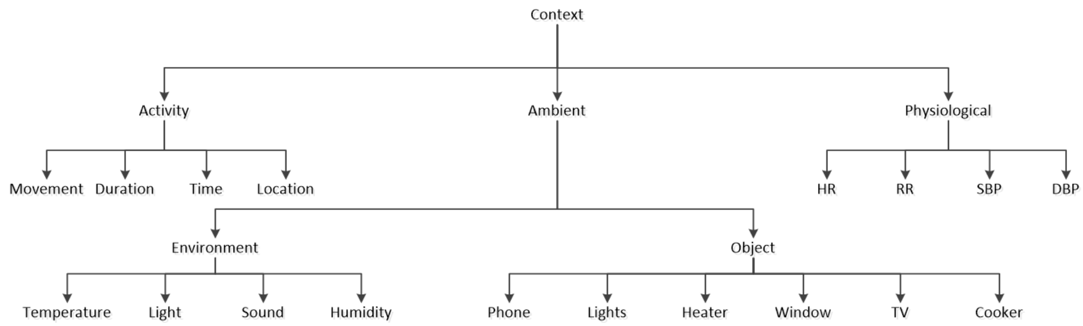


Figure 6.3: Grouped contexts for CBR

Figure 6.3 presents the structured contexts that are used in our CBR engine. In CBR, a case is the basic unit of data representation and processing. It usually consists two main parts: the *Features* of the situation, which consists of the perceived context; and the *Solution*, which consists of the goal to be achieved and the corresponding task to be accomplished. Note that, in the prototype of the CARA system, an interpreted solution involves both anomaly detection and home automation.

A notable improvement of our context model for CBR is the introduction of grouped features. The hierarchical context model was populated through observations done in the process of system evaluation. It turns out that not all the context is relevant to a specific situation. For example, if hypertension is detected during the night when the subject is watching TV, it certainly does not matter whether the lights are on or off or the humidity of the room is high or low. In order to improve the accuracy of case retrieval, we use different context groups and their group similarities to achieve the goals of the reasoning task. We divide relevant features into three groups: *Activity Features* which consists of the contexts used for recognizing user daily activities; *Ambient Features* which contains two sub-groups of contexts representing the home environment and the usage of home appliances; *Physiology Features* which includes physical readings obtained from wearable sensors. Instead of using a local similarity value of each individual feature for case retrieving, we calculate the group similarity based on local similarities and dynamic weights. This can boost the performance of our reasoning engine in dealing with real world problems.

6.4 Fuzzy Rule-based Reasoning (FRBR)

The declarative logical framework we use for knowledge representation and RBR in the CARA system is that of fuzzy logic, based on the fuzzy set theory proposed by Lotfi Zadeh [Zadeh, 1965]. The way that people think is inherently fuzzy. The way that we perceive the world is continually changing and cannot always be defined in true or false statements. Built on this theory, fuzzy logic is useful when working with vague, ambiguous, imprecise, noisy or missing information. Because of similar characteristics of sensor data, we adopt fuzzy logic to build a rule-based reasoning engine which represents domain knowledge in the system. A fuzzy reasoning system consists of three main parts: fuzzy sets, rules, and an inference engine. These components and the general architecture is shown in Figure 6.4.

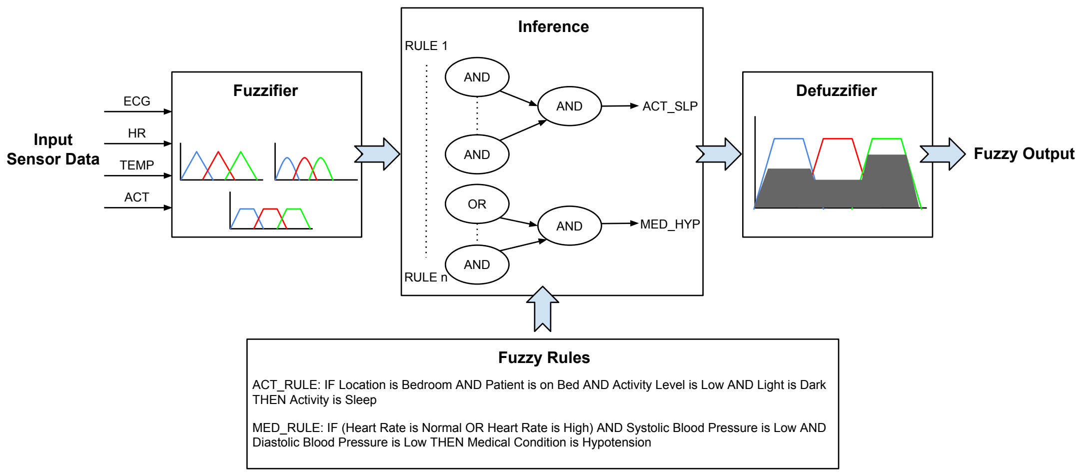


Figure 6.4: A generalized fuzzy logic system

The process of fuzzy reasoning can be broken down into three main steps [Bělohlávek and Klir, 2011]. The first of these is the fuzzification, this uses defined membership functions to process the inputs and to fuzzify them. These fuzzified inputs are then used in the second part, the rule-based inference system. This system uses previously defined linguistic rules to generate a fuzzy output. The fuzzy output is then defuzzified in the final process: defuzzification. This process will provide a real number as an output.

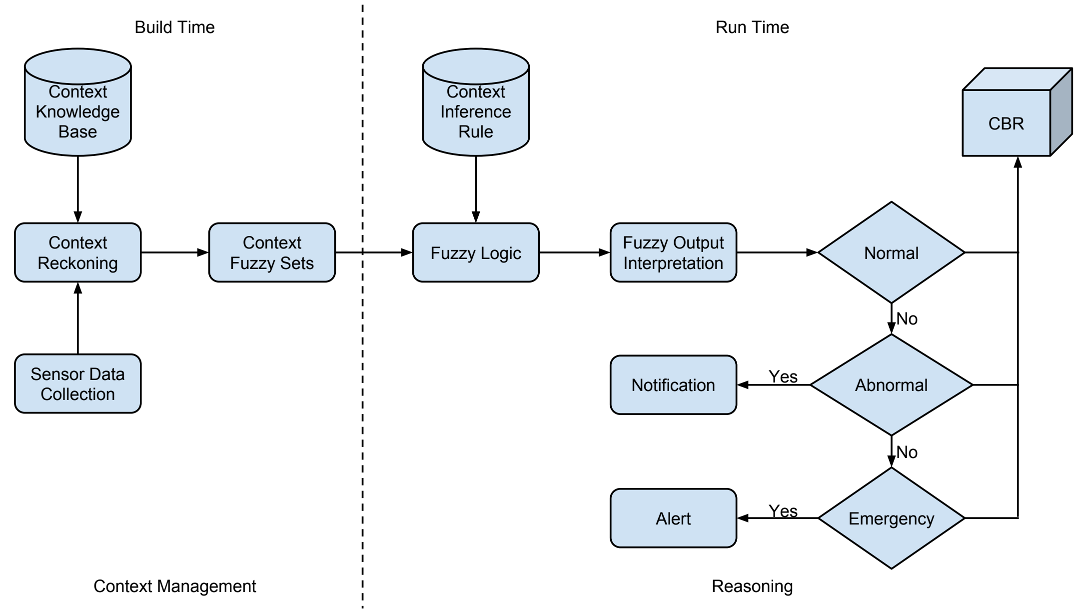


Figure 6.5: The workflow of the FRBR

Figure 6.5 presents the workflow of the CARA fuzzy reasoning engine. Low level contexts are obtained from raw data collected from sensors and are mapped into fuzzy sets according to predefined membership functions. After that, fuzzy rules loaded from the inference rule database are applied to fuzzy input sets to generate high level contexts. Finally, the rule engine identifies the current situation of the patient (normal, abnormal or emergency) based on the combination of high level contexts. The fuzzy output is stored for provenance and forwarded to assist the CBR component for making final decisions.

Fuzzy relations among the fuzzy sets are represented in terms of fuzzy rules in the reasoning process. These rules are stated as linguistic rules that relate different fuzzy sets and numbers. The general form of these rules are: "*if* x *is* A *then* y *is* B," where x and y are fuzzy numbers in the fuzzy sets A and B respectively. These fuzzy sets are defined by membership functions. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Inputs are combined logically using the logical operators to produce output response values for all expected inputs. The final conclusion is then combined into a logical sum for each membership function. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp value that best represents, and consistently represents the fuzzy set. The output can also be used as the confidence value, which indicates the strength of each output membership function [Zadeh, 1965]. An abstract view of the relations of all the fuzzy sets that implicate fuzzy rules is shown in Figure 6.6. As we can see, low level entities are gathered to generate high level contexts which are then used to infer a final decision.

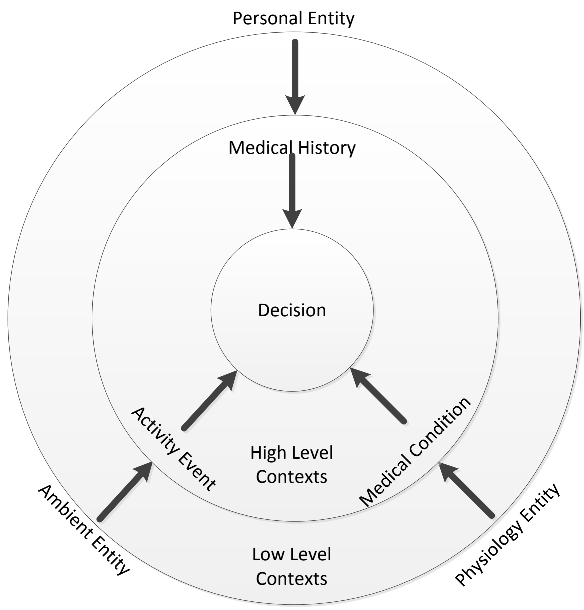


Figure 6.6: The abstract view of fuzzy relations

The hybrid architecture analyses the context information (derived from sensors and other information sources) using both CBR and FRBR. FRBR supplements CBR with expert insight and the ability to disambiguate between cases using domain knowledge. In line with the goal of transparency, rules are described, as much as possible, by means of linguistic terms using user-friendly

Table 6.2: Sample rules for generating high level contexts

Medical Context Generating Rules
IF(Age is Elderly or Middle Age) and Systolic Blood Pressure is Very High and Diastolic Blood Pressure is Very High **THEN** Medical is Pre-Hypertension **IF** Systolic Blood Pressure is Low and Diastolic Blood Pressure is Low **THEN** Medical is Hypotension
Event Context Generating Rules
IF On Bed and In Bedroom and Low Activity Level and Light is Dark and Sound is Mute **THEN** Activity is Sleeping **IF** TV On and In Living Room and (Low Activity Level or Normal Activity Level) and (Sound is Regular or Loud) **THEN** Activity is Watching TV

Table 6.3: Sample rules for anomaly detection in the smart home environment

Medical Context Associated Rules
IF SystolicBloodPressure is VeryHigh and DynamicBloodPressure is VeryHigh **THEN** Situation is Emergency **IF** Activity is not Exercising and (HeartRate is VeryHigh or RespirationRate is VeryHigh) **THEN** Situation is Alert
Event Context Associated Rules
IF Activity is Sleeping and (TV is ON or Cooker is ON or Lights is ON) **THEN** Situation is Warning **IF** (Activity is Eating or Activity is Cooking or Activity is Bathing or Activity is Exercising) and Time is Night and Lights is OFF **THEN** Situation is Alert

range-type values derived from expert knowledge. The FRBR approach allows the system to make use of imperfect data (e.g. lacks precision, is noisy or ambiguous) and can apply rules to reach a conclusion. In fact, it tries to model human-like decision making based on contextual information but does so in a more explicit way, mechanically and much faster. An example of generating high level context rules is given in Table 6.2, and Table 6.3 demonstrates some situation assessment rules associated with high level contexts. More detailed fuzzy rules can be found in Appendix D: Examples of Fuzzy Rules for Inference. Figure 6.7 presents the user interface for fuzzy rules and membership functions management in the CARA system. One notable aspect of our fuzzy reasoning engine is that all the hierarchical fuzzy rules and membership functions can be specified by a medical expert or a particular caregiver in linguistic terms, and the system will automatically translate them into computable objects. Thus a non-computer expert, can add the domain knowledge to our reasoning system. Furthermore, such rules and membership functions can also be modified by a patient under supervision.

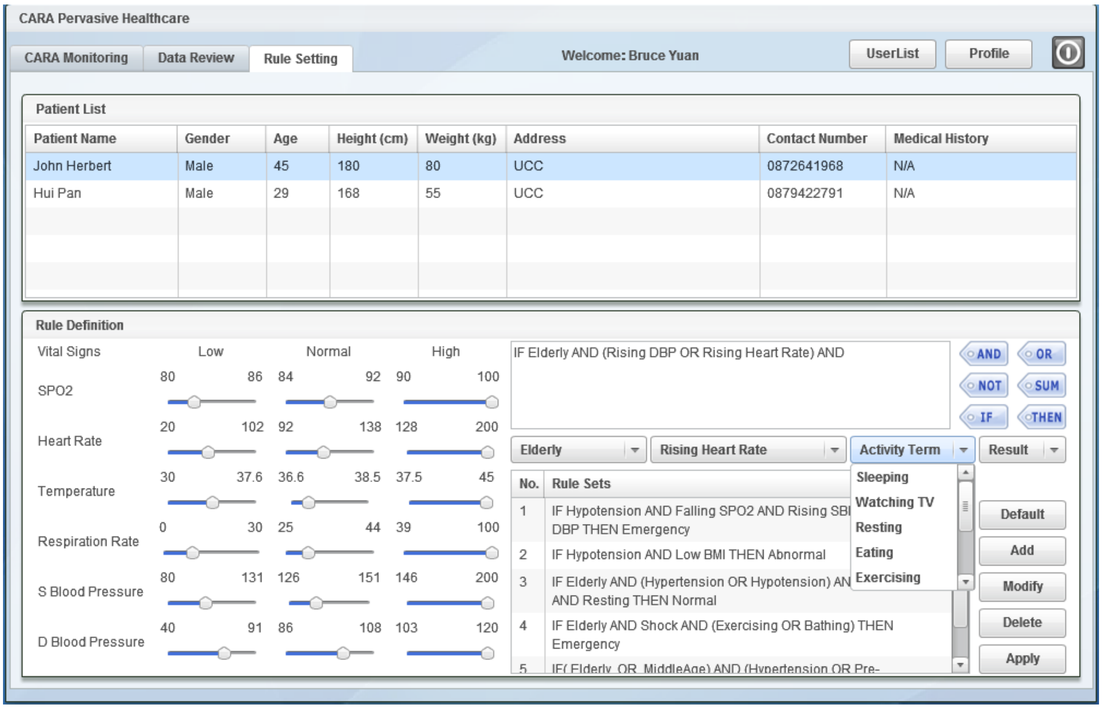


Figure 6.7: GUI for fuzzy sets and rules management

6.4.1 Semantic-based Rule Conflict Detection

Desired accountability of the FRBR engine largely depends on the coherence of the fuzzy rules. However, because the rules are configured manually by experts, conflicts may be introduced unintentionally into the rule set when a new rule is defined. Conflicts among these rules may result in an unexpected result from the reasoning engine. For example, a simple scenario of conflict is:

> **Rule 1: IF** Time is Night and Activity is not Sleeping **THEN** Action is Turn On Lights

Rule 2: IF Time is Night and Activity is Watching TV **THEN** Action is Turn Off Lights

For a specific fuzzy input, multiple rules may be eligible to be triggered. Assume that a person is watching TV during the night, the second rule is in conflict with the first one, because the conditional parts are matched while the action parts are non-compatible. There are many reasons why this kind of collision exists. The most reasonable one is that the rule designer does not have a clear and complete understanding of the previously defined rules when adding new rules to the rule base. This situation actually happens quite often when the number of rules in the rule base is increasing. So it is necessary to detect and remove incompatible rules to prevent the above scenario from happening. If conflicts among rules exist, the rule designer can amend the conflicting rule to prevent an undesired action taking place.

In this dissertation, a semantic-based logical mechanism is used for detecting conflict and redundancy among fuzzy rules; this is the approach that addresses the consistency issue. One of the existing conflict resolution strategies deals with rule conflicts dynamically during the run-time by choosing which one of the rules should be fired if more than one rule match working memory contents [Kuchar and Yang, 2000]. However, they only consider how the rules are prioritized but not how the rules are designed at the first place. As a consequence, it is sometimes difficult to maintain the rule base since more conflicting rules will be added into the rule base. This could also have impact on the performance of the rule engine. To reduce the redundancy of our rule base as well as to improve the run-time efficiency of our rule engine, we design and implement a static approach that checks for conflict and inconsistency whenever a new rule is defined and gives feedback to the user whether the new rule is eligible to be added, so that there are never any incompatible rules in the fuzzy rule base.

Conflict occurs when the objectives of two or more rules can not be simultaneously satisfied. The definition of conflicting rules is as follows:

Definition 1. Conflicting Rules

Given a rule base which has N fuzzy rules:

$R_i : if(C_i)then(A_i) \quad 1 <= i <= N$

We say that two pieces of rules Ri and Rj are conflict if Ai conflicts with A_j and $C_i \wedge C_j$ is satisfiable.

Based on this definition, we distinguish two types of conflict between rules:

- **Possible Condition Conflict**: we suppose A_i conflicts with A_j, the conjunction of C_i and C_j is satisfiable in some ways that Ai and Aj can both be fired.
- **Probable Condition Conflict**: we suppose A_i conflicts with A_j, C_i and C_j are logically equivalent, thus A_i and A_j are always simultaneously fired.

We also identify implausible rules that are logically correct in design but are worth to be reviewed by rule designer in the real-world application.

- **Implausible Rule**: Consider the following two rules:
  $R_1 : ifC_ithenA_i \quad R_2 : ifC_jthenA_i$
  where $C_j = \neg C_i$. Consequently, either C_i or C_j will be true, A_i is always fired. Thus, the rules are flagged to the user for review.

When detecting any possible conflict in the rule design phase, the system sends a notification with all possible conflicting rules to the user and lets the user decide whether to add the new rule into the rule base or not. As for the probable conflict and anomalies, the system is supposed to alert the user and prevent the new rule from being added to the rule base since there's a definite error that will cause troubles. Figure 6.8 illustrates the flow chart of the conflict detection mechanism.

The algorithm we proposed to solve the conflict detection problem involves three steps. The first step is to normalize the fuzzy rules. The second step is to check the compatibility between the action parts of the new rule and existing rules, and the third step is to check for different types of conflicts between rules.

The principle of the rule normalization is to transform the conditional part of a rule into disjunctive normal form (DNF) [Hazewinkel, 2001] and its negation form (NF) (e.g. Given a conditional formula $A \wedge B$, its NF is $\neg(A \wedge B)$).

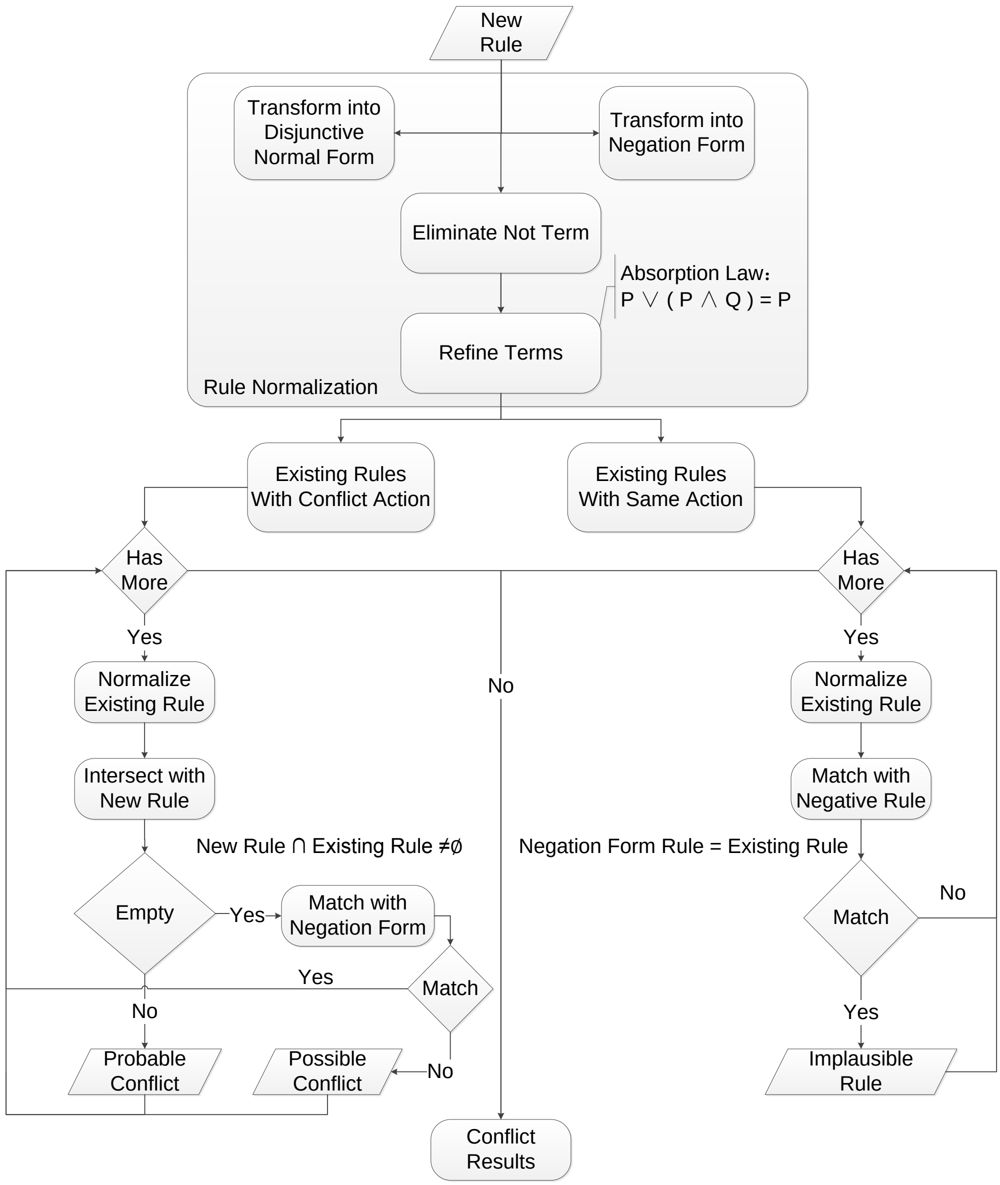


Figure 6.8: The workflow of the semantic-based conflicts detection approach

In this way, the condition part of the rule can be represented by a set of terms on which one or more actions are applied. The enforcement of another action on a sub-set of this set can provoke a conflict (the compatibility of action parts must be checked first). Based on Definition 2, any propositional formula can be expressed in DNF. Similarly, any propositional formula can be transformed into its NF as well.

Respecting characteristics of fuzzy rules that rules consist of linguistic terms which indicate the membership function of fuzzy sets, functionally complete logical connectives: conjunction, disjunction and negation $(\wedge, \vee, \neg)$ can be mapped into linguistic terms: AND, OR, NOT respectively.

Definition 2. DNF Transformation

Given an arbitrary conditional expression C, we can transform it into a disjunctive normal form:

$C = P_1 \vee P_2 \vee ... \vee P_n$

Where $P_1, P_2, ..., P_n$ are either:

- Literals which represent the membership function of fuzzy sets
- Statements of the form $Q_1 \wedge Q_2 \wedge ... \wedge Q_n$

Use law of Conjunction Distributes over Disjunction to convert any $Q_i = (R_1 \vee R_2 \vee ... \vee R_n)$, until all the $Q_1, Q_2, ..., Q_n$ are literals.

Every conjunctive form $Q_1, Q_2, ..., Q_n$ is referred as a clause in the algorithm.

However, the rule-based reasoning system is designed based on a Closed World Assumption [Duan and Cruz, 2011]. As a result, any other circumstances firing the action part of the rule besides the circumstances considered in the system are out of scope of the system. The system only considers the equivalence of the terms that has been designed in the set. The underlying model of the rule-based reasoning engine is designed based on fuzzy logic, which is a superset of conventional(Boolean) logic that has been extended to handle the concept of partial truth values between "completely true" and "completely false". The importance of fuzzy logic derives from the fact that most modes of human reasoning and especially common sense reasoning are approximate rather than exact. It may be considered that apparent conflicts among fuzzy rules would be handled by the fuzzy logic, since every term is a matter of degree in fuzzy sets. However, in fact, conflicting rules will result in a lower membership degree of the fuzzy output set which can lead to undesirable confidence value. Besides, not all natural phenomena considered in our reasoning system are fuzzy. Some of the terms are intended to have discrete values (e.g. ON/OFF,OPEN/CLOSE, or Temperature = HIGH-/MEDIUM/LOW). As a result, it is necessary to eliminate possible conflicts in the process of designing fuzzy rules.

In order to apply our conflict detection algorithm to all rules with apparent conflicts, we assume every term in the set is exclusive, and consequently, each specific clause of a rule is non-compatible with each other. It is inappropriate to simply convert a term to its logic negation form. For example, in a logical formula, the negation form of $Temperature\ is\ not\ High$ is presented as: $\neg(Temperature\ is\ High)$. While according to the terms of the set *Temperature*{*High, Medium, Low* }, the clause $Temperature\ is\ not\ High$ is the equivalent of the clause $Temperature\ is\ Medium$ or $Temperature\ is\ Low$. It may cause a conflict between rules that contain these clauses. In this case, the negation form of the clause should be presented as:

$$Temperature\ is\ not\ High = Temperature\ is\ Low \vee Temperature\ is\ Medium$$

Hence we can eliminate a NOT term by converting negative term into a positive term according to the corresponding set. The final sets of DNF and NF terms are refined using the Absorption Law:

$$P \vee (P \wedge Q) = P$$

This eliminates the redundant clauses produced during the transformation. An example of rule normalization is listed in Table 6.4:

Since the final output result of fuzzy rule-based reasoning is used for anomaly detection and home automation, the action part of fuzzy rules is built upon either the set of *Situation* {*Warning, Alert, Emergency*} or the set of *Automation* {*Turn on TV, Turn off Cooker, Open Window, ...*}, or the combination of both. To distinguish the conflict between actions of two fuzzy rules, we create a conflict table (see Table 6.5) representing the compatibility of actions so that the conflict between actions can be checked based on this table.

In the second step of the algorithm, we traverse the rule base to check the action of the existing rule against the action of the new rule. All the rules that

Table 6.4: An example of rule normalization process

Original Rule: **IF** (Activity is not Sleeping or Cooker is ON) and Time is Night **THEN** Situation is Alert and Automation is Turn OFF Cooker
DNF Terms: (Activity is not Sleeping and Time is Night) or (Cooker is ON and Time is Night)
NF Terms: (Activity is Sleeping or Time is not Night) and (Cooker is OFF or Time is not Night)
Finalized DNF Terms: (Activity is Resting and Time is Night) or (Activity is Watching TV and Time is Night) or (Activity is Bathing and Time is Night) or (Activity is Cooking and Time is Night) or (Activity is ... and Time is Night) or (Cooker is ON nd Time is Night)
Finalized NF Terms: (Activity is Sleeping and Cooker is OFF) or (Time is Morning) or (Time is Afternoon) or (Time is Evening)

Table 6.5: The compatibility of the action of rules (N: Non-compatible)

Action Pair	**Situation**			**Automation**									
	S_1	S_2	S_3	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
S_1:Warning		N	N										
S_2:Alert	N		N										
S_3:Emergency	N	N											
A_1:TV ON					N								
A_2:TV OFF				N									
A_3:Cooker ON							N						
A_4:Cooker OFF						N							
A_5:Heater ON									N				
A_6:Heater OFF								N					
A_7:Window Open											N		
A_8:Window Close										N			
A_9:Light ON													N
A_{10}:Light OFF												N	

have conflicting actions or the same action as the new rule are fetched and are going to be checked in the next step of the algorithm to detect the conflict between rules. There can be multiple terms in the action part of each rule. We call an action part that has multiple terms as a *composite-action*. The conflict between the single action of any two composite-actions may result in the conflict between these two composite-actions. Definition 3 gives a formal description of the conflict between two composite-actions. Two composite-actions are conflicting if there is at least one pair of actions in conflict.

Definition 3. Conflict Composite-actions

For two rules R_i and R_j:

$R_i : if(C_i)then(A_{i1}; A_{i2}; ...; A_{in})$

$R_j : if(C_j)then(A_{j1}; A_{i2}; ...; A_{jm})$

Their action parts conflict if and only if there is at least one pair of actions (A_{ip}, A_{jq}): A_{ip} conflicts with A_{jq} $(1 \leq p \leq n, 1 \leq q \leq m)$.

The last step is to identify different types of conflicts based on the output of the last two steps and to notify the rule designer with the conflict detection result. According to Definition 1, two rules are non-compatible if they establish the action and its opposite. Further we define the two types of conflicts: {*Possible Conflict, Probable Conflict*} and implausible rules in our conflict detection algorithm. Conflict detection is achieved by checking the intersection of the set of clauses in the DNF/NF between the new rule and existing rules. The implementation of this mechanism is described in Algorithm 3.

Considering the characteristics of fuzzy logic [Zadeh, 1965], a single input data could contribute to multiple fuzzy terms. In this case, multiple rules could be triggered during the inference process. For instance, let's consider the following two simple fuzzy rules:

Rule 1: IF Temperature is Low **THEN** Turn On Heater

Rule 2: IF Temperature is Medium **THEN** Turn Off Heater

Assuming the fuzzy membership function of Temperature set is designed to be: $High[30-50], Medium[15-35], Low[0-20]$, an input value of temperature:

Possible Conflict and Probable Conflict Detection:

input : Set of clauses of the new rule N_{dnf} in DNF, N_{nf} in NF
Collection of existing rules C_e with conflict action

output: Collection of possible conflicting rules C_h
Collection of probable conflicting rules C_c

```
begin
    foreach existing rule e in C_e do
        Normalize condition expression of e into set of clauses E_dnf in DNF;
        if N_dnf ∩ E_dnf ≠ ⊘ then
            C_c.add(e);
            Notify user for probable conflict;
        else if N_nf ≠ E_dnf then
            C_h.add(e);
            Notify user for possible conflict;
        else
            No conflict detected;
            continue;
        end
    end
    return C_c,C_h;
end
```

Implausible Rules Detection:

input : Set of clauses of the new rule N_{nf} in NF
Collection of existing rules C_e with same action

output: Collection of implausible rules C_i

```
begin
    foreach existing rule e in C_e do
        Normalize condition expression of e into set of clauses E_dnf in DNF;
        if N_nf = E_dnf then
            C_i.add(e);
            Notify user for the anomaly;
        else
            No anomaly detected;
            continue;
        end
    end
    return C_i;
end
```

Algorithm 3: Detecting different types of conflicts and anomalies

18 belongs to both Medium and Low sets. As a result, both rule 1 and rule 2 are fired by the fuzzy inference engine. However, only one action is definitized as the final output according to the degree of membership values of the fuzzy input and output sets. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Since the mechanism we designed is used to check conflict among rules semantically, the characteristics of fuzzy logic is not a concern for conflict detection in our case.

6.5 Case-based Reasoning (CBR)

CBR is a paradigm for combining problem-solving and learning that has become one of the most successful applied methods of artificial intelligence. CBR overcomes the difficulty of knowledge acquisition, can reason over domains that have not been fully understood or modelled and learn over time. The basic principle of CBR is to retrieve former, already solved problems similar to the current one and to attempt to modify those solutions to fit the current problem. The underlying idea is the assumption that similar problems have similar solutions. CBR has several advantages over traditional knowledge-based systems: it reduces the knowledge acquisition effort, requires less maintenance effort, improves over time and adapts to changes in the environment [Riesbeck and Schank, 2013]. These characteristics make it an ideal approach for the CARA system to detect anomalies or diagnose an illness from observed attributes. Figure 6.9 shows the standard CBR cycle described by [Aamodt, 1994].

Generally, a CBR application can be described by a cycle composed of the following four processes:

1. *RETRIEVE* the most similar case or cases.
2. *REUSE* the information and knowledge in that case to solve the problem.
3. *REVISE* the retrieved solution.
4. *RETAIN* the parts of this experience that is useful for future problem solving.

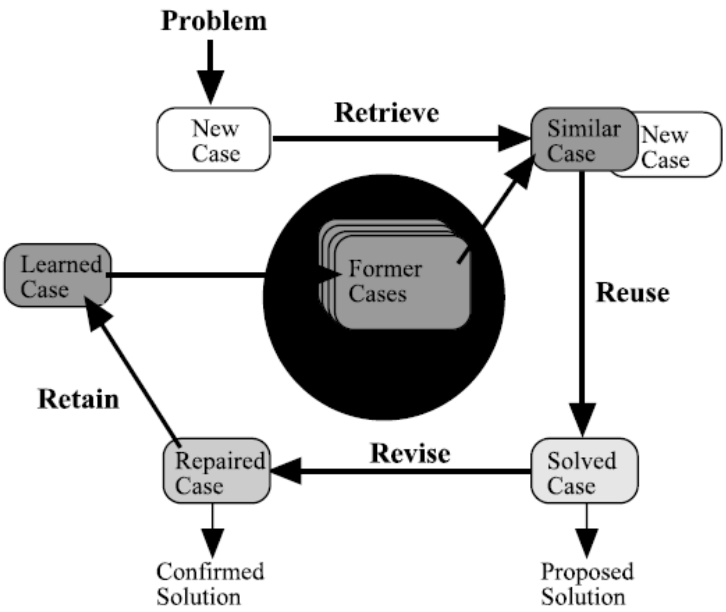


Figure 6.9: The standard CBR cycle

A new case represents a description of a problem and also defines the query. It is used to retrieve the most similar cases from the collection of previous cases. Similarity measure algorithms are applied to the case retrieval task. The similarity measures involved in case retrieving depend very much on the application domain. One commonly used method is the nearest neighbour retrieval [Ram and Wiratunga, 2011]. The case that has the maximum similarity value is retrieved as the most possible solution to the query case. In simple terms, a case that matches the query case on n number of features, will be retrieved rather than a case which matches on k number of features where $k < n$. Some features that are considered more important in a problem solving situation may have their importance denoted by weighting these features more heavily in the matching algorithm.

In terms of case adaptation (reuse and revision), if there are no important differences between a current and a similar case, a simple solution transfer is sufficient. Sometimes only a few substitutions are required, but in other situations, the adaptation is a very complicated process. There are no general methods for the modification of the retrieved cases to fit the actual design problem [Koiranen et al., 1998]. Our approach is to integrate a fuzzy rule based reasoning mechanism into the process of case adaptation.

6.5.1 Query Sensitive Retrieving

The first and also the most important step in the CBR cycle is the retrieval of previous cases that can be used to solve the current problem. Improving retrieval performance through more effective algorithms for similarity assessment has been the focus of a considerable amount of research. As one of the most popular algorithms for similarity measure, the k nearest neighbour (K-NN) method has been widely explored and evaluated in case retrieving. However, the normal K-NN algorithm for case retrieving has limitations as pointed out in [Aggarwal, 2001]. Finding the nearest neighbours in a high-dimensional space raises the following issues:

1. Lack of contrast: Two high-dimensional objects are unlikely to be very similar in all the dimensions.

2. Statistical sensitivity: The data points are rarely uniformly distributed, and for a pair of cases, there may be only relatively few features that are statistically significant for comparing those objects.

In our reasoning framework, a case query usually contains the following features listed in Table 6.6. The high-dimensional features of each query are unlikely to be uniformly similar and it is certain in some cases that some of the features are more important than the rest. To address these problems, we construct, together with context awareness, a query sensitive mechanism for the similarity measure. The term *Query Sensitive* means that the similarity measure changes depending on the current query object. In particular, the weights of features used for the similarity measure are automatically adjusted for each query. Specifically, we apply fuzzy rules to the input query and use the crisp value of the fuzzy output to dynamically adjust weights. We expect this method to be more accurate than the simple K-NN method for case retrieving. The query sensitive similarity measure

Table 6.6: Features of the case involved in a query

Features	**Type**
Activity	Enum Activity
Duration	Integer
Location	Enum Location
Time	Date
Day	Integer
Temperature	Double
Light	Double
Sound	Double
Humidity	Double
TV	Boolean
Heater	Boolean
Windows	Boolean
Lights	Boolean
Cooker	Boolean
Heart Rate	Integer
Respiration Rate	Integer
Systolic Blood Pressure	Integer
Diastolic Blood Pressure	Integer

function employed by our reasoning framework is shown in Equation 6.1.

$$Sim_g(Q,P) = \frac{\sum_{k=1}^{n} W_k * Sim_l(Q_k, P_k)}{\sum_{k=1}^{n} W_k} \tag{6.1}$$

In this formula, Sim_g (global similarity) of Q (query) and P (past case) is calculated based on Sim_l (local similarity) of Q_k (feature k of the query) and P_k (feature k of the past case) and the dynamic weight of the feature W_k. If k is the feature of a query, we use the term *weighted* to denote any function mapping W_k (weight of k) to the binary set $\{0,1\}$. We can readily define this function using fuzzy logic. Given a query Q, and a block of fuzzy rules F_{rules}, we can define a weighted function $W_{Q,F_{rules}} \rightarrow \{0,1\}$as follows:

$$W_{Q,F_{rules}}(k) = \begin{cases} MAX f(k,r) & \forall r, r \in F_{rules}, \\ & r\ is\ triggered\ during\ inference\ involving\ k \\ Default & otherwise \end{cases} \tag{6.2}$$

Where $f(k,r)$ is the degree of fuzzy output of rule r involving feature k. The weight of feature k is determined by the maximum value of $f(k,r)$ if one or multiple rules involving feature k are fired by the inference engine, otherwise, it is set to the default value of 0.1 if no rule involving feature k is fired. For instance, assuming a fuzzy rule, "if (Activity is Sleeping or Activity is Resting or Activity is Watching TV or Activity is Toileting) and (Systolic Blood Pressure is VeryHigh or Diastolic Blood Pressure is VeryHigh) then Situation is Alert" is evaluated and triggered during the reasoning process, and the crisp value of the output fuzzy membership function Situation {Alert} is 0.65. The weight of Systolic Blood Pressure used for the similar case retrieval is set to 0.65. As the result, the final weight of each feature of the query is dynamically adjusted according to the fuzzy output.

6.5.2 Similarity Weighted Voting

K most similar cases are retrieved after the query sensitive retrieving algorithm is applied to similarity measurement. In our system, for anomaly detection, the solutions of retrieved cases are supposed to be classified into *Normal, Warning, Alert, Emergency* groups. Normally, the possible solution of the given query can be predicted from the most similar case. However, under certain circumstance, the prediction maybe vary from the solution of the most similar case, but rather depends on the majority solutions of the retrieved cases. For instance, for a new case query, 5 most similar cases are retrieved for case reuse. Among them, the solution of the top one, which owns the similarity value of 0.87, is *Normal*. The solutions of the rest of cases are *Warning*, and the similarity values are 0.83, 0.79, 0.74, 0.56 respectively. In this case, we consider the predicted solution is most likely to be *Warning*. To determine the possible situation of the query, a similarity weighted voting mechanism is used in the voting decision during prediction. The

principle of similarity weighted voting method is to use the similarity value of each retrieved case as the weight to vote for the most reasonable solution. In other word, every nearest neighbour has a different influence on the prediction according to its similarity to the query. It is achieved in the following steps (the details of the similarity weighted voting algorithm are shown is Algorithm 4).
1. Classify K retrieval result into different groups according to the solution label.
2. Calculate total similarity of all retrieved cases.
3. Get the sum of similarity of each group.
4. Calculate the average of similarity of each group.
5. Use the group similarity to vote for prediction, the highest vote of the solution group is assigned as the predicted solution.
6. Calculate the confidence value of the predicted result according to the average similarity and total similarity.

To distinguish the predicted result from past cases, we apply a threshold to the confidence value of the predicted solution, which is used as a controller to balance the detection rate and the false alarm rate of the rule engine. The threshold ε can be set by the user. If a user chooses $\varepsilon = 0$, the rule engine takes into account all possible solutions in past cases$P$, and the determination of the solution for a unique query Q in the given P relies, in this case, on the voting result. Otherwise, the threshold ε can be considered as a level of decidability: if there exists no case $C, C \in P$ such that $Conf(Q, C) \geqslant \varepsilon$, then there is no already-solved problem sufficiently similar to Q and no solution can be proposed. In this case, we introduce the fuzzy adaptation model to deal with the uncertainty. A core idea of our reasoning framework is that domain knowledge, which is represented by fuzzy rules and fuzzy sets, is applied to both case retrieving and case adaptation.

6.5.3 Fuzzy Adaptation Model

As we discussed before, the CBR mechanism is capable of making analogies based on previous experience. However, to perform its reasoning functions, it requires sufficient solved cases in a case base as the reasoning resources. It can only work with a supervised learning scheme where previous cases have already been recorded and labelled with solutions. Labelling solutions can be a tedious and

```
input  : Collection of cases from Retrieval Result
output: Predicted Solution
begin
    votes ⟵ New HashMap;
    counts ⟵ New HashMap;
    foreach case c of the retrieval result do
        solution = getSolution(c);
        similarity = getSimilarity(c);
        totalSim += similarity;
        if votes.containsKey(solution) then
            votes.put(solution, votes.get(solution) + similarity);
            counts.put(solution, counts.get(solution) + 1);
        else
            votes.put(solution, similarity);
            counts.put(solution, 1);
        end
        highestVoteSoFar ⟵ 0.0;
        predictedSolution ⟵ NULL;
        foreach entity e of votes do
            if e.getValue() ⩾ highestVoteSoFar then
                highestVoteSoFar = e.getValue();
                predictedSolution = e.getKey();
            end
            averageSim = highestVoteSoFar / counts;
            pow = highestVoteSoFar / totalSim;
            confidence = Math.pow(averageSim, pow);
            predictedSolution.setConfidence(confidence);
        end
    end
end
```

Algorithm 4: Similarity weighted voting for prediction time consuming task. Also, the previous cases need to be comprehensive enough for retrieving, otherwise the reasoning task could still fail in not finding any similar case in the case base.

In this dissertation, a fully unsupervised learning mechanism is developed by adopting an adaptation technique for CBR derived from fuzzy logic based intelligent reasoning and modelling. As a result, the solution of a new case is determined or revised based on the fuzzy reasoning output when there are not enough solved cases in the case base. This can enhance CBR with the adaptation ability of domain knowledge, in which problems and solutions are, in many cases, described by means of the fuzzy sets and rules. The steps of constructing the fuzzy adaptation model assisting CBR are:

1. Configure the fuzzy reasoning engine by setting up fuzzy sets and rules.
2. Traverse the case base to find K-NN similar cases.
3. Make a prediction based on weighted median of similarity.
4. Apply the fuzzy adaptation if the confidence of the prediction is low.
5. Use the fuzzy output to revise the solution of the present case.

Step 1 is performed only once to configure the fuzzy membership function and register fuzzy rules. Steps 2-4 are performed every time a CBR cycle starts. Note the fuzzy reasoning mechanism is applied, if and only if the CBR method can not find a similar solution for the present query. In other words, the confidence value of retrieving the result is lower than the confidence threshold which is set to 90% in our case. The result of the fuzzy output is then used as an alternative solution making use of domain knowledge to alleviate the lack of experience. However, the rules designed by the user are unlikely to cover all circumstances. For a new case where no rule is applicable and no similar case is existed in case base, we label the solution as unknown and provide an option for the user to manually label the case solution as an alternative approach of case revision.

6.5.4 Case Provenance

Provenance is an annotation to explain how a particular result has been derived; such provenance information can be used to better identify the process that was used to reach a particular conclusion [Szomszor and Moreau, 2003]. The benefit of data provenance is widespread since it provides a proof of the derived data and a record of its history, which allows people to view the derivation data and make observations about its quality and reliability.

In CBR, memory of prior problems and solutions plays a crucial role, where new solutions are generated by retrieving and adapting prior solutions, and are added to the case base for future use. However, standard CBR systems do not remember the provenance of the cases in their case base [Leake and Whitehead, 2007]. Noteworthy, a provenance recording module is integrated into our reasoning framework, so that users can have a deeper understanding of how a particular solution was derived from previous cases. This also supports the goal of transparency.

In this dissertation, we investigate the use of a simple provenance-based method for tracing the source of the reasoning output. Considering the following requirements for provenance collection that were proposed by [Frew and Bose, 2002], we implemented a basic case provenance module for the reasoning framework:

- A standard provenance representation is required so that data lineage can be communicated reliably between systems (currently there is no standard lineage format).
- Automated provenance recording is essential since humans are unlikely to record all the necessary information manually.
- Unobtrusive information collecting is desirable so that current working practices are not disrupted.

Generally, the provenance component works seamlessly with the reasoning engine. Whenever the reasoning engine derives a new case in its case base, its provenance trace, including the cases from which it was derived and the adaptation rules used to derive it, is automatically recorded and stored along with the case itself. More specifically, the most similar k cases (e.g. for k-NN with $k > 1$) retrieved from the case base are kept as the provenance for the new case, and if the fuzzy adaptation is involved in producing the new case, any fuzzy rule that is triggered by the case adaptation is added to the provenance. Case provenance data are stored in a customized XML format for data review and further for sharing with other components or systems. As shown in Code 6.1, each new case is assigned a case ID and the creation time. Its provenance consists of similar cases with their similarity value and adaptation rules with their confidence value.

Code 6.1: Case provenance format

```
<Case id="int" timestamp="time">
  <Provenance>
    <SimilarCase>
        <ID>int</ID>
        <Similarity>double</Similarity>
    </SimilarCase>
     ...
    <AdaptationRule>
        <ID>int</ID>
        <Confidence>double</Confidence>
    </AdaptationRule>
     ...
  </Provenance>
</Case>
```

The presentation of case provenance can be useful as an explanation of the reasoning conclusion to the user. Figure 6.10 gives an example of the UI demonstrating case provenance (as highlighted in the red block). Furthermore, provenance considerations could contribute not only to understanding the derived case but also to assessing the case quality. Similar to RBR, it is important to access confidence in the solutions of a CBR system. It is commonplace in rule-based systems to assign confidence values to rules, and to estimate the confidence of conclusions based on their derivations. For CBR systems, the quality of solutions may be estimated based on the quality of the original case and adaptation procedures applied to revise the case solution. In future work, a provenance-based metric can be put in place to estimate the case quality. For example, by examining the length of the generation chain (the number of intermediate cases generated from an initial case before generating the current solution) as well as the confidence of

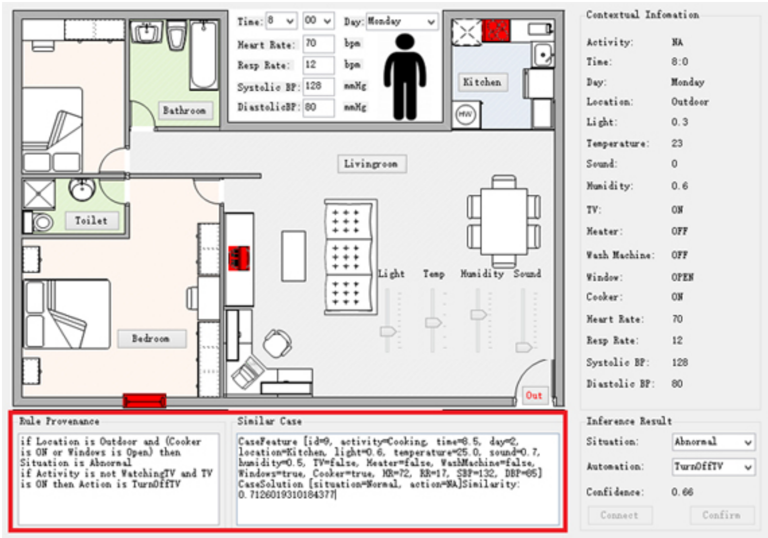


Figure 6.10: The demonstration of the case provenance

adaptation rules, the reasoning system could estimate the adaptation-based case quality as part of assessing confidence in a solution.

6.6 Implementation and Evaluation

It is difficult to evaluate the CARA system in its entirety without extensive field deployment and analysis. Issues including medical (e.g. need for support from medical institutes and healthcare professionals), ethical (e.g. privacy problems concerning confidential healthcare data) and practical (e.g. limited number of medical devices and smart home sensors) make a field experiment infeasible at present.

As a result, we have conducted realistic laboratory experiments to evaluate the correctness of the context-aware hybrid reasoning framework in a pervasive healthcare environment and report the results in this section. In our testing scenario, the system deployed consists of a remote healthcare sever, a wireless sensor network (WSN) and client applications. At the first test stage, real-time vital signs of the patient are collected from the wearable BioHarness sensor while environmental sensing is simulated by an android application, which is developed to reflect the changes of the ambient environment when smart home sensors are

Figure 6.11: Android application for the simulation of a smart home environment

still under development. In the future, a case study would be examined with the full deployment of the wireless sensor network (WSN) in a real-world home environment. Biomedical parameters currently considered and used are: heart rate, blood oxygen level, systolic and diastolic blood pressure, body temperature, and respiration rate. Ambient contexts include: time, space, activity, duration associated with a subject's activity, environmental readings (e.g. temperature, light, noise and humidity), and object interactions (e.g. usage of TV, cooker, phone, and status of the heater, window and lights). Figure 6.11 illustrates the screenshots of the prototype Android simulator in the testing scenario.

Firstly, the fuzzy-based reasoning engine is implemented in the CARA system to provide real-time intelligence for prediction in various healthcare situations. The context-aware hybrid reasoning framework enhances the previous fuzzy rule-based reasoning engine with a learning ability by incorporating a novel CBR model. The CBR implementation is based on the jCOLIBRI:CBR Framework supported by [Asanchezrg et al., 2012]. The wireless connection between the sensor network and client application is built using Bluetooth, and the application

Table 6.7: Results of various reasoning approaches

ε	True Positive	False Positive	True Negative	False Negative	Accuracy
Common CBR					
0.7	32	3	189	38	84.35%
0.6	32	3	189	38	84.35%
Improved CBR with Fuzzy Dynamic Weights					
0.7	54	7	185	16	91.22%
0.6	47	5	187	23	89.31%
Proposed CBR with Fuzzy Dynamic Weights and Fuzzy Rules Adaptation					
0.7	54	5	187	16	91.98%
0.6	47	2	190	23	90.46%

is also connected to the home gateway which transmits real-time data to the remote healthcare server.

Comprehensive tests have been carried out in our lab to evaluate performance of the implemented software solution. While the smart home test-bed is still under construction, we simulate the behaviour of a person living in a realistic home environment based on the typical daily routine of an elderly person which is summarized from interviews. This provides us with high level *activity contexts*. In addition, simulation of *ambient contexts* models changes of light, room temperature, sound and humidity. *Physiology contexts* and *personal contexts* are derived from the BAN and the personal medical history respectively.

All the contexts are used to build the input query for CBR. They are also mapped into fuzzy sets and enforced by applying consistent rules which refer to domain knowledge. The system then produces the final decision which indicates the current situation of the subject. The case base used for testing contains 262 cases, among them, 192 are normal cases and 70 are abnormal cases. We evaluated the proposed hybrid approach against the conventional CBR approach and evolving CBR approach using dynamic weights in case retrieval. Given the broad range of these trials, we are able to evaluate the accuracy of the reasoning outcomes over a wide range of situations. The results are shown in Table 6.7.

To simplify the evaluation process for anomaly detection, we only consider a two-class prediction problem (normal or abnormal), in which the outcomes are labelled either as positive or negative. If the outcome of a prediction is abnormal

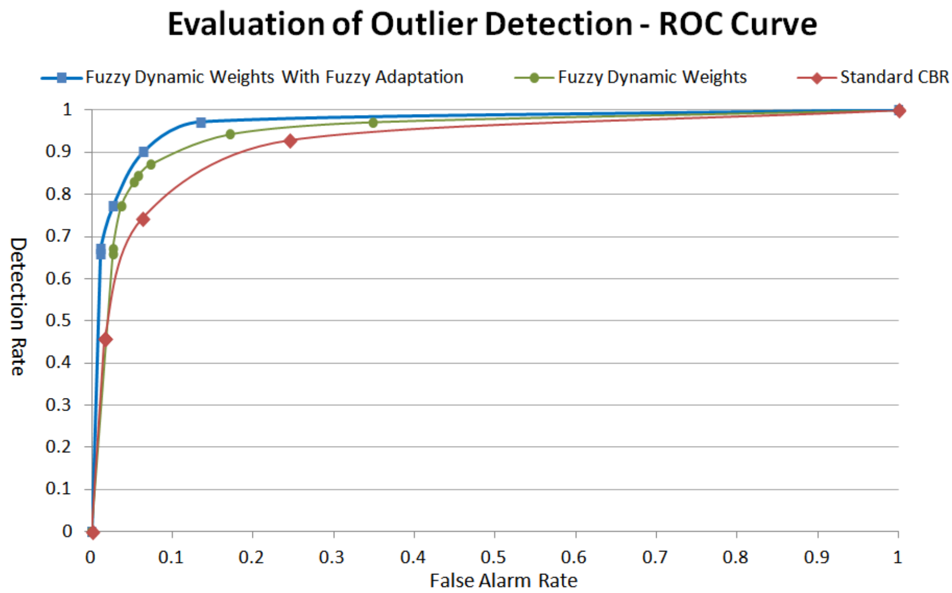


Figure 6.12: ROC space of three different approaches for anomaly detection

and the actual situation is also abnormal, then it is called a true positive (TP). However, if the actual situation is normal then it is said to be a false positive (FP). Conversely, a true negative (TN) has occurred when both the prediction outcome and the actual situation are normal, and false negative (FN) is when the prediction outcome is normal while the actual situation is abnormal. As we discussed in the previous section, we adjust the threshold for the confidence value to get a trade-off between detection rate and false alarm rate. The contingency table above can derive several evaluation metrics e.g. true positive rate (recall), false positive rate (fall-out), true negative rate (specificity), positive predictive value (precision). It turns out that accuracy is not a sufficient metric for the evaluation of anomaly detection. Since most of the cases are normal, even if it predicts every situation as normal, the accuracy could still be very high. As a result, we introduce the idea of receiver operating characteristic (ROC) from signal detection theory [Swets, 1996] to evaluate our reasoning framework. By calculating the true positive rate and false positive rate, we are able to draw a ROC curve as shown in Figure 6.12.

Each prediction result or instance of a confusion matrix represents one point in ROC space. The best possible prediction method would yield a point in the

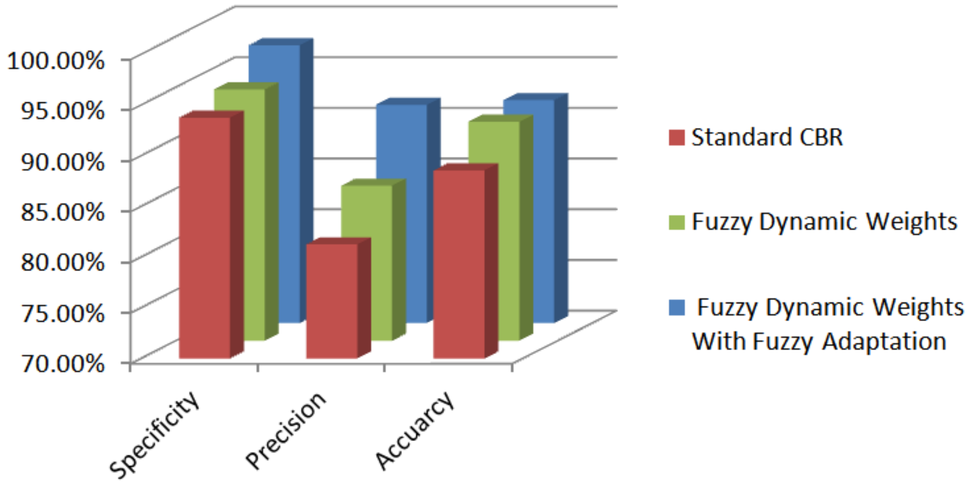


Figure 6.13: Best prediction performances of three different approaches

upper left corner at coordinate (0,1), called a perfect classification. So any point closer to that would be considered as a better approach. It is clear that the proposed approach is the best prediction method for anomaly detection. The best performance of each approach is compared and presented in Figure 6.13, where the proposed approach gives 97.4% Specificity, 91.5% Precision and 92.6% Accuracy when the threshold ε for the confidence value of the predicted solution is set to 0.7 while the normal CBR approach only gives 93.7% Specificity, 81.2% Precision and 88.5% Accuracy at a confidence threshold value of 0.8.

To measure the execution performance of our approach, we added a time checking function. A start time is noted before calling a method, and then the finish time is noted after calling the method, providing a measure of the execution time for each task. We measured a 10-fold cross-validation using several case bases with different amounts of normal and abnormal cases. The summarized test results are shown in Table 6.8.

To evaluate the reliability of the reasoning system, we added 7 conflict rules into the rule base which contains 42 rules in total. The performance of the proposed approach was impacted in consequence. Leave-one-out approach (i.e. one case is taken out at a time to match against the remaining cases in the case base) and a confidence threshold value of 0.7 were applied to evaluate the

Table 6.8: Inference performance for various amounts of cases

Total Amount of Cases	Normal	Abnormal	Time Per Cycle (ms)
262	192	70	1925
200	150	50	1256
100	72	28	507
50	39	11	182

Table 6.9: Evaluation metrics concerning consistency issue

Hybrid Context-aware Reasoning	Specificity	Precision	Accuracy
With Inconsistent Rules	91.2%	78.6%	85.3%
Without Inconsistent Rules	97.4%	91.5%	92.6%

hybrid reasoning approach against the same approach with inconsistent rules. The result shown in Table 6.9 proves that the accuracy of the reasoning system can be affected by the inconsistency of the rule base. Thus, by using the semantic-based rule validation mechanism to ensure the consistency of the rule base, one can improve the accuracy of the reasoning system.

Although the reasoning tasks mostly rely on the computational power of the client device, it is clear that the response time of our rule engine is in direct proportion to the amount of cases being checked and the complexity of rules. We notice that the CBR mechanism gets computationally expensive as the size of the case base increases. If we have the system running for weeks and months, producing many thousands of cases, then it would become unacceptable in terms of efficiency for the user. To relieve the problem, the following approaches can be studied in future work. Firstly, a regular maintenance scheme is critical to remove redundancy from the case base. Secondly, the cloud-based data analysis infrastructure can be utilized to provide a reasonable solution for big data mining by sub-dividing the case base and then allocating subsets of cases to different task processes to allow parallel processing.

6.7 Conclusion

In this dissertation, we have developed a comprehensive context-aware hybrid reasoning framework that integrates fuzzy rule-based reasoning with a case-based model to achieve automated intelligence for pervasive healthcare in a smart home environment. The advantage of our approach is the fact that it performs fully unsupervised learning and with the minimum input from the domain expert. This is achieved by adopting context models for case representation, dynamic weights and hierarchic similarity measurement for case retrieving, and an intelligent fuzzy adaptation method for case revision. By combining these concepts, the proposed reasoning framework makes the CARA system capable of handling uncertain knowledge and using context in order to analyse the situation in a changing environment. Case study for evaluation of this hybrid reasoning framework is carried out under simulated but realistic smart home scenarios. The results indicate the feasibility of the framework for effective at-home monitoring.

When the reasoning system is used to provide technology-driven assistive healthcare, there is a need for the system to be as intelligent and sophisticated as possible, while also being as transparent and accountable as possible for both subject and caregivers. To support transparency, the rules used for anomaly detection in a smart home environment are given in a structured natural language, allowing both subject and caregiver to inspect and modify. Given that the rules are being devised for each individual case, it is important for accountability to detect and report any inconsistency or possible conflict among the rules. A semantic-based approach is used to examine the rules for inconsistency and possible conflicts, and indicate this to the user. This is especially crucial in a pervasive environment where new sensors may be incorporated, and where any new rules being added should be checked with respect to all the existing rules. When the reasoning system is in use, an outcome of the system might be unexpected. In this case, a provenance mechanism is developed whereby the rule executed or the case-based derivation can be provided to explain the outcome of the reasoning system. This also supports both transparency and accountability in the sophisticated reasoning component of CARA.

Chapter 7

Real-time Remote Monitoring

7.1 Introduction

The main design goal of the CARA system is to provide a pervasive real-time intelligent at-home healthcare solution. Recognizing this, we have built a solution of remote monitoring that support scenarios of using CARA to deliver healthcare data and reasoning decisions. In this chapter we describe a scenario where continuous monitoring of the patient and home environment is done in a non-intrusive way via wireless sensor networks, and a telecare function provides interaction between the patient and the remote caregiver through real-time video communication while the patient is being monitored remotely. The remote monitoring component, including its recording and playback facilities, is integrated with the CARA system and can make use of CARA's intelligent analysis result. This interactive user friendly approach provides an introduction to the technology for an elderly person, and can lead to incremental incorporation of the technology. Implemented as a rich internet application means that it is available to a remote caregiver just using any web browser. Thus, the CARA system can be accessed from any internet-connected PC or appropriate smart device. The results of the experiments illustrate the effectiveness of the system in monitoring a patient within a home environment.

7.1.1 Incremental Use of the CARA system

The CARA system can be used in different ways, varying from fully automatic real-time at-home monitoring of patient vital signs resulting in automated response, to use as a non-automatic assistant for remote real-time consultation with a specialist. While the fully automated system is the ultimate goal, it is recognized, this may be too disruptive initially for both patient and caregiver. Since healthcare is about more than just immediate application of the most advanced technology, we have built a solution that focuses on an incremental introduction of CARA as a pervasive healthcare system. This solution supports scenarios where the wireless sensors are initially introduced to the patient under supervision in a clinic, while the real-time consultation is provided by a remote healthcare specialist (Figure 7.1), and a scenario where the wireless sensors are used at home

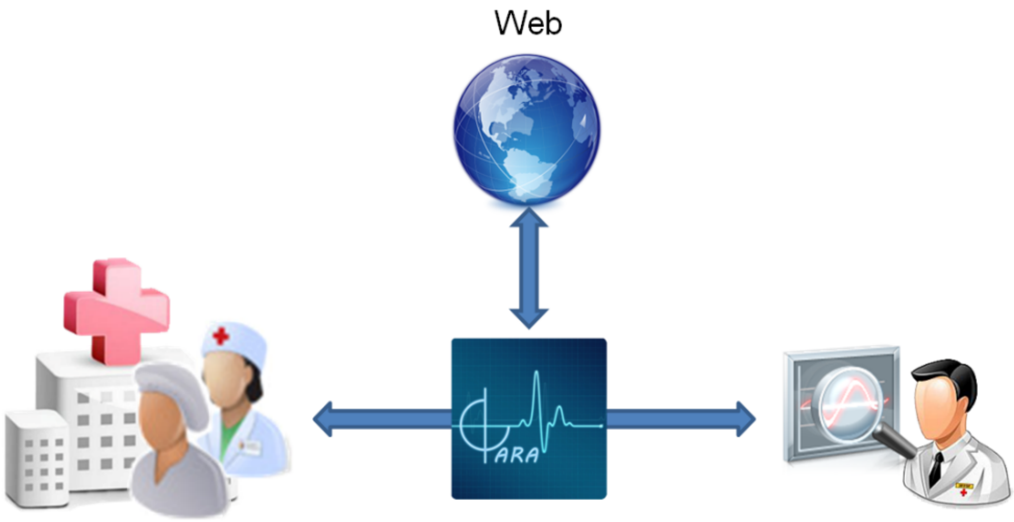


Figure 7.1: Remote monitoring under supervision in a clinic

under remote supervision (using a two-way video link) for a real-time interactive monitoring session with a caregiver (Figure 7.2). This use of CARA is over a short interval of time and is fully supervised with guidance for the patient in the wearing of the BAN and the use of the interactive remote monitoring system. It makes effective use of time for both the patient and remote specialist, and furthermore a facility in CARA to record both video and associated sensor data allows the session information to persist.

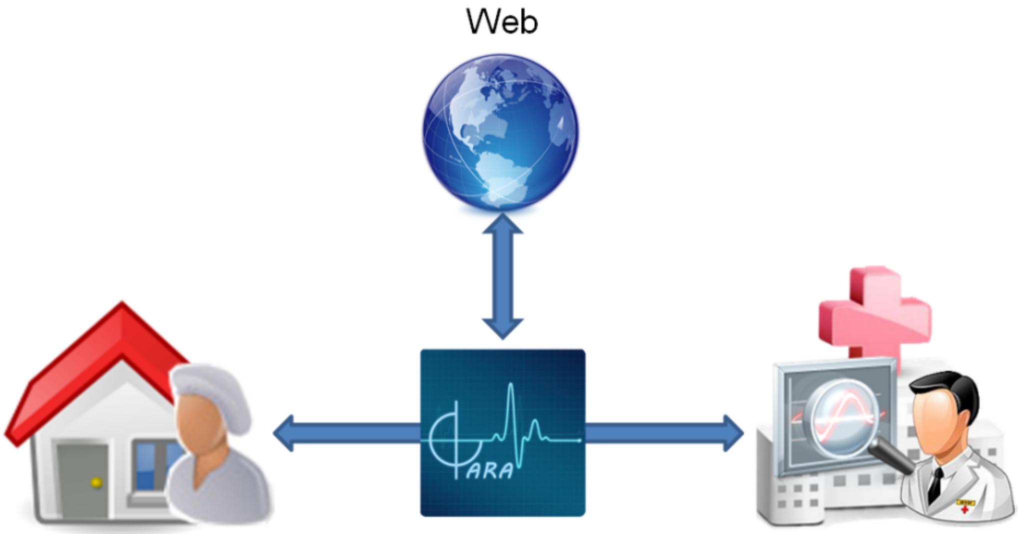


Figure 7.2: Real-time interactive monitoring under remote supervision

This restricted use of the CARA system is also important from a number of viewpoints. An inherent problem with all small wearable wireless sensors is the noise of various kinds, and this results in data errors. Real-time visualized monitoring along with the sensor recordings allows the consultant to disambiguate erroneous readings from significant readings. Furthermore, it avoids the medical, legal and social issues associated with introducing new models of healthcare, and instead is an alternative, less-disruptive approach. This use of the system provides an immediately practicable solution that respects current healthcare practice and the experience of both patient and caregiver, and leads to incremental incorporation of the technology.

Although this use of the CARA system is promising, there may also be a need to continuously and automatically monitor a patient during normal daily activities. This can be achieved by monitoring a patient through wireless sensor networks in an unobtrusive way without supervision, recording healthcare data for later review and analysis, and reasoning with all available knowledge for situation assessment (Figure 7.3). By integration with the CARA system, real-time remote monitoring can take advantage of the sophisticated reasoning and data management capabilities of the system.

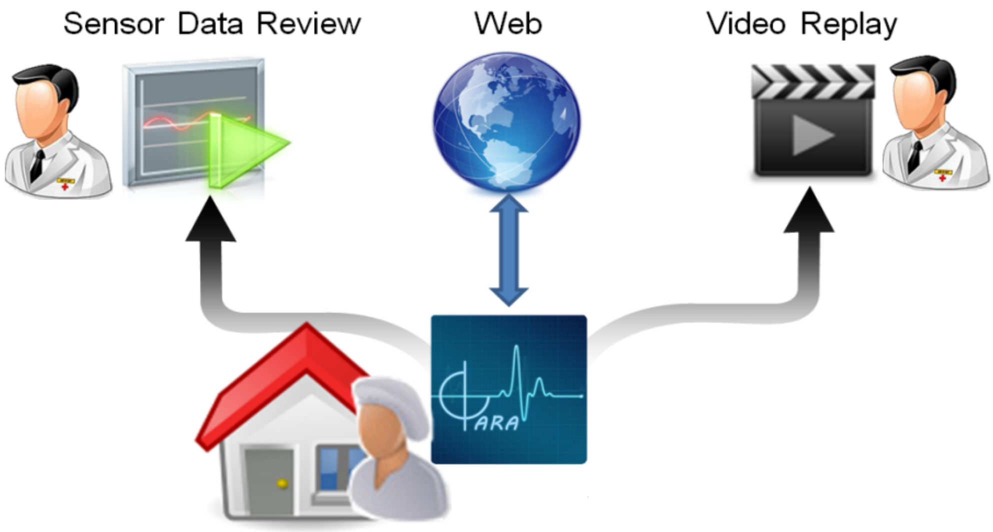


Figure 7.3: Real-time automated remote monitoring

7.2 Rich Internet Application

The Remote Monitoring component is designed as a web application using RIA (Rich Internet Applications) technologies. In traditional web applications, there is a limit to the interactivity that can be added to a single page. This often leads to delays, during which time users may get tired of waiting and doctors may waste valuable consulting time. With RIA technologies, the client and the server can communicate without any page refreshes. In this way, web applications can support more complex and diverse user interactivity within a single screen. This allows real time user interaction, which is essential for our system.

There are plenty of RIA frameworks available for developing web applications typically delivered by way of a web browser, a browser plug-in, and extensive use of JavaScript. Adobe Flash, JavaFX, and Microsoft Silverlight are currently the three most common platforms, with desktop browser penetration rates around 96%, 76%, and 66%, respectively (as estimated by [Seltzer, 2010]). Generally, users need to install a software framework using the computer's operating system before launching the application, which typically downloads, updates, verifies and executes the RIA. This is the major difference from HTML5/JavaScript based alternatives like Ajax that use built-in browser functionality to implement comparable interfaces. Regardless of the fact that recent trends show that plug-in-based frameworks are in the process of being replaced by HTML5/JavaScript based alternatives, the need for plug-in-based RIAs for accessing video capture and distribution has not diminished. For that reason, we decided to use Adobe Flash as the platform to build the remote monitoring application. The client application is implemented by Apache Flex, formerly Adobe Flex, which is a software development kit (SDK) for the development and deployment of cross-platform rich internet applications based on the Adobe Flash platform.

Figure 7.4: Flash user interface of the user login in the CARA system

The login user interface of the client application is illustrated is Figure 7.4. Since it is a Flash application, it is compatible with most operating systems and mobile platforms. Users can login to the system from PCs, laptops or smart devices that have an internet connection. The only tool needed to launch the application is a web browser with Adobe Flash plug-in installed.

7.3 Scenarios and Their Implementation

Research has revealed that telecare, monitoring a patient's health in their own home, can be used safely with active patients in place of the standard clinic visit [Whiteman, 2012]. The emergence of telecare adds a new paradigm in healthcare, where the patient is monitored between physician office visits. This has been shown to significantly reduce hospitalizations, while improving the patient's quality of life [Breen, 2011]. Telecare can also benefit patients where traditional delivery of health services is affected by distance and lack of local specialist clinicians to deliver services. Generally, there are two different telecare approaches:

- Real-time model: a telecommunications link allows instantaneous interaction between patients and caregivers. Video-conferencing equipment is one of the most widespread forms of real-time telecare. With the availability of better and cheaper communication channels, direct two-way audio and video streaming between clients is leading to lower costs.
- Store-and-review model: digital images, video, audio, observations of daily living and healthcare data are captured and stored on the server; then at a convenient time they are retrieved by caregivers at another location where they are studied and reviewed.

The CARA system provides remote at-home monitoring services that support both of the telecare models. It allows the incremental use of the system and thereby encourages the adoption of the technology. Two use case scenarios concerning different telecare models for remote monitoring are presented and their implementations are discussed in this section.

7.3.1 Real-time Interactive Remote Monitoring

The first scenario involves real-time at-home monitoring under remote supervision by a caregiver. Real-time sensor data are collected and transferred to a remote caregiver, who might be any suitable healthcare worker, specialist or medical consultant. The monitoring session also involves the use of a two-way video link whereby the patient and remote caregiver can communicate with each other. This is an important aspect of making this scenario low-disturbance and non-stressful, and thereby gaining acceptance for the technology.

Specifically, the client application running on a home gateway gathers information from smart home sensors and medical equipment through wireless connections. The sensor data are then transmitted to the remote server in real-time through the Action Message Format (AMF) protocol [Adobe, 2011]. The caregiver can login to the system and select the patient who is eligible to be monitored. A video-conferencing component is integrated with real-time remote monitoring,

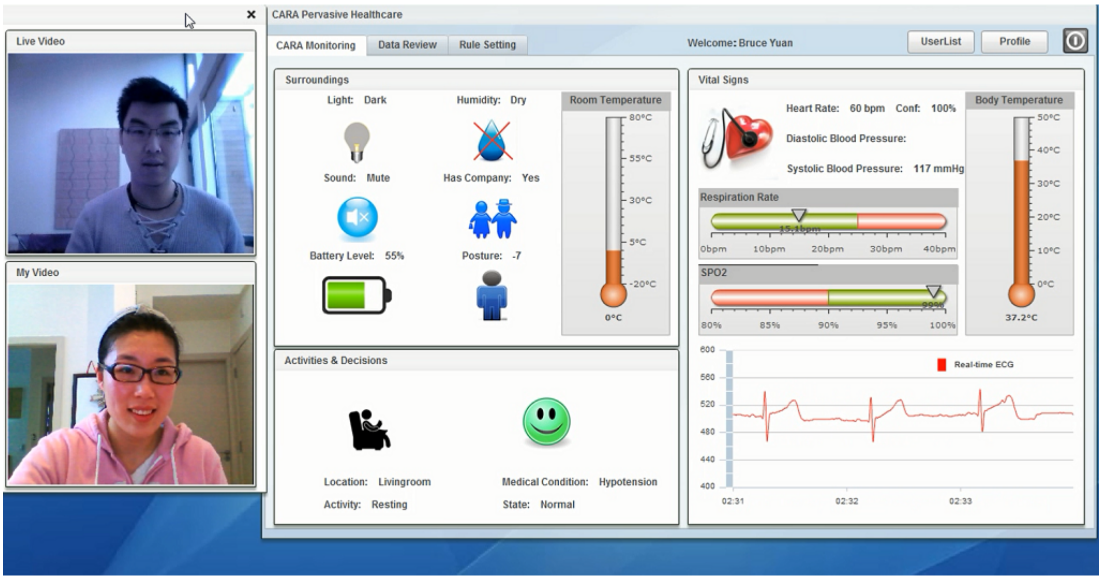


Figure 7.5: Real-time interactive remote monitoring

which can aid in an interactive examination and allow reassurance for patients who are new to the system. In this case, continuous monitoring of the patient is carried out by the remote caregiver in real-time with a live video communication channel established as shown in Figure 7.5. Real-time sensor data are published along with the video stream through the Flash Media Server (FMS) to the remote client of a caregiver. (The FMS is a proprietary data and media server from Adobe Systems). The live video is captured and encoded by Flash Media Live Encoder, and streamed to FMS. The video streams are then broadcast to the remote client through the internet. Moreover, a social utility is also implemented in the application which allows the user and caregiver to communicate with each other by sending messages. It can be used during or outside of the monitoring session.

7.3.2 Automated Remote At-home Monitoring

The second scenario, following easily from the first, involves the most innovative use of the CARA system that is a later stage of the incremental incorporation of pervasive healthcare technology into medical practice. This scenario is where the system is analyzing the real-time sensor data, reasoning with all the contexts to identify critical patient conditions and forwarding healthcare data to the healthcare server for inspection by caregivers. This makes use of the reasoning technology described in the previous chapter.

The fully automated intelligence of the system is achieved by incorporating real-time remote monitoring with the activity recognition application and the reasoning framework. In this scenario, the patient is monitored in a smart home environment without any supervision while wearing BAN continuously. Information around the patient is gathered and analyzed to detect any possible anomalies (as demonstrated in Figure 7.6).

The contextualization involves the processing of raw data coming from smart home sensors and wearable wireless sensors, producing higher level information. While the activity recognition application keeps tracking the movement of the patient and providing activity contexts to the system. Note that, dynamic activities (e.g. walking stairs, washing hands, sweeping) can be identified directly by the trained classification models, while other static activities (e.g. watching TV, sleeping, toileting) can be inferred from the posture of the body and relative ambient context. The contexts can be used by the reasoning system to identify the current state of the patient using the domain knowledge (in terms of fuzzy sets and rules) and previous experience (in terms of case bases). The reasoning engine executes in real-time and can offer immediate notification of critical conditions. The vital signs of the patient and all the available contexts are transmitted along with the reasoning output to the healthcare server, where they are stored and can be examined or reviewed by a remote caregiver at the proper time.

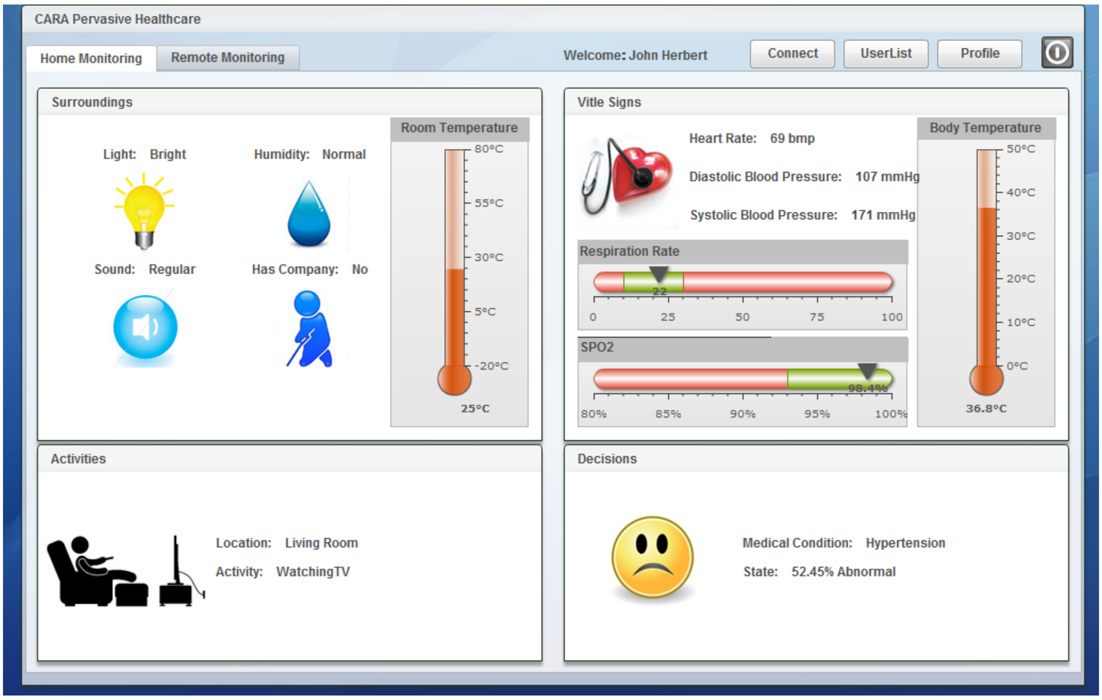


Figure 7.6: Automated remote monitoring of the patient without supervision

7.4 Healthcare Data Review

An important use of the CARA system for the caregiver is the ability to record and review the real-time patient monitoring session. It is very convenient for the caregiver to record a monitoring session and then review both the video stream and associated real-time vital sign data at any subsequent time. This plays a part in encouraging the automated use of the CARA system, where long-term at-home real-time vital sign data may be reviewed by health professionals. The healthcare data review function supporting this approach includes the sensor data review and video session replay.

7.4.1 Sensor Data Review

The sensor data review function allows the caregiver to analyze the broad context of sensor readings in order to distinguish critical from non-critical situations. By clicking the review button on the main user interface which brings up the data review interface, the caregiver is able to define parameters for reviewing the sensor data, for example: selecting patient profile, defining the duration of data record, setting data priority, choosing types of sensors, etc. Once the caregiver sets up all these options, they can retrieve the sensor data review chart as shown in Figure 7.7. It shows the recorded sensor readings from the sensor database for

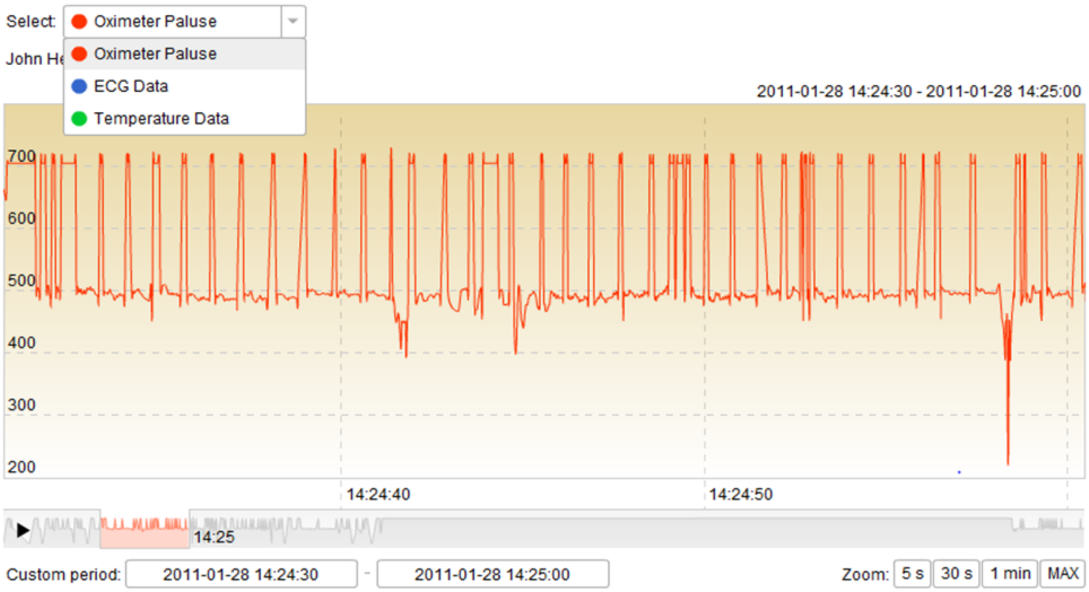


Figure 7.7: Sensor data review chart

the selected patient over a certain time period. The chart is implemented in flash using a third party web chart API [amChart, 2011]. It supports zooming and scrolling functions so that users can adjust the graph easily and analyze the data in an efficient manner. A playback function is also integrated into the chart which enables the user to play the sensor data graph at any speed.

An added functionality of the data review application is the ability to highlight readings with abnormal situations that demand attention. This allows a caregiver to easily locate relevant data and this assists diagnosis. Moreover, the medical specialist is able to annotate sensor data streams to record his/her findings while reviewing the data.

7.4.2 Video Session Replay

The video session replay function is designed for the caregiver to review the recorded patient's live video along with the associated real-time sensor data. This function is integrated in the real-time remote monitoring system. Whenever the patient's live video stream is published on FMS, it is also recorded as a flash live video file (FLV) on the server. To distinguish the video file and to synchronize with the sensor data, several correlations of the video must be recorded into the database as well (e.g. video start time, end time, patient profile).

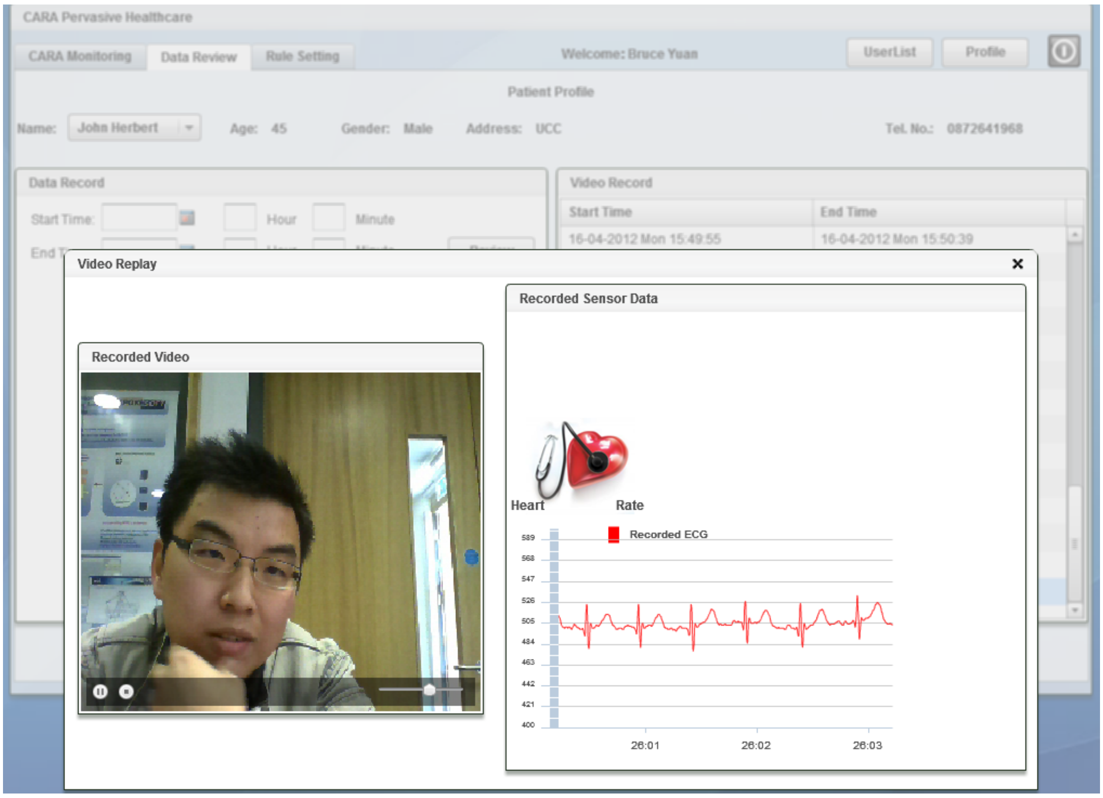


Figure 7.8: User interface of video replay function

To launch the video replay application, the caregiver needs to select a patient in the data review interface. Then all recorded video related to the patient will be listed on the screen with start and end timestamps. The caregiver is able to replay any of them simply by clicking on the listed item. To synchronize the video session with the recorded sensor data, the system searches the sensor data in the database by timestamps and plays the video with the selected sensor data as shown in Figure 7.8. This use of data review can assist caregivers with vital context. In an environment with many sources of data (some of which may be unreliable), the context including video, the log of sensor readings with timestamps and patient's profile can be used to disambiguate the real critical conditions from false alarms.

7.5 System Evaluation

The evaluation of the remote monitoring system concerns several non-functional design and implementation issues.

Software Compatibility

Software compatibility can refer to the compatibility that a particular software runs on a particular CPU architecture. Software compatibility can also refer to the ability of the software to run on a particular operating system. Normally, the remote monitoring application is compiled for different CPU architectures and operating systems with a variety of browsers to allow it to be compatible with the different systems.

Table 7.1: Software compatibility of the remote monitoring application

Operating Systems	**Browsers**				
	IE	Firefox	Chrome	Opera	Safari
Windows	✓	✓	✓	✓	✓
Mac	✓	✓	✓	✓	✓
Ubuntu	✓	✓	✓	✓	✗
Linux	✓	✓	✓	✓	✗
Android	✗	✓	✓	✗	✗
iOS	✗	✗	✗	✗	✗

We have tested the remote monitoring application in various operating systems and browsers. Table 7.1 shows the software compatibility of the application. As expected, the application is compatible with most popular desktop/mobile operating systems and web browsers, with some exceptions for Safari and Opera web browsers. However, it does not work on iOS devices with all the browsers since iOS still does not support Flash.

System Performance

Two experiments were conducted to test the performance of the remote monitoring system. The first experiment is carried out to test the concurrency performance of the server for live video communication. FMS was engaged to handle and broadcast the live video streams. However, the starter version of FMS (free) we used is limited to supporting only 10 simultaneous connections for live adaptive media streaming. As a consequence, we tested our interactive remote monitoring

Table 7.2: System capacity of the remote monitoring server

Num. of Users	CPU Speed	CPU Usage	Memory Usage
2	3.08 GHz	46%	4.3 GB (27%)
4	3.11 GHz	67%	6.7 GB (42%)
8	3.13 GHz	79%	7.5 GB (47%)

application with 2, 4 and 8 users sharing the live video in pairs simultaneously. The physical server used to host the system was an Alienware M14R laptop, with 2.30 GHz Intel Core i7 CPU and 16 GB memory installed. The CPU and memory usage of the machine was recorded and is listed in Table 7.2. As we can see from the result, the use of hardware resources increased with the increasing number of users. The server (hosted on a laptop) can barely support 10 users using the system and streaming live video to the server simultaneously. Further experiments need to be conducted by hosting the system on an enterprise server or deploying it in the cloud infrastructure would give a more comprehensive evaluation.

The second experiment is to evaluate the impact of the potential delay of the networks which can affect the server response time. We tested the remote monitoring client application through localhost, wireless local area network (WLAN) and wireless internet respectively. Results indicating the data transmission delay in milliseconds are shown in Figure 7.9. The testing scenario utilizing the internet has taken place under different bandwidth of the broadband (e.g. 150Mb/10Mb, 50Mb/5Mb, 10Mb/1Mb). Clearly, the data transmission delay is inversely proportional to the bandwidth of the internet connection, which is unavoidable under the current approach. However, this delay does not substantially affect the overall performance of the system.

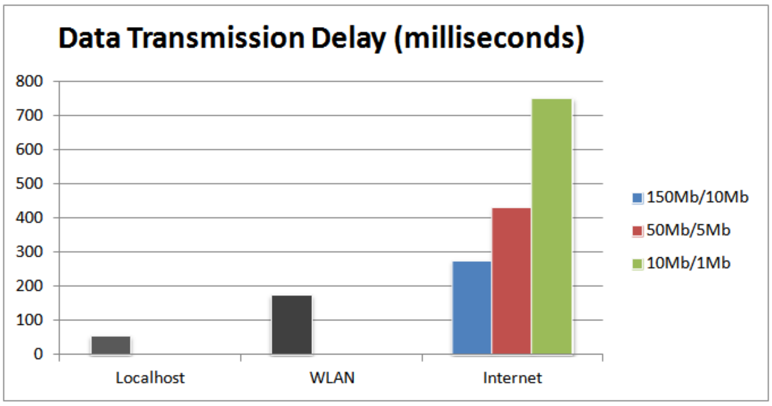


Figure 7.9: Data transmission delay with various networks

Security & Privacy Issues

Basic security and privacy issues are taken into consideration in the design of the remote monitoring system as well. Password control allows only an authorized user to log in to the CARA system. Authority management is implemented in the system to achieve privacy control, which means different users can access different functions of the system according to their authorities. For example, the medical consultant can view the patient's history data and define rules for the individual patient while only the patient can start the remote monitoring session.

7.6 Conclusion

The CARA pervasive healthcare system is designed to provide an innovative technical solution for automated at-home healthcare. It is recognized, however, such a change from current practice may be unacceptable, and an incremental introduction of technology may be the best approach to the successful use of the CARA system. Following this approach, the remote monitoring system is developed for delivering healthcare services which support scenarios where the wireless sensors are initially introduced to the patient under supervision in a clinic (where the real-time monitoring is provided by a remote medical consultant), and a scenario where the wireless sensors are used at home under remote supervision of a caregiver. These scenarios provide a non-stressful introduction of the technology, and gain acceptance for more advanced scenarios such as non-interactive, at-home automated patient monitoring using the WSN.

Important aspects of the remote monitoring include: real-time data sharing between patient and caregiver; interactive remote consultation; and replay, review and annotation of the comprehensive monitoring data by the medical professional. The offline review and annotation can provide the basis for improving the automated intelligent analysis of the CARA system.

Chapter 8

Conclusions

This concluding chapter is divided into two sections. It summarises the main achievements of the research, provides some insights from user trials and experimental evaluations, and outlines a number of interesting research directions for future work.

8.1 Research Achievements

Presented in this thesis is the Context-aware Real-time Assistant (CARA) architecture for pervasive healthcare. It is designed to provide a context-aware infrastructure which can make effective use of wireless sensor network (WSN) technology for innovative real-time monitoring, analysis and diagnosis within a pervasive home environment. It achieves this through the development of a set of CARA distributed components which collect real-time sensor readings, provide correlated context to the context-aware reasoning engine for the inference of patient conditions and deliver healthcare data to a remote caregiver in real-time. A cloud-based data analysis framework also plays an important part in processing machine learning tasks to ease the burden on the mobile client. This research explored the use of various integrated software frameworks to provide intelligent healthcare services in a home environment, which should lead to extended independent living and better quality of life. In the case of the research contributions, the following was achieved:

- **A WSN for a smart home environment**: A network of sensors (worn, carried, and environmental) is an ideal technology platform for detecting and responding to health-relevant parameters such as physiology readings, physical activity and ambient readings. The WSN presented in chapter 4 consists of a wireless body area network (BAN) that can monitor various vital signs while providing real-time feedback to the patient and a remote caregiver; smart home sensors deployed in a patient's home environment to provide real-time and extended monitoring of activity and wellbeing; as well as a smartphone carried by the user to detect body movement. The WSN can deliver a long-term home monitoring service to assist in condition diagnosis and identification of changes in a person's behaviour pattern. Furthermore, the always-on nature of the WSN means that it can detect and respond to anomalies in a timely manner. In particular, the BAN can provide notice of significant shifts in critical physiological parameters in order to prevent a health crisis. Other quality-of-life issues, such as privacy, dignity, and convenience, are supported and enhanced by the ability to unobtrusively provide services in the patient's own home.

- **Smartphone based activity recognition**: A robust system of activity of daily living (ADL) recognition is discussed in chapter 5, which provides the activity context for the CARA system using hierarchical classification by combining threshold-based methods and multi-classifier machine learning algorithms. An Android smartphone and a motion sensor are attached to the thigh and chest, respectively, to track the movement of the body. Human movement can be naturally represented through hierarchies, such as motion and motionlessness. Firstly, a threshold-based mechanism was used to separate the sensing data into two groups: static and dynamic. Static activities are identified based on the posture of the body which is calculated from 3D-acceleration, and dynamic activities are classified by using adapted machine learning models. By utilizing the cloud infrastructure, the system provides high scalability and availability for data analysis and classification model optimization. Data processing and classification algorithms are implemented in the smartphone for real-time activity monitoring while the data analysis and evaluation are done off-line in the cloud. The experimental results compare favourably with other work using multiple body sensors [Cook, 2012]. This shows a lot of promise for using a smartphone and just one body sensor as an alternative to multiple body sensors. Moreover, the performance of our approach shows a significant improvement in comparison to an approach utilizing a single smartphone [Zhang et al., 2010].

- **A context-aware hybrid reasoning framework**: A novel context-aware hybrid reasoning framework is developed that integrates fuzzy rule-based reasoning with a case-based model to achieve automated intelligence for pervasive healthcare in a smart home environment. By combining these concepts, the reasoning framework presented in chapter 6 makes CARA capable of handling uncertain knowledge and using contexts in order to analyse the situation more precisely. The advantage of the approach is that it performs fully unsupervised learning and with the minimum input from the domain expert. The evaluation results show that, comparing with other approaches(e.g. rule-based reasoning and normal case-based reasoning), it significantly improves the performance of the reasoning engine in terms of efficiency, accuracy and flexibility. This is achieved by adopting context models for case representation, dynamic weights and hierarchic similarity measurement for case retrieving, and intelligent fuzzy rule-based assistance for case adaptation. Furthermore, a semantic-based rule validation mechanism is applied to automatically check for conflicts between rules to ensure the correctness and consistency of the reasoning engine.

- **Real-time interactive remote monitoring**: The design goal of CARA is to enable improved healthcare services through the intelligent use of the wireless remote monitoring of patient vital signs, supplemented by rich contextual information. To support this, the remote monitoring system presented in chapter 7 provides a technical solution for automated remote monitoring of elderly people and soliciting expert feedback on healthcare data being streamed from their homes. It can be used in different ways, varying from fully automatic analysis of real-time patient vital signs and intelligent reasoning based upon contexts resulting in automated response, to a non-automatic assistant for remote real-time clinical analysis by the medical professional. Currently, the latter use of the system is more viable, as it avoids the inherent problem of data errors in wearable sensors, and also avoids the medical, legal and social issues associated with the automated intelligent healthcare solution. A novel aspect of remote monitoring is the use of web technologies and a video link as the basis for the transparent, incremental introduction of the CARA System.

- **Healthcare data analysis utilizing a cloud infrastructure**: Like many other fields, the pervasive healthcare system is looking to cloud computing as a means to improve the quality of service and efficiency of operations while reducing costs. With cloud computing, the limitations of traditional data mining mechanisms for healthcare could be minimized (e.g. high cost of servers and network, low network bandwidth, and limited system resources in mobile devices). In this dissertation, a cloud-based data analysis framework is developed taking advantage of cloud computing to support the analysis of healthcare data of the CARA system in a more efficient and reliable way. As presented in chapter 5, the healthcare data analysis framework, implementing machine learning mechanisms for activity recognition, is deployed in the cloud environment to optimize classification models using different machine learning algorithms in parallel. Data processing and storage are thus moved from the local device to the powerful and centralized computing platform located in the cloud. Data analysis results are then accessed over the internet connection through a local thin client. The evolving cloud-based machine learning mechanism makes the CARA system more customizable and self-adaptive.

Overall, the results presented and discussed in this thesis support the viability of integrating CARA into the everyday routine of the patient to produce effective solutions to pervasive healthcare problems.

8.2 Future Work

The pervasive healthcare paradigm is constantly under development. Despite the discussion of the achievements of this work, there are some areas which will require future research work. Future work may extend CARA with incremental incorporation of therapeutic resources and local healthcare services to enhance the pervasive healthcare paradigm. Presented are a number of subject areas which would enhance the healthcare system within a pervasive environment:

Medical BAN integrated in CARA involves Tyndall 25mm sensors and the Zephyr BioHarness, which are capable of measuring ECG, heart rate, respiration rate, blood oxygen, body temperature and position in space in real-time. Although they provide essential readings for medical interpretation, there is an improvement to augment the BAN with a larger collection of medical sensors, such as a beat-to-beat blood pressure measurement device (e.g. Portapres [Finapres, 2013], NIBP100D [BIOPAC System Inc., 2013]), in order to obtain a more complete medical picture. As we all know, blood pressure is one of the most important vital signs. High blood pressure may increase the risk of heart disease and are often related to other diseases and medical conditions. As a consequence, continuous blood pressure monitoring is an essential prerequisite for any study on blood pressure variability. Current intrusive cuff-based procedures are less acceptable for research projects in a clinical setting, and recently developed devices able to record blood pressure on a beat-by-beat basis in a non-invasive fashion may offer additional functionality to the BAN.

The WSN continuously monitors the home environment as well as patient vital signs in an unobtrusive manner. A large amount of data being made available in real-time raises a number of issues including the occurrence of noise, unreliable data; intermittent communication; limited battery power. Although, the quality of sensor data is not a major concern of this thesis, it is necessary to develop a data management component in association with sensing devices to enhance the CARA architecture within a real-world pervasive environment. The data management agent needs to be designed to ensure that all the essential data can be gathered while at the same time avoiding false alarms or getting overwhelmed by inaccurate data. It can be either deployed on the base-station of the WSN or the home gateway to acquire the sensor data, calibrate readings, filter unreliable values, note important values, and compress data if required. Concurrently sensing devices can be updated and improved as new sensors are constantly becoming available. For example, a small bluetooth enabled wearable sensor could take place of the smartphone, and thereby avoid dependency on carrying of a smartphone.

The smart home we designed for evaluating the CARA architecture is constructed in a constrained laboratory setting. Despite the fact that it is designed as a "living laboratory" and deployed in a real home environment, it is only feasible for use in laboratory conditions in a specific house. A major hurdle in implementing the system outside of trials is the adaptivity and customizability of the smart home sensors. Since there is no such thing as two exactly similar home, the CARA architecture should be improved to dynamically support various smart homes. To do so, the designer should be able to outline the floor map of a smart home and deploy sensors on it through a graphical user interface. Rather than using a fixed number of sensors, customized sensor settings should be recorded for different smart homes. A robust infrastructure is required to interpret the configuration of smart home sensors so that dynamic context modelling and context-aware reasoning can be achieved. In this way, the system could become more flexible and practical.

The activity recognition component employs the rule-base reasoning mechanism depending on two thresholds to separate the static and dynamic activities. Regardless of the fact that the chosen threshold values work well in this thesis, they may not be generally suitable for other cases and for different sensors. New algorithms need to be explored in future work, which can separate motion and motionless activities automatically, without the need for static thresholds. Also, the number of features extracted from raw sensor data for machine learning is fixed in this approach. It is necessary to assess the value of different features during the learning process so that the machine learning mechanism can be improved by assigning weights to features or by dynamically adjusting the number of features involved. Furthermore, the current approach recognizes an activity by detecting the single movement in a fixed time window, it does not consider the inter-relation between movements. To improve the usability of the system, different approaches could be investigated to recognize more complex activities (e.g. cooking, making coffee, and doing laundry) by detecting a series of movements within a period of time. Instead of using a fixed length sliding window, a dynamic sliding window could be adopted, which automatically determines the start and end of an activity according to the dynamic activity threshold.

The context-aware reasoning framework presented in this thesis may be enhanced to take into account more comprehensive and sophisticated medical rules as well as to develop a knowledge discovery mechanism which can extract new knowledge from previous solved cases and structure new rules. This will increase its level of autonomy thus becoming an invaluable tool within the pervasive paradigm. The remote monitoring system is designed for the visualization of the state of patients and surroundings, and also supports real-time interaction between the patient and a remote caregiver. It can be further developed to integrate with existing social networks (e.g. Facebook, Twitter) sharing healthcare related information among patients, families and caregivers. Moreover, it could be connected to the local emergency services in the case of emergency situations which may require the immediate attention of the medical staff.

Finally, the cloud-based data analysis framework presented assigns data analysis task to multiple machine learning worker roles running in parallel. It uses a static allocation strategy whereby each machine learning worker role is deployed to run with a fixed number of instances sharing the same amount of hardware resources. However, it turns out the computational requirement of each worker role is different due to different machine learning algorithms and size of datasets. A dynamic approach could be explored to dynamically allocate computing resources to specific worker role by adjusting the number of instances so that the processing time of different worker roles can be balanced. This dynamic mechanism promotes maximum use of computing resources while minimizing the cost. It will contribute to improving the level of synchronisation between mobile users and cloud servers in real-time as well as increase the availability and scalability of the system.

A confluence of technology developments has led to the possibility of realizing a vision of pervasive healthcare. The CARA architecture presented in this thesis highlights the potential of automated at-home monitoring and context-aware reasoning for healthcare by utilizing the hardware and software resources within a ubiquitous computing environment. All CARA components cooperate to deliver intelligent healthcare services based on the real-world context. Future pervasive healthcare systems may build on the ideas and techniques of CARA to deliver high levels of patient care, and empower individuals and their families for self-care and health management.

Appendix A: List of Publications

1. Bingchuan Yuan and John Herbert: *Real-time interactive medical consultation using a pervasive healthcare architecture*, the 9th international conference on toward useful services for elderly and people with disabilities: smart homes and health telematics (ICOST 2011), Pages 215-219
2. Bingchuan Yuan and John Herbert: *Non-intrusive Movement Detection in CARA Pervasive Healthcare Application*, the 2011 international conference on wireless networks (ICWN 2011), Pages 360-366
3. Bingchuan Yuan and John Herbert: *Web-based real-time remote monitoring for pervasive healthcare*, the 2011 IEEE International Conference on pervasive computing and communications workshops (PERCOM 2011), Pages 625-629
4. Bingchuan Yuan and John Herbert: *Fuzzy CARA - A Fuzzy-Based Context Reasoning System For Pervasive Healthcare*, Procedia Computer Science (ANT 2012), Volume 10, Pages 357-365
5. Bingchuan Yuan and John Herbert: *A Fuzzy-Based Context Modeling and Reasoning Framework for CARA Pervasive Healthcare*, Impact Analysis of Solutions for Chronic Disease Prevention and Management Lecture Notes in Computer Science (ICOST 2012), Volume 7251, Pages 254-257
6. Bingchuan Yuan and John Herbert: *Context-aware Hybrid Reasoning Framework for Pervasive Healthcare*, Journal of personal and ubiquitous computing, 2013, Volume 18, Issue 4, pp 865-881

7. Bingchuan Yuan and John Herbert: *Hybrid Reasoning Framework for CARA Pervasive Healthcare*, inclusive society: health and wellbeing in the community, and care at home lecture notes in computer science (ICOST 2013), Volume 7910, Pages 126-133

8. Bingchuan Yuan and John Herbert: *Transparency Issues in a Hybrid Reasoning Architecture for Assistive Healthcare*, AASRI Procedia 2013, volume 4, Pages 268-274

9. Bingchuan Yuan and John Herbert: *Smartphone-based Activity Recognition Using Hybrid Classifier*, in proceedings of the 4th international conference on pervasive and embedded computing and communication Systems (PECCS 2014)

10. Bingchuan Yuan and John Herbert:*A Cloud-based Mobile Data Analytics Framework - Case Study of Activity Recognition Using A Smartphone*, in proceedings of the 2nd IEEE international conference on mobile cloud computing, services, and engineering (MoblieCloud 2014)

11. Bingchuan Yuan and John Herbert:*Accountability in a Context-aware Smarthome Healthcare Reasoning System*, in proceedings of the 38th IEEE international computers, software & applications conference (COMPSAC 2014)

Appendix B: Activity Patterns with Accelerometer Signals

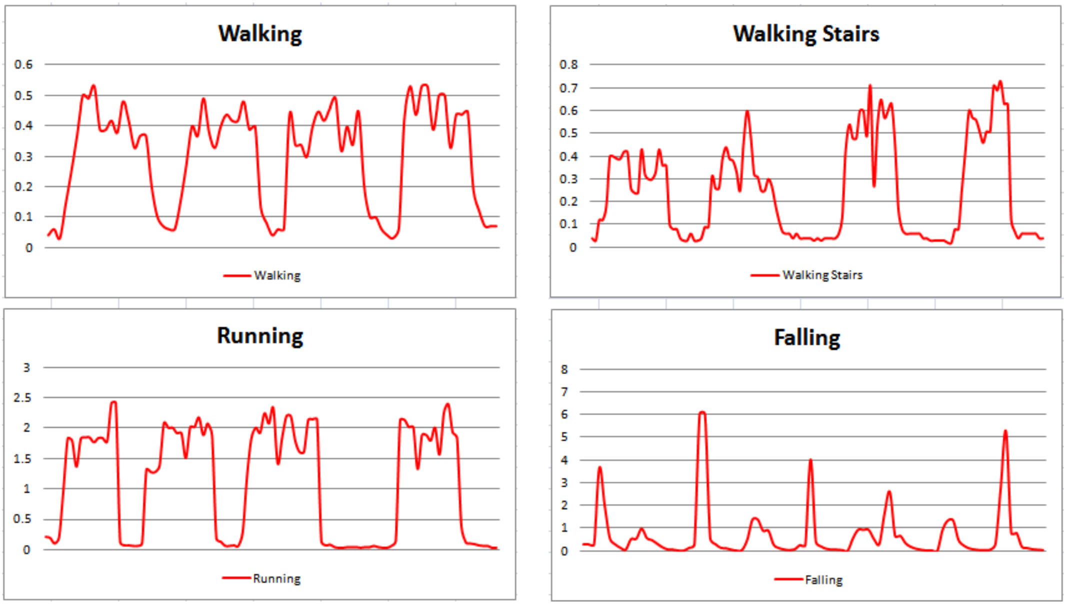


Figure 1: The trunk accelerometer signal of different activities

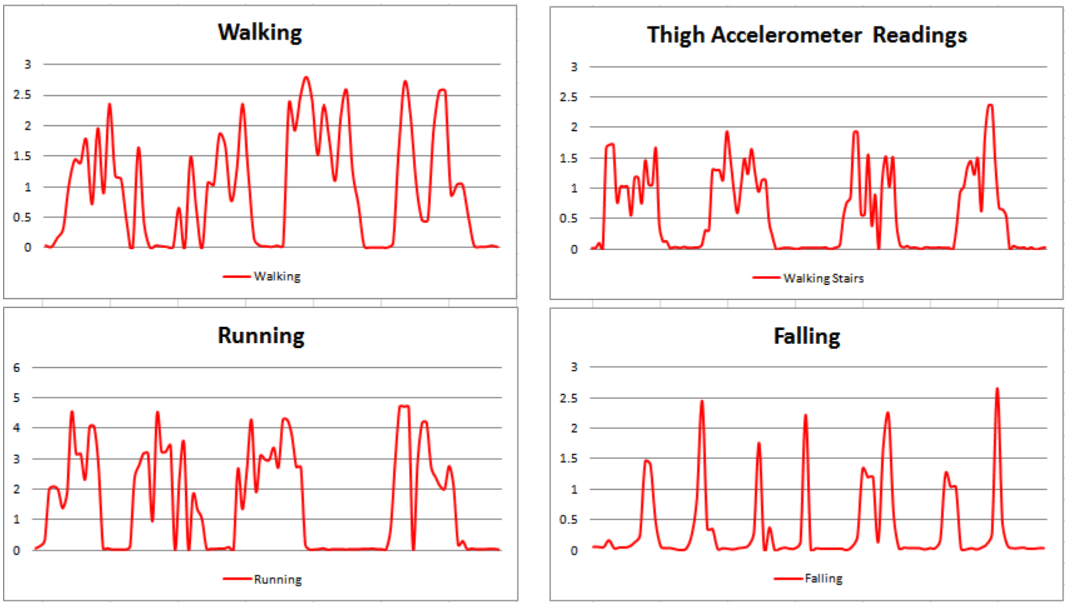


Figure 2: The thigh accelerometer signal of different activities

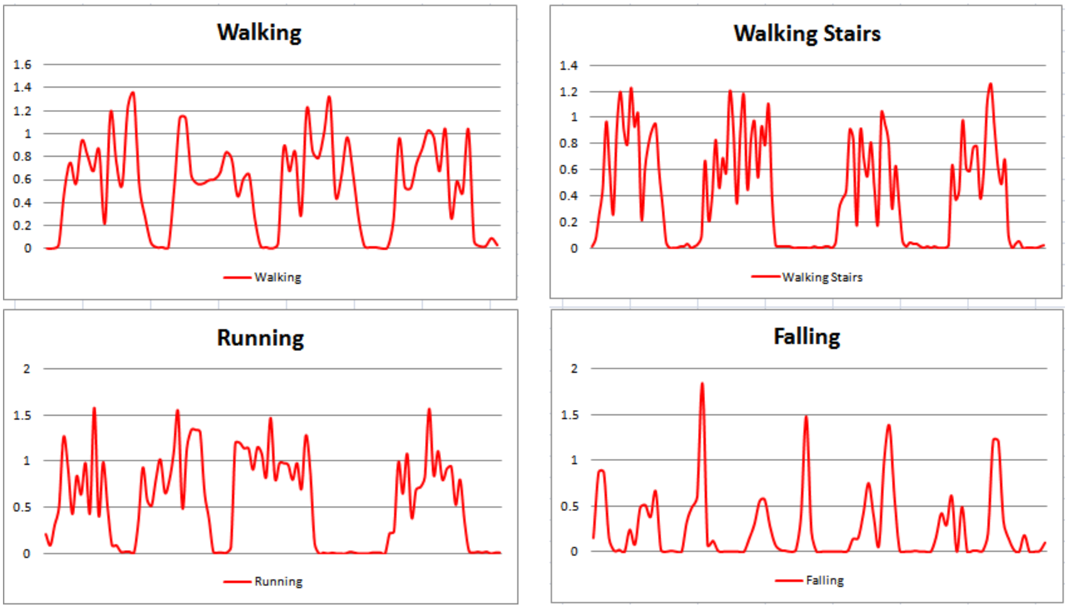


Figure 3: The thigh gyroscope signal of different activities

Appendix C: Details of Activity Recognition Results

The three activity classification models were evaluated using Weka Machine Learning Toolkit. Details of the evaluation results are listed as follows:

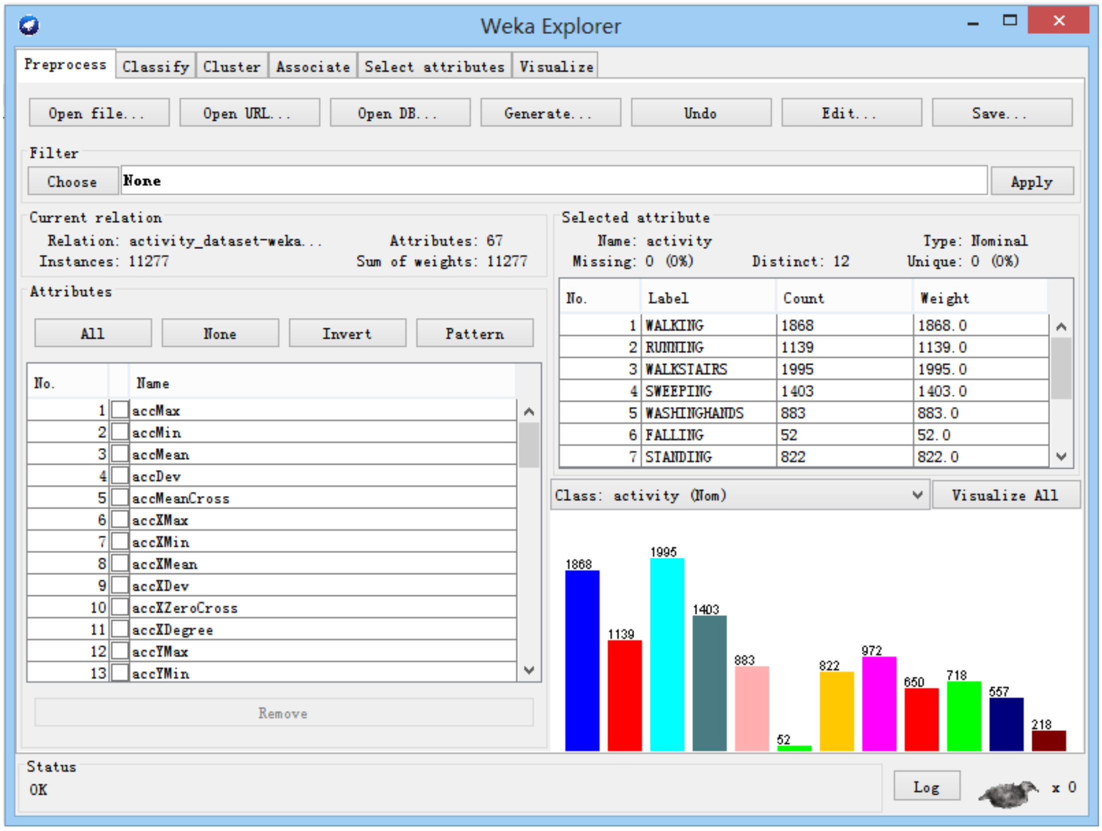


Figure 4: The Weka Explorer Window with adapted activity dataset

Default Model

Naive Bayes

Time taken to build model: 0.09 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	8697	85.1145 %
Incorrectly Classified Instances	1521	14.8855 %
Kappa statistic	0.8303	
Mean absolute error	0.0249	
Root mean squared error	0.1527	
Relative absolute error	17.0675 %	
Root relative squared error	56.5294 %	
Coverage of cases (0.95 level)	87.7275 %	
Mean rel. region size (0.95 level)	8.8055 %	
Total Number of Instances	10218	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.917	0.069	0.724	0.917	0.810	0.775	0.972	0.886	WALKING
0.736	0.011	0.904	0.736	0.811	0.793	0.983	0.906	RUNNING
0.669	0.057	0.761	0.669	0.712	0.643	0.944	0.801	WALKSTAIRS
0.876	0.005	0.960	0.876	0.916	0.906	0.990	0.968	SWEEPING
0.972	0.013	0.873	0.972	0.920	0.914	0.989	0.972	WASHINGHAND
0.812	0.010	0.359	0.812	0.498	0.536	0.981	0.553	FALLING
0.969	0.000	1.000	0.969	0.984	0.983	0.995	0.987	STANDING
0.948	0.001	0.980	0.948	0.964	0.961	0.994	0.978	SITTING
0.943	0.003	0.953	0.943	0.948	0.945	0.984	0.943	LYING
0.935	0.000	0.993	0.935	0.963	0.961	0.995	0.975	BENDING
0.961	0.000	0.998	0.961	0.979	0.978	0.994	0.978	LEANINGBACK
0.983	0.007	0.705	0.983	0.821	0.829	1.000	0.978	ROLLING
0.851	0.027	0.863	0.851	0.852	0.828	0.977	0.910	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1541	17	118	4	0	0	0	0	0	0	0	0	a = WALKING
3	910	311	0	0	13	0	0	0	0	0	0	b = RUNNING
580	74	1452	3	0	35	0	0	0	0	0	25	c = WALKSTAIRS
1	5	15	1153	104	22	0	0	0	0	0	16	d = SWEEPING
0	0	5	15	810	2	0	0	0	0	0	1	e = WASHINGHANDS
2	1	2	0	1	56	0	0	0	0	0	7	f = FALLING
0	0	1	6	7	0	494	1	0	0	1	0	g = STANDING
0	0	1	2	0	5	0	583	17	0	0	7	h = SITTING
0	0	0	0	0	14	0	0	529	0	0	18	i = LYING
0	0	0	10	6	2	0	11	9	548	0	0	j = BENDING
0	0	2	8	0	4	0	0	0	4	444	0	k = LEANINGBACK
0	0	0	0	0	3	0	0	0	0	0	177	l = ROLLING

Default Model

Bayes Net

Time taken to build model: 0.81 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9792	95.8309 %
Incorrectly Classified Instances	426	4.1691 %
Kappa statistic	0.9524	
Mean absolute error	0.007	
Root mean squared error	0.0793	
Relative absolute error	4.822 %	
Root relative squared error	29.3555 %	
Coverage of cases (0.95 level)	96.9368 %	
Mean rel. region size (0.95 level)	8.5527 %	
Total Number of Instances	10218	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.963	0.017	0.918	0.963	0.940	0.928	0.997	0.987	WALKING
0.938	0.003	0.980	0.938	0.958	0.953	0.999	0.996	RUNNING
0.916	0.017	0.934	0.916	0.925	0.905	0.995	0.982	WALKSTAIRS
0.976	0.003	0.977	0.976	0.976	0.973	0.999	0.994	SWEEPING
0.964	0.002	0.982	0.964	0.973	0.970	0.999	0.993	WASHINGHAND
0.841	0.003	0.644	0.841	0.730	0.734	0.996	0.862	FALLING
0.984	0.000	1.000	0.984	0.992	0.992	1.000	0.999	STANDING
0.993	0.001	0.992	0.993	0.993	0.992	1.000	0.998	SITTING
0.984	0.000	0.998	0.984	0.991	0.991	1.000	1.000	LYING
0.997	0.001	0.986	0.997	0.992	0.991	1.000	0.998	BENDING
0.991	0.000	1.000	0.991	0.996	0.995	1.000	0.999	LEANINGBACK
0.983	0.003	0.868	0.983	0.922	0.922	1.000	0.993	ROLLING
0.958	0.008	0.960	0.958	0.959	0.951	0.998	0.991	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1617	3	57	1	0	2	0	0	0	0	0	0	a = WALKING
0	1160	69	0	0	8	0	0	0	0	0	0	b = RUNNING
142	16	1986	3	0	7	0	0	0	0	0	15	c = WALKSTAIRS
1	2	7	1284	9	5	0	0	0	0	0	8	d = SWEEPING
0	0	4	24	803	1	0	0	0	0	0	1	e = WASHINGHANDS
1	3	3	0	0	58	0	1	1	0	0	2	f = FALLING
0	0	0	0	6	1	502	1	0	0	0	0	g = STANDING
0	0	0	2	0	2	0	611	0	0	0	0	h = SITTING
0	0	0	0	0	3	0	1	552	4	0	1	i = LYING
0	0	0	0	0	0	0	2	0	584	0	0	j = BENDING
0	0	0	0	0	0	0	0	0	4	458	0	k = LEANINGBACK
0	0	0	0	0	3	0	0	0	0	0	177	l = ROLLING

Default Model

C4.5 Decision Tree

Time taken to build model: 1.42 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	10005	97.9154 %
Incorrectly Classified Instances	213	2.0846 %
Kappa statistic	0.9762	
Mean absolute error	0.0041	
Root mean squared error	0.057	
Relative absolute error	2.7912 %	
Root relative squared error	21.0918 %	
Coverage of cases (0.95 level)	98.3069 %	
Mean rel. region size (0.95 level)	8.5234 %	
Total Number of Instances	10218	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.982	0.006	0.972	0.982	0.977	0.972	0.993	0.973	WALKING
0.985	0.002	0.984	0.985	0.985	0.983	0.993	0.979	RUNNING
0.970	0.007	0.974	0.970	0.972	0.965	0.987	0.967	WALKSTAIRS
0.976	0.004	0.974	0.976	0.975	0.972	0.990	0.960	SWEEPING
0.986	0.001	0.984	0.986	0.985	0.984	0.996	0.989	WASHINGHAND
0.623	0.001	0.768	0.623	0.688	0.690	0.846	0.561	FALLING
0.996	0.000	0.994	0.996	0.995	0.995	0.998	0.992	STANDING
0.997	0.001	0.990	0.997	0.994	0.993	1.000	0.996	SITTING
0.991	0.000	1.000	0.991	0.996	0.995	0.998	0.997	LYING
0.985	0.001	0.986	0.985	0.985	0.985	1.000	0.980	BENDING
0.989	0.000	0.993	0.989	0.991	0.991	0.999	0.995	LEANINGBACK
0.950	0.001	0.950	0.950	0.950	0.949	0.985	0.925	ROLLING
0.979	0.003	0.979	0.979	0.979	0.976	0.992	0.974	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1650	2	24	4	0	0	0	0	0	0	0	0	a = WALKING
0	1219	12	1	1	4	0	0	0	0	0	0	b = RUNNING
36	8	2105	16	2	0	0	0	0	0	0	2	c = WALKSTAIRS
5	1	12	1285	8	2	0	0	0	1	0	2	d = SWEEPING
0	2	0	9	821	0	0	0	0	0	0	1	e = WASHINGHANDS
4	4	7	2	0	43	3	1	0	1	0	4	f = FALLING
0	0	0	0	0	1	508	1	0	0	0	0	g = STANDING
0	0	0	0	0	1	0	613	0	1	0	0	h = SITTING
1	0	0	0	0	1	0	3	556	0	0	0	i = LYING
0	0	0	2	2	1	0	1	0	577	3	0	j = BENDING
0	0	0	0	0	0	0	0	0	5	457	0	k = LEANINGBACK
2	3	1	0	0	3	0	0	0	0	0	171	l = ROLLING

Default Model
K-Star (Instance-based)
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	10053	98.3852 %
Incorrectly Classified Instances	165	1.6148 %
Kappa statistic	0.9815	
Mean absolute error	0.0027	
Root mean squared error	0.0504	
Relative absolute error	1.8432 %	
Root relative squared error	18.6487 %	
Coverage of cases (0.95 level)	98.5809 %	
Mean rel. region size (0.95 level)	8.3823 %	
Total Number of Instances	10218	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.997	0.005	0.976	0.997	0.986	0.984	1.000	0.997	WALKING
0.978	0.000	0.999	0.978	0.989	0.987	1.000	1.000	RUNNING
0.974	0.005	0.982	0.974	0.978	0.972	0.999	0.996	WALKSTAIRS
0.990	0.004	0.975	0.990	0.982	0.980	0.999	0.994	SWEEPING
0.998	0.001	0.989	0.998	0.993	0.993	1.000	0.999	WASHINGHAND
0.609	0.001	0.875	0.609	0.718	0.728	0.981	0.858	FALLING
0.998	0.001	0.979	0.998	0.988	0.988	0.999	0.995	STANDING
0.995	0.001	0.986	0.995	0.990	0.990	1.000	0.997	SITTING
0.996	0.001	0.989	0.996	0.993	0.992	1.000	0.999	LYING
0.995	0.001	0.988	0.995	0.991	0.991	1.000	1.000	BENDING
0.994	0.000	0.994	0.994	0.994	0.993	1.000	1.000	LEANINGBACK
0.872	0.000	1.000	0.872	0.932	0.933	1.000	0.998	ROLLING
0.984	0.003	0.984	0.984	0.984	0.981	0.999	0.996	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1675	0	0	3	0	0	2	0	0	0	0	0	a = WALKING
1	1210	24	2	0	0	0	0	0	0	0	0	b = RUNNING
35	1	2113	19	0	0	0	1	0	0	0	0	c = WALKSTAIRS
4	0	3	1303	3	0	2	1	0	0	0	0	d = SWEEPING
0	0	0	1	831	0	0	1	0	0	0	0	e = WASHINGHANDS
1	0	6	3	5	42	5	1	3	2	1	0	f = FALLING
0	0	0	0	0	0	509	1	0	0	0	0	g = STANDING
0	0	0	0	0	1	1	612	0	1	0	0	h = SITTING
0	0	0	0	0	1	0	0	559	1	0	0	i = LYING
0	0	0	0	0	1	0	1	0	583	1	0	j = BENDING
0	0	0	0	0	0	0	0	0	3	459	0	k = LEANINGBACK
0	0	5	6	1	3	1	3	3	0	1	157	l = ROLLING

```
Default Model
MultilayerPerceptron (Neural Network)
Time taken to build model: 248.36 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances        9901               96.8976 %
Incorrectly Classified Instances       317                3.1024 %
Kappa statistic                          0.9646
Mean absolute error                      0.0061
Root mean squared error                  0.0665
Relative absolute error                  4.1997 %
Root relative squared error             24.6109 %
Coverage of cases (0.95 level)          97.3772 %
Mean rel. region size (0.95 level)       8.7729 %
Total Number of Instances            10218
=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0.983    0.006    0.969      0.983    0.976      0.971    0.997     0.978    WALKING
0.981    0.001    0.993      0.981    0.987      0.986    0.994     0.989    RUNNING
0.973    0.008    0.971      0.973    0.972      0.965    0.995     0.985    WALKSTAIRS
0.978    0.002    0.986      0.978    0.982      0.979    0.995     0.989    SWEEPING
0.988    0.002    0.981      0.988    0.984      0.983    0.994     0.989    WASHINGHAND
0.725    0.004    0.562      0.725    0.633      0.635    0.943     0.632    FALLING
0.996    0.000    0.994      0.996    0.995      0.995    0.998     0.996    STANDING
0.998    0.002    0.975      0.998    0.986      0.986    1.000     0.999    SITTING
0.765    0.000    0.993      0.765    0.864      0.865    0.896     0.806    LYING
0.997    0.008    0.886      0.997    0.938      0.936    0.998     0.994    BENDING
0.983    0.000    1.000      0.983    0.991      0.991    0.990     0.985    LEANINGBACK
0.978    0.002    0.880      0.978    0.926      0.926    1.000     0.976    ROLLING
0.969    0.004    0.971      0.969    0.969      0.966    0.990     0.975    Weighted Avg.
=== Confusion Matrix ===
    a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
 1651    0   26    2    0    0    1    0    0    0    0    0 |    a = WALKING
    5 1214    9    1    0    4    1    0    1    0    0    2 |    b = RUNNING
   39    3 2111    9    1    0    0    1    0    0    0    5 |    c = WALKSTAIRS
    4    2   15 1287    1    2    0    4    0    0    0    1 |    d = SWEEPING
    0    0    5    5  823    0    0    0    0    0    0    0 |    e = WASHINGHANDS
    4    3    4    1    1   50    1    1    0    0    0    4 |    f = FALLING
    0    0    1    0    0    0  508    1    0    0    0    0 |    g = STANDING
    0    0    0    0    0    1    0  614    0    0    0    0 |    h = SITTING
    0    0    0    0   13   28    0    8  429   71    0   12 |    i = LYING
    0    0    0    0    0    0    0    1    1  584    0    0 |    j = BENDING
    0    0    0    0    0    4    0    0    0    4  454    0 |    k = LEANINGBACK
    0    0    3    0    0    0    0    0    1    0    0  176 |    l = ROLLING
```

Filtered Model

Naive Bayes

Time taken to build model: 0.11 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9143	98.2696 %
Incorrectly Classified Instances	161	1.7304 %
Kappa statistic	0.9803	
Mean absolute error	0.0031	
Root mean squared error	0.0495	
Relative absolute error	2.0922 %	
Root relative squared error	18.2715 %	
Coverage of cases (0.95 level)	99.0864 %	
Mean rel. region size (0.95 level)	8.525 %	
Total Number of Instances	9304	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.998	0.004	0.981	0.998	0.990	0.988	1.000	1.000	WALKING
0.916	0.002	0.984	0.916	0.949	0.943	0.999	0.993	RUNNING
0.973	0.012	0.950	0.973	0.961	0.952	0.998	0.994	WALKSTAIRS
0.997	0.001	0.992	0.997	0.994	0.994	0.999	0.995	SWEEPING
0.997	0.000	1.000	0.997	0.999	0.999	1.000	0.999	WASHINGHAND
0.962	0.001	0.862	0.962	0.909	0.910	1.000	0.908	FALLING
0.996	0.000	1.000	0.996	0.998	0.998	1.000	1.000	STANDING
0.998	0.000	1.000	0.998	0.999	0.999	1.000	1.000	SITTING
0.994	0.000	0.996	0.994	0.995	0.995	0.999	0.997	LYING
0.993	0.000	1.000	0.993	0.997	0.996	0.999	0.998	BENDING
0.996	0.000	1.000	0.996	0.998	0.998	1.000	1.000	LEANINGBACK
1.000	0.000	0.983	1.000	0.991	0.991	1.000	1.000	ROLLING
0.983	0.003	0.983	0.983	0.983	0.980	0.999	0.996	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1513	0	3	0	0	0	0	0	0	0	0	0	a = WALKING
0	968	89	0	0	0	0	0	0	0	0	0	b = RUNNING
29	13	1752	2	0	3	0	0	0	0	0	2	c = WALKSTAIRS
0	3	0	1239	0	1	0	0	0	0	0	0	d = SWEEPING
0	0	0	2	773	0	0	0	0	0	0	0	e = WASHINGHANDS
0	0	1	0	0	50	0	0	0	0	0	1	f = FALLING
0	0	0	2	0	0	492	0	0	0	0	0	g = STANDING
0	0	0	0	0	0	0	608	1	0	0	0	h = SITTING
0	0	0	0	0	3	0	0	542	0	0	0	i = LYING
0	0	0	2	0	1	0	0	1	576	0	0	j = BENDING
0	0	0	2	0	0	0	0	0	0	456	0	k = LEANINGBACK
0	0	0	0	0	0	0	0	0	0	0	174	l = ROLLING

Filtered Model

Bayes Net

Time taken to build model: 0.56 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9303	99.9893 %
Incorrectly Classified Instances	1	0.0107 %
Kappa statistic	0.9999	
Mean absolute error	0.0001	
Root mean squared error	0.0046	
Relative absolute error	0.0344 %	
Root relative squared error	1.7077 %	
Coverage of cases (0.95 level)	100 %	
Mean rel. region size (0.95 level)	8.3423 %	
Total Number of Instances	9304	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	WALKING
0.999	0.000	1.000	0.999	1.000	0.999	1.000	1.000	RUNNING
1.000	0.000	0.999	1.000	1.000	1.000	1.000	1.000	WALKSTAIRS
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	SWEEPING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	WASHINGHAND
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	FALLING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	STANDING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	SITTING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	LYING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	BENDING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	LEANINGBACK
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	ROLLING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1516	0	0	0	0	0	0	0	0	0	0	0	a = WALKING
0	1056	1	0	0	0	0	0	0	0	0	0	b = RUNNING
0	0	1801	0	0	0	0	0	0	0	0	0	c = WALKSTAIRS
0	0	0	1243	0	0	0	0	0	0	0	0	d = SWEEPING
0	0	0	0	775	0	0	0	0	0	0	0	e = WASHINGHANDS
0	0	0	0	0	52	0	0	0	0	0	0	f = FALLING
0	0	0	0	0	0	494	0	0	0	0	0	g = STANDING
0	0	0	0	0	0	0	609	0	0	0	0	h = SITTING
0	0	0	0	0	0	0	0	545	0	0	0	i = LYING
0	0	0	0	0	0	0	0	0	580	0	0	j = BENDING
0	0	0	0	0	0	0	0	0	0	458	0	k = LEANINGBACK
0	0	0	0	0	0	0	0	0	0	0	174	l = ROLLING

Filtered Model

C4.5 Decision Tree

Time taken to build model: 0.95 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9215	99.0434 %
Incorrectly Classified Instances	89	0.9566 %
Kappa statistic	0.9891	
Mean absolute error	0.0019	
Root mean squared error	0.0391	
Relative absolute error	1.3008 %	
Root relative squared error	14.4413 %	
Coverage of cases (0.95 level)	99.1509 %	
Mean rel. region size (0.95 level)	8.3898 %	
Total Number of Instances	9304	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.997	0.001	0.993	0.997	0.995	0.994	0.999	0.993	WALKING
0.991	0.001	0.994	0.991	0.993	0.992	0.996	0.990	RUNNING
0.990	0.003	0.989	0.990	0.989	0.987	0.995	0.984	WALKSTAIRS
0.993	0.003	0.982	0.993	0.988	0.986	0.996	0.974	SWEEPING
0.994	0.001	0.994	0.994	0.994	0.993	0.998	0.986	WASHINGHAND
0.654	0.000	0.895	0.654	0.756	0.764	0.863	0.661	FALLING
0.996	0.000	0.998	0.996	0.997	0.997	0.998	0.996	STANDING
0.995	0.000	0.995	0.995	0.995	0.995	0.998	0.990	SITTING
0.991	0.000	0.993	0.991	0.992	0.991	0.995	0.985	LYING
0.995	0.000	0.998	0.995	0.997	0.996	0.998	0.994	BENDING
0.993	0.000	0.991	0.993	0.992	0.992	0.999	0.986	LEANINGBACK
0.948	0.001	0.948	0.948	0.948	0.947	0.975	0.901	ROLLING
0.990	0.001	0.990	0.990	0.990	0.989	0.996	0.983	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1511	0	5	0	0	0	0	0	0	0	0	0	a = WALKING
0	1048	5	3	0	1	0	0	0	0	0	0	b = RUNNING
7	2	1783	5	0	0	0	0	0	0	1	3	c = WALKSTAIRS
2	0	3	1234	2	1	0	0	0	1	0	0	d = SWEEPING
0	0	0	5	770	0	0	0	0	0	0	0	e = WASHINGHANDS
1	3	3	6	1	34	0	0	0	0	0	4	f = FALLING
0	0	0	0	2	0	492	0	0	0	0	0	g = STANDING
0	0	1	0	0	0	0	606	2	0	0	0	h = SITTING
0	0	0	0	0	0	0	1	540	0	2	2	i = LYING
0	0	0	2	0	0	0	0	0	577	1	0	j = BENDING
0	0	1	1	0	0	1	0	0	0	455	0	k = LEANINGBACK
0	1	2	0	0	2	0	2	2	0	0	165	l = ROLLING

Filtered Model

K-Star (Instance-based)

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	9217	99.0649 %
Incorrectly Classified Instances	87	0.9351 %
Kappa statistic	0.9894	
Mean absolute error	0.0015	
Root mean squared error	0.0383	
Relative absolute error	1.0472 %	
Root relative squared error	14.1422 %	
Coverage of cases (0.95 level)	99.2046 %	
Mean rel. region size (0.95 level)	8.3548 %	
Total Number of Instances	9304	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.999	0.002	0.988	0.999	0.993	0.992	1.000	1.000	WALKING
0.980	0.000	1.000	0.980	0.990	0.989	1.000	1.000	RUNNING
0.990	0.003	0.986	0.990	0.988	0.985	1.000	0.999	WALKSTAIRS
0.997	0.001	0.991	0.997	0.994	0.993	1.000	0.997	SWEEPING
1.000	0.001	0.990	1.000	0.995	0.994	1.000	1.000	WASHINGHAND
0.635	0.000	0.917	0.635	0.750	0.762	0.996	0.848	FALLING
1.000	0.000	0.992	1.000	0.996	0.996	1.000	1.000	STANDING
1.000	0.001	0.989	1.000	0.994	0.994	1.000	0.997	SITTING
0.994	0.001	0.991	0.994	0.993	0.992	1.000	0.999	LYING
1.000	0.000	0.995	1.000	0.997	0.997	1.000	1.000	BENDING
1.000	0.000	0.993	1.000	0.997	0.997	1.000	1.000	LEANINGBACK
0.885	0.000	1.000	0.885	0.939	0.940	1.000	0.997	ROLLING
0.991	0.001	0.991	0.991	0.990	0.989	1.000	0.998	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1514	0	0	2	0	0	0	0	0	0	0	0	a = WALKING
3	1036	18	0	0	0	0	0	0	0	0	0	b = RUNNING
14	0	1783	3	0	0	0	1	0	0	0	0	c = WALKSTAIRS
1	0	1	1239	2	0	0	0	0	0	0	0	d = SWEEPING
0	0	0	0	775	0	0	0	0	0	0	0	e = WASHINGHANDS
0	0	1	2	6	33	3	1	2	2	2	0	f = FALLING
0	0	0	0	0	0	494	0	0	0	0	0	g = STANDING
0	0	0	0	0	0	0	609	0	0	0	0	h = SITTING
0	0	0	0	0	0	0	2	542	1	0	0	i = LYING
0	0	0	0	0	0	0	0	0	580	0	0	j = BENDING
0	0	0	0	0	0	0	0	0	0	458	0	k = LEANINGBACK
0	0	5	4	0	3	1	3	3	0	1	154	l = ROLLING

Filtered Model
MultilayerPerceptron (Neural Network)
Time taken to build model: 222.79 seconds
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	9183	98.6995 %
Incorrectly Classified Instances	121	1.3005 %
Kappa statistic	0.9852	
Mean absolute error	0.0029	
Root mean squared error	0.0419	
Relative absolute error	1.9903 %	
Root relative squared error	15.4492 %	
Coverage of cases (0.95 level)	98.9897 %	
Mean rel. region size (0.95 level)	8.5805 %	
Total Number of Instances	9304	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.999	0.001	0.994	0.999	0.997	0.996	1.000	0.998	WALKING
0.998	0.000	0.996	0.998	0.997	0.997	1.000	1.000	RUNNING
0.993	0.001	0.994	0.993	0.993	0.992	1.000	0.999	WALKSTAIRS
0.995	0.001	0.995	0.995	0.995	0.994	1.000	0.999	SWEEPING
1.000	0.000	0.997	1.000	0.999	0.999	1.000	1.000	WASHINGHAND
0.712	0.003	0.587	0.712	0.643	0.644	0.980	0.782	FALLING
1.000	0.000	0.996	1.000	0.998	0.998	1.000	1.000	STANDING
1.000	0.001	0.985	1.000	0.993	0.992	1.000	1.000	SITTING
0.859	0.000	0.998	0.859	0.923	0.922	0.932	0.879	LYING
1.000	0.004	0.948	1.000	0.973	0.972	1.000	1.000	BENDING
0.991	0.000	1.000	0.991	0.996	0.995	0.998	0.993	LEANINGBACK
0.983	0.002	0.900	0.983	0.940	0.939	1.000	0.984	ROLLING
0.987	0.001	0.988	0.987	0.987	0.986	0.996	0.990	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1515	0	0	1	0	0	0	0	0	0	0	0	a = WALKING
0	1055	2	0	0	0	0	0	0	0	0	0	b = RUNNING
7	2	1788	3	0	0	0	0	0	0	0	1	c = WALKSTAIRS
1	0	3	1237	1	1	0	0	0	0	0	0	d = SWEEPING
0	0	0	0	775	0	0	0	0	0	0	0	e = WASHINGHANDS
1	2	6	1	1	37	0	0	0	0	0	4	f = FALLING
0	0	0	0	0	0	494	0	0	0	0	0	g = STANDING
0	0	0	0	0	0	0	609	0	0	0	0	h = SITTING
0	0	0	0	0	22	0	9	468	32	0	14	i = LYING
0	0	0	0	0	0	0	0	0	580	0	0	j = BENDING
0	0	0	0	0	2	2	0	0	0	454	0	k = LEANINGBACK
0	0	0	1	0	1	0	0	1	0	0	171	l = ROLLING

Adapted Model
Naive Bayes
Time taken to build model: 0.18 seconds
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	11024	97.7565 %
Incorrectly Classified Instances	253	2.2435 %
Kappa statistic	0.9747	
Mean absolute error	0.004	
Root mean squared error	0.0578	
Relative absolute error	2.692 %	
Root relative squared error	21.2616 %	
Coverage of cases (0.95 level)	98.6078 %	
Mean rel. region size (0.95 level)	8.524 %	
Total Number of Instances	11277	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.997	0.007	0.965	0.997	0.981	0.977	1.000	0.999	WALKING
0.913	0.001	0.986	0.913	0.948	0.943	0.999	0.993	RUNNING
0.956	0.011	0.949	0.956	0.952	0.942	0.997	0.991	WALKSTAIRS
0.991	0.001	0.991	0.991	0.991	0.990	0.998	0.996	SWEEPING
0.990	0.001	0.991	0.990	0.990	0.990	1.000	0.998	WASHINGHAND
0.885	0.002	0.719	0.885	0.793	0.796	0.999	0.789	FALLING
0.991	0.000	0.995	0.991	0.993	0.993	0.999	0.997	STANDING
0.996	0.000	1.000	0.996	0.998	0.998	1.000	1.000	SITTING
0.992	0.001	0.982	0.992	0.987	0.986	0.999	0.996	LYING
0.986	0.000	1.000	0.986	0.993	0.993	0.999	0.999	BENDING
0.996	0.000	1.000	0.996	0.998	0.998	1.000	1.000	LEANINGBACK
0.977	0.001	0.938	0.977	0.957	0.957	1.000	0.990	ROLLING
0.978	0.004	0.978	0.978	0.978	0.974	0.999	0.995	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1863	0	4	1	0	0	0	0	0	0	0	0	a = WALKING
0	1040	94	0	0	5	0	0	0	0	0	0	b = RUNNING
65	12	1907	3	0	6	0	0	0	0	0	2	c = WALKSTAIRS
0	3	3	1390	6	1	0	0	0	0	0	0	d = SWEEPING
0	0	0	5	874	0	4	0	0	0	0	0	e = WASHINGHANDS
0	0	1	0	0	46	0	0	0	0	0	5	f = FALLING
2	0	0	2	2	0	815	0	0	0	0	1	g = STANDING
0	0	0	0	0	0	0	968	3	0	0	1	h = SITTING
0	0	0	0	0	1	0	0	645	0	0	4	i = LYING
0	0	0	1	0	1	0	0	8	708	0	0	j = BENDING
0	0	1	0	0	0	0	0	0	0	555	1	k = LEANINGBACK
0	0	0	0	0	4	0	0	1	0	0	213	l = ROLLING

Adapted Model

Bayes Net

Time taken to build model: 1.11 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	11222	99.5123 %
Incorrectly Classified Instances	55	0.4877 %
Kappa statistic	0.9945	
Mean absolute error	0.0009	
Root mean squared error	0.0264	
Relative absolute error	0.5906 %	
Root relative squared error	9.7136 %	
Coverage of cases (0.95 level)	99.7694 %	
Mean rel. region size (0.95 level)	8.3851 %	
Total Number of Instances	11277	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.998	0.002	0.988	0.998	0.993	0.992	1.000	1.000	WALKING
0.994	0.000	1.000	0.994	0.997	0.997	1.000	1.000	RUNNING
0.984	0.001	0.995	0.984	0.990	0.987	1.000	1.000	WALKSTAIRS
0.998	0.000	0.998	0.998	0.998	0.998	1.000	1.000	SWEEPING
0.997	0.000	0.998	0.997	0.997	0.997	1.000	1.000	WASHINGHAND
0.981	0.001	0.836	0.981	0.903	0.905	1.000	0.987	FALLING
0.998	0.000	0.998	0.998	0.998	0.997	1.000	1.000	STANDING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	SITTING
0.998	0.000	1.000	0.998	0.999	0.999	1.000	1.000	LYING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	BENDING
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	LEANINGBACK
0.986	0.001	0.973	0.986	0.979	0.979	1.000	0.998	ROLLING
0.995	0.001	0.995	0.995	0.995	0.994	1.000	1.000	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1864	0	3	1	0	0	0	0	0	0	0	0	a = WALKING
0	1132	6	0	0	1	0	0	0	0	0	0	b = RUNNING
22	0	1964	1	0	4	0	0	0	0	0	4	c = WALKSTAIRS
0	0	1	1400	0	1	0	0	0	0	0	1	d = SWEEPING
0	0	0	1	880	0	2	0	0	0	0	0	e = WASHINGHANDS
0	0	0	0	0	51	0	0	0	0	0	1	f = FALLING
0	0	0	0	2	0	820	0	0	0	0	0	g = STANDING
0	0	0	0	0	0	0	972	0	0	0	0	h = SITTING
0	0	0	0	0	1	0	0	649	0	0	0	i = LYING
0	0	0	0	0	0	0	0	0	718	0	0	j = BENDING
0	0	0	0	0	0	0	0	0	0	557	0	k = LEANINGBACK
0	0	0	0	0	3	0	0	0	0	0	215	l = ROLLING

Adapted Model

C4.5 Decision Tree

Time taken to build model: 1.7 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	11107	98.4925 %
Incorrectly Classified Instances	170	1.5075 %
Kappa statistic	0.983	
Mean absolute error	0.003	
Root mean squared error	0.0492	
Relative absolute error	2.0261 %	
Root relative squared error	18.1164 %	
Coverage of cases (0.95 level)	98.6876 %	
Mean rel. region size (0.95 level)	8.4508 %	
Total Number of Instances	11277	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.990	0.002	0.990	0.990	0.990	0.988	0.995	0.982	WALKING
0.985	0.001	0.993	0.985	0.989	0.988	0.996	0.986	RUNNING
0.983	0.005	0.979	0.983	0.981	0.977	0.992	0.974	WALKSTAIRS
0.981	0.003	0.981	0.981	0.981	0.978	0.994	0.978	SWEEPING
0.984	0.002	0.982	0.984	0.983	0.982	0.997	0.982	WASHINGHAND
0.635	0.001	0.733	0.635	0.680	0.681	0.868	0.562	FALLING
0.991	0.001	0.985	0.991	0.988	0.988	0.996	0.973	STANDING
0.996	0.000	0.999	0.996	0.997	0.997	0.999	0.998	SITTING
0.991	0.001	0.985	0.991	0.988	0.987	0.996	0.974	LYING
0.989	0.001	0.992	0.989	0.990	0.990	0.999	0.993	BENDING
0.989	0.001	0.987	0.989	0.988	0.988	0.994	0.977	LEANINGBACK
0.945	0.001	0.954	0.945	0.949	0.948	0.973	0.905	ROLLING
0.985	0.002	0.985	0.985	0.985	0.983	0.994	0.978	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1850	0	15	1	0	0	2	0	0	0	0	0	a = WALKING
0	1122	12	1	0	3	0	0	0	0	0	1	b = RUNNING
16	4	1962	7	0	1	0	0	0	0	1	4	c = WALKSTAIRS
0	0	5	1377	11	5	2	0	0	3	0	0	d = SWEEPING
0	0	0	5	869	0	6	0	0	2	1	0	e = WASHINGHANDS
0	2	6	2	0	33	2	0	2	1	1	3	f = FALLING
2	0	0	0	4	0	815	0	1	0	0	0	g = STANDING
0	0	1	0	0	0	0	968	2	0	1	0	h = SITTING
0	0	1	0	0	1	0	0	644	0	2	2	i = LYING
0	0	0	7	1	0	0	0	0	710	0	0	j = BENDING
0	0	2	2	0	1	0	1	0	0	551	0	k = LEANINGBACK
0	2	1	2	0	1	0	0	5	0	1	206	l = ROLLING

```
Adapted Model
K-Star (Instance-based)
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances       11082              98.2708 %
Incorrectly Classified Instances       195               1.7292 %
Kappa statistic                          0.9805
Mean absolute error                      0.0029
Root mean squared error                  0.0524
Relative absolute error                  1.9458 %
Root relative squared error             19.2931 %
Coverage of cases (0.95 level)          98.4836 %
Mean rel. region size (0.95 level)       8.3703 %
Total Number of Instances            11277
=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
0.991    0.006    0.972      0.991    0.981      0.978    1.000     0.998     WALKING
0.970    0.000    1.000      0.970    0.985      0.983    1.000     1.000     RUNNING
0.971    0.005    0.975      0.971    0.973      0.967    0.999     0.997     WALKSTAIRS
0.982    0.002    0.989      0.982    0.985      0.983    1.000     0.997     SWEEPING
0.989    0.002    0.978      0.989    0.983      0.982    1.000     0.996     WASHINGHAND
0.615    0.000    0.970      0.615    0.753      0.772    0.995     0.803     FALLING
1.000    0.002    0.970      1.000    0.985      0.984    1.000     0.997     STANDING
1.000    0.001    0.989      1.000    0.994      0.994    1.000     1.000     SITTING
0.998    0.001    0.985      0.998    0.992      0.991    1.000     0.998     LYING
1.000    0.000    0.993      1.000    0.997      0.996    1.000     1.000     BENDING
1.000    0.000    0.993      1.000    0.996      0.996    1.000     0.998     LEANINGBACK
0.858    0.000    1.000      0.858    0.923      0.925    0.999     0.991     ROLLING
0.983    0.003    0.983      0.983    0.982      0.980    1.000     0.997     Weighted Avg.
=== Confusion Matrix ===
    a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
 1852    0   10    0    0    0    6    0    0    0    0    0 |    a = WALKING
    1 1105   32    1    0    0    0    0    0    0    0    0 |    b = RUNNING
   47    0 1937    4    0    0    1    4    0    0    2    0 |    c = WALKSTAIRS
    6    0    1 1378   14    0    4    0    0    0    0    0 |    d = SWEEPING
    0    0    0    0  873    0    7    0    0    2    1    0 |    e = WASHINGHANDS
    0    0    2    1    5   32    6    1    2    3    0    0 |    f = FALLING
    0    0    0    0    0    0  822    0    0    0    0    0 |    g = STANDING
    0    0    0    0    0    0    0  972    0    0    0    0 |    h = SITTING
    0    0    0    0    0    0    0    1  649    0    0    0 |    i = LYING
    0    0    0    0    0    0    0    0    0  718    0    0 |    j = BENDING
    0    0    0    0    0    0    0    0    0    0  557    0 |    k = LEANINGBACK
    0    0    4   10    1    1    1    5    8    0    1  187 |    l = ROLLING
```

Adapted Model

MultilayerPerceptron (Neural Network)

Time taken to build model: 270.44 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	11079	98.2442 %
Incorrectly Classified Instances	198	1.7558 %
Kappa statistic	0.9802	
Mean absolute error	0.0037	
Root mean squared error	0.0494	
Relative absolute error	2.5295 %	
Root relative squared error	18.1934 %	
Coverage of cases (0.95 level)	98.8029 %	
Mean rel. region size (0.95 level)	8.6119 %	
Total Number of Instances	11277	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.991	0.003	0.985	0.991	0.988	0.986	1.000	0.999	WALKING
0.993	0.001	0.994	0.993	0.993	0.993	0.999	0.997	RUNNING
0.980	0.003	0.984	0.980	0.982	0.978	0.999	0.996	WALKSTAIRS
0.989	0.002	0.989	0.989	0.989	0.988	0.999	0.997	SWEEPING
0.984	0.002	0.982	0.984	0.983	0.982	0.997	0.991	WASHINGHAND
0.692	0.002	0.632	0.692	0.661	0.660	0.949	0.746	FALLING
0.993	0.001	0.986	0.993	0.989	0.988	0.999	0.996	STANDING
1.000	0.001	0.989	1.000	0.994	0.994	1.000	1.000	SITTING
0.894	0.000	0.997	0.894	0.942	0.941	0.962	0.913	LYING
1.000	0.002	0.968	1.000	0.984	0.983	1.000	0.999	BENDING
0.993	0.000	0.996	0.993	0.995	0.994	0.997	0.989	LEANINGBACK
0.959	0.003	0.878	0.959	0.917	0.916	0.991	0.965	ROLLING
0.982	0.002	0.983	0.982	0.982	0.981	0.997	0.990	Weighted Avg.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
1851	0	16	0	0	0	1	0	0	0	0	0	a = WALKING
0	1131	7	0	0	1	0	0	0	0	0	0	b = RUNNING
25	1	1955	7	0	0	0	2	0	0	1	4	c = WALKSTAIRS
0	1	4	1388	6	0	2	0	0	1	0	1	d = SWEEPING
1	0	0	3	869	0	8	0	0	1	1	0	e = WASHINGHANDS
1	3	3	1	1	36	0	0	0	0	0	7	f = FALLING
1	0	0	0	4	0	816	1	0	0	0	0	g = STANDING
0	0	0	0	0	0	0	972	0	0	0	0	h = SITTING
0	0	0	0	5	18	0	8	581	21	0	17	i = LYING
0	0	0	0	0	0	0	0	0	718	0	0	j = BENDING
0	0	0	0	0	2	1	0	0	1	553	0	k = LEANINGBACK
0	2	1	4	0	0	0	0	2	0	0	209	l = ROLLING

Appendix D: Examples of Fuzzy Rules for Inference

The following rules are used in the fuzzy reasoning engine for inference:

$ActivityRule_1$: fuzzyEngine.parseRule("if Activity is Sleeping and (TV is ON or Cooker is ON or Lights is ON) then Situation is Abnormal");

$ActivityRule_2$: fuzzyEngine.parseRule("if Activity is Sleeping and Location is not Bedroom then Situation is Abnormal");

$ActivityRule_3$: fuzzyEngine.parseRule("if Activity is Sleeping and Duration is ExtremlyLong then Situation is Abnormal");

$ActivityRule_4$: fuzzyEngine.parseRule("if Activity is Resting and (Location is Bathroom or Location is Kitchen) then Situation is Abnormal");

$ActivityRule_5$: fuzzyEngine.parseRule("if (Activity is Resting or Activity is WatchingTV) and Duration is VeryLong then Situation is Abnormal");

$ActivityRule_6$: fuzzyEngine.parseRule("if Activity is Eating and (Location is Bathroom or Location is Bedroom) then Situation is Abnormal");

$ActivityRule_7$: fuzzyEngine.parseRule("if (Activity is Eating or Activity is Cooking or Activity is Bathing or Activity is Exercising and Time is Night and Lights is OFF then Situation is Abnormal");

$ActivityRule_8$: fuzzyEngine.parseRule("if (Activity is Eating or Activity is WatchingTV or Activity is Chatting or Activity is Cooking or Activity is Bathing or Activity is Exercising)and Time is LateNight then Situation is Abnormal");

$ActivityRule_9$: fuzzyEngine.parseRule("if (Activity is Toileting or Activity is Bathing) and Duration is not Short then Situation is Abnormal");

$ActivityRule_10$: fuzzyEngine.parseRule("if Activity is Toileting and Location is not Bathroom then Situation is Abnormal");

$ActivityRule_11$: fuzzyEngine.parseRule("if (Activity is Toileting and Time is Night and Lights is OFF then Situation is Abnormal");

$ActivityRule_12$: fuzzyEngine.parseRule("if Activity is Cooking and Cooker is OFF then Situation is Abnormal");

$ActivityRule_13$: fuzzyEngine.parseRule("if Location is Outdoor and (Cooker is ON or Windows is Open) then Situation is Abnormal");

$ActivityRule_14$: fuzzyEngine.parseRule("if Location is Outdoor and Time is LateNight then Situation is Abnormal");

$ActivityRule_15$: fuzzyEngine.parseRule("if Location is Outdoor and Duration is VeryLong then Situation is Abnormal");

$MedicalRule_1$: fuzzyEngine.parseRule("if Activity is not Exercising and (HeartRate is VeryHigh or RespirationRate is VeryHigh) then Situation is Abnormal");

$MedicalRule_2$: fuzzyEngine.parseRule("if (Activity is Sleeping or Activity is Resting or Activity is WatchingTV or Activity is Toileting) and SystolicBloodPressure is High and DynamicBloodPressure is High) then Situation is Abnormal");

$MedicalRule_3$: fuzzyEngine.parseRule("if SystolicBloodPressure is VeryHigh and DynamicBloodPressure is VeryHigh then Situation is Abnormal");

$MedicalRule_4$: fuzzyEngine.parseRule("if (HeartRate is VeryLow or RespirationRate is VeryLow) and (SystolicBloodPressure is VeryLow or DynamicBloodPressure is VeryLow) then Situation is Abnormal");

References

Aamodt, A. (1994). Cased-based reasoning: foundation issues. *AICOM*, page 3959. 26, 126

Aarts, E. and Marzano, S. (2003). *The New Everyday: Visions of Ambient Intelligence*. 010 Publishers. 34

Adami, A., Pavel, M., Hayes, T., and Singer, C. (2010). Detection of movement in bed using unobtrusive load cell sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):481–490. 22

Adobe (2011). Amf 3 specification category: Actionscript serialization. Adobe Systems Inc. 149

Aggarwal, C. (2001). Re-designing distance functions and distance-based applications for high dimensional data. *SIGMOD Record*, page 1318. 128

Ahmadi, H., Pham, N., Ganti, R., Abdelzaher, T., Nath, S., and Han, J. (2010). Privacy-aware regression modeling of participatory sensing data. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 99–112. 63

Ahuja, S., Mani, S., and Zambrano, J. (2012). A survey of the state of cloud computing in healthcare. *Network and Communication Technologies*, 1(2):12. 29

Alwan, M. and Nobel, J. (2008). State of technology in aging services according to field experts and thought leader. Technical report, Center for Aging Services Technologies. 2, 3 amChart (Accessed: December 2011). Javascript/html5 charts for modern browsers. www.amCharts.com. 152

Aminian, M. and Naji, H. (2013). A hospital healthcare monitoring system using wireless sensor networks. *J Health Med Inform*, 4(121):2. 16

Arnrich, B., Mayora, O., Bardram, J., and Trster, G. (2010). Pervasive healthcare: paving the way for a pervasive, user-centered and preventive healthcare model. *Methods Inf Med*, 49(1):67–73. 1

Asanchezrg, Belend, Jarecio, and Pagoncal (Accessed: September 2012). jcolibri: Cbr framework. Available online at `http://sourceforge.net/projects/jcolibri-cbr`. 137

Atmel (Accessed: October 2013). Available online at `http://www.atmel.com/products/microcontrollers/default.aspx`. 46

Aztiria, A., Izaguirre, A., and Augusto, J. (2010). Learning patterns in ambient intelligence environments: a survey. *Artificial Intelligence Review*, 34(1):35–51. 23

Badlani, A. and Bhanot, S. (2011). Smart home system design based on artificial neural networks. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1. 24

Bali, R., Troshani, I., Goldberg, S., and Wickramasinghe, N. (2013). Pervasive computing and healthcare. In *Pervasive Health Knowledge Management*, Healthcare Delivery in the Information Age, pages 7–13. Springer New York. 7

Bao, L. and Intille, S. (2004). Activity recognition from user-annotated acceleration data. In *Proceedings of Pervasive Computing, LNCS 3001*, pages 1–17. 72

Bardram, J. (2008). Pervasive healthcare as a scientific discipline. *Methods Inf Med*, 47(3):178–185. 2

Bassiliades, N., Governatori, G., and Paschke, A. (2011). *Rule-Based Reasoning, Programming, and Applications: 5th International Symposium, RuleML*

2011-Europe, Barcelona, Spain, July 19-21, 2011, Proceedings, volume 6826. Springer. 104

Bělohlávek, R. and Klir, G. (2011). *Concepts and fuzzy logic*. MIT Press. 113

Benzo, R., Karpman, C., and Kluka, E. (2014). Gait speed and activities of daily living disability in individuals with copd. *Am J Respir Crit Care Med*, 189:A5854. 18

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180. 23

Biacino, L. (2002). Fuzzy logic, continuity and effectiveness. *Archive for Mathematical Logic*, 7(41):643–667. 25

BIOPAC System Inc. (Accessed: October 2013). Nibp100d noninvasive blood pressure monitoring system. Available online at `http://www.biopac.com/Product_Spec_PDF/NIBP100D.pdf`. 162

Biosensics (Accessed: October 2013). Innovations in wearable sensors for healthcare. Available online at `http://www.biosensics.com`. 45, 47

Black, A., Car, J., Pagliari, C., Anandan, C., Cresswell, K., Bokun, T., McKinstry, B., Procter, R., Majeed, A., and Sheikh, A. (2011). The impact of ehealth on the quality and safety of health care: a systematic overview. *PLoS medicine*, 8(1). 14

Bosch (Accessed: October 2013). Health Buddy `http://www.bosch-telehealth.com/content/language1/html/5578_ENU_XHTML.aspx`. 21

Brdiczka, O., Reignier, P., and Crowley, J. (2005). Supervised learning of an abstract context model for an intelligent environment. In *Proceedings of the 2005 joint conference on smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, volume 121. 25

Breen, C. (2011). *The role of telecare in supporting carers of older people*. Centre for Ageing Research and Development in Ireland (CARDI). 148

Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., and Srivastava, M. (2006). Participatory sensing. In *Proceedings of ACM SenSys Workshop. World-Sensor-Web*. 60

Caldeira, J., Rodrigues, J., and Lorenz, P. (2012). Toward ubiquitous mobility solutions for body sensor networks on healthcare. *Communications Magazine, IEEE*, 50(5):108–115. 16

Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., and Peterson, R. (2006). People-centric urban sensing. In *Proceedings of the 2nd annual international workshop on Wireless internet*. 60

Carretero, S., Stewart, J., Centeno, C., Barbabella, F., Schmidt, A., Lamontagne-Godwin, F., and Lamura, G. (2012). Can technology-based services support long-term care challenges in home care? *Analysis of evidence from social innovation good practices across the EU CARICT Project Summary Report*. 15

Chan, M., Hariton, C., Ringeard, P., and Campo, E. (1995). Smart house automation system for the elderly and the disabled. In *Proceedings of the 1995 IEEE international conference on systems, man and cybernetics*. 24

Charlon, Y., Bourennane, W., Bettahar, F., and Campo, E. (2013). Activity monitoring system for elderly in a context of smart home. *IRBM*, 34(1):60–63. 18

Chattopadhyay, S., Banerjee, S., Rabhi, F., and Acharya, U. (2013). A case-based reasoning system for complex medical diagnosis. *Expert Systems*, 30(1):12–20. 26

Chen, C., Kehtarnavaz, N., and Jafari, R. (2014). A medication adherence monitoring system for pill bottles based on a wearable inertial sensor. In *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'14)*, pages 4983–4986. 18

Chen, D., Liu, Z., Wang, L., Dou, M., Chen, J., and Li, H. (2013). Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems. *Mobile Networks and Applications*, 18(5):651–663. 16

Chen, X., Liu, J., Han, J., and Xu, H. (2010). Primary exploration of mobile learning mode under a cloud computing environment. In *Proceedings of the International Conference on E-Health Networking, Digital Ecosystems and Technologies*, volume 2. 30

Chiang, H., Chen, Y., and Lee, T. (2013). Design and realization of a knowledge-based framework for personalized home healthcare systems. In *Automatic Control Conference (CACS), 2013 CACS International*, pages 180–185. 3

Chipcon (Accessed: October 2013). Available online at `http://focus.ti.com/docs/prod/folders/print/cc2420.html`. 46

Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., LaMarca, A., LeGrand, L., Rahimi, A., Rea, A., Klasnja, P., Koscher, K., Landay, J., Lester, J., and Wyatt, D. (2008). The mobile sensing platform: an embedded activity recognition system. In *IEEE Pervasive Computing*, pages 32–41. 59, 70

Chu, Y. (2013). *Intelligent prompting systems for people with cognitive disabilities*. PhD thesis, University of Rochester. 18

Codagnone, C. (2009). *Reconstructing the Whole: Present and Future of Personal Health Systems*. Deliverable D.6.2 PHS2020. 1

CodeBlue (Accessed: October 2013). Codeblue: Wireless sensors for medical care. Harvard Sensor Networks Lab. Available online at `http://fiji.eecs.harvard.edu/CodeBlue`. 45, 47

Cook, D. (2012). Learning setting-generalized activity models for smart spaces. In *Proceedings of the IEEE Intelligent Systems*, pages 32–38. 102, 160

Cornelius, C., Kapadia, A., Kotz, D., Peebles, D., Shin, M., and Triandopoulos, N. (2008). Anonysense: Privacy-aware people-centric sensing. In *Proceedings of the 6th International Conference on Mobile Systems*, Applications and Services, pages 211–224. 63

Corporation, I. (Accessed: October 2013). Intel PHS6000 `http://www.intel.com/corporate/healthcare/emea/eng/healthguide/index.htm`. 15, 21

Couto, R., A.D., R., and Endler, M. (2012). *Context Management for Distributed and Dynamic Context-Aware Computing*. Springer London. 3

Cuervo, E., Balasubramanian, A., Cho, D., Wolman, A., Saroiu, S., Ranveer, C., and Bahl, P. (2010). Maui: Making smartphones last longer with code offload. In *Proceedings of the 8th ACM MobiSys*. 62

Dey, A. and Abowd, G. (1999). Towards a better understanding of context and context-awareness. Technical report, GaTech. 36

Dishongh, T. and McGrath, M. (2010). *Wireless Sensor Networks for Healthcare Applications*. Artech House. 14, 18

Docobo (Accessed: October 2013). Doc@Home `http://www.docobo.co.uk`. 21

Duan, Y. and Cruz, C. (2011). Formalizing semantic of natural language through conceptualization from existence. *International Journal of Innovation, Management and Technology*, 2(1):37–42. 121

Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., and Burgelman, J. (2010). Scenarios for ambient intelligence in 2010. Technical report, Technical report `http://cordis.europa.eu/ist/istagreports.htm`. 3, 22

Ejigu, D., Scuturici, M., and Brunie, L. (2007). Semantic approach to context management and reasoning in ubiquitous context-aware systems. In *Digital Information Management, 2007. ICDIM'07. 2nd International Conference on*, volume 1, pages 500–505. IEEE. 27

Equivital (Accessed: October 2013). Mobile human monitoring. Available online at `http://www.equivital.co.uk`. 45, 47

Fan, C., Chang, P., Lin, J., and Hsieh, J. (2011). A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification. *Applied Soft Computing*, 11(1):632 – 644. 105

Ferzli, R. and Khalife, I. (2011). Mobile cloud computing educational tool for image/video processing algorithms. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop*. 30

Finapres (Accessed: October 2013). The ambulatory finapres technology solution. Available online at `http://www.finapres.com/Products/Portapres`. 162

Flowers, D. and Yang, Y. (2010). *Microchip MiWi Wireless Networking Protocol Stack*. Microchip application note AN1066. 52

Fortino, G., Li, X., Lin, X., Mayora, O., Natalizio, E., and Yuce, M. (2014). Wireless technology for pervasive healthcare. *Mobile Networks and Applications*, 19(3):273–275. 1

Frew, J. and Bose, R. (2002). Lineage issues for scientific data and information. In *Data provenance/derivation workshop*. 134

Gannot, I., Litvak, D., and Zigel, Y. (2013). System for automatic fall detection for elderly people. 18

Ganti, R., Pham, N., Tsai, Y., and Abdelzaher, T. (2008). Poolview: Stream privacy for grassroots participatory sensing. In *Proceedings of the 6th ACM Conference on Embedded network Sensor Systems*, pages 281–294. 63

Gao, H. and Zhai, Y. (2010). System design of cloud computing based on mobile learning. In *Proceedings of the 3rd International Symposium on Knowledge Acquisition and Modeling*. 30

Hazewinkel, M. (2001). Disjunctive normal form. *Encyclopedia of Mathematics, Springer*. 119

Hoang, D. and Chen, L. (2011). Mobile cloud for assistive healthcare. In *Proceedings of the 2010 IEEE Asia-Pacific Services Computing Conference*. 29

Hu, X., Wang, B., and Ji, H. (2013). A wireless sensor network-based structural health monitoring system for highway bridges. *Computer-Aided Civil and Infrastructure Engineering*, 28(3):193–209. 16

IEEE-BAN (Accessed: October 2013). IEEE 802.15 is the Working Group for WPAN and Task Group 6 is the group responsible for BANs. `http://www.ieee802.org/15/pub/TG6.html`. ix, 44, 45, 46

InMedica (2011). The World Market for Telehealth A Quantitative Market Assessment. 20

Intille, S. (2013). Closing the evaluation gap in ubihealth research. *IEEE Pervasive Computing*, 12(2):76–79. 15

Jensen, J., Peter, B., and Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*. 5

Kailanto, H., Hyvrinen, E., Hyttinen, J., and Institute, R. (2008). Mobile ecg measurement and analysis system using mobile phone as the base station. In *Second International Conference on Pervasive Computing Technologies for Healthcare*. 18

Khan, A., Lee, Y., Lee, S., and Kim, T. (2010). Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *5th International Conference on Future Information Technology*, pages 1–6. 19, 59

Kmiecik, L. (2013). Cloud centered, smartphone based long-term human activity recognition solution. Technical report, Department of Computing, Imperial College London. 93

Koiranen, T., Virkki-Hatakka, T., Kraslawski, A., and Nystrm, L. (1998). Hybrid, fuzzy and neural adaptation in case-based reasoning system for process equipment selection. *Computers & Chemical Engineering*, 22(0):997–1000. 127

Kopec, D., Kabir, M., Reinharth, D., Rothschild, O., and Castiglione, J. (2003). Human errors in medical practice: Systematic classication and reduction with automated information systems. *Journal of Medical Systems*, 27(4):297–313. 29

Kosko, B. (1992). *Neural Networks and Fuzzy Systems-A Dynamic Systems Approach to Machine Intelligence*. Prentice Hall. 110

Krishnan, N., Colbry, D., Juillard, C., and Panchanathan, S. (2008). Real time human activity recognition using tri-axial accelerometers. In *Sensors, Signal and Information Precessing Workshop*. 19

Kuchar, J. and Yang, L. (2000). A review of conflict detection and resolution modeling methods. *Intelligent Transportation Systems, IEEE Transactions on*, 1(4):179–189. 118

Kushwaha, N., Kim, M., Kim, D., and Cho, W. (2004). An intelligent agent for ubiquitous computing environments: smart home ut-agent. In *Proceedings of the 2nd IEEE workshop on software technologies for future embedded and ubiquitous systems*. 26

Kwapisz, J., Weiss, G., and Moore, S. (2010). Activity recognition using cell phone accelerometers. In *Proceedings of the 4th International Workshop on Knowledge Discovery from Sensor Data*, pages 10–18. 19, 20, 59, 69

Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. (2010). A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150. 25, 57

Leake, D. and Whitehead, M. (2007). Case provenance: The value of remembering case sources. In *Proceedings of the Seventh International Conference on Case-Based Reasoning*, pages 194–208. 134

Lee, S., Jung, P., and Song, S. (2013). Hybrid indoor location tracking for pedestrian using a smartphone. In *Robot Intelligence Technology and Applications 2012*, pages 431–440. Springer. 18

Lewis, C. (2009). Emerging trends in medical device technology: Home is where the heart monitor is. 15

Liang, X., Barua, M., Chen, L., Lu, R., Shen, X., Li, X., and Luo, H. (2012). Enabling pervasive healthcare through continuous remote health monitoring. *Wireless Communications, IEEE*, 19(6):10–18. 5

Lu, C. and Fu, L. (2009). Robust location-aware activity recognition using wireless sensor network in an attentive home. In *IEEE Transactions on Automation Science and Engineering*, pages 598–609. 18

Lum, W. (2002). A context-aware decision engine for content adaptation. In *PERVASIVE Computing*. 23

Lyons, G., Culhane, K., Hilton, D., Grace, P., and Lyons, D. (2005). A description of an accelerometer-based mobility monitoring technique. *Medical Engineering and Physics*, 27:497–504. 73, 77

Ma, T. (2005). Context-aware implementation based on cbr for smart home. In *Proceedings of the IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, pages 112–115. 105

Makris, P., Skoutas, D., and Skianis, C. (2013). A survey on context-aware mobile and wireless networking: On networking and computing environments' integration. *Communications Surveys & Tutorials, IEEE*, 15(1):362–386. 22

Marco, A., Casas, R., Falco, J., Gracia, H., Artigas, J., and Roy, A. (2008). Location based services for elderly and disabled people. In *Computer Communications*, volume 31, pages 1055–1066. 18

Maurer, U., Samilagic, A., Siewiorek, D., and Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *Proceedings of the International Workshop on Wearable and Implantable Sensor Networks*. 19, 59

MayoClinic (Accessed: October 2014). MayoClinic.com `http://www.mayoclinic.com/health/electrocardiogram/MY00086`. 15

McCullagh, P. and Augusto, J. (2011). The potential to facilitate health and wellness. *Internet of Things*, pages 59–68. 14

McKinstry, B. (2009). Lothian telehealth project: Long term conditions collaborative. Technical report, The Scottish Government, Edinburgh. Avaiable online at http://www.scotland.gov.uk/Resource/Doc/263175/0078713.pdf. 15

Microchip Technology Inc. (2012). *ZENA Wireless Adapter User's Guide*. 56

Mihailidis, A. and Bardram, J. (2010). *Pervasive computing in healthcare*. CRC Press. 5

Mileo, A. (2010). Support for context-aware monitoring in home healthcare. *Journal of Ambient Intelligence and Smart Environment*, pages 49–66. 104, 109

Nguyen, T., Woo, Y., and Choi, D. (2009). Ccbr: Chaining case based reasoning in context-aware smart home. In *Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on*, pages 453–458. 111

Nkosi, M. and Mekuria, F. (2011). Cloud computing for enhanced mobile health applications. In *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*. 29

Nusca, A. (2009). Smartphone vs. feature phone arms race heats up; which did you buy? ZDNet. 57

O'Flynn, B., Bellis, S., Mahmood, K., Morris, M., Duffy, G., Delaney, K., and OMathuna, C. (2004). A 3-d miniaturised programmable transceiver. In *XXVIII International Conference of IMAPS*, pages 343–347. 46

Oliveira, L. and Rodrigues, J. (2011). Wireless sensor networks: a survey on environmental monitoring. *Journal of communications*, 6(2):143–151. 16

OPNET (Accessed: October 2014). OPNET Modeler http://www.opnet.com. 15

Padovitz, A., Loke, S., and Zaslavsky, A. (2008). The ecora framework: A hybrid architecture for context-oriented pervasive computing. *Pervasive and mobile computing*, 4(2):182–215. 26

Pang, Z., Chen, Q., and Zheng, L. (2009). A pervasive and preventive healthcare solution for medication noncompliance and daily monitoring. In *Second International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–9. 18

Pecht, M. and Jaai, R. (2010). A prognostics and health management roadmap for information and electronics-rich systems. *Microelectronics Reliability*, 50(3):317–323. 22

Peiris, V. (2013). Highly integrated wireless sensing for body area network applications. *SPIE Newsroom. doi*, 10(2.1201312). 16

Pietschmann, S., Mitschick, A., Winkler, R., and Meißner, K. (2008). Croco: Ontology-based, cross-application context management. In *Semantic Media Adaptation and Personalization, 2008. SMAP'08. Third International Workshop on*, pages 88–93. IEEE. 28

Pung, H., Gu, T., Xue, W., Palmes, P., Zhu, J., Tang, C., and Chung, N. (2009). Context-aware middleware for pervasive elderly homecare. *IEEE Journal on Selected Areas in Communications*, 27(4):510–523. 3

Ram, A. and Wiratunga, N. (2011). Case-based reasoning research and development. In *Proceedings of the 19th International Conference on Case-Based Reasoning*, pages 444–445. 127

Ranganathan, A. and Campbell, R. H. (2003). An infrastructure for context-awareness based on first order logic. In *Personal Ubiquitous Computing*. 23

Rantz, M., Skubic, M., Koopman, R., Phillips, L., Alexander, G., Miller, S., and Guevara, R. (2011). Using sensor networks to detect urinary tract infections in older adults. In *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on*, pages 142–149. IEEE. 22

Rashidi, P. and Mihailidis, A. (2013). A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590. 23

Ravi, N., Dandekar, N., Mysore, P., and Littman, M. (2005). Activity recognition from accelerometer data. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20. 19

Riboni, D. and Bettini, C. (2009). Context-aware activity recognition through a combination of ontological and statistical reasoning. In *Ubiquitous Intelligence and Computing*, pages 39–53. Springer. 25, 28

Riesbeck, C. and Schank, R. (2013). *Inside case-based reasoning*. Psychology Press. 104, 126

Rodrigues, R., Huber, M., Lamura, G., et al. (2012). Facts and figures on healthy ageing and long-term care. *Itävalta: European Centre for Social and Wellfare policy and Research: Vienna*. 12

Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., and Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE pervasive computing*, 1(4):74–83. 27

Sager, I. (2012). Before iphone and android came simon, the first smartphone. Bloomberg Businessweek (Bloomberg). 57

Sain, M., Lee, H., and Wan, Y. (2010). Designing context awareness middleware architecture for personal healthcare information system. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 2, pages 1650–1654. 4

Samsung Inc. (Accessed: October 2013). An android smartphone: Samsung galaxy siii. Description Available online at `http://en.wikipedia.org/wiki/Samsung_Galaxy_S_III`. 69

Schlender, B. (2003). Intel andy grove: The next battles in tech. *Fortune*, pages 80–81. 13

Schmidt, R. and Gierl, L. (2001). Cased-based reasoning for medical knowledge-based systems. *International Journal of Medical Informatics*, pages 355–367. 105

Schwickert, L., Becker, C., Lindemann, U., Maréchal, C., Bourke, A., Chiari, L., Helbostad, J., Zijlstra, W., Aminian, K., and Todd, C. (2013). Fall detection with body-worn sensors. *Zeitschrift für Gerontologie und Geriatrie*, 46(8):706–719. 18

Seltzer, L. (2010). *Rich Internet Applications: The Next Frontier of Corporate Development*. eWeek. 146

Shimmer (Accessed: October 2013). Wireless sensor platform that can record and transmit physiological and kinematic data in real-time. Shimmer Research. Available online at `http://www.shimmer-research.com`. 16, 45, 47

Skelton, D. and McLaughlin, A. (1996). *Train functional ability in old age*, volume 82. Physiotherapy. 64

Smarr, L. (2012). Quantifying your body: A how-to guide from a systems biology perspective. *Biotechnology Journal*, 7(8):980–991. 5

Stankovski, V. and Tmkoczy, J. (2006). Application of decision trees to smart homes. In *Designing smart homes. The role of artificial intelligence*, pages 132–145, Berlin. Springer. 25

Stefan, D., Das, B., Krishnan, N., Thomas, B., and Cook, D. (2012). Simple and complex activity recognition through smart phones. In *Proceedings of the 8th International Conference on Intelligent Environments*, pages 214–221. 20, 59, 69

Sudan, R., Salter, M., Lynch, T., and Jacobs, D. (2011). Bariatric surgery using a network and teleconferencing to serve remote patients in the veterans administration health care system: feasibility and results. *The American Journal of Surgery*, 202(1):71–76. 21

Swets, J. (1996). *Signal detection theory and ROC analysis in psychology and diagnostics*. Lawrence Erlbaum Associates, Mahwah, NJ. 139

Szomszor, M. and Moreau, L. (2003). Recording and reasoning over data provenance in web and grid services. In *In Int. Conf. on Ontologies, Databases and Applications of Semantics, volume 2888 of LNCS*, pages 603–620. 133

Tang, W., Hu, C., and Hsu, C. (2010). A mobile phone based homecare management system on the cloud. In *Proceedings of the 3rd International Conference on Biomedical and Informatics*, volume 6. 29

Tapia, E., Intille, S., Haskell, W., Larson, K., and Friedman, R. (2007). Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers & Physics*. 19

Tapia, E., Intille, S., Haskell, W., Larson, K., and Friedman, R. (2008). Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers & Physics*, pages 37–40. 59

Tapia, E., Intille, S., and Larson, K. (2004). Activity recognition in the home setting using simple and ubiquitous sensors. In *Proceedings of PERVASIVE*. 19

Tharion, W., Potter, A., Duhamel, C., Karis, A., Buller, M., and Hoyt, R. (2013). Real-time physiological monitoring while encapsulated in personal protective equipment. *Journal of Sport and Human Performance*, 1(4). 18

Thomas, L., Little, L., Briggs, P., McInnes, L., Jones, E., and Nicholson, J. (2013). Location tracking: views from the older adult population. *Age and ageing*, 42(6):758–763. 18

Triantafyllidis, A., Koutkias, V., Chouvarda, I., and Maglaveras, N. (2013). A pervasive health system integrating patient monitoring, status logging, and social sharing. *Biomedical and Health Informatics, IEEE Journal of*, 17(1):30–37. 18

Vainio, A., Valtonen, M., and Vanhala, J. (2008). Proactive fuzzy control and adaptation methods for smart homess. In *IEEE Intell Syst*, volume 23, pages 42–49. 26

Varshney, U. (2003). Pervasive healthcare. *Computer*, 36:138–140. 1

Varshney, U. (2009). *Pervasive Healthcare Computing*. Springer US. 4

Wakefield, B., Holman, J., Ray, A., Scherubel, M., Adams, M., Hills, S., and Rosenthal, G. (2012). Outcomes of a home telehealth intervention for patients with diabetes and hypertension. *Telemed J E Health*, 18(8):575–579. 21

Wallace, M. and Stamou, G. (2002). Towards a context aware mining of user interests for consumption of multimedia documents. In *Proceedings of the IEEE Multimedia and Expo*. 23

Wang, C., Chiang, C., Lin, P., Chou, Y., Kuo, I., Huang, C., and Chan, C. (2008). Development of a fall detecting system for the elderly residents. In *Second International Conference on Bioinformatics and Biomedical Engineering*, pages 1359–1362. 18

Wang, H., Zheng, H., J.C., A., Martin, S., Mulvenna, M., Carswell, W., Wallace, J., Jeffers, P., Taylor, B., and McSorley, K. (2010). Monitoring and analysis of sleep pattern for people with early dementia. In *1st Workshop on Knowledge Engineering, Discovery and Dissemination in Health (KEDDH10), co-located with IEEE Bioinformatics and Biomedicine Conference*, Hong Kong, China. 21

White, D., Wagenaar, R., and Ellis, T. (2001). Monitoring activity in individuals with parkinsons disease: A validity study. *Journal of Neurological Physical Therapy*, pages 12–21. 64

Whiteman, H. (2012). Telehealth can safely replace follow-up clinic visits, study shows. Medical News Today. 148

Witten, I., Hall, M., and Frank, E. (2011). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 3rd edition edition. 80, 102

Xue, W., Pung, H., Ng, W., and Gu, T. (2008). Data management for context-aware computing. In *Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on*, volume 1, pages 492–498. IEEE. 27

Yang, Y. (2010). *Microchip MiWi P2P Wireless Protocol*. Microchip application note AN1204. ix, 52, 53

Yin, C., David, B., and Chalon, R. (2009). Use your mobile computing devices to learn - contextual mobile learning system design and case studies. In *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology*. 30

Yoon, S., Ye, W., Heidemann, J., Littlefield, B., and Shahabi, C. (2011). Swats: Wireless sensor networks for steamflood and waterflood pipeline monitoring. *Network, IEEE*, 25(1):50–56. 16

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 3(8):338–353. 113, 115, 124

Zeifman, M. and Roth, K. (2011). Nonintrusive appliance load monitoring: Review and outlook. *Consumer Electronics, IEEE Transactions on*, 57(1):76–84. 23

Zephyr Inc. (Accessed: October 2013). Wireless professional heart rate monitor & physiological monitor with bluetooth. Zephyr. Available online at `http://www.zephyranywhere.com/products/bioharness-3/`. xii, 6, 45, 47, 49, 69

Zhang, J., Song, G., Wang, H., and Meng, T. (2011). Design of a wireless sensor network based monitoring system for home automation. In *Future Computer Sciences and Application (ICFCSA), 2011 International Conference on*, pages 57–60. IEEE. 22

Zhang, S., McCullagh, P., Nugent, C., and Zheng, H. (2010). Activity monitoring using a smart phone's accelerometer with hierarchical classification. In *Proceedings of the 6th International Conference on Intelligent Environments*, pages 158–163. 102, 160

Zhao, W., Sun, Y., and Dai, L. (2010). Improving computer basis teaching through mobile communication and cloud computing technology. In *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering*, volume 1. 30

Ziekow, H., Goebel, C., Struker, J., and Jacobsen, H. (2013). The potential of smart home sensors in forecasting household electricity demand. In *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, pages 229–234. IEEE. 23