Department of Electrical and Computer Engineering
The University of Auckland
New Zealand

Context-aware Activity Recognition for Elderly Healthcare using Wearable and Sensors Embedded in Environment

Mohd Halim Mohd Noor

A thesis submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Systems Engineering
The University of Auckland, April 2017

Abstract

Smart home is a promising solution for the aging population who requires assistance but prefers living independently at home. Smart home is a manifestation of pervasive computing which incorporates multimodal sensors, actuators, devices, information, and communication technologies to gather different information about the environment and its users. One of the main characteristics of pervasive computing is context awareness. In this thesis, we present the development of a context-aware activity recognition system using wearable and sensors embedded in environment to continuously monitoring activities of daily living of elderly people. The proposed system fuses contextual information about user's physical activity, location and interactions with objects in the environment using ontology, and accurately recognize the activities. To develop a robust physical activity recognition, we propose a novel adaptive sliding window segmentation to select a more effective window segmentation of acceleration signals. In addition, we propose an activity transition diagram to be integrated into the activity classification algorithm to validate the activity transition after window classification. To overcome the limitation of ontology in dealing with uncertainty due to missing sensor data, we propose a novel reasoning algorithm that integrates ontological reasoning mechanism with Dempster-Shafer theory of evidence. The algorithm provides support for handling uncertainty by quantifying uncertainty while aggregating contextual information and produce a degree of belief to facilitate a more robust decision making in activity recognition. To further enhance the recognition accuracy of the system, we present the integration of user context into the activity ontology to handle uncertainty due to missing sensor data. In addition, the approach allows additional and more precise inference of activities and recognizes activities that do not involve interaction with objects in the environment.

Acknowledgements

I would like to express my deepest gratitude towards my principal supervisor Professor Zoran Salcic and my co-supervisor Dr. Kevin I-Kai Wang for their positive encouragement, constructive ideas and invaluable advice. Under their supervision and guidance, I was able to learn and grow tremendously over the progressive years. And I am most grateful for their motivation, professional and immense knowledge, and endless comments on and editing of my English writing. Thank you!

I would like to acknowledge the hardware/firmware support provided by Lead Electronic Engineer Akshat Bisht. This enabled the design and development of the Wireless Sensor Network in the Intelligent Environment laboratory, which played a major part in the evaluation of the proposed system.

I would also like to thank my fellow colleagues and friends of the Embedded Systems Research Group, especially Muhammad Faraz Shaikh and Mohd Nazrin Muhammad for their candid opinions and inputs. Finally, and most importantly, I thank my wife Zuraidah Sukadi, my children and my parents for their continuous love and support. Without their understanding this would have been a difficult journey to embark.

Contents

Publications

M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, “Enhancing ontological reasoning with uncertainty handling for activity recognition,” *Knowledge-Based Systems*, vol. 114, pp. 47–60, Dec. 2016.

M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, “Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer,” *Pervasive and Mobile Computing*, Sep. 2016.

M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, “Dynamic sliding window method for physical activity recognition using a single tri-axial accelerometer,” in 2015 *IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 102–107.

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
The ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Chapter 3 and 4

Mohd Halim Mohd Noor, Zoran Salcic, Kevin I-Kai Wang, Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer, Pervasive and Mobile Computing, Available online 23 September 2016, ISSN 1574-1192, http://dx.doi.org/10.1016/j.pmcj.2016.09.009.

Nature of contribution by PhD candidate	Main author/Majority of the work
Extent of contribution by PhD candidate (%)	90

CO-AUTHORS

Name	Nature of Contribution
Mohd Halim Mohd Noor	Propose the original idea, do the theoritical analysis, simulation and experimental works, and draft the original paper.
Zoran Salcic	Main supervisor, advise research direction, help with theoritical analysis, paper structure and presentation, and proofreading.
Kevin I-Kai Wang	Co-supervisor, advise research direction, help with theoritical analysis, paper structure and presentation, and proofreading.

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Mohd Halim Mohd Noor		12/12/2016
Zoran Salcic	Z Salcic	12/12/2016
Kevin I-Kai Wang		12/12/2016

Last updated: 19 October 2015

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
The ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Chapter 5

Mohd Halim Mohd Noor, Zoran Salcic, Kevin I-Kai Wang, Enhancing ontological reasoning with uncertainty handling for activity recognition, Knowledge-Based Systems, Volume 114, 15 December 2016, Pages 47-60, ISSN 0950-7051, http://dx.doi.org/10.1016/j.knosys.2016.09.028.

Nature of contribution by PhD candidate	Main author/Majority of the work
Extent of contribution by PhD candidate (%)	90

CO-AUTHORS

Name	Nature of Contribution
Mohd Halim Mohd Noor	Propose the original idea, do the theoritical analysis, simulation and experimental works, and draft the original paper.
Zoran Salcic	Main supervisor, advise research direction, help with theoritical analysis, paper structure and presentation, and proofreading.
Kevin I-Kai Wang	Co-supervisor, advise research direction, help with theoritical analysis, paper structure and presentation, and proofreading.

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Mohd Halim Mohd Noor		12/12/2016
Zoran Salcic	Z-Salcic	12/12/2016
Kevin I-Kai Wang		12/12/2016

THE UNIVERSITY OF AUCKLAND
SCHOOL OF GRADUATE STUDIES

Co-Authorship Form

Graduate Centre
The ClockTower – East Wing
22 Princes Street, Auckland
Phone: +64 9 373 7599 ext 81321
Fax: +64 9 373 7610
Email: postgraduate@auckland.ac.nz
www.postgrad.auckland.ac.nz

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Mohd Halim Mohd Noor, Zoran Salcic and Kevin I-Kai Wang, Dynamic sliding window method for physical activity recognition using a single tri-axial accelerometer, 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), Auckland, 2015, pp. 102-107, doi: 10.1109/ICIEA.2015.7334092

Nature of contribution by PhD candidate	Main Author/Majority of the work
Extent of contribution by PhD candidate (%)	90

CO-AUTHORS

Name	Nature of Contribution
Mohd Halim Mohd Noor	Propose the original idea, do the theoritical analysis, simulation and experimental works, and draft the original paper.
Zoran Salcic	Main supervisor, advise research direction, help with theoritical analysis, paper structure and presentation, and proofreading.
Kevin I-Kai Wang	Co-supervisor, advise research direction, help with theoritical analysis, paper structure and presentation, and proofreading.

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Mohd Halim Mohd Noor		12/12/2016
Zoran Salcic		12/12/2016
Kevin I-Kai Wang		12/12/2016

1

Introduction

1.1 Aging Population

The world's population is aging. Longer lifespans and lower birth rates have led to the increase in the number and proportion of elderly population. According to World Health Organization, it is estimated that there will be 2 billion people of age 60 and above by 2050 [1]. In New Zealand, the composition of population pyramid is changing, with the older age group widening. The projection of elderly population aged over 65 by the year 2051 is 1.14 million, which is 25% of the population [2]. In Europe, it is expected that the elderly population of European Union (EU27) aged 65 years and over to rise to 30% in 2060 [3]. In the United States, the elderly population aged 65 years and over is projected to increase to 20.9% in 2050 [4]. The world's fertility rate and life expectancy is illustrated in Figure 1.1 [1]. Clearly, in general, the trend of growth rate is decreasing while the trend of life expectancy is increasing. It is projected that by the year 2050, the growth rate will drop to 2.25 children per woman and the life expectancy will increase to 77.06 years. By the year 2050, the proportion of elderly population is projected to be above 20% for all regions except Africa as shown in Figure 1.2 [1].

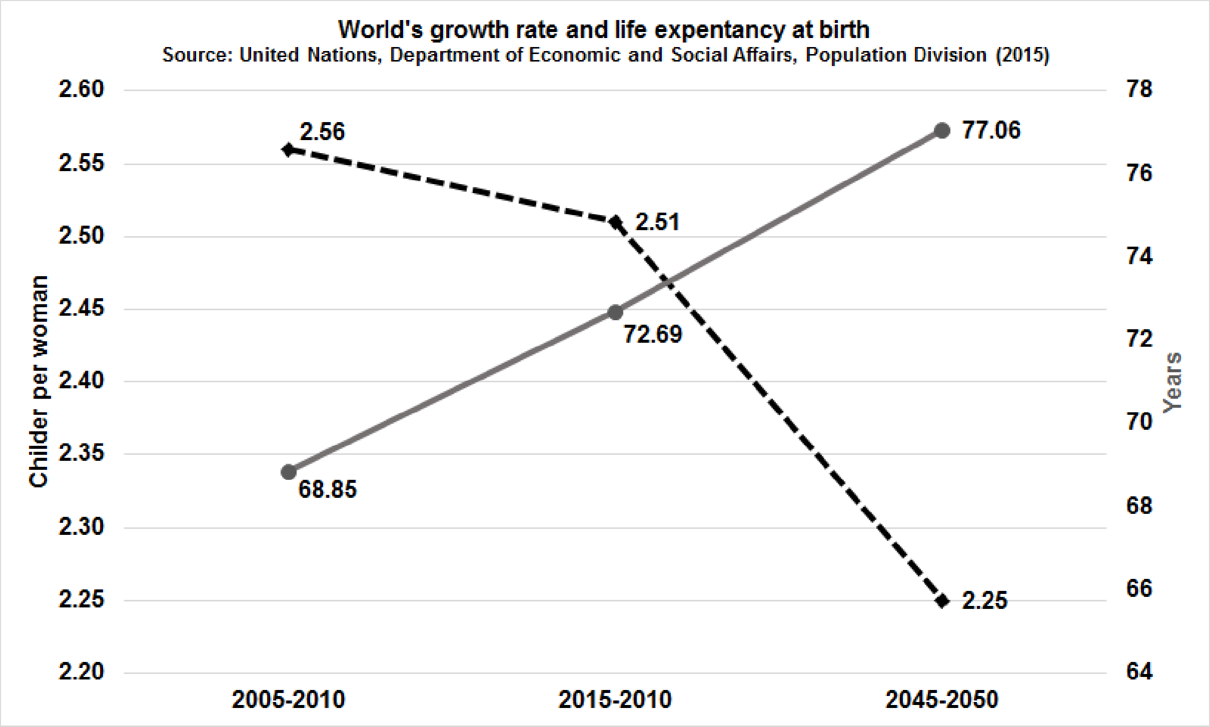


Figure 1.1: The trend of fertility rate and longer life expectancy from 2005 to 2050.

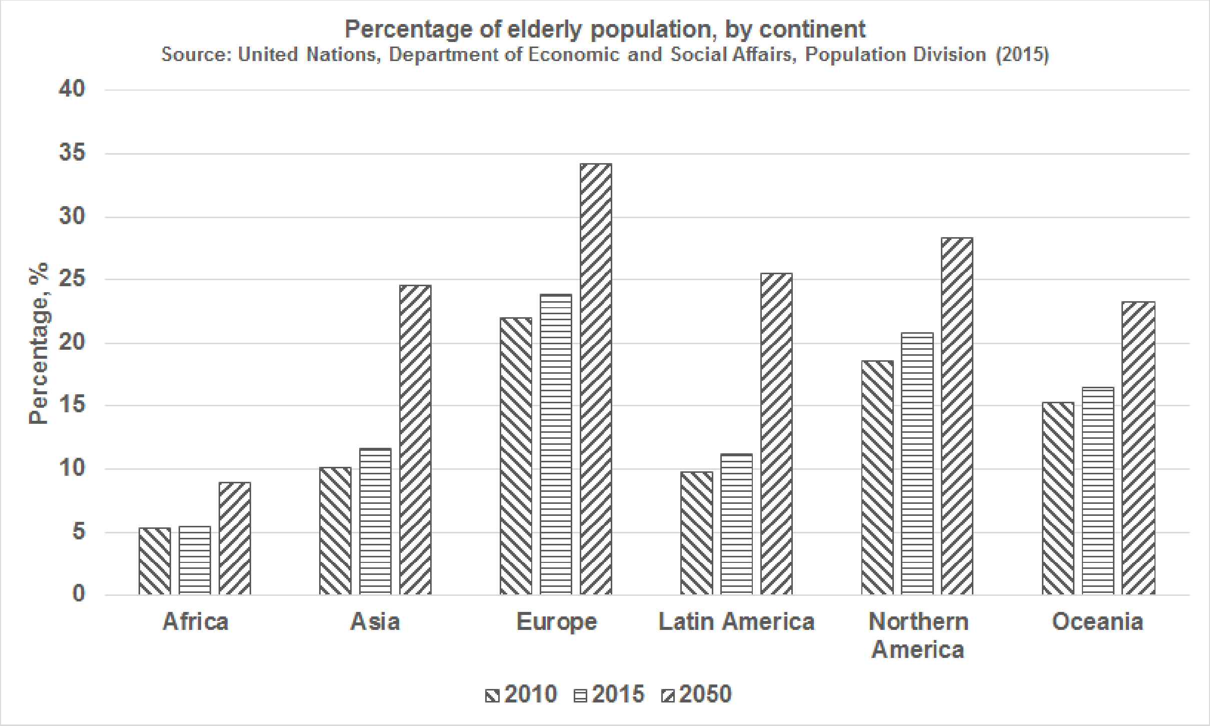


Figure 1.2: The growing elderly population (aged 60 and over).

Aging and dependent population is becoming a major social and economic issue. Previous studies have reported that chronic diseases such as Alzheimer's disease, osteoporosis and heart disease are more prevalent among elderly people [5], [6]. With the increase of elderly population, rise in healthcare cost with insufficient and ineffective care are becoming an issue in the future. According to the Ministry of Health New Zealand, people aged over 65 are the high users of general practice services with an average visit of six or seven times a year. Hospital discharges have been increasing for elder people, from 5.57% in 2006 to 6.30% in 2013 [7]. It is also projected that the cost of elderly healthcare will be experiencing the greatest growth. In United States, the healthcare costs for elderly aged 65 and over is three to five times greater than the people aged less than 65, and the cost of public healthcare is projected to increase by 21% by 2020 [8].

Elders who are dependent and vulnerable due to cognitive and physical impairment require assistance in their activities of daily living (ADL). Also, caregivers become overburdened with continuous monitoring responsibilities. As a result, it affects the timely assistance the elders require and deserve. On the other hand, most elderly people prefer to remain independent rather than to be moved in to a home care [9], [10]. In fact, a study has shown that people fear more losing independence in old age than death [11]. Therefore, it is of the utmost importance to develop technologies and services which can enable elderly people to live independently and happily, and at the same time enhancing their quality of life.

1.2 Motivation and Objectives

A retirement village contains several homes of elderly people who are dependent and vulnerable in different aspects due to cognitive and physical impairment. The residents are living alone with limited assistance. A system is context aware if it exploits information about the context of its user and adapts itself to provide relevant services in order to improve the user experience [12]. Therefore, a context-aware system is a potential way to monitor the residents' ADL in the village and alert healthcare provider in the case of emergency. Furthermore, it is desirable for the system to act upon any ongoing anomalous situation by providing reminder to the residents such as to switch off stove after cooking. Ultimately, the system will allow the elders to live independently and at the same time enhancing their quality of life.

Let us look at an example scenario that can be used as the motivation for our work. An elderly person is living alone in a house which has two bedrooms, a living room with a sofa, a toilet, a bathroom and a kitchen with a dining table. This person has high blood pressure and diabetes. Normally the person wakes up around 7 a.m. in the morning and takes medication for high blood pressure after toileting. Breakfast happens about 8 a.m. and is followed by a recommended 30 minutes exercise around 10 a.m. Lunch is between 1 and 2 p.m. after some routine housework such as cleaning and working with computer. Dinner is usually around about 7 p.m. after taking insulin. Afterwards, the person watches television or reads before going to sleep at 11 p.m. On average, (s)he visits the toilet 15 times a day. This person's movement and vital signs such as temperature, heart rate, blood pressure and blood glucose are monitored continuously using wearable sensors. Sensors are distributed all over the living environment and connected to the smart home system which implements a context-aware system for processing. The sensors are passive infrared (PIR) sensor or motion detector to provide user's location in the house, and door sensor, temperature sensor, force sensor, water detector and fire sensor to provide the states of objects as shown in Figure 1.3. The information from the sensors will be used to infer the activities of the elderly.

Based on the scenario, a context-aware system must be able to provide monitoring services and healthcare support to the elderly. In realizing this goal, a context-aware system must be aware of the normal and abnormal situations by learning the activity patterns of the elderly using context information acquired from wearable sensors and sensors deployed in the environment. However, activity recognition is facing a number of challenges. Activity patterns of an elderly are different from one to another due to different lifestyles, habits and health conditions and as such have their own way of performing activities. Specifically, activities can be carried out with a high degree of freedom in relation to the way and the sequential order they are performed. There is no strict constraint on the order and duration of individual physical actions. Furthermore, multiple activities can be carried out at the same time such as making coffee while heating up food.

Wearable sensors such as accelerometers are proved by a number of researchers to be effective in monitoring physical activities. Typically, the accelerometers are worn on various parts of the body such as wrist, waist and ankle to capture hand motions and ambulatory activities. However, multiple accelerometers are not feasible for long-term activity monitoring because they impede elderly's daily living activities due to multiple attachments to the body. Furthermore, wearable sensors are not suitable for monitoring activities that involve complex physical motions and multiple interactions with the environment [13], [14]. Embedded sensors in a smart home provide contextual information about the environment which can be used to infer the activity being undertaken. For instance, a sensor embedded or installed on a chair can indicate if a person is sitting on it. However, the contextual information is emerging from heterogeneous sources with different contextual representation. Furthermore, uncertainties are always present due to sensor errors, communication failures and variability in human activities.

The aim of this thesis is to develop a context-aware activity recognition system for elderly healthcare using wearable and sensors embedded in the environment. The main goal has been carried out through the following three research objectives.

1. To develop a robust wearable sensor-based activity recognition system using a single accelerometer that provides context information on the physical activity of the user.
2. To develop an algorithm that handles uncertainty due to missing sensor data in ontology-based activity recognition.
3. To develop a sensor fusion technique that combines context information from wearable and ambient sensors.

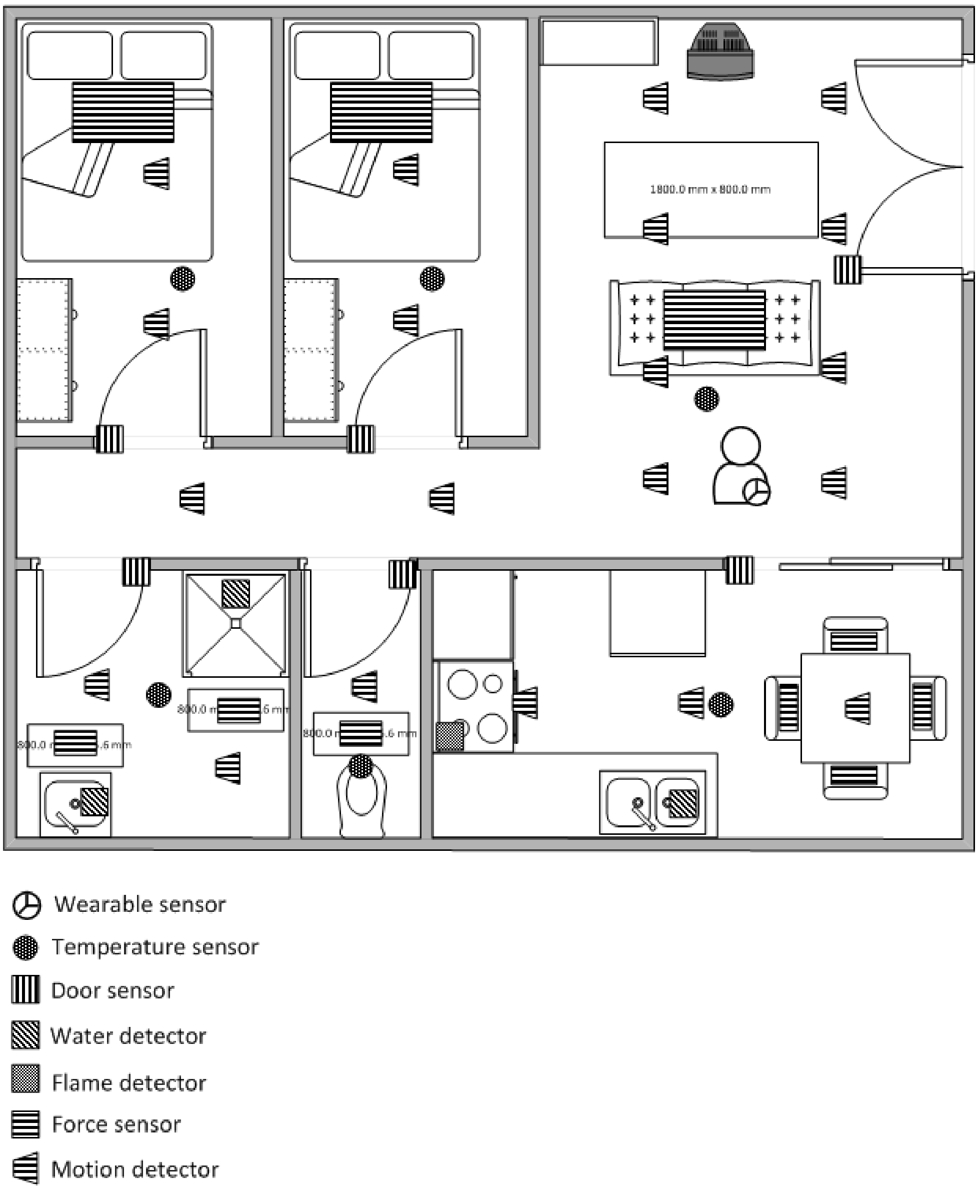


Figure 1.3: Sensor distribution in smart home.

1.3 Contributions

The main contributions of this thesis are presented as follows:

- We propose a novel adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. The approach can adaptively change the window size to detect not only static and dynamic, but also transitional activity signals of varying periods as the segmentation window is being evaluated. The window size is dynamically adjusted based on the continuous evaluation of the activity signals. As a result, a more effective window size can be selected for signal segmentation to achieve more accurate classification (Chapter 3).
- We propose a physical activity transition model in the form of state transition diagram to model the temporal dependence of physical activities. The approach includes the integration of the transition diagram into the classification algorithm to validate the activity transition. As a result, the recognition accuracy is improved (Chapter 4).
- We propose a novel Description Logic-based ontological reasoning algorithm for activity recognition in smart environments. The algorithm resolves uncertainty due to missing sensor data while combining context information, and hence supporting the decision making process in order to improve the reasoning performance (Chapter 5).
- We investigate the fusion of wearable and ambient sensors for recognizing activities in smart environments. The approach exploits the advantages of both types of sensing to resolve uncertainty due to sensor observation errors. The approach includes the proposal of ontology-based sensor fusion methodology (Chapter 6).

1.4 Thesis Outline

The following chapters are included in this thesis:

- Chapter 2 reviews the areas of study relevant to the development of context-aware activity recognition system in smart home environment that include sensing, activity modeling and recognition approaches.

- Chapter 3 describes a novel approach to activity signal segmentation in which, the window size is adaptively adjusted according to signal information to achieve the most effective segmentation. This is significant because fixed window size will not produce good segmentation due to the diversity of activity signals, which will lead to misclassification. We describe the characterization of activity signals and the rationale behind the introduction of adaptive sliding window.
- Chapter 4 describes the proposed activity transition diagram to model the temporal dependence of physical activities. The approach exploits the temporal dependence and improves the recognition accuracy by validating the activity transition after window classification. The integration of transition model enables the system to become more robust.
- Chapter 5 describes a novel reasoning algorithm which integrates Description Logic-based ontological reasoning with Dempster-Shafer theory to handle data uncertainty due to missing sensor data. This is significant because traditional ontological reasoning can only infer an activity when all the contextual information that define the activity is asserted. If one of the contexts is missing, ontology will not be able to infer the activity and as a result, the recognition accuracy is reduced. We describe the modeling of uncertainty in ontological reasoning and the evidential operations.
- Chapter 6 describes the proposal of an activity recognition system that uses an integrated wearable sensor and dense sensing approaches. The approach can not only resolve uncertainty due to missing sensor data, but also allows additional and more precise inference of information about the activity being recognized. The approach can also infer activities which do not involve user-object interaction context.
- Chapter 7 give conclusions about the presented context-aware activity recognition system and provides some directions for the future work.

2

Background Research and System Architecture

2.1 Introduction

Elderly population is increasing in every part of the world. This brings a need for more healthcare options. This chapter discusses the roles of smart homes in supporting independent living of the elderly populations. Firstly, the overview of smart homes and the essential contexts in recognizing situations in a home environment are presented. Then, we discuss human activity recognition from the aspect of sensing, activity modeling and recognition approaches. Finally, we summarize the chapter by describing the proposed context-aware activity recognition system for elderly healthcare.

2.2 Smart Homes

Smart home is a promising solution for the aging population, assisting and providing services to the elders living alone at home [15]–[17]. Smart home technology aims to provide better quality of life and to ensure elderly to live independently through automated appliance control and assistive services. Home appliances and devices can be controlled to execute tasks remotely. Household electricity usage can be reduced by the ambient intelligence system. The intelligent monitoring system can be used to ensure the safety of inhabitants in the environment. Specifically, the objectives of a smart home is to provide comfort, security and healthcare services to the inhabitants [15].

User comfort can be achieved through automation of home appliances and remote access and control of home environment. Home appliances automation is an intelligent system that utilizes human activity and behavior information. Based on the information, home appliances are automated to make life easier and more comfortable by facilitating the inhabitants' daily activities. Furthermore, energy usage can be optimized by controlling unattended home appliances. Remote access and control provides a platform for users to control and monitor home appliances from distant and remote location and hence, make life more convenient. Smart homes are vulnerable to security threats such as hacking, phishing and firmware alteration. Security services ensure user privacy by enforcing system security enforce through user and device authentication.

Elderly people who are living alone with limited assistance can be worrying for family members. Healthcare services in smart homes provide continuous monitoring of the inhabitants' health conditions and overall well-being. The physiological state and vital signals are acquired to identify the health conditions, as well as to generate warnings and alarms if necessary. Long-term data can be analyzed to predict any potential risks in the future. In addition, daily activities are monitored to detect abnormal behavior or deviations from the routine. It has been suggested that smart home technology has positive effects on care efficiency and cost. However, despite the positive prospects, smart home technology has not been widely accepted and adopted. Furthermore, the technology readiness level of smart homes in general is still at low as reported in [18]. Thus, efforts are needed for research in smart homes especially in the domain of fall detection and health and activity monitoring. One area that needs special attention is the development of “intelligent” algorithms for sensor data analysis and interpretation [19].

2.3 Context-aware System for Elderly Healthcare

Smart home is a manifestation of pervasive computing that involves incorporating multimodal sensors, actuators, devices, information, and communication technologies [20]. Sensors gather different types of information about the environment including the inhabitants, deliver that information to automatic systems and caregivers to control the environment or to healthcare professionals to monitor their behavior and health conditions. Pervasive computing is a paradigm in which computing can occur using any device in any location to access and exchange information. One of the main characteristics of pervasive computing is context awareness [21], [22]. A system is context aware if it exploits information about the context of its user and adapts itself to provide relevant services in order to improve the user experience [12]. The term context has been defined by many researchers−Dey et al. [12] defined context as: “Any information that can be used to characterize the situation of a person, place, or object that is considered relevant to the interaction between a user and an application”. Context is not raw data obtained from a sensor. Context is generated by processing raw sensor data.

Context can be any measurable information which can affect the behavior of the system. Typically, contexts are acquired through low-cost and low-power sensor nodes wirelessly connected and working together in the environment and such networks are called wireless sensor networks (WSN). These sensor nodes are attached to specific locations and objects to provide context information about the users’ location and users’ interaction with objects. Sensor nodes are also attached to the users to create an interface to humans, which allows them to capture information related to body movements and health conditions and send this information for further processing. The interconnection of the sensors into a system is called body area sensor network (BASN).

Context information is gathered from a large variety of sensors that differ in their sensor output, sampling rate and semantic level. Some sensors provide fast and real-time raw data which has to be interpreted before being useful to the application. For examples, inertial sensors provide continuous acceleration and angular velocity data of body motions at high frequency for estimating the position and orientation of body segments [23] and recognizing human activities [24]. Whereas information such as user profiles do not need further interpretation and are updated rarely. In [25], social networks are used to retrieve personal information such as gender and age for context-aware multimedia recommender. Moreover, context information can also be derived from existing context information. For instance, distance between two objects can be determined using their coordinates provided by GPS [12]. Therefore, context-aware system needs to formally represent the contexts in order to ease the development and reduce the complexity of its application. It is also necessary for consistency checking before sensor data and contexts are being processed [26], [27]. The methodology of organizing and storing contexts is called context modelling. The ability to classify and infer situations is important in order for the system to adapt itself and react to different circumstances. A situation is typically defined by the activities that are occurring in a specific location and at a specific time or for a certain period of time [28], [29]. Therefore, activity context is an essential context in context-aware system for elderly healthcare. However, context information derived from sensors, called low-level context, is less meaningful, trivial, vulnerable to small changes, and uncertain. In order to better understand and deduce new knowledge about the environment, the context information should be interpreted to acquire activity information that can describe the situation of environment [30]. The process of identifying the state of the inhabitant in a smart home is called human activity recognition.

2.4 Human Activity Sensing

Human activity recognition is a process of discovering an activity pattern and recognizing the activity from a series of observations acquired by various types of sensors [31]. With respect to the type of sensors, sensing approaches to activity recognition can be classified into two groups. The first group is referred to as vision-based activity recognition, which is based on the use of vision sensors to record the events in the environment. The sensor data are a sequence of digitized images. The approach utilizes video processing techniques to analyze the visual observations and recognize the ongoing activities. The recognition process includes low-level features extraction, action descriptions from low-level features and semantic interpretations from primitive actions [32], [33]. However, vision-based sensors in the context of smart homes are debatable as they are perceived as recording devices that intrude resident's privacy [34], [35]. The second group of approaches is referred to as sensor-based activity recognition, which is based on the use of sensor network technologies for activity recognition. The sensor data are time series of parameter values and state changes. These approaches utilize data fusion, machine learning techniques and formal knowledge engineering methods for activity recognition [35].

The sensing approach of sensor-based activity recognition can be further classified into two categories. The first approach makes use of wearable sensors, either dedicated sensors attached to human body directly or indirectly (i.e. sensors embedded into clothes, wristwatches, eyeglasses etc.) or those available on the portable devices like mobile phones. The sensors generate signals when the user performs activities, where their characteristics describe the person's movement or physiological state. Numerous sensors have been studied to determine their effectiveness in activity recognition applications and can be categorized into biosensors and inertial sensors. Biosensors such as electrocardiogram (ECG) are found to be useful in discriminating walking-related activities [36]. In [37], electroencephalography (EEG) is used to provide information which can be integrated into classification system to improve activity and gesture recognition accuracy. In [38], body temperature of soldiers is monitored to detect hypothermia. Similar works include [39], [40]. In [39], heart rate measurement is acquired to monitor the dynamic regulation of the heart, while [40] measures knee joint movement to assess its functional ability. Inertial sensors, specifically accelerometers, are the most frequently used in monitoring human activity. They measure acceleration of an entity along sensitive axes. Accelerometers are found to be effective in recognizing physical activities such as walking, running, sitting and standing [35]. They can also be used for detection of fall [41].

Wearable sensor-based activity recognition suffers from limitations and is not suitable for monitoring ADL which involve multiple interaction with the environment. For example, preparing meal activity involves complex hand motions and object interactions which would be difficult to recognize by using wearable and especially inertial sensors only. It is also not sufficient to differentiate even simple activities such as working with computer and having meals [35]. As a result, another approach which makes use of sensors attached to objects has emerged. The approach which is referred to as dense sensing approach attaches sensors to objects in the environment to capture the user-object interactions. The approach performs activity recognition through the inference of user-object interaction. An object being used provides clues about the activity being undertaken. In this way, activities can be recognized from data collected from sensors that monitor the interaction with objects in the environment.

Dense sensing makes applications such as smart environments possible, whereby it is adopted to realize smart homes [15], [16], [20]. The deployed sensors capture an inhabitant movement and environmental events and these data are processed to infer the undergoing activities. This approach is called “dense sensing-based activity recognition”. In [42], environmental state-change sensors are used to collect information on user-object interaction to recognize activities such as preparing meal, toileting and grooming. In [43], four binary sensors, motion detectors, break-beam sensors, pressure mats and contact switches, are used to track the inhabitants’ movements and recognize their activities. Similar work is found in [44] where sensors such as reed switches, pressure mats, mercury contacts, PIR, float sensors and temperature sensors are used for activity recognition. In [45], accelerometers are attached to objects such as plates and cups, and reed switches are attached to fridge, dishwasher and drawers for activity recognition.

2.5 Activity Modeling and Recognition

Activity modeling and recognition can be classified into two approaches, data-driven approaches and knowledge-driven approaches. Data-driven approaches use learning-based (machine learning) techniques to model the activity patterns by extracting specific features from sensor data. In knowledge-driven approaches, prior knowledge is exploited to build semantic activity model by using knowledge engineering techniques, and then reason on it with input sensor data.

2.5.1 Data-driven Approaches

There are two categories of data-driven approaches, generative modeling and discriminative modeling. Generative modeling attempts to model the probability distribution over the sensor observations and activity labels, and performs classification using Bayes theorem to calculate the posterior probability. Naïve Bayes is a simple classification model based on Bayes' theorem with independence assumption between the features, and has been used extensively for activity recognition [46]–[49]. The classifier models all features that characterize the activity by estimating their joint probabilities, and chooses an activity with the maximum posterior probability. In the literature, naïve Bayes produces good classification accuracies. However, the performance might reduce if the independence assumption is broken [50]. Furthermore, naïve Bayes is not capable of modeling temporal information which is important in activity recognition [35]. Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional probabilities in the form of directed acyclic graph. The model encodes the dependencies among the variables and the causal relationships, allowing more accurate activity classification [51]–[54].

Hidden Markov Model (HMM) is a probabilistic approach that can model temporal information. HMM is composed of a finite set of hidden states and observations, where each state has a probability distribution over state transitions and the possible observations. HMM have been successfully used in the domain of activity recognition due to its ability to model temporal information [55]–[58]. In [57], the activities are the hidden states and the observations are the user-object interactions derived from the sensor data. In [55], two-layer HMM is developed for recognizing hospital-staff activities. The first layer comprises two HMMs, where people interactions and object interactions are used as inputs. The outputs of the models are fed to the second layer HMM to produce the classification. A problem with the use of traditional HMMs is that the state duration is modeled with fixed distribution. Hidden semi-Markov model can be used to explicitly model the state duration [59], and, hence improving the performance of the classifier. Despite the successful implementation in activity recognition, due to its strict independence assumptions on the observations, an HMM is not capable of capturing transitive dependencies of the observations. Furthermore, an HMM may not be able to recognize all the possible observation sequences consistently without significant training process [57].

Discriminative modeling attempts to model the dependence of the activity labels on the sensor observations. The k-Nearest Neighbors (k-NN) is a non-parametric, instance-based learning that performs classification based on similarity measure called distance function. The approach has been used in activity recognition and provide very good results [48], [60]. A drawback of k-NN is the cost of computation because classifications require the entire training set to be compared with the new instance, and, hence, not feasible for mobile devices [24]. Decision Tree is a hierarchical model where the training data is partitioned into nodes and branches. Each branch from the root to a leaf node represents the classification rule. The generated classification rules are easy to understand and interpret, and useful for analysing the feature extraction and sensor performances [46]. [46], [47], [61], [62] have used Decision Tree in classifying human activities from the body acceleration and angular velocity data. Support vector machine (SVM) is a classification algorithm that searches for the linear optimal decision boundary to separate the training data of one class from another. SVM has been applied in classifying physical activities [47], [63] and in detecting falls [64]. A variant of SVM called Probabilistic-SVM is proposed and provides a probability vector instead of a single predicted output to deal with transitional activity signals such as stand-to-sit and sit-to-lie [65]. Conditional random field (CRF) is a discriminative classifier that includes temporal information by modeling the conditional probability of the state sequence rather than the joint probability of the states. As such, the classifier eliminates the independence assumption which allows CRF to incorporate complex features of the observation sequence. CRF has been used in classifying activity from body acceleration data [66], [67] and embedded sensors [68]. Van Kasteren et al. [69] recorded a dataset consisting of 14 sensors embedded in a three-room apartment for 28 days. Using the dataset, a study has been conducted to compare the offline and online performance of HMM and CRF model for activity recognition in terms of accuracy. The results show that CRF achieved an overall accuracy of 95.6% which is 1.1% higher than HMM in offline experiment. However, the overall accuracy for HMM (94.4%) is higher than CRF (88.2%) in online experiment.

2.5.2 Knowledge-driven Approaches

Numerous knowledge-driven approaches have been developed such as key-value, markup schemes, object-based, logic-based and ontology-based modeling [12], [27], [70]. Key-value model is the simplest form of context modeling. It uses simple key-value pairs to define the list of attributes and their values describing context information used by context-aware applications. Markup scheme is an improvement over the key-value modeling technique whereby the key-value pairs are stored under appropriate tags, and as a result it allows efficient data retrieval. An example of popular markup scheme modeling is Composite Capabilities/Preference Profiles (CC/PP) [71]. Further model validation is available through schema definitions such as eXtensible Markup Language (XML). However, these approaches are not able to capture variety of context types, relationships, dependencies, timeliness and quality of context information. These approaches also exhibit lack of consistency checking and reasoning capabilities [70]. Object based technique uses object oriented concepts to model data using class hierarchies and relationships. The advantages of using this technique are encapsulation, re-usability and easy integration with context-aware systems. However, it does not have reasoning capabilities and validation of the model is difficult due to the lack of standards and specification [12], [70].

Logic based modeling uses facts, expressions and rules to represent context information. Rules can be used to express policies, constraints and preferences. Therefore, reasoning is possible in logic based modeling. Furthermore, the model representation can be developed by employing interactive graphical techniques to allow non-technical users to add rules and logic to the system which makes it highly flexible. However, logic based modeling lacks standards, specification and validation and is strongly coupled with applications, which reduces its re-usability and applicability [12]. Ontology-based modeling is a formal and explicit way of specifying and representing domain knowledge through formal axioms and constraints. It uses semantic technologies to represent and organize context information according to their relationship into hierarchical structures which are understandable to both human and machine. Ontology-based models have several advantages over other models [12], [50], [70], [72]. Firstly, they allow the domain knowledge to be decoupled from the operational knowledge.

Next, they have strong support through standardization such as Resource Description Framework Schema (RDF) and Web Ontology Language (OWL) and hence a variety of development tools are available. The current recommendation is OWL 2 which is an extended version of OWL. The main reasons OWL is recommended as the context modeling mechanism are interoperability among other context-aware systems, high-level of inference/reasoning support, more expressive and World Wide Web Consortium (W3C) support for standards and specification. Finally, rules which are tightly integrated into reasoning can be expressed via Semantic Web Rule Language (SWRL). Despite the advantages of ontology-based technique, ontological reasoning is computationally expensive, support for modeling temporal information is minimal and they cannot deal with uncertainty.

2.6 Dealing with Uncertainty in Activity Recognition

Uncertainty is always present in ambient intelligence environment [73]. Uncertainty may arise due to errors in sensor measurements, missing activations or communication failures. Activities performed by persons are carried out in different sequences and with different durations depending on the persons' habits and lifestyle. Hence, uncertainty may also arise as a result of variability in human activities. As such, uncertainty significantly influences the decision making in activity recognition. Uncertain sensor data is normally associated with incompleteness, imprecise, inaccurate, timeliness and incongruent [50], [70], [74].

Numerous approaches have been used for reasoning under uncertainty. Probabilistic theory is the most widely used method in dealing with uncertainty. In probabilistic theory, the likelihood of an event is represented by means of a non-negative value called probability. Probabilistic theory such as Bayes theorem can model the reliability of sensor data by learning the correlation between sensor data and the activity to be detected as presented in [75]. Moreover, Bayesian networks are well suited for resolving conflicting data by representing causal relationships between context information and activities through conditional probabilities. In [53], contexts derived from embedded sensors are fused using Bayesian network to achieve location-aware activity recognition. Hidden Markov Model is another probabilistic approach which can be used for sensor fusion as reported in [59]. However, incompleteness of sensor data would reduce the performance of probabilistic approaches because they requires sufficient and representative datasets to obtain reliable predictive models [35], [50].

Data-driven approaches such as support vector machines (SVM) and Decision Trees model the classification boundary rather than the joint probability of the observation variables and class labels. SVM can deal with missing sensor data by defining a risk function to incorporate uncertainty due to missing data into the predictive model [76]. An enhancement to the Decision Tree-building algorithm is reported in [77], which uses a numeric weighting scheme when splitting the data to deal with incompleteness. Imprecise sensor data can be dealt with fuzzy logic. Fuzzy logic assigned a membership value to an element which quantifies the degree of membership of the element in a given set. For example, a range of sensor data in ontology can be represented by a fuzzy membership function [78]. Fuzzy logic can also be used to quantify the relevance of sensor data with time by defining a temporal decaying function [79]. Dempster-Shafer (DS) theory is an evidence theory based on belief function, which can be used to combine context information derived from multiple sensors to calculate the degree of belief of an activity. In DS theory, the universal set comprises all possible states of an entity. Hence, the belief function can explicitly represent any ambiguity or ignorance about what is being observed such as incompleteness of sensor data [80]. Table 2.1 summarizes the aforementioned approaches in resolving uncertainty.

Table 2.1: Reasoning under uncertainty approaches.

	Bayesian model	Decision Tree & SVM	Fuzzy logic	Dempster-Shafer theory
Incompleteness		Y		Y
Imprecise			Y	
Inaccuracy	Y	Y		
Timeliness			Y	
Incongruent	Y	Y		Y

2.7 System Architecture

As previously introduced, the objective is to develop a context-aware activity recognition system for elderly healthcare that is capable of inferring the ADL in the home to support independent living. In this thesis, an activity recognition system which combines physical activity recognition using a single wearable sensor and dense sensing-based activity recognition is proposed. Our proposed approach harnesses the best of both sensing approaches to achieve a robust and comprehensive context-aware activity recognition system which has the ability to elicit and reason context information from the user and environment (user-object interaction and location). In principle, dense sensing-based activity recognition recognizes the ADL by classifying the sensor data of the user-object interaction and location in the environment. The wearable sensor-based activity recognition provides context information about the user in the form of physical activity which can be used to resolve uncertainties in the activity recognition such as distinguishing between sitting on a sofa and lying on a sofa. The proposed activity recognition system is graphically depicted in Figure 2.1.

The lower layer (SENSORS) includes wearable sensor and ambient sensors embedded in the environment. The wearable sensor is based on a single tri-axial accelerometer. The embedded tri-axial accelerometer provides data about the mobility and activity information about the user in the form of body acceleration measurement. A machine learning-based classifier processes the sensor data by building a feature vector to predict the user's physical activity (Chapter 3). In addition, a physical activity transition model is integrated into the recognition system to validate the activity transition. As a result, the recognition system becomes more robust (Chapter 4). The classification output represents the contextual data about the user's physical activity. The ambient sensors capture the raw contextual data by using various types of sensors. The contextual data represents the events occurring in the environment such as user-object interaction context and location context. The contextual data also includes the time of the sensor data (timestamp) that describes the temporal relationship between events. The data provided by the ambient sensors and wearable sensor-based activity recognition are communicated through a wireless connection to the base station, to be mapped into the activity ontology. The activity ontology is a description of shared concepts and their relationship within the domain of activity recognition. Then, the ontological reasoner is used to reason the information represented in the ontology to infer the activity being performed in the environment. The ontological reasoner is integrated with DS theory to handle uncertainty due to missing sensor data (Chapter 5). The fusion of the contextual information from user and environment allows the system to resolve uncertainty during inference process achieving a robust context-aware activity recognition system (Chapter 6).

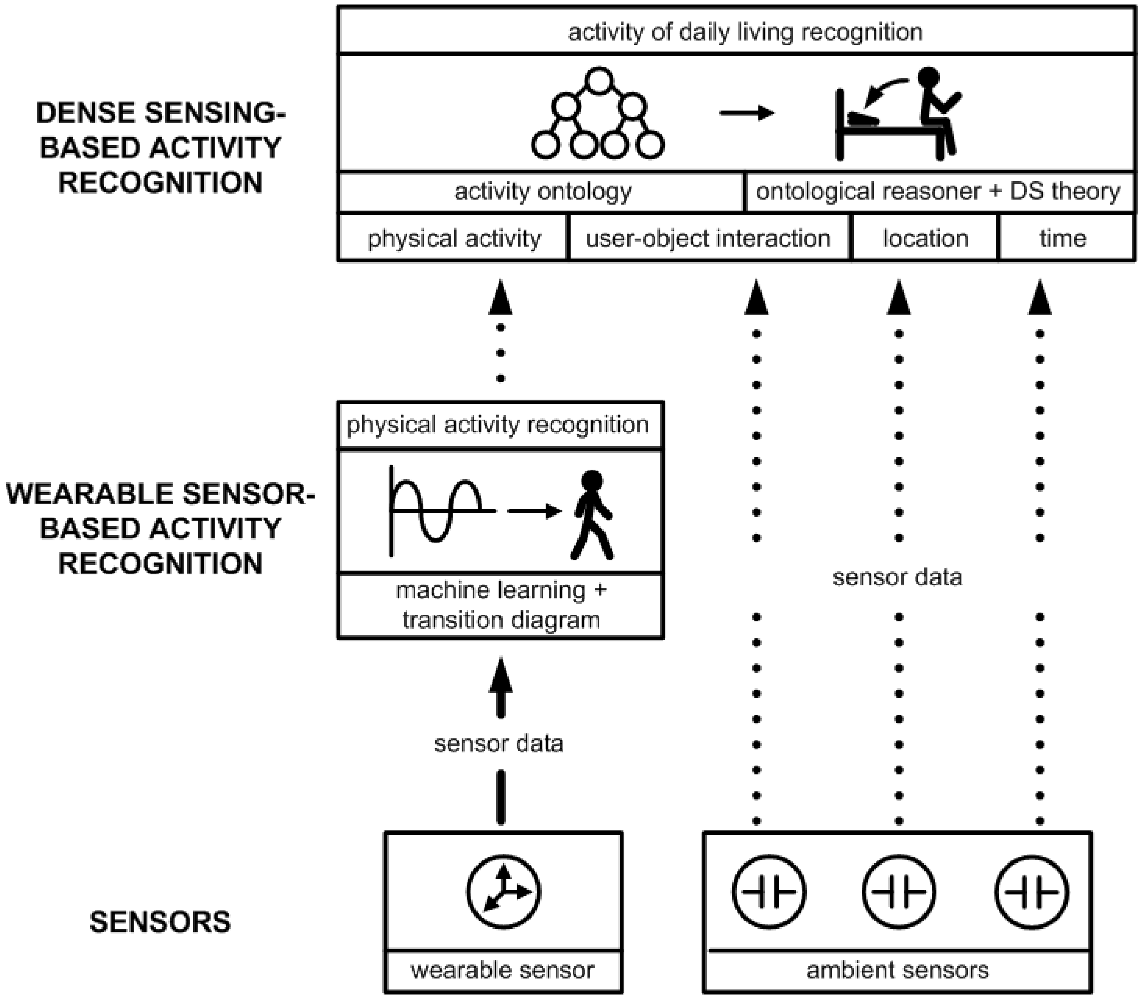

DENSE SENSING-BASED ACTIVITY RECOGNITION
activity of daily living recognition
activity ontology
ontological reasoner + DS theory
physical activity
user-object interaction
location
time
WEARABLE SENSOR-BASED ACTIVITY RECOGNITION
physical activity recognition
machine learning + transition diagram
sensor data
sensor data
SENSORS
wearable sensor
ambient sensors

Figure 2.1: Architecture of context-aware activity recognition system

2.8 Conclusions

In this chapter, the impacts of aging population and the viable technology to support independent living have been presented. The overview of activity recognition system in terms of sensing technology and data processing techniques are reviewed. Following this, the proposed system architecture is described. In the next chapter, the wearable sensor-based activity recognition is presented. A novel adaptive sliding window segmentation and activity transition model for robust physical activity recognition is described. Following this, the dense sensing-based activity recognition is presented. A novel reasoning algorithm that features ontological reasoning mechanism of Description Logic and uncertainty management due to missing sensor data is described. Finally, the proposed sensor fusion methodology is described to achieve a hybrid approach to activity recognition. In each chapter, the state-of-the-art and research challenges are described.

3

Adaptive Sliding Window for Physical Activity Recognition

3.1 Introduction and Problem Formulation (Physical Activity)

This chapter introduces the methodology of the wearable sensor-based activity recognition system. The ability to gather the physical activity and physiological information is a crucial feature in context-aware activity recognition system. Particularly, monitoring physical activity is a task of high interest within elderly care applications because physical activity is associated with physical functional state [81]. Furthermore, transitional activities such as rising from a chair and sitting down is a prerequisite for maintaining independent living. Difficulties in performing these activities can limit independence and lead to a less active lifestyle and a subsequent deterioration in health [82]–[84]. Falls cause two thirds of fatal death in elderly people aged 65 years or older [85], and they are the most common type of accidents among the elders [86]. Most falls occurred during postural transition activities such as from standing to sitting and vice versa and when initiating walking [87], [88]. Several features of sit-to-stand or stand-to-sit performance have been associated with falls or fall risks such as transition duration and number of successful attempts [86], [89]. Therefore, it is important to recognize transitional activities so that early preventive measures can be provided to prevent fall incidents. Also, physical activity is an essential information in context-aware activity recognition system to infer ADL [24]. ADL information can be used by the system to react and adapt to the circumstance of the user, allowing preventive measures to be taken if necessary.

Activity recognition usually segments the sensor signals into windows for the successive features extraction and classification. The size of the segmented windows is empirically selected based on past experiments and hardware limitations for specific types of activity recognition. Majority of approaches used window size in the range of 2s to 6.7s while a few of them used larger window size such as 10s and 12.8s [90]–[93]. As a result, the developed techniques may not be applicable to be trained for recognizing different activities. In addition, misclassifications could still happen especially for transitional activities. This is due to the fact that the length of transitional activity signals varies depending on the time to complete the activity [92], [93]. Evidently, sliding window with a fixed size is not an effective approach for activity recognition system. This is the motivation for our proposed approach in which the window size is dynamically adapted during classification, based on certain characteristics in the signal, to better capture signals of different activities.

In this thesis, a systematic adaptive signal segmentation approach is developed for physical activity recognition based on the use of a single tri-axial accelerometer. The foundation of adaptive sliding window approach is presented. The approach can detect not only static and dynamic, but also transitional activity signals of varying period as the segmentation window is being evaluated and its size adapted dynamically. The window size is adaptively adjusted based on the continuous evaluation of the activity signals. As a result, a more effective window size can be selected for segmentation to achieve more accurate classification.

3.2 Related Works

3.2.1 Existing Signal Segmentation Approaches

In activity recognition, signal segmentation is a technique of dividing a large signal into smaller segments for processing and has direct impact on the quality of feature extraction and classification accuracy [94]. At the same time, it also determines suitability of the approach for real-time operation. Numerous techniques have been proposed for signal segmentation. Santos et al. [95] proposed an adaptive sliding window approach to improve segmentation of human action sequences for activity recognition. In the approach, window size and time shift are dynamically adjusted based on entropy feedback to improve the classification results. However, the experiments do not involve transitions between activities such as stand-to-sit, sit-to-lie and lie-to-sit. Furthermore, the algorithm could be computationally expensive since shorter time shifts would increase the rate of classifications per second. Kozina and Lustrek [96] proposed a segmentation algorithm that searches for significant differences between consecutive samples which is defined by the reduction of the samples' values exceed certain threshold. The threshold is determined by the difference of average maximum and minimum values of a set of samples. Bifet & Gavalda [97] proposed a segmentation algorithm that can adapt the window size according to the determination of concept drift (change in data stream) in which, the window size is increased when the data values in the window are stable (low concept drift) to include more training instances. Otherwise, the window size is decreased. The change detection is based on mean difference of two sub-windows is greater than a given threshold. However, the algorithms are sensitive to noise such as abnormal high or low peaks which is very common in acceleration data. Núñez et al. [98] proposed the OnlineTree2 algorithm which uses an adaptive windowing technique to induce improved Decision Tree by evaluating the performance of the Decision Tree. Sheng et al. [99] proposed an adaptive time window method to extract features from quasi-periodic signals more accurately for activity recognition. The method uses pitch extraction algorithms to achieve more effective segmentation. The experiments involve dynamic and static activities only. Activity-defined techniques detect changes in activity and take the initial and end time as segmentation boundaries. Then, the specific activity in the window is identified. In [100] and [101], wavelet analysis is used to detect changes in frequency characteristics which indicate changes in activity. In [102], a changing point which is defined by the change in action from static activity to dynamic activity and vice versa is detected by calculating the displacement of sensor data, and from this point the window segmentation is set and classified. Event-defined techniques locate specific events such as heel strikes and toe-offs to segment a signal [103], [104]. The detection of events is achieved by filtering the signals to produce resultant signals which indicate the location of the events. In [105], wavelet analysis is used to detect the heel strikes and toe-offs events. Jasiewicz et al. [106] uses foot linear accelerations and foot sagittal angular velocity to detect the events. Benocci et al. [107] detects walking tasks on loaded conditions by identifying gait cycle through heel strike events. Symbol-based method is used to detect heel strikes and toe-offs events in [108]. Ignatov and Strijov [109] proposed a segmentation algorithm for activity recognition that defines the segmentation boundaries by extracting the fundamental period of the signals. Sliding window is the most widely used technique in activity recognition due to its simplicity. It segments the signal into a window of fixed size for features extraction and classification. Then the window is shifted to segment new sensor data with a degree of overlap. A degree of 50% would shift the window by half of its size, which means 50% of the previous data are included in the window. A degree of 0% means that the windows are not overlapping. Various window sizes from 0.1s to 12.8s have been used in previous studies [90]–[93]. However, in our study we have found that fixed sliding window is not an effective segmentation approach for activity recognition because the lengths of transitional activities are varies from one to another. A small window size could split an activity signal while large window size could contain multiple activity signals. Both cases could lead to suboptimal information for an activity classification algorithm.

3.2.2 Physical Activity Recognition Systems

Inertial sensors, specifically accelerometers, are the most frequently used, and they are found to be effective in monitoring physical activities such as walking, running, standing and sitting [35]. In [46], [64], [67], [110], [111], multiple accelerometers were placed on different parts of body to investigate the performance of the sensors at various body parts for activity recognition. Their findings provide a strong case for the use of accelerometers in activity recognition. Some of the studies utilized accelerometers with other sensors such as barometer, electrocardiogram (ECG) and GPS [36], [47]. Multiple attachment of sensors allows the systems to recognize complex activities such as cooking, grooming and cleaning with high accuracy. However, these systems are not feasible for long-term activity monitoring because they impede subject's daily physical activities due to multiple attachments to the body [13]. The use of single accelerometer has been investigated for activity recognition with encouraging results [36], [47], [63], [112]–[115]. The focus of the studies is to recognize physical activities such as ambulation activities (walking, running), body postures (sitting, standing) and postural transitions. In [36], [47], [63], [112], [113], [116], fixed window size in the range of 2.56s to 10s is used to classify the activity signals. In [114], [115], wavelet transform is used to decompose the raw acceleration signals to extract wavelet features such as low-frequency components and wavelet coefficients using 2.56s and 10.24s window with 50% overlapping. The results show wavelet transform can discriminate the activities effectively. However, according to [90], time-frequency features outperform the wavelet features in the performed experiments. All the aforementioned works do not consider transitional activities in their studies.

Transitional activities are usually disregarded in activity recognition since the number and length of transition windows is relatively lower and shorter than for other activities as reported in [24]. A number of systems have been proposed which consider transitional activities in the classification. In [117], an algorithm is designed to compute tilting angles using signals from three wearable sensors. The computed tilt angles are used to classify walking, body postures and postural transitions such as stand-to-sit and sit-to-stand. However, these systems require attachment of multiple sensors to the body. Ahanathapillai et al. [118] utilized a single accelerometer worn on the wrist to recognize walking, sitting, stand-to-sit and sit-to-stand. Using k-NN, accuracy rate of 89% was achieved. Window size is not mentioned in the study. Khan et al. [119] utilized a single accelerometer to recognize physical activities including postural transitions such as stand-to-sit, sit-to-lie and stand-to-walk with an accuracy of 97.9%. However, fixed window size of 3.2s is used which can give rise to an increase in false negative rate especially when the main focus is ambulation activities and body postures rather than transitions.

Reyes-Ortiz et al [65] presented the Transition-Aware Human Activity Recognition to deal with transitional activities: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie and lie-to-stand. 2.56s window size and 50% fixed window overlap is used to classify the signals from an accelerometer and a gyroscope attached to the waist. The approach uses heuristic filtering technique to filter a sequence of classification in the form of probability vectors to recognize transitional activities by measuring the length of the signal activation. A transitional activity is determined if the signal activation does not exceed a threshold. However, the system does not distinguish between the different transitional activities in which the transitional activities are classified as postural transition. The proposed approach achieved overall accuracy of 96.7%. Gupta and Dallas [48] introduced new features to effectively capture the characteristics of transitional activities: stand-to-sit/sit-to-stand and stand-to-kneel-to-stand. The features are mean trend, windowed mean difference, variance trend, windowed variance difference. These features further break the fixed window size of 6s into 0.5s sub-windows with no overlap, and extract the characteristics of the signals within the sub-windows. They also evaluated features called detrended fluctuation analysis coefficient, uncorrelated energy and maximum difference acceleration to capture the correlation and uncorrelation between signals. The features were chosen ahead of other features such as mean, variance and energy by Relief-Feature (Relief-F) selection algorithm and wrapper-based sequential forward floating search (SFFS). The proposed approach achieved overall accuracy of 98%. However, stand-to-sit and sit-to-stand are not distinguished by the system in which the activities are classified as a single class (sit-to-stand/stand-to-sit). Table 3.1 shows the comparison of the related works and our proposed approach in terms of wearable sensing component and activities recognized.

All presented existing approaches used sliding window technique with various fixed window sizes and degrees of overlapping without discussing criteria for selecting window size. The impact of window size on the performance of activity recognition system has been investigated in [92], [93]. The results show a variation in accuracy between the different window sizes, with transitions being most often misclassified. In this work, the transitional activities are targeted by adaptively adjusts the size of segmentation based on the signal information. Therefore, a more effective segmentation can be selected to achieve more accurate classification. We validate the proposed algorithm with an internal and public datasets. The datasets contain physical activities and transitional activity signals of different lengths which is required for the evaluation of the system. Additionally, we have implemented the state-of-the-art approach described in [48] to compare with the proposed approach. This work is, to the best of our knowledge, the first to propose an adaptive sliding window to deal with transitional activities.

Table 3.1: Comparison of different related works that deal with transitional activities.

Referenced work	Wearable sensing component	Activities recognized
Reyes-Ortiz and colleagues	Single wearable sensor: accelerometer and gyroscope.	Seven activities: walking, walking upstairs, walking downstairs, standing, sitting, lying down and postural transition.
Gupta and Dallas	Single wearable sensor: accelerometer.	Six activities: walking, running, jumping, staying stationary, sit-to-stand/stand-to-sit and stand-to-kneel-to-stand.
Adaptive sliding window (this dissertation)	Single wearable sensor: accelerometer.	Ten activities: walking, standing, sitting, lying face-up, lying face-down, stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, falling.

3.3 Characterization of Activity Signals

A key factor in signal segmentation is to select a suitable window size for activity classification. Window size is important because it needs to capture necessary characteristics of a signal in order to achieve correct detection/classification. Short windows could slice an activity signal into multiple separate windows. Thus a truncated signal lacks the full information to describe the activity. On the other hand, larger window size could contain multiple activity signals which could also lead to misinterpretation of physical activities. The most effective window size depends on the type of signals being evaluated because different activities have different periods of motion. The scenario is shown in Figure 3.1. The signal contains three activity signals

(A1, A2 and A3) with varying length. The fixed sliding window with 50% overlapping is employed to classify the activities. As shown in Figure 3.1, only signal A2 is shorter than the window size while signals A1 and A3 are longer. Therefore, the signals are not fully segmented by window W1, W2 and W4. In both cases, misclassification could happen because the windows do not have optimal information of the signals.

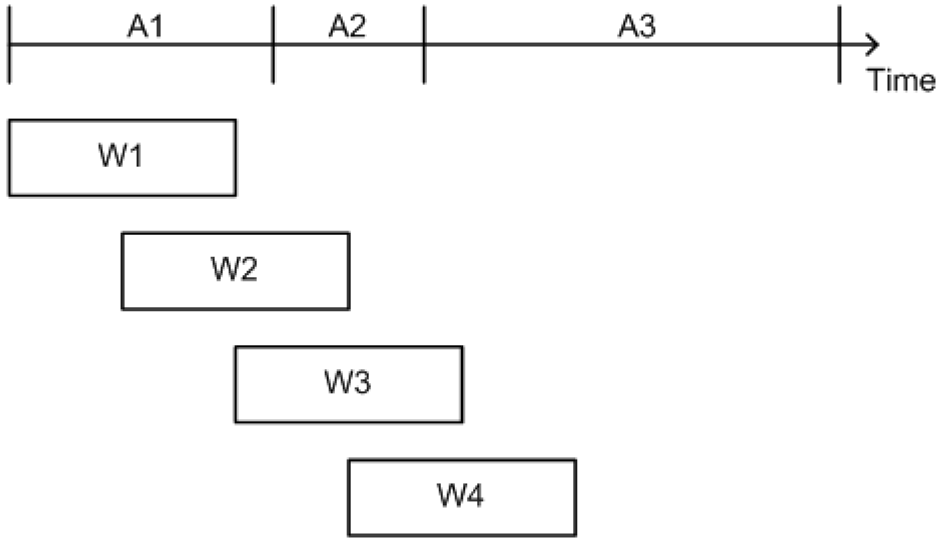
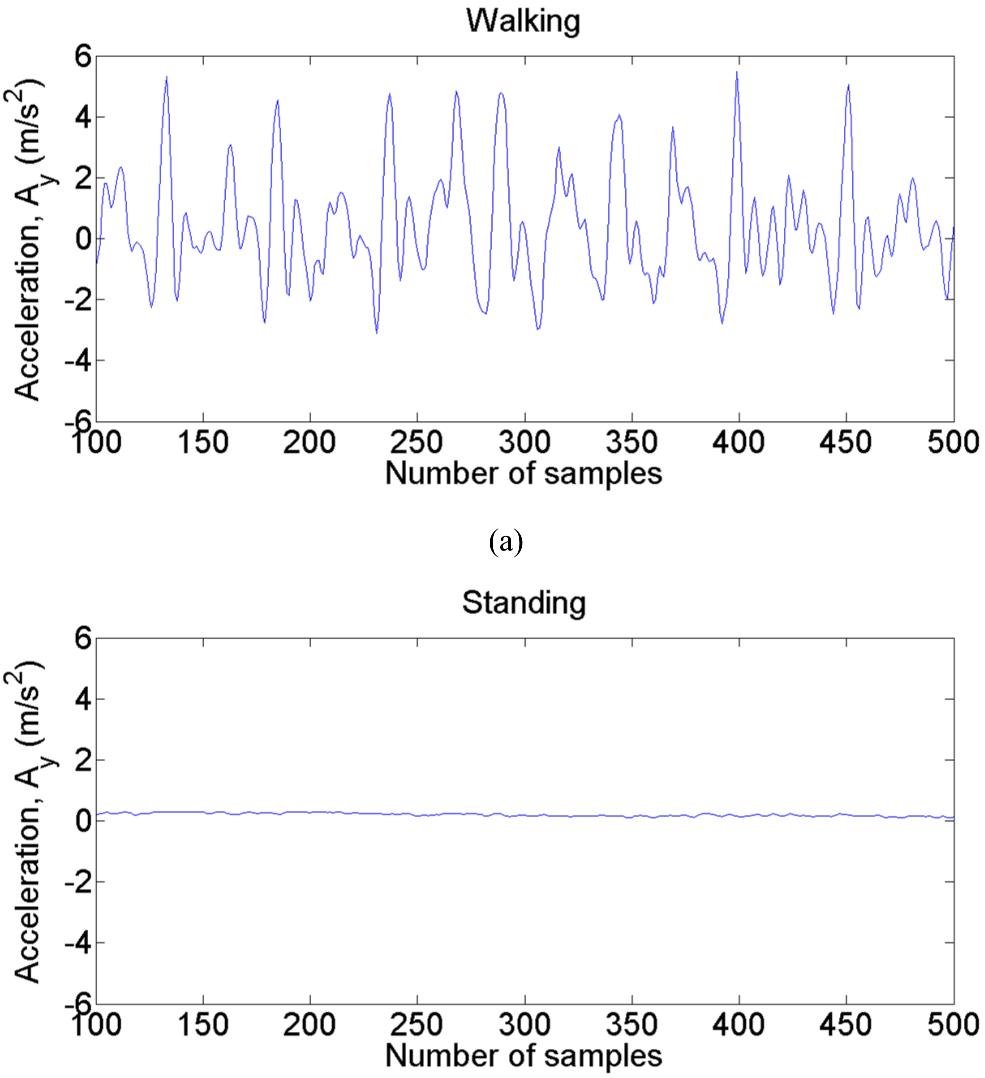


Figure 3.1: Activity classification with fixed sliding window.

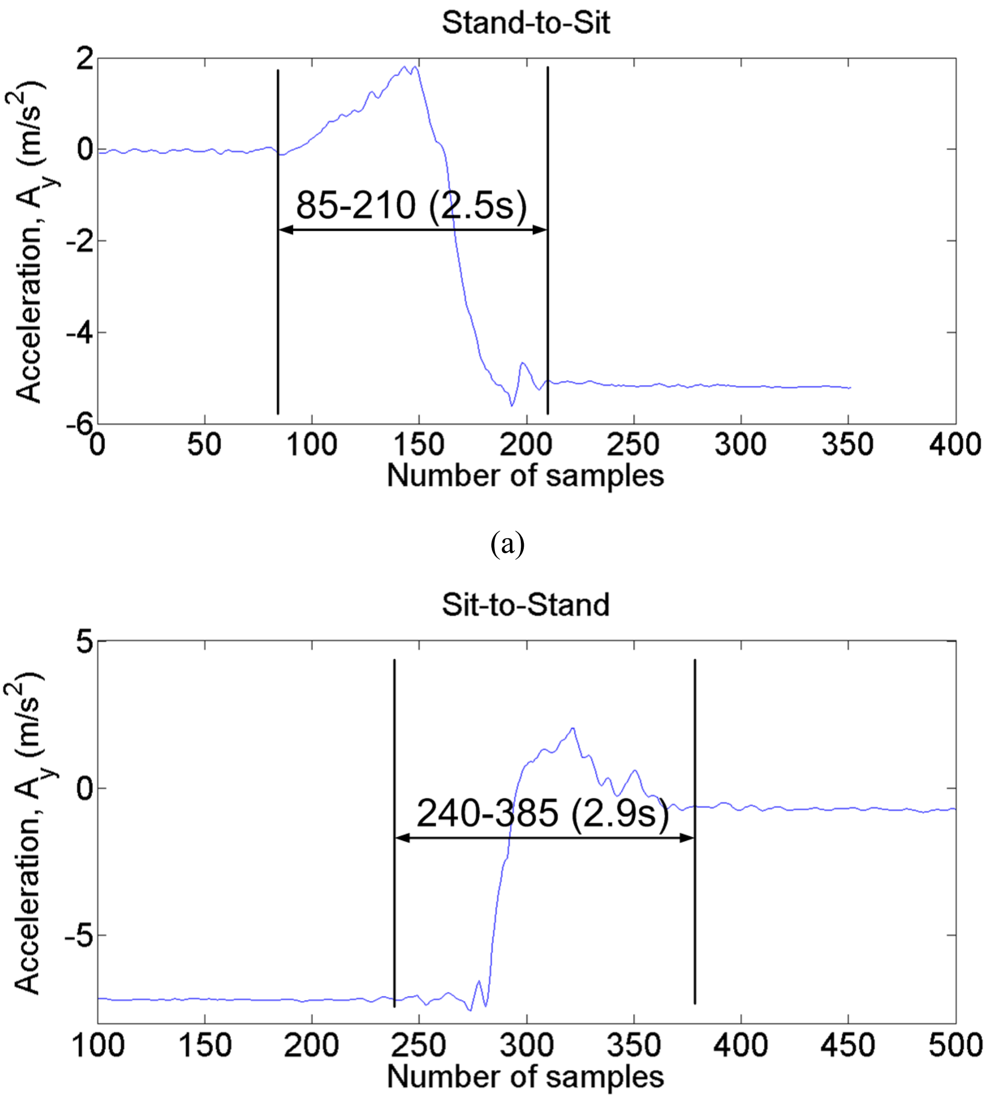
To demonstrate the differences in signal characteristics and motion periods, three scenarios of activity signal are considered and illustrated in Figure 3.2 and Figure 3.3. The signals are generated at 50Hz by an accelerometer attached to the right waist. In the first scenario, the signals are generated by dynamic activity. Dynamic activity signal exhibits periodic behavior with high frequency components and the trend is generally flat. An acceleration signal along the horizontal axis, A_y of a dynamic activity (walking) is illustrated in Figure 3.2(a). The second scenario involves segmenting signals generated by static activities. Since static activities do not involve much body movement, the generated signals have almost constant magnitude values and very low frequency components. Therefore, the trend of the signals is also generally flat. Figure 3.2(b) illustrates the acceleration signal along the horizontal axis, A_y of the standing activity.



(a)

(b)

Figure 3.2: Acceleration signals for (a) walking and (b) standing.



(a)

(b)

Figure 3.3: Acceleration signals for (a) stand-to-sit and (b) sit-to-stand.

In the third scenario, the signals have low frequency components and the magnitude is either increasing or decreasing. Furthermore, the signal length is varying from one to another because some activities take longer time to complete. This type of signal is generated by transitional activities. For example, from the position of standing to sitting, the trend of the acceleration signal along the horizontal axis is decreasing before the magnitude is stabilized at -5m/s^2, and it takes 2.5s (125 samples) to complete as illustrated in Figure 3.3(a). The flat signal indicates the person is in a sitting position. Conversely, when the person is getting up, the trend of the signal is increasing before the magnitude stabilizes at 0m/s^2, and it takes 0.4s (20 samples) longer to complete as shown in Figure 3.3(b). Evidently, fixed size window is not the most effective approach to achieve accurate activity recognition due to the diverse characteristics and periods of different activity signals.

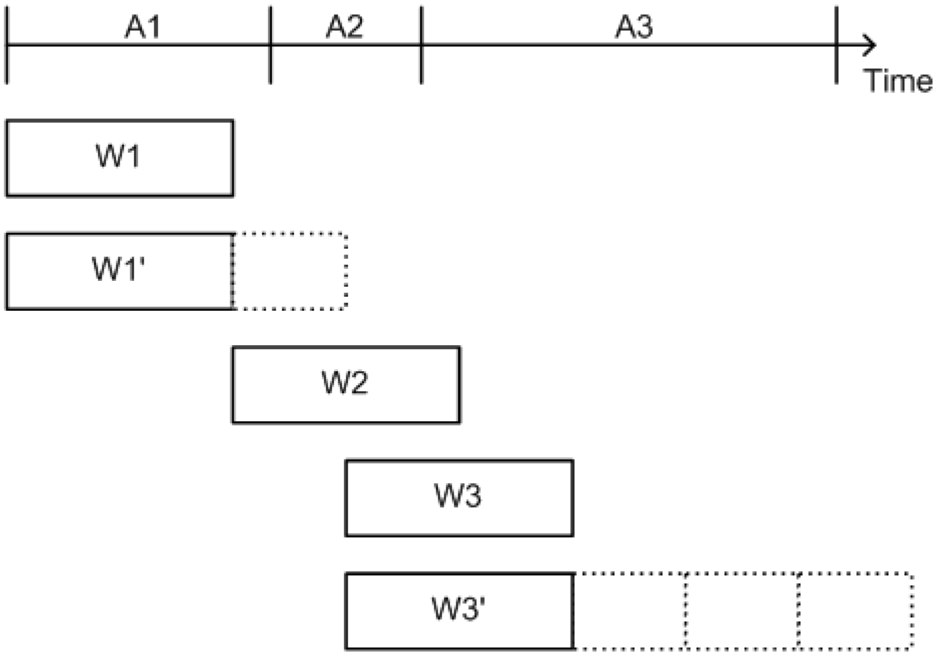


Figure 3.4: Activity classification with adaptive sliding window.

In this thesis, we present the adaptive sliding window algorithm for physical activity recognition as shown in Figure 3.4. In Figure 3.4, activity signals of varying length are being classified by employing adaptive sliding window. The algorithm has an initial size of window used for segmentation which can be expanded dynamically to accommodate more samples if the signal is deemed longer than the current window size. The scenario is shown in Figure 3.4, in which windows W1' and W3' are the actual segmentation windows expanded from W1 and W3, respectively, since signals A1 and A3 are longer than initial window size. In this way, a more effective segmentation for classification can be achieved. The key challenges are the criteria for triggering window size expansion, how to adapt the window size to capture the whole signal and how to determine the most effective window size.

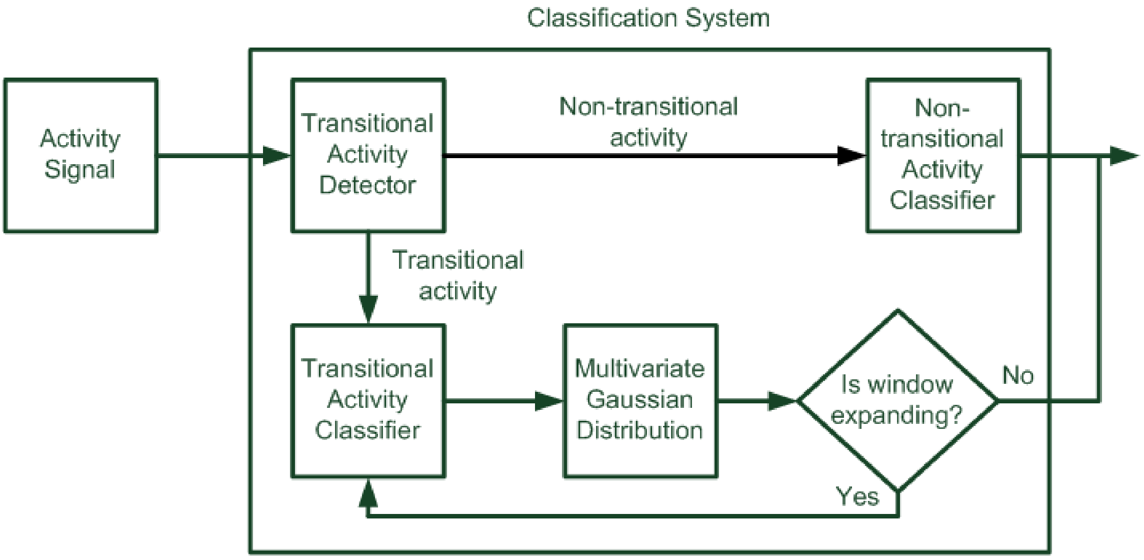

Classification System
Activity Signal
Transitional Activity Detector
Non-transitional activity
Non-transitional Activity Classifier
Transitional activity
Transitional Activity Classifier
Multivariate Gaussian Distribution
Is window expanding?
No
Yes

Figure 3.5: The block diagram of physical activity recognition system.

3.4 System Overview

The block diagram of the proposed physical activity recognition system is given in Figure 3.5. The activity signal from accelerometer is pre-processed for noise filtering. Then, relevant features are extracted from the signal for activity classification. The classification system consists of three classifiers: transitional activity detector, non-transitional activity classifier and transitional activity classifier. All three classifiers are implemented as Decision Tree. The implementation of the classifiers is described in Section 3.6.2. The function of transitional activity detector is to differentiate transitional activity signal from static/dynamic activity signals by processing the signal in the fixed initial window size. When transitional activity signal is identified, adaptive sliding window is executed and the signal is classified by transitional activity classifier. Then, the window is expanded to determine the most effective segmentation by calculating the probability of the segmented signal belong to a particular activity given a set of features, which is classified by transitional activity classifier. Multivariate Gaussian distribution is used to calculate the probability. The window will be expanded as long as the calculated probability is increasing in each iteration. If the signal is non-transitional activity, fixed sliding window is executed and the signal is classified by non-transitional activity classifier.

3.5 Adaptive Sliding Window

The proposed algorithm requires transitional activity signals to be detected in order to trigger the usage of adaptive sliding window technique. Therefore, a feature that can effectively capture the acceleration trend (increasing or decreasing) needs to be determined for identifying the transitional activity. The process of selecting the features for transitional activity detector is described in Section 3.6.2. The pseudo-code is shown in Listing 3.1. The algorithm starts with an initial (default) window size for signal segmentation and classification. The algorithm first distinguishes non-transitional activity (static/dynamic activity) and transitional activity as the windows are being evaluated as defined by line 1-2. Detection of transitional activity signal is performed for every window classification. If a transitional activity window is detected, adaptive sliding window algorithm is executed to expand the window size and segment the transitional activity signal. Otherwise, the window will be classified by the non-transitional (dynamic/static) activity classifier as defined by lines 4. Lines 6–16 define the adaptive sliding window process, which are executed whenever transitional activity signal is detected. The algorithm starts with extracting features to be evaluated by transitional activity classifier, and then calculates the probability density function (PDF) of the classified activity which is used to determine the most effective window segmentation.

Probability density function of d-dimensional data (features) $\mathbb{x} = \{x\}$ given an activity, A_j, denoted by $p\left(\mathbb{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_\mathbf{j}\right)$ is calculated by using multivariate Gaussian distribution, which allows correlation between multiple features and their relevance to the problem to be modeled [120] as follows. Probability density function is the likelihood that a signal belongs to a particular activity, which is used to determine the most effective window segmentation.

$$p\left(\mathbb{x}|A_j\right) \propto p\left(\mathbb{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) = \frac{1}{(2\pi)^{n/2}\left|\boldsymbol{\Sigma}_j\right|^{1/2}} e^{-\frac{1}{2}\left(\mathbb{x}-\boldsymbol{\mu}_j\right)^T \Sigma^{-1}\left(\mathbb{x}-\boldsymbol{\mu}_j\right)} \tag{3.1}$$

where n is the dimension of the feature vector. μ_j is the mean matrix and Σ_j is the covariance matrix corresponding to the features extracted from the window. Parameters μ_j and Σ_j are estimated from the training datasets as follows:

$$\boldsymbol{\mu_j} = \frac{1}{N_j}\sum_{x \in A_j} x \tag{3.2}$$

$$\boldsymbol{\Sigma_j} = \frac{1}{N_j}\sum_{x \in A_j} xx^T - \boldsymbol{\mu_j}\boldsymbol{\mu_j^T} \tag{3.3}$$

where N_j is the total number of observations belonging to activity, A_j. The same features used by the regular classifiers are used to model the distribution.

Based on the recognition result and PDF, the window may be expanded to capture a longer duration transitional activity signal. Window expansion algorithm is an iterative process in which the window size is expanded by the amount defined by an expansion factor (ef) of the initial window size as defined by line 1. The expansion factor in the range of $0 \leq ef \leq 1$ is predefined to determine the size of window expansion. A value of one $(ef = 1)$ indicates that the window is expanded by the size of the initial window. In each iteration, the features of the signal are computed for the transitional activity classifier to be evaluated. Then the PDF of the window corresponding to the activity is calculated. The window will continue expanding until the most effective window segmentation is found. The most effective window segmentation is the window with the highest probability density function value.

One window expansion scenario is illustrated in Figure 3.6. Figure 3.6 shows that the window has been expanded two times. $W_{i,0}$, $W_{i,1}$ and $W_{i,2}$ denote the initial window and the windows after each expansion. The PDF for each window is denoted by $p_{S_{i,j},0}$, $p_{S_{i,j},1}$ and $p_{S_{i,j},2}$. In this scenario, since PDF value of $W_{i,2}$ is lower than the PDF value of $W_{i,1}$, the window expansion operation is stopped at $W_{i,2}$ and $W_{i,1}$ is determined as the most effective window segmentation.

Listing 3.1: Adaptive sliding window

w is the size of initial window in number of samples.
of is the window overlapping factor in the range of $0 \leq of \leq 1$.
ef is the window expansion factor in the range of $0 \leq ef \leq 1$.
k is the number of window expansions, where $k = 0,1,\dots,k_{max}$ and k_{max} is the maximum number of allowed window expansion.
$N_{i,k}$ is the number of samples in window i after k expansion, where $N_{i,0} = w$.
$S_{i,j}$ is a sample in window i, where $j = 0,1,2,\dots,N_{i,k} - 1$ is the sample index within the window.
$p_{S_{i,j},k}(\mathbb{x}|A)$ is the probability density function of extracted features, x, from samples of window i after k expansion given an activity A.
p_{max} is the maximum probability density function value for a window expansion to determine the most effective window segmentation.
$round(i)$ is rounding i to the nearest integer.

```
calculate features of the signal
execute transitional activity detector
if non-transitional activity then
    activity, $A$ = execute non-transitional (dynamic/static) activity classifier
else
    while $k \leq k_{max}$ do
        calculate features of the signal
        activity, $A$ = execute transitional activity classifier
        calculate probability density function, $p_{S_{i,j},k}(\mathbb{x}|A)$
        if $A$ is not changed or $p_{S_{i,j},k} > p_{max}$ then
            $p_{max} = p_{S_{i,j},k}$
            $N_{i,k} = w + round(ef \times w) \times k$
        else
            stop window expansion
        end if
    end while
end if
$S_{i+1,0} = S_{i,v}$ where $v = \left(N_{i,k} - 1\right) - round(of \times w)$
```

Three conditions are defined as stopping conditions of the window expansion. Firstly, classification of the current iteration is found to be different than the initial classification. Initial classification is defined by the transitional activity classifier in the first iteration of window expansion. If classification result is changed in the next iteration, it is assumed the window contains another activity signal and hence affecting the classification. Secondly, the computed probability density function of current iteration is lower than of the previous iteration. This indicates the window contains other activity signal which is the reason of the smaller PDF value. Lastly, the window reaches its maximum number of expansions and the window stop expanding. After window classification process is finished, line 18 shifts the window forward to segment new samples with an overlapping factor (of). The overlapping factor determines the number of samples from current window to be overlapped by the next window. In other words, the new window will contain some samples from the previous window. The overlapping factor is in the range of $0 \leq of \leq 1$.

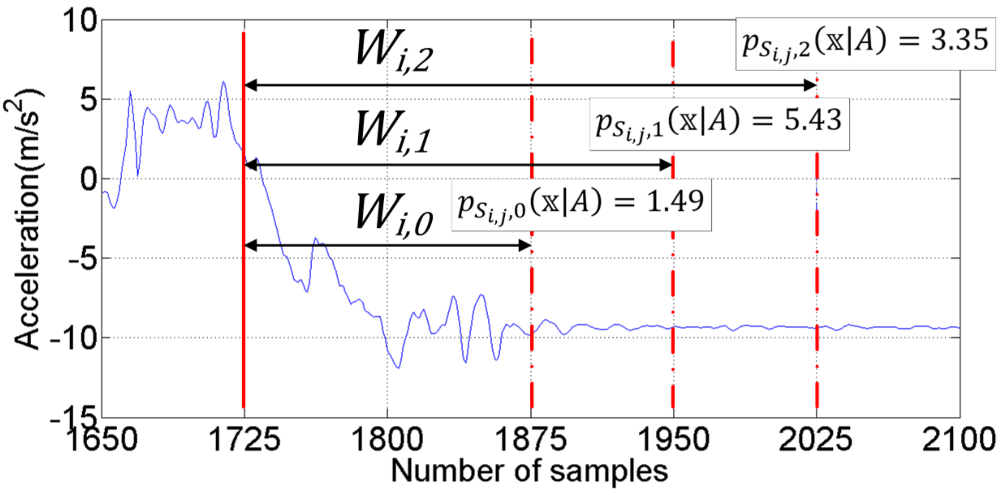


Figure 3.6: Window expansion scenario.

3.6 Experimental Setup for Physical Activity Recognition

3.6.1 Device and Data Collection

Digital tri-axial accelerometer is a sensing device which can measure the acceleration in three mutually orthogonal directions. Virtenio Preon32 wireless sensor node with a digital tri-axial accelerometer is used in this research for data acquisition. The accelerometer is configured to collect acceleration in the range of ± 4.0g at a sampling rate of 50Hz during the experiments. Previous study results show that sampling rate beyond 20Hz increases recognition accuracy by just 1% and without further improvements beyond 50Hz [49]. Therefore, sampling rate of 50Hz is considered to be sufficient. A single tri-axial accelerometer worn on the right waist achieved the highest recognition accuracy in a single sensor comparison study [46], [49], [121] and hence is used in this research. The sensor position with its coordinate system is illustrated in Figure 3.7.

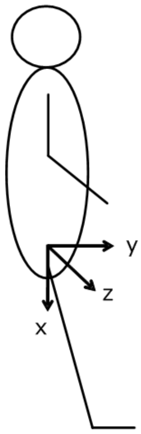


Figure 3.7: Sensor position and its coordinate system.

The accelerometer measures acceleration along X-axis or vertical axis (A_x), Y-axis or horizontal axis (A_y) and Z-axis or sideway axis (A_z). The low pass filter with 0.5Hz cutoff frequency f_c is applied to separate the acceleration force from gravity force [122]. The separation process produces linear acceleration (LA_i) and is performed for each axis to generate LA_x, LA_y, and LA_z. Firstly, the gravity force, $\hat{g}$ is estimated. Then, the acceleration force is subtracted with the estimated gravity force as follows.

Table 3.2: Summary of the thirteen signals.

Signal	Description
A_x	Vertical axis (X-axis)
A_y	Horizontal axis (Y-axis)
A_z	Sideway axis (Z-axis)
A_{xy}	Vertical plane
A_{yz}	Horizontal plane
A_{xyz}	Resultant plane
LA_x	Linear acceleration of vertical axis
LA_y	Linear acceleration of horizontal axis
LA_z	Linear acceleration of sideway axis
LA_{xy}	Linear acceleration of vertical plane
LA_{yz}	Linear acceleration of horizontal plane
LA_{xyz}	Linear acceleration of result acceleration
TA	Tilt angle of the body trunk

$$\hat{g}_t^i = \alpha g_{t-1}^i + (1-\alpha)A_t^i \; where \; i = x, y, z \tag{3.4}$$

where initial value of gravity force, g_0^i is set as: $g_0^x = -9.8m/s^2 \quad g_0^y = 0m/s^2 \quad g_0^z = 0m/s^2$. ∝ is defined as follows.

$$\propto = \frac{\Delta t}{\Delta t + \tau} \tag{3.5}$$

Δt is the sampling period and $\tau = 1/2\pi f_c$ is the time-constant or the filter response in the time domain. Linear acceleration is obtained as follows.

$$LA_t^i = A_t^i - \hat{g}_t^i \; where \; i = x, y, z \tag{3.6}$$

Acceleration and linear acceleration in the horizontal plane (A_{yz} and LA_{yz}), vertical plane (A_{xy} and LA_{xy}) and resultant acceleration (A_{xyz} and LA_{xyz}) are derived from raw and linear acceleration. For instance, acceleration in horizontal plane, A_{yz} is obtained as follows.

$$A_{yz} = \sqrt{A_y^2 + A_z^2} \tag{3.7}$$

The characteristic of tilt angle (TA) of body trunk signal also had been investigated. The tilt angle (TA) of the body trunk can be derived by $cos^{-1} LA_x/LA_{xyz}$. In total, thirteen signals including the raw acceleration, linear acceleration, horizontal and vertical plane acceleration, resultant acceleration and tilt angle are investigated. Table 3.2 list all the thirteen signals.

Six healthy volunteers (4 males, age: 33 ± 2.2 years, 1 female, age: 33 years), including a kid (female, age: 10 years) were asked to wear the tri-axial accelerometer on their right waist. Each subject was asked to perform the activities described in Table 3.3 in their own preferred style and pace. No specific instructions were given about how to perform the activities. All activities were performed continuously for a single trial in a house which consisted of a corridor, a lounge and a bedroom. The length of the corridor and the distance from a room to another is about 10m − 15m. Each volunteer was asked to conduct each experiment five times in their own pace. This internal dataset is referred as IELAB.

Table 3.3: Classification of physical activities.

Dynamic	Static	Transitional
Walking	Standing	Stand-to-Sit
	Sitting	Sit-to-Stand
	Lying Face-up	Sit-to-Lie
	Lying Face-down	Lie-to-Sit
		Falling

A public (SBHAR) [65] dataset is also used in the experiment to validate the proposed algorithm. SBHAR dataset contains activity signals collected gathered from a smartphone inertial sensors (accelerometer and gyroscope). 30 subjects were asked to perform six basic activities. The position of the device is different to our experiment where it was attached to the front waist instead of right waist. The dataset includes six transitional activities: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie and lie-to-stand. Stand-to-lie is actually a sequence of other two transitional activities (stand-to-sit and sit-to-lie). Similarly for lie-to-stand. The labels were defined between the end and the start of consecutive static activities. The experiments generated 5 hours of data. Table 3.4 shows the average, standard deviation and maximum durations of the transitional activities for both datasets. We randomly chose 10 out of 30 data to train the model for both datasets. Then, the model was tested on the 30 data. Only accelerometer signals are considered for the purpose of this research.

Table 3.4: Average ± standard deviation duration and maximum duration of IELAB and SBHAR datasets transitional activities.

	IELAB		SBHAR	
Transitional activity	Average ± standard deviation duration (s)	Maximum duration (s)	Average ± standard deviation duration (s)	Maximum duration (s)
Stand-to-Sit	2.96 ± 0.61	4.60	3.41 ± 0.8	4.50
Sit-to-Stand	2.34 ± 0.43	3.30	2.57 ± 0.5	3.70
Sit-to-Lie	3.22 ± 0.76	5.30	4.12 ± 0.8	7.10
Lie-to-Sit	3.01 ± 0.69	5.20	3.69 ± 0.7	7.30
Stand-to-Lie	N.A.	N.A.	4.95 ± 1.4	8.40
Lie-to-Stand	N.A.	N.A.	3.72 ± 0.8	11.20
Falling	2.99 ± 0.64	4.30	N.A.	N.A.

3.6.2 Pre-processing and Feature Selection

Moving average filter is applied to remove high frequency noise. As shown in Table 3.5, thirteen features are extracted for activity recognition. The slope of signal is calculated by using linear regression technique, which fits a straight line through the signal. Mean Trend and Windowed Mean Difference introduced by [48] describes the trend of mean values over the window. The window is divided into N sub-windows with no overlap. Then the mean of each sub-window, μ_i is calculated. Mean Trend and Windowed Mean Difference are computed as follows.

$$|\mu T| = \sum_{i=2}^{N} |\mu_i - \mu_{i-1}| \tag{3.8}$$

$$|\mu D| = \sum_{i=1}^{N} |\mu - \mu_i| \tag{3.9}$$

In this study, we have investigated the variants of the features by not taking the absolute difference in order to obtain the trend (increasing or decreasing) of signal as follows.

$$\mu T, \mu D = \begin{cases} > 0\ trend\ of\ signal\ is\ increasing \\ < 0\ trend\ of\ signal\ is\ decreasing \end{cases} \tag{3.10}$$

Table 3.5: Initial set of features.

Feature	Key	Formulation
Average	Avg	$\mu = \frac{1}{N}\sum_{i=1}^{N} s_i$
Standard Deviation	Std	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \mu)}$
Skewness	SK	$E\left[\left(\frac{s-\mu}{\sigma}\right)^3\right]$
Signal Magnitude Area	SMA	$\frac{1}{N}\sum_{i=1}^{N}\lvert s_{x_i}\rvert + \lvert s_{y_i}\rvert + \lvert s_{z_i}\rvert$
Slope	M	$\frac{N\sum_{i=1}^{N}(s_i \times i) - \sum_{i=1}^{N} s_i \sum i}{N\sum_{i=1}^{N} {s_i}^2 - (\sum i)^2}$
Absolute Slope	AM	$\lvert \mathrm{M} \rvert$
Spectral Energy	E	$\sum_{i=1}^{N}\lvert S_i\rvert^2$
Mean Trend	$\lvert\mu T\rvert, \mu T$	$\mu T = \sum_{i=1}^{N} \mu_i - \mu_{i-1}$ $\lvert\mu T\rvert = \sum_{i=1}^{N}\lvert\mu_i - \mu_{i-1}\rvert$
Windowed Mean Difference	$\lvert\mu D\rvert, \mu D$	$\mu D = \sum_{i=1}^{N} \mu - \mu_i$ $\lvert\mu D\rvert = \sum_{i=1}^{N}\lvert\mu - \mu_i\rvert$
Maximum	Max	$max(s_i)$
Minimum	Min	$min(s_i)$

All thirteen features are extracted from each of the thirteen signals. Therefore, a total of 169 features are extracted from the acceleration, linear acceleration and tilt angle signals. Waikato Environment for Knowledge Analysis (WEKA) toolkit is used to analyze the features [123]. Features are extracted from the acceleration and linear acceleration data over the sliding window. Najafi et al. investigated the correlation of temporal postural duration with falling risk in elderly people [86], and it is found that the average of postural duration is 2.95s. Therefore, the initial window size is set to 3s (150 samples). We have chosen overlapping factor (of) of 0.5 (75 samples) and expansion factor (ef) of 0.5 (75 samples). The three classifiers are implemented as Decision Tree. Decision Tree is chosen in this study due to its short execution and training time [49]. Furthermore, Decision Tree is found to give the highest levels of classification accuracy according to [90]. ID3 algorithm is used to construct the Decision Tree classifiers [124]. The algorithm determines the threshold value that gives the best separation of samples to effectively distinguish between classes, in this case activities. This is achieved by finding the value of threshold that maximizes the information gain. Given a set of samples, $S = \langle s_1, s_2, s_3, \dots, s_n \rangle$ of attribute (in this case feature) A, the information gain ratio is calculated as follows.

$$G(S, A) = H(S) - \sum_{v \in values(A)} \frac{S_v}{S} H(S_v) \tag{3.11}$$

where $H(S)$ is the entropy of the given samples. S_v is a subset of S for which has value v and $H(S_v)$ is the entropy of S_v. Given a set of classes, $C = \langle c_1, c_2, \dots, c_n \rangle$, entropy is given as follows.

$$H(S) = \sum_{i=1}^{n} -p_i \, log_2 \, p_i \tag{3.12}$$

where p_i is the proportion of S belonging to class c_i. The objective is to find the threshold value that most effectively split samples. This is achieved by finding the value that maximizes the information gain ratio. The implementation of the classifiers are described as follows.

For the Transitional Activity Detector, the signals (acceleration and linear acceleration) are segmented by the fixed initial window size and divided into non-transitional activity and transitional activity signals. The thirteen features are calculated using the segmented signals, and Relief-F is used to select the most relevant features against target activities. Relief-F is chosen because of its speed and simplicity [125]. Then, ID3 algorithm is used to construct the Decision Tree classifier.

For training the Non-transitional (Dynamic/Static) Activity Classifier, all non-transitional activity signals are segmented and divided into walking, standing, sitting, lying face-up and lying face-down classes. The thirteen features are calculated using the fixed 3s signal segments, and Relief-F is used to select the most relevant features against target activities. ID3 algorithm is used to construct the Decision Tree classifier.

For the Transitional Activity Classifier, all transitional activity signals are segmented and divided into stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit and falling. Transitional activity signals are of different durations. Therefore, the size of the window to calculate the features varies from one signal to another. The algorithm expands the window by expansion factor, ef which is 1.5s (75 samples). Hence, the window size to calculate the features are 3s (150 samples), 4.5s (225 samples), 6s (300 samples) etc. depending on the length of the signals. Relief-F is used to select the most relevant features against target activities. Then ID3 algorithm is used to construct the Decision Tree classifier. The features of each classifier for both datasets are given in Table 3.6. Notice that the selected features may be different due to varying sensor location and types of activities.

Table 3.6: Features of the Decision Tree classifier for IELAB and SBHAR datasets.

	IELAB	SBHAR
Transitional Activity Detector	AM A_y	$\|\mu D\|$ A_y
Non-transitional Activity Classifier	Avg LA_{yz}, Avg A_y	Avg LA_{xy}, Avg A_y
Transitional Activity Classifier	μT A_y, Avg A_y	M A_y, Avg A_y

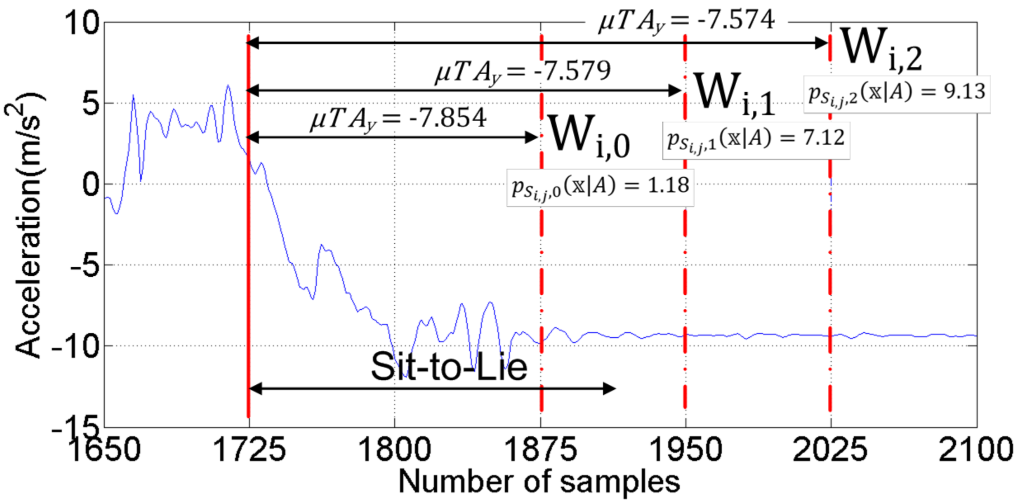


Figure 3.8: The values of μT Ay for each window expansion.

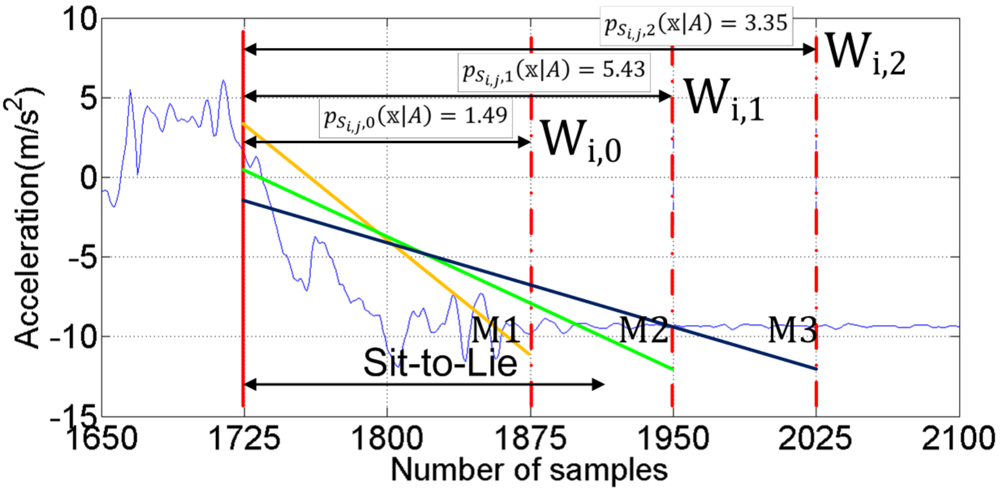


Figure 3.9: The fitted line through acceleration Y signal for each window expansion.

During window expansion process, multivariate Gaussian distribution is used to determine the most effective window expansion based on the probability density function. Two features from Decision Tree classifier (Avg LA_{yz} and Avg A_y) are used to model the distribution. μT A_y is not effective to capture the trend of the activity signal when determining the most effective window expansion. An example of this scenario is shown in Figure 3.8. As can be seen in Figure 3.8, μT A_y values of $W_{i,1}$ (-7.579) and $W_{i,2}$ (-7.574) are almost the same, and the PDF value is increasing due to the flat trend of the signal. As a result, the window continues expanding and the most effective window segmentation cannot be identified. Therefore, M (slope) A_y is selected to model the distribution since it has the highest information gain after $\mu T\ A_y$ during feature selection process. Figure 3.9 shows how the slope of fitted lines can effectively capture the trend of an activity signal. Three lines, M1, M2 and M3 are fitted through the activity signal for each window expansion. As can be seen, the slope of the fitted line is decreasing as more samples (of other activity signal) are segmented by the window and as a result the PDF value is decreasing.

3.6.3 Physical Activity Recognition

We have applied the proposed algorithm to develop a physical activity recognition system. The system is implemented in MATLAB. Gupta and Dallas (later referred to as GD approach) [48] have introduced new features to effectively recognize transitional activities. The features are mean trend, windowed mean difference, variance trend, windowed variance difference, detrended fluctuation analysis coefficient, energy uncorrelated and maximum difference acceleration. Naïve Bayes and k-NN are used to recognize the activities with a fixed window size of 6s. Window size of 6s was used assuming it was long enough to segment transitional activity signals. We implemented the GD approach using the Naïve Bayes classifier since it achieved better accuracy when classifying transitional activities. We have compared their approach with the proposed adaptive sliding window approach in terms of recognition accuracy.

3.7 Results and Discussion

3.7.1 IELAB: Intelligent Environment Lab Dataset

We compute and tabulate the accuracy of the recognition from the values of true positive (TP) and false negative (FN) to evaluate the performance of the proposed approach. Recall or true positive rate is the number of windows that are correctly classified and is given by

$$Recall = (TP/(TP + FN))\times 100 \tag{3.13}$$

In addition to the recall, we calculated the precision and F-score metrics [24] as follow.

$$Precision = TP/(TP + FP) \tag{3.14}$$

$$F - score = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{3.15}$$

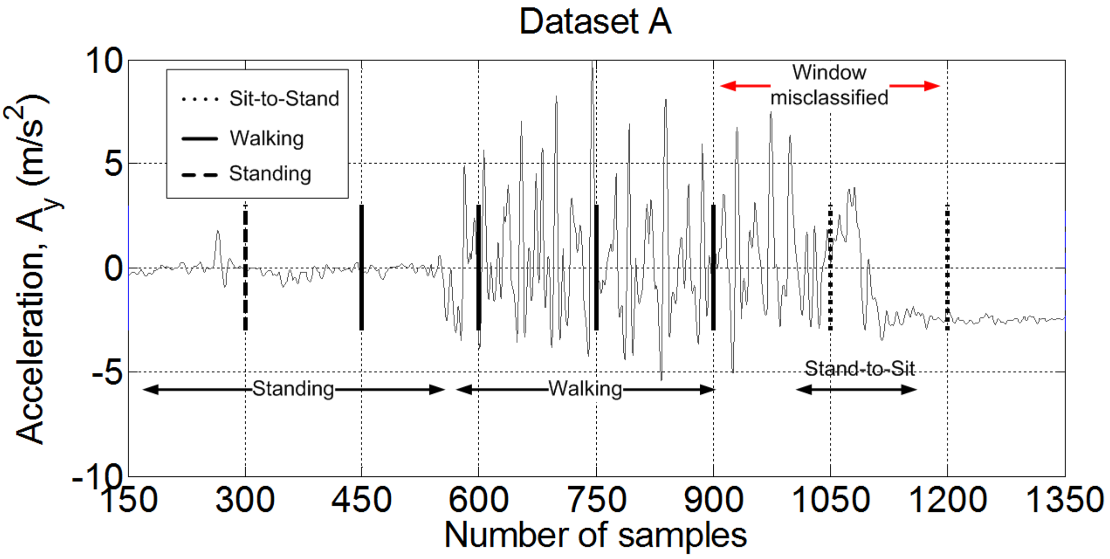
The additional metrics are given in Figure 3.11. Table 3.7 compares the accuracy of activity recognition system using adaptive sliding window (AW) approach against the existing GD approach. Table 3.8 and Table 3.9 show the performance of the approaches by means of confusion matrices. The recognition accuracy of individual activities, transitional activities, non-transitional activities and the overall accuracy are compared and analyzed. In general, GD approach performed reasonably well in classifying most activities and achieved an overall accuracy of 89.9%, which is 3.1% lower than the proposed AW approach. For the AW approach, classification accuracy is over 90% for every individual activity except walking and falling.

Table 3.7: Comparison of accuracy of activity recognition.

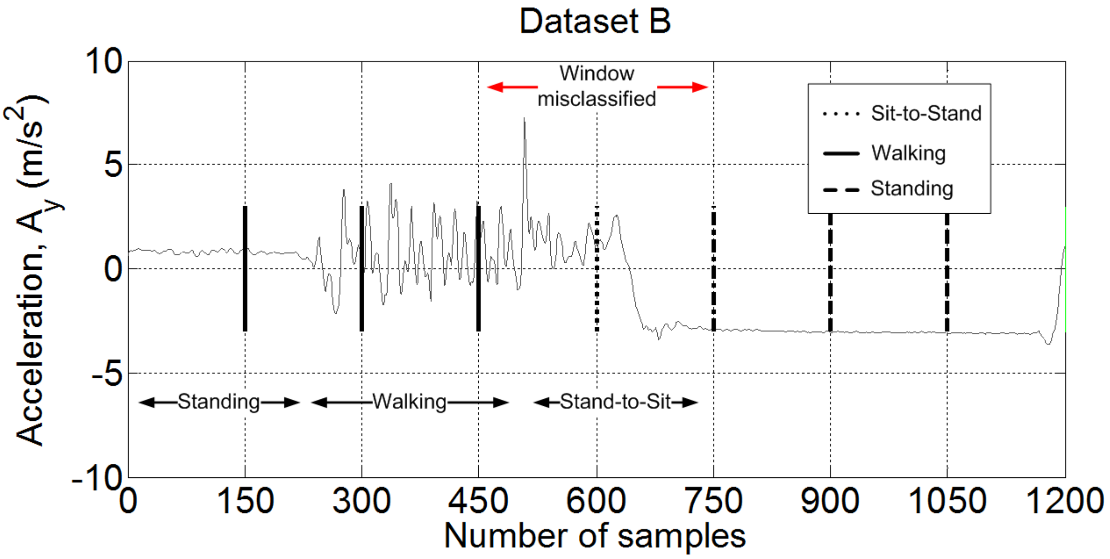
	Recall (Transitional)	Recall (Non-transitional)	Overall Recall
GD Approach	88.7%	90.5%	89.9%
AW Approach	93.6%	92.9%	93.0%

As for classifying transitional activities, GD approach achieved 88.7% accuracy. Majority of the transitional activities can be classified with 91.2%-97.6% accuracy range with the exception of stand-to-sit which was poorly classified with only 82.7% accuracy. This reflects the fact that GD approach cannot handle activity signal with varying length as shown in Figure 3.10. The length of stand-to-sit signal in Figure 3.10(a) is about 2s (100 samples) while stand-to-sit signal in Figure 3.10(b) is about 3.5s (175 samples). As can be seen in Figure 3.10(a), walking signal occupies almost half of the window, which leads to misclassification. Conversely, AW approach achieved recognition accuracy of 93.6% in transitional activities. It successfully detected the activities and adapted the window size to accommodate activity signals of varying lengths. However, GD approach achieved slightly higher accuracy, about 1.9% and 2.7%, in classifying Lie-to-Sit and Sit-to-Lie respectively than AW approach. This is because, in the experiments, AW approach failed to detect transitional activity signal at the beginning and hence adaptive sliding window is not applied. As a result, the window being processed is wrongly classified. It is also observed that in a few experiments, the algorithm failed to determine the best window segmentation due to the over-expansion of the window, which leads to misclassification of the activity. Out of 210 window expansion operations (adaptive window mode), only in 9 cases the windows are not correctly expanded.

In total, 6.4% of transitional activity windows were misclassified by AW approach while GD approach misclassified 11.3% of transitional activity windows. This demonstrates that adaptive sliding window segmentation is significantly more effective in classifying transitional activities. Based on our previous observation, activities are very often misclassified during activity transitions due to ambiguous signal characteristics caused by some minor motion behavior change. These can be clearly observed in dynamic behaviors such as walking and falling, as well as standing which is the preceding activity of walking. In AW approach, these activities (walking, falling and standing) are classified relatively poor with only an average of 91.1% accuracy.



(a)



(b)

Figure 3.10: Stand-to-sit with varying length.

Table 3.8: Confusion matrix of GD approach.

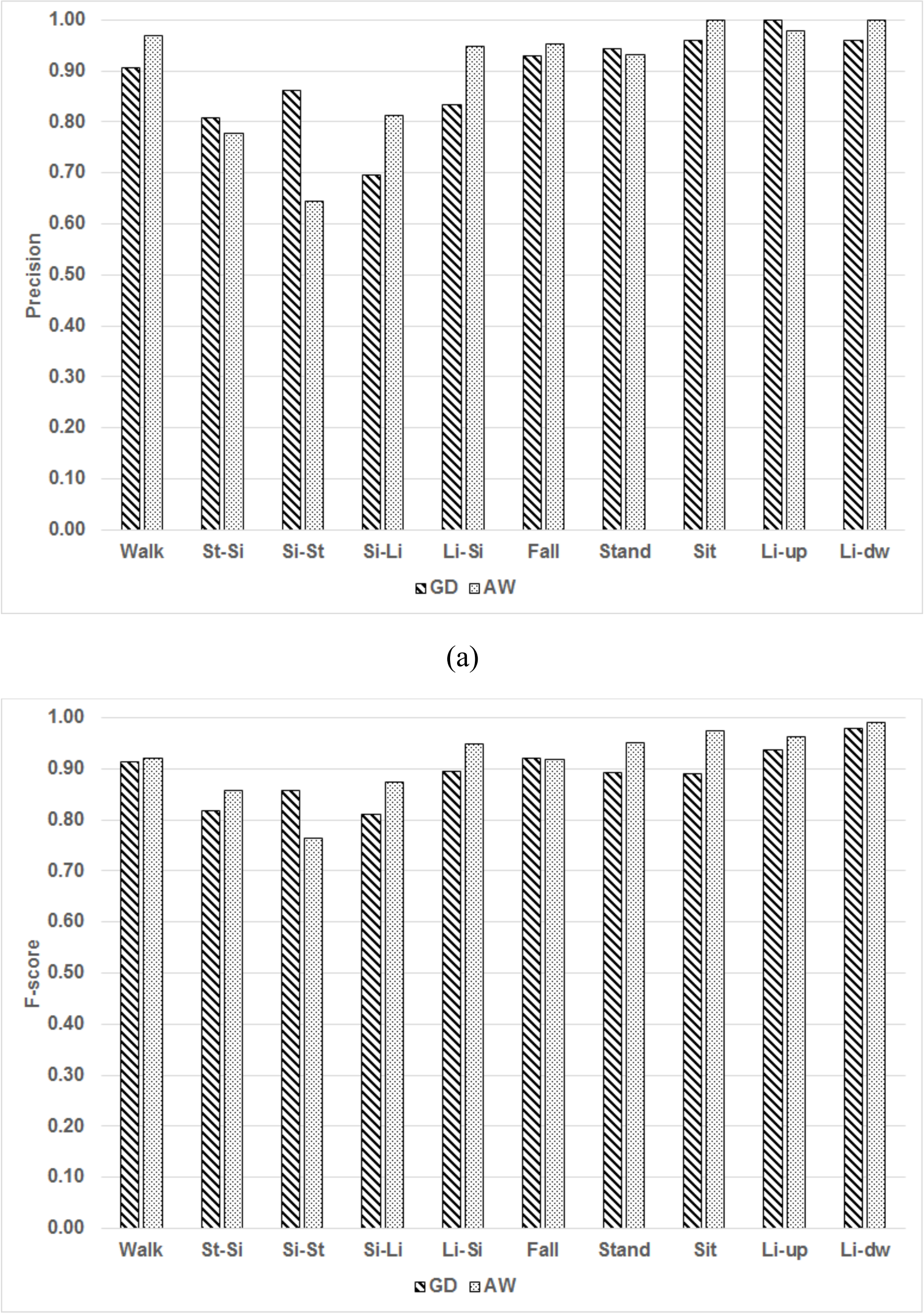
	Walk	St-Si	Si-St	Si-Li	Li-Si	Fall	Stand	Sit	Lie-Up	Lie-Dw	Count	Recall
Walk	**183**	10	3	0	0	3	0	0	0	0	**199**	92.0%
St-Si[a]	3	**67**	0	11	0	0	0	0	0	0	**81**	82.7%
Si-St[b]	8	0	**69**	0	3	1	0	0	0	0	**81**	85.2%
Si-Li[c]	0	0	0	**41**	1	0	0	0	0	0	**42**	97.6%
Li-Si[d]	0	0	1	0	**30**	0	0	0	0	0	**31**	96.8%
Fall	1	0	0	0	0	**52**	0	0	0	4	**57**	91.2%
Stand	7	0	2	0	0	0	**50**	0	0	0	**59**	84.7%
Sit	0	6	5	1	0	0	3	**73**	0	0	**88**	83.0%
Lie-Up[e]	0	0	0	6	2	0	0	3	**83**	0	**94**	88.3%
Lie-Dw[f]	0	0	0	0	0	0	0	0	0	**98**	**98**	100%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-sit
e. Lying face-up
f. Lying face-down

Table 3.9: Confusion matrix of AW approach.

	Walk	St-Si	Si-St	Si-Li	Li-Si	Fall	Stand	Sit	Lie-Up	Lie-Dw	Count	Recall
Walk	**446**	21	39	0	0	0	4	0	0	0	**510**	87.5%
St-Si[a]	0	**84**	1	1	0	2	0	0	0	0	**88**	95.5%
Si-St[b]	2	1	**83**	0	2	0	0	0	0	0	**88**	94.3%
Si-Li[c]	0	0	0	**56**	0	0	0	0	3	0	**59**	94.9%
Li-Si[d]	0	0	0	0	**37**	0	0	0	2	0	**39**	94.9%
Fall	4	0	4	0	0	**62**	0	0	0	0	**70**	88.6%
Stand	3	1	0	0	0	0	**138**	0	0	0	**142**	97.2%
Sit	2	1	2	0	0	0	6	**215**	0	0	**226**	95.1%
Lie-Up[e]	0	0	0	12	0	0	0	0	**223**	0	**235**	94.9%
Lie-Dw[f]	3	0	0	0	0	1	0	0	0	**212**	**216**	98.1%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-sit
e. Lying face-up
f. Lying face-down



(a)

(b)

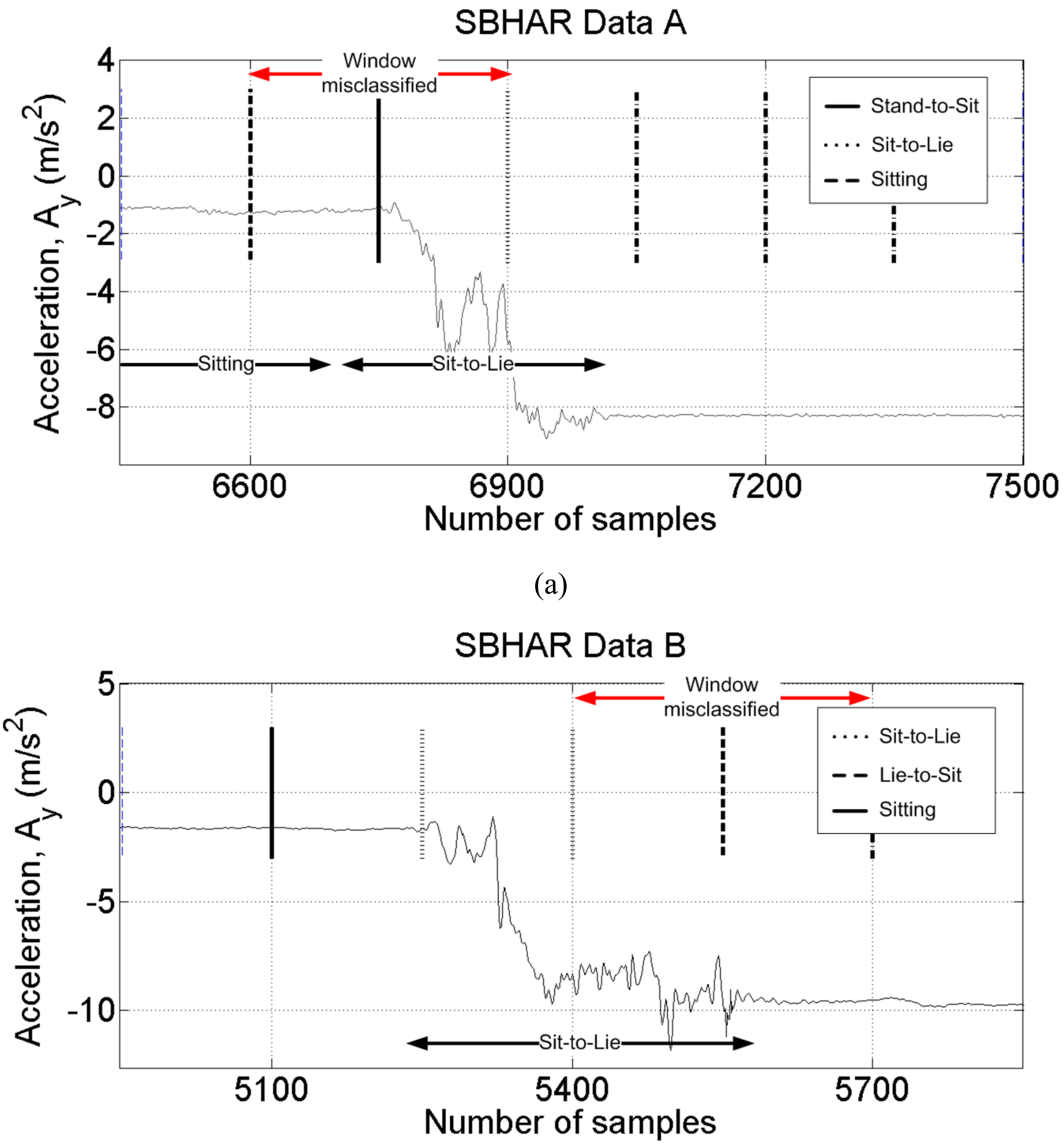
Figure 3.11: Comparison of (a) precision and (b) F-score between GD and AW for IELAB dataset.

3.7.2 SBHAR: Smartphone-based HAR Dataset

Table 3.10 compares the accuracy of activity recognition system using GD and AW on the dataset. The recognition accuracy of individual activities, transitional activities, non-transitional activities and the overall accuracy are compared and analyzed. GD approach performed reasonably well in classifying the activities and achieved overall accuracy of 91.9% which is 3.8% lower than AW. For AW approach, the classification accuracies for all activities are above 90% except stand-to-sit and sit-to-lie, achieving overall recognition accuracy of 95.7%. As for transitional activities, AW performed better in classifying all transitional activities than GD approach. Figure 3.12 illustrates examples of window misclassifications by GD approach. In Figure 3.12(a), the window (sit-to-lie) is misclassified as stand-to-sit because the window contains large portion of sitting signal. In Figure 3.12(b), the length of sit-to-lie signal is about 7s (350 samples) while the window size is 6s. As can be seen in the figure, the signal is not completely contained in the window, which leads to misclassification of the subsequent window. This reflects the fact that GD approach is not effective in classifying activity signal with varying length. Table 3.11 and Table 3.12 are the confusion matrices of GD and AW approaches. However, GD approach performed slightly better in classifying Lie-to-Sit than AW approach, achieving slightly higher accuracy about 0.2%. This is because, in few experiments, the algorithm failed to determine the best window segmentation due to the under-expansion of the window, which leads to misclassification of the activity. Out of 240 window expansion operations (adaptive window mode), only in 21 cases the windows are not correctly expanded. Overall, AW achieved recognition accuracy of 90.3% which is 2.1% higher than GD. It is also observed that, misclassification often occurred when classifying sitting signals due to the pattern of the signals' is similar to standing and lying face-up. This is shown in Table 3.11 and Table 3.12, whereby the percentages of sitting misclassified as standing or lying-face-up are 6.4% and 7.1% for AW and GD respectively. Figure 3.13 shows the comparison of precision and F-score between GD and AW.

Table 3.10: Comparison of accuracy of activity recognition.

	Recall (Transitional)	Recall (Non-transitional)	Overall Recall
GD Approach	88.2%	92.4%	91.9%
AW Approach	90.3%	96.1%	95.7%



(a)

(b)

Figure 3.12: Examples of window misclassification by GD approach.

Table 3.11: Confusion matrix of GD approach.

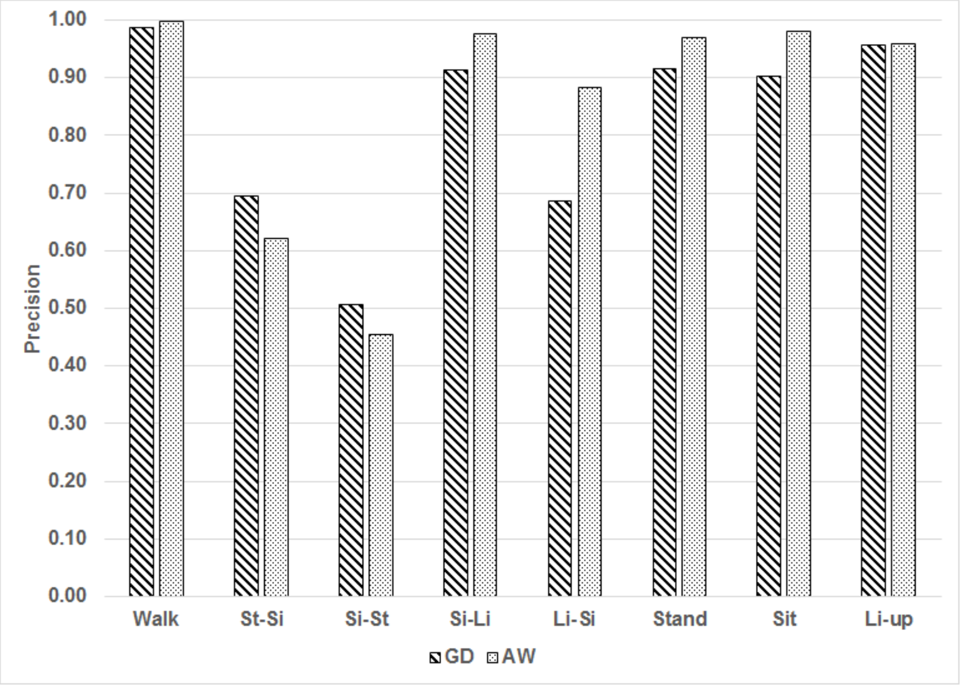
	Walk	St-Si	Si-St	Si-Li	Li-Si	Stand	Sit	Lie	Count	Recall
Walk	**1718**	19	58	0	1	15	1	0	**1812**	94.8%
St-Si[a]	5	**87**	1	6	0	0	0	0	**99**	87.9%
Si-St[b]	1	2	**80**	0	10	0	0	0	**93**	86.0%
Si-Li[c]	0	8	0	**94**	2	0	0	2	**106**	88.7%
Li-Si[d]	1	0	7	0	**90**	0	0	2	**100**	90.0%
Stand	12	0	8	0	0	**362**	3	0	**385**	94.0%
Sit	5	7	2	3	8	11	**276**	12	**324**	85.2%
Lie	0	2	2	0	20	7	26	**351**	**408**	86.0%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-Sit

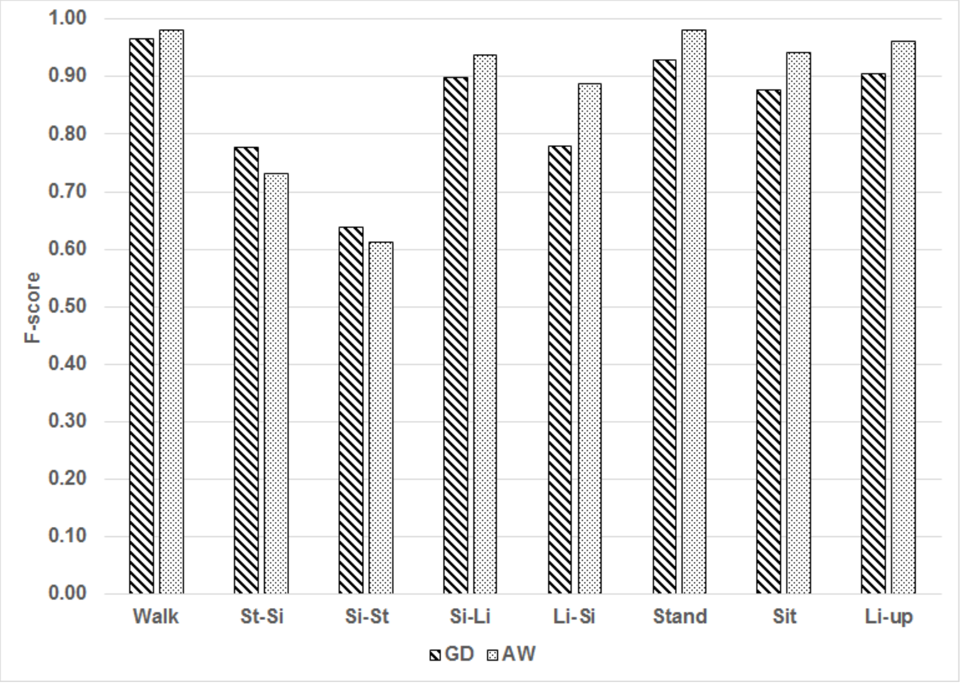
Table 3.12: Confusion matrix of AW approach.

	Walk	St-Si	Si-St	Si-Li	Li-Si	Stand	Sit	Lie	Count	Recall
Walk	**3341**	48	68	0	0	1	11	0	**3469**	96.3%
St-Si[a]	0	**116**	13	0	0	1	0	0	**130**	89.2%
Si-St[b]	3	1	**92**	0	3	0	0	0	**99**	92.9%
Si-Li[c]	0	10	2	**120**	0	0	0	1	**133**	90.2%
Li-Si[d]	0	2	9	0	**99**	0	0	1	**111**	89.2%
Stand	2	1	4	0	0	**1004**	1	0	**1012**	99.2%
Sit	1	6	9	2	5	12	**684**	36	**755**	90.6%
Lie	1	3	5	1	5	16	1	**897**	**929**	96.6%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-Sit



(a)



(b)

Figure 3.13: Comparison of (a) precision and (b) F-score between GD and AW for SBHAR dataset.

3.8 Conclusions

We proposed a novel adaptive sliding window technique for segmentation of activity signal acquired from a tri-axial accelerometer to overcome the limitations of fixed-size sliding window used in existing works. In the proposed approach the window size is adaptively adjusted and increased gradually, starting with initial size, based on signal information to achieve the more effective window segmentation compared to fixed-size window approaches. In this study, we demonstrated the performance of the approach on two datasets in which one of them is a public dataset. The employed datasets were generated by different subjects with different styles and pace. It was observed that the system can classify different activities performed by different subjects with excellent accuracy. The results showed that the proposed approach effectively segments activity signals resulting in better classification accuracy in a wide range of activities. The approach specifies small initial window size, which is able to segment dynamic and static activity signals, and expand window size dynamically to accommodate transitional activity signals which is longer than the current window size. The approach determines the optimum window size automatically as the signal is being evaluated. As a result, the window contains the right information when performing classification. The results showed that AW achieved 93.0% overall accuracy, which is 3.1% better than existing GD approach. AW achieved an overall accuracy of 95.7%, which is 3.8% better than GD approach when tested on SBHAR dataset.

4

Physical Activity Transition Model

4.1 Introduction

A sequence of physical activities is characterized by the temporal dependence of the activities. For example, the possible activities after the standing position are standing, walking and stand-to-sit. However, the Decision Tree is not capable of modeling the temporal dependence of a sequence of activities. In this chapter, the transition model of physical activity in the form of an activity transition diagram (ATD) is proposed. The role of the transition diagram in the classification system is explained. The transition model is integrated into the activity classification algorithm resulting in a higher recognition accuracy.

4.2 Related Works

The finite state machine has been used for activity recognition. Kerr et al. [126] proposed a novel approach to activity recognition by extracting the signatures that are shared between all training instances of an activity. Using the signatures, a generic finite state machine (FSM) is generated for activity recognition. In [127], a hand recognition system is proposed by using FSMs. The FSMs are developed according to the specific hand motion pattern. Similar work is found in [128] whereby the states represent the action patterns. Langensiepen et al. [129] proposed a fuzzy finite state machines to model the activities in an office. The model incorporates ambient sensor data as the input and a set of fuzzy rules to define the state transitions. In [130], a fuzzy finite state machine is improved by learning the fuzzy rule set that best suits the training data.

A stochastic state machine such as hidden Markov model (HMM) has been used model action sequences for activity recognition [131]. Lee and Cho [132] proposed a two-layer HMM to recognize short-term and long-term activity in real time. The first layer of the HMM recognizes the short-term activity while the second layer recognizes the activities with longer time period. Similar work is found in [133], whereby the first layer of the HMM recognizes the activity class and the actual activity is recognized by the second layer of the HMM. Kozina et al. [134] proposed a new architecture for activity recognition to recognize ADL, exercise activities and seven transitional activities using three accelerometers. The architecture consists of three layers, in which knowledge-based and machine learning classifiers are implemented in the first two layers. The outputs of the classifiers are aggregated and fed to the top layer to correct the final decision of the recognized activity using Hidden Markov Model by filtering the spurious or ambiguous transitions between activities. Unlike existing works, this dissertation utilizes the transition diagram to enhance the performance of the activity recognition. The model represents the possible transitions of the activities providing support to the classification system in order to improve the recognition accuracy.

4.3 Integration of Activity Recognition with Transition Model

In order to make classification of activity more robust, a further enhancement of activity recognition is proposed by the integration of a transition model of physical activities represented by an ATD, in the activity recognition system as shown in Figure 4.1. The ATD is a part of the state validator of the activity recognition system. The role of the state validator is to provide feedbacks to the system in order to improve the accuracy of classification. The state validator consists of invalid activity transition detector, three state buffers and ATD. The function of invalid activity transition detector is to check the validity of an activity transition. State buffers are to store the three consecutively classified activities, the current one and two immediately preceding ones. The classification system provides the recognized activity for every classified window to the state validator. Each time the current activity is updated, the activity transition validity is checked by the invalid activity transition detector. In the case of an invalid activity transition, multivariate Gaussian distribution is applied to re-classify the signal. Next possible activities are acquired from ATD to aid the re-classification process.

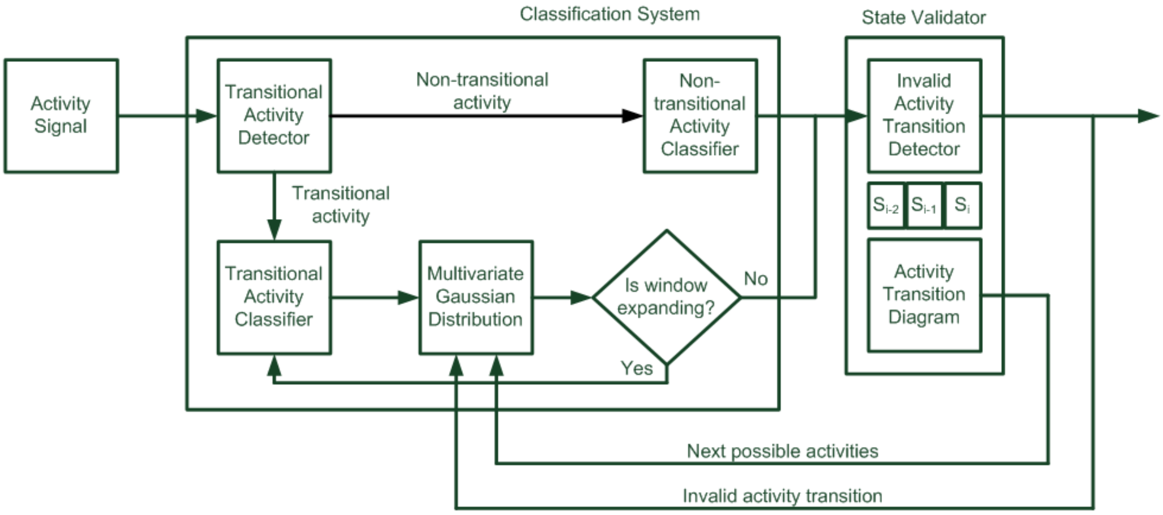

Classification System
State Validator
Activity Signal
Transitional Activity Detector
Non-transitional activity
Non-transitional Activity Classifier
Invalid Activity Transition Detector
Transitional activity
S_{i-2}
S_{i-1}
S_i
Transitional Activity Classifier
Multivariate Gaussian Distribution
Is window expanding?
No
Yes
Activity Transition Diagram
Next possible activities
Invalid activity transition

Figure 4.1: The block diagram of physical activity recognition system.

4.4 Activity Transition Diagram

The physical activity transition model in the form of activity transition diagram is proposed as illustrated in Figure 4.2. All activities are represented as states, and the transitions define conditions under which we consider changes of the states. These conditions are not depicted in the figure and will be explained in the following text. The state transitions reflect the possible transitions between activities. For example, from standing position, a person can perform either walking or sitting. There are two possible scenarios of invalid activity transition which can be detected by the state validator as illustrated in Figure 4.3. The shaded windows are misclassified. The first scenario involves the occurrence of invalid activity transition due to misclassification of current window, $W_{i,k}$. As shown in Figure 4.3, activity transition from walking activity ($W_{i-1,k}$) to sit-to-stand activity ($W_{i,k}$) is an invalid transition, which will be detected by the state validator. In the second scenario, the current window, $W_{i,k}$ is correctly classified, but violation of activity transition is caused due to misclassification of the previous window, $W_{i-1,k}$. As shown in Figure 4.3, walking activity ($W_{i,k}$) is correctly classified but an invalid activity transition is detected (falling to walking) due to misclassification on $W_{i-1,k}$. However, no invalid activity transition was detected from previous window because walking to falling is a valid transition. Note that, the state validator can only detect invalid activity transition by checking the activity transition from the previous window to current window. But it does not know which window is being misclassified.

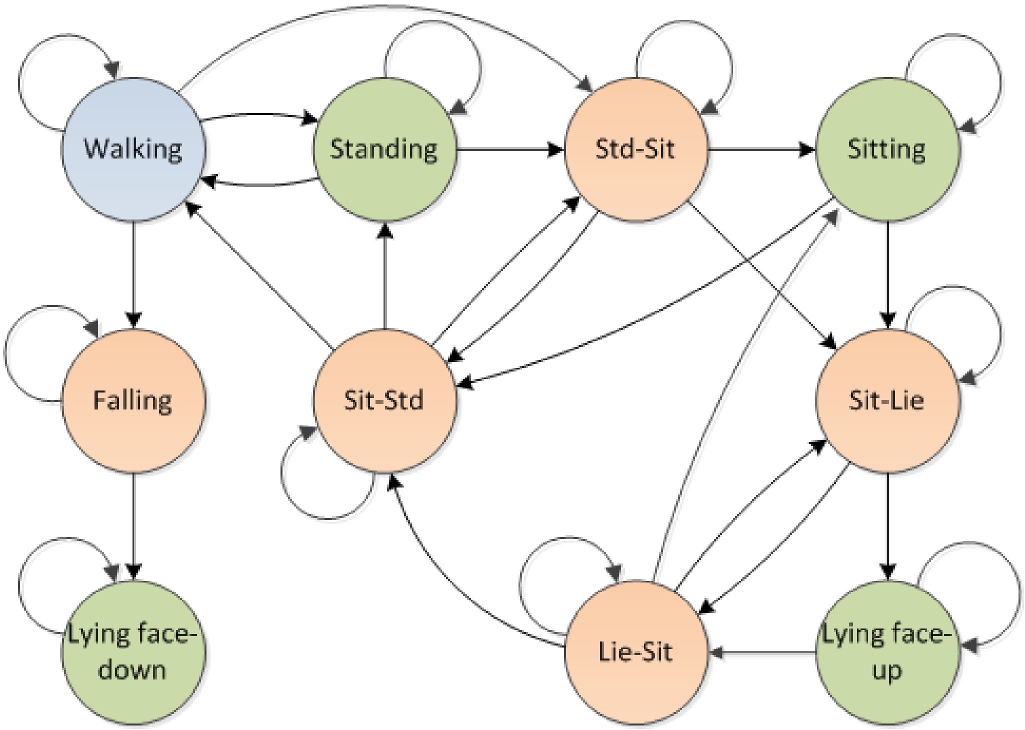


Figure 4.2: Activity transition diagram.

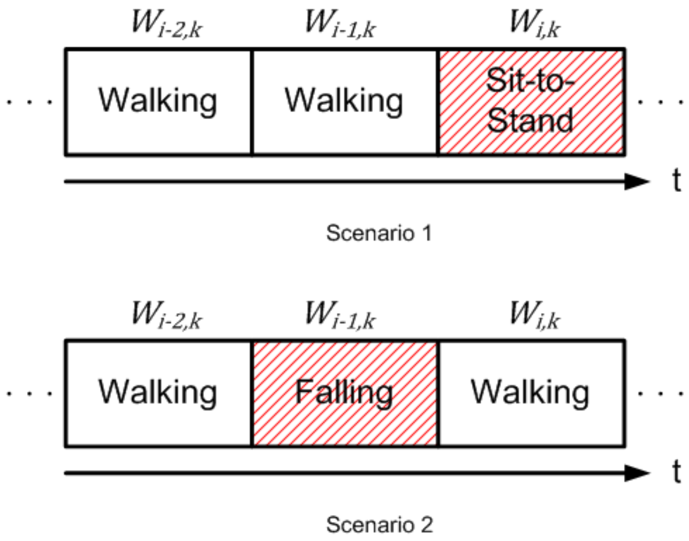


Figure 4.3: Invalid activity transition scenarios.

Listing 4.1: Re-classification of windows

$W_{x,k}$ is a window after k expansion being classified where $x = i - 1, i$, with $W_{i,k}$ representing the current window.

A is a list of N possible valid activities, with a_j represents a single activity where $j = 1 \dots N$

p_{max} is the maximum probability density function value for a possible activity to determine the activity (state) of the window.

```
if invalid activity transition is detected then
    for $x = i - 1$ until $i$ do
        calculate features of $W_{x,k}$
        $A$ = get next possible activities of $A_{W_{x-1,k}}$
        for all $a_j$ in $A$ do
            $p_{a_j}$ = calculate probability density function of $a_j$
            if $p_{a_j} > p_{max}$ then
                $p_{max} = p_{a_j}$
                $A_{W_{x,k}} = a_j$
            end if
        end for
    end for
end if
```

In the case of invalid activity transition, the state validator will notify the classification system to re-perform classification on the windows. The re-classification algorithm is given in Listing 4.1. The algorithm begins by calculating features of window $W_{i-1,k}$ as defined by line 3. Then, using the ATD, all possible next activities of the previous window (i.e. $W_{i-2,k}$) are acquired for reclassifying $W_{i-1,k}$ as defined by line 4. Using scenario 2 as an example, the algorithm will acquire next possible activities of window $W_{i-2,k}$ (walking) which are walking, falling, standing and stand-to-sit. Next, for each next possible activity, the probability density functions of extracted features are calculated using the multivariate Gaussian distribution (lines 5-6) explained in Section 3.5. Among the possible next activities, the one that has the highest probability density function value will now be assigned as the state of window $W_{i-1,k}$. Then, the same process is repeated for window $W_{i,k}$. Notice that, the algorithm performs re-classification on the current window and its previous window to handle both invalid transition scenarios without having any prior knowledge on which one is misclassified. In order not to invalidate all previous activity transitions, the algorithm acquires only valid next activities of the previous window to perform re-classification.

4.5 Results and Discussion

We have compared the adaptive sliding window with and without ATD in terms of recognition accuracy to investigate the effectiveness of the integration of ATD in the classification algorithm. The performance evaluation is performed using IELAB dataset and the SBHAR dataset.

4.5.1 IELAB: Intelligent Environment Lab Dataset

We compute and tabulate the (recall) accuracy of the recognition from the values of true positive (TP) and false negative (FN) to evaluate the performance of the proposed approach. We also calculated the precision and F-score metrics as illustrated in Figure 4.4. Table 4.1 compares the accuracy of activity recognition system using adaptive sliding window approach with ATD (AW-TD) against the AW approach. Table 4.2 and Table 4.3 show the performance of the approaches by means of confusion matrices. The recognition accuracy of individual activities, transitional activities, non-transitional activities and the overall accuracy are compared and analyzed. There is an additional improvement to the overall recognition accuracy when the state validator is integrated into the activity recognition system and acts as a feedback. In general, classification accuracy of AW-TD approach is over 90% for every individual activity. For AW approach, although the overall accuracy is only marginally lower (2.4% less) compared with AW-TD, the standard deviation in classification accuracy is higher, which are 3.4% for AW and 2.1% for AW-TD. This indicates AW approach is less accurate in some activities while AW-TD provides good classification in every case.

The recognition accuracy of walking has seen significant improvement whereby more than half of the misclassification have been corrected. From the observation, the state validator successfully detected the invalid transition when a window is misclassified and the window classification is corrected by the classification system. The overall recognition accuracy is also improved with lower standard deviation, in which recognition accuracy for all activities are well above 90%. However, recognition accuracy of sitting and standing has been decreased by 1.3% and 0.7% respectively. This is because the windows were incorrectly re-classified by multivariate Gaussian distribution.

Table 4.1: Comparison of accuracy of activity recognition.

	Recall (Transitional)	Recall (Non-transitional)	Overall Recall
AW Approach	93.6%	92.9%	93.0%
AW-TD Approach	95.3%	95.4%	95.4%

Table 4.2: Confusion matrix of AW approach.

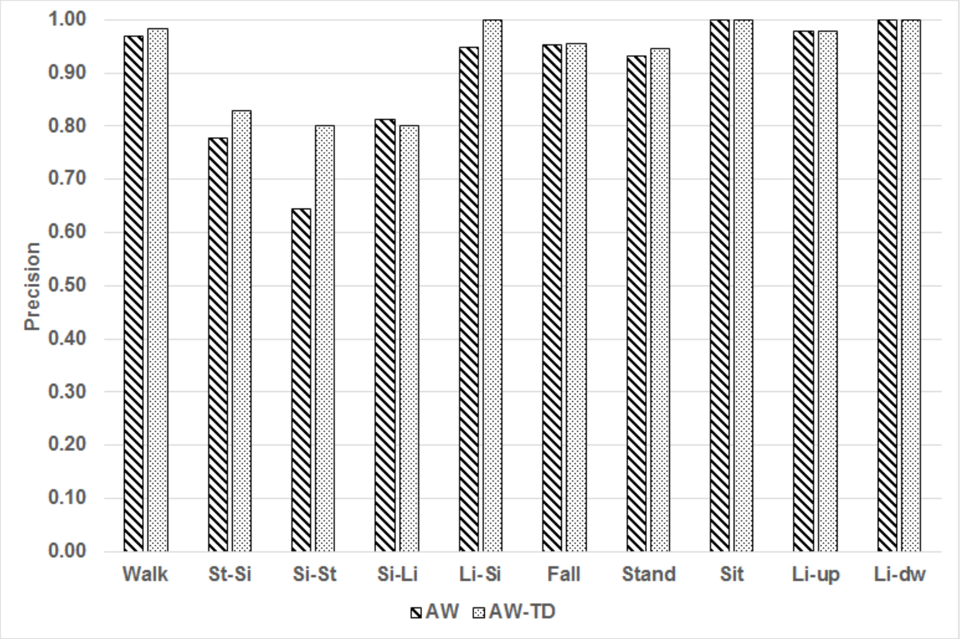
	Walk	St-Si	Si-St	Si-Li	Li-Si	Fall	Stand	Sit	Lie-Up	Lie-Dw	Count	Recall
Walk	**446**	21	39	0	0	0	4	0	0	0	**510**	87.5%
St-Si[a]	0	**84**	1	1	0	2	0	0	0	0	**88**	95.5%
Si-St[b]	2	1	**83**	0	2	0	0	0	0	0	**88**	94.3%
Si-Li[c]	0	0	0	**56**	0	0	0	0	3	0	**59**	94.9%
Li-Si[d]	0	0	0	0	**37**	0	0	0	2	0	**39**	94.9%
Fall	4	0	4	0	0	**62**	0	0	0	0	**70**	88.6%
Stand	3	1	0	0	0	0	**138**	0	0	0	**142**	97.2%
Sit	2	1	2	0	0	0	6	**215**	0	0	**226**	95.1%
Lie-Up[e]	0	0	0	12	0	0	0	0	**223**	0	**235**	94.9%
Lie-Dw[f]	3	0	0	0	0	1	0	0	0	**212**	**216**	98.1%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-sit
e. Lying face-up
f. Lying face-down

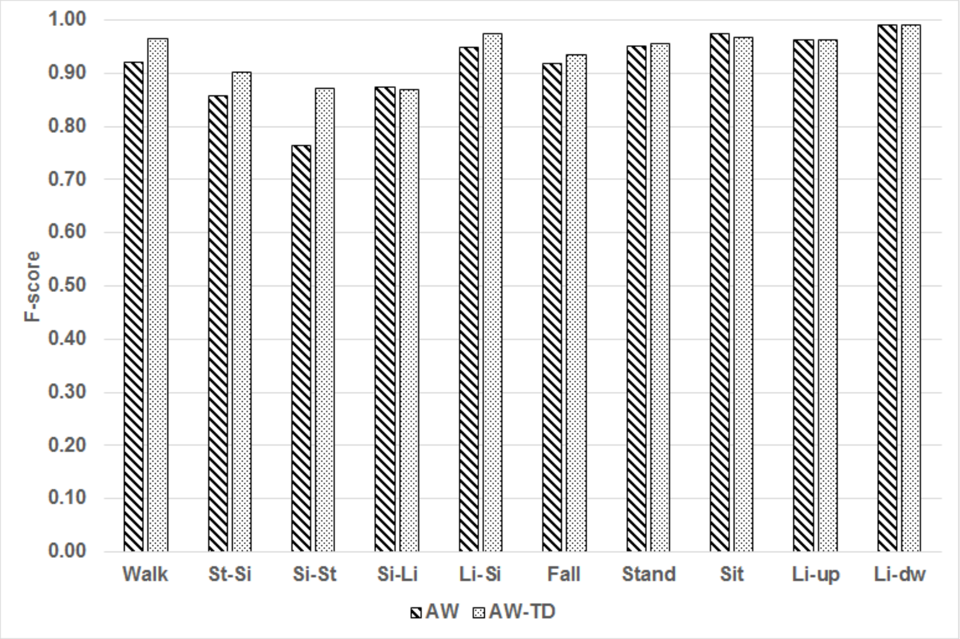
Table 4.3: Confusion matrix of AW-TD approach.

	Walk	St-Si	Si-St	Si-Li	Li-Si	Fall	Stand	Sit	Lie-Up	Lie-Dw	Count	Recall
Walk	**484**	10	11	0	0	1	4	0	0	0	**510**	94.9%
St-Si[a]	0	**87**	0	1	0	0	0	0	0	0	**88**	98.9%
Si-St[b]	2	1	**84**	1	0	0	0	0	0	0	**88**	95.5%
Si-Li[c]	0	0	0	**56**	0	0	0	0	3	0	**59**	94.9%
Li-Si[d]	0	0	0	0	**37**	0	0	0	2	0	**39**	94.9%
Fall	4	2	0	0	0	**64**	0	0	0	0	**70**	91.4%
Stand	2	3	0	0	0	0	**137**	0	0	0	**142**	96.5%
Sit	0	2	8	0	0	0	4	**212**	0	0	**226**	93.8%
Lie-Up[e]	0	0	0	12	0	0	0	0	**223**	0	**235**	94.9%
Lie-Dw[f]	0	0	2	0	0	2	0	0	0	**212**	**216**	98.1%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-Sit
e. Lying face-up
f. Lying face-down



(a)



(b)

Figure 4.4: Comparison of (a) precision and (b) F-score between AW and AW-TD for IELAB dataset.

4.5.2 SBHAR: Smartphone-based HAR Dataset

Table 4.4 compares the accuracy of activity recognition system using AW-TD and AW on the dataset. The recognition accuracy of individual activities, transitional activities, non-transitional activities and the overall accuracy are compared and analyzed. In general, the overall recognition accuracy is improved when state validator is integrated into the activity recognition system. For the AW-TD, the classification accuracies for all activities are above 90%, achieving overall recognition accuracy of 96.5% while AW approach achieved an overall recognition accuracy of 95.7%. This can be seen by the lower standard deviation for AW-TD (2.7%) compared to AW (3.9). However, the recognition accuracy for standing slightly decreased by 0.4%. This is because, the windows were incorrectly re-classified by multivariate Gaussian distribution. As for transitional activities, AW-TD performed better in classifying all transitional activities than AW approach, achieving recognition accuracy of 95.1% which is 4.8% higher than AW. Table 4.5 and Table 4.6 are the confusion matrices of AW and AW-TD approaches. Figure 4.5 shows the comparison of precision and F-score between AW and AW-TD.

Table 4.4: Comparison of accuracy of activity recognition.

	Recall (Transitional)	Recall (Non-transitional)	Overall Recall
AW Approach	90.3%	96.1%	95.7%
AW-TD Approach	95.1%	96.6%	96.5%

Table 4.5: Confusion matrix of AW approach.

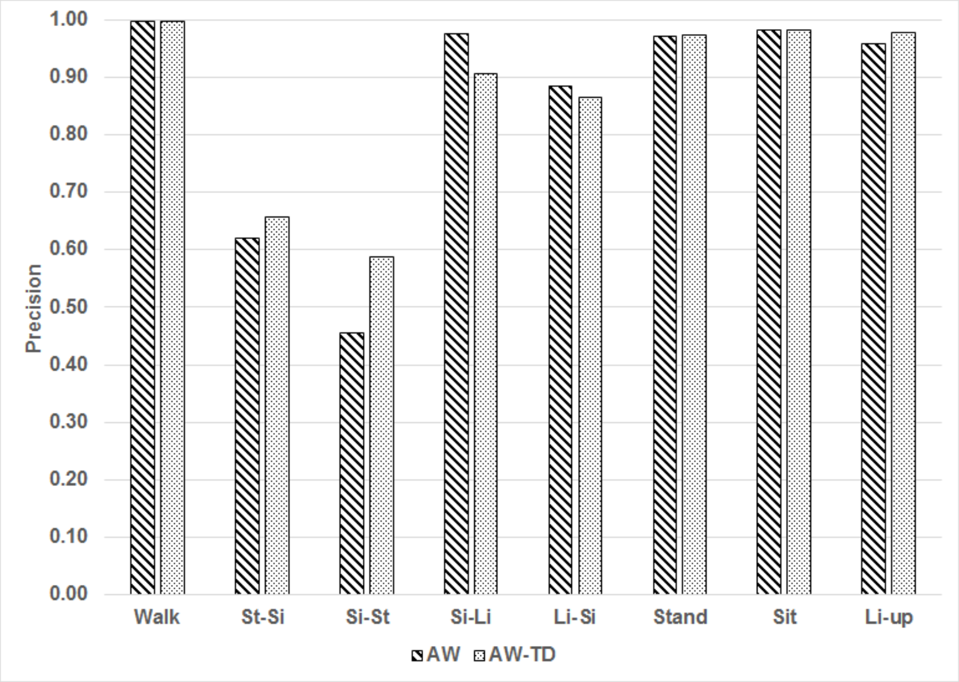
	Walk	St-Si	Si-St	Si-Li	Li-Si	Stand	Sit	Lie	Count	Recall
Walk	**3341**	48	68	0	0	1	11	0	**3469**	96.3%
St-Si[a]	0	**116**	13	0	0	1	0	0	**130**	89.2%
Si-St[b]	3	1	**92**	0	3	0	0	0	**99**	92.9%
Si-Li[c]	0	10	2	**120**	0	0	0	1	**133**	90.2%
Li-Si[d]	0	2	9	0	**99**	0	0	1	**111**	89.2%
Stand	2	1	4	0	0	**1004**	1	0	**1012**	99.2%
Sit	1	6	9	2	5	12	**684**	36	**755**	90.6%
Lie	1	3	5	1	5	16	1	**897**	**929**	96.6%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-Sit

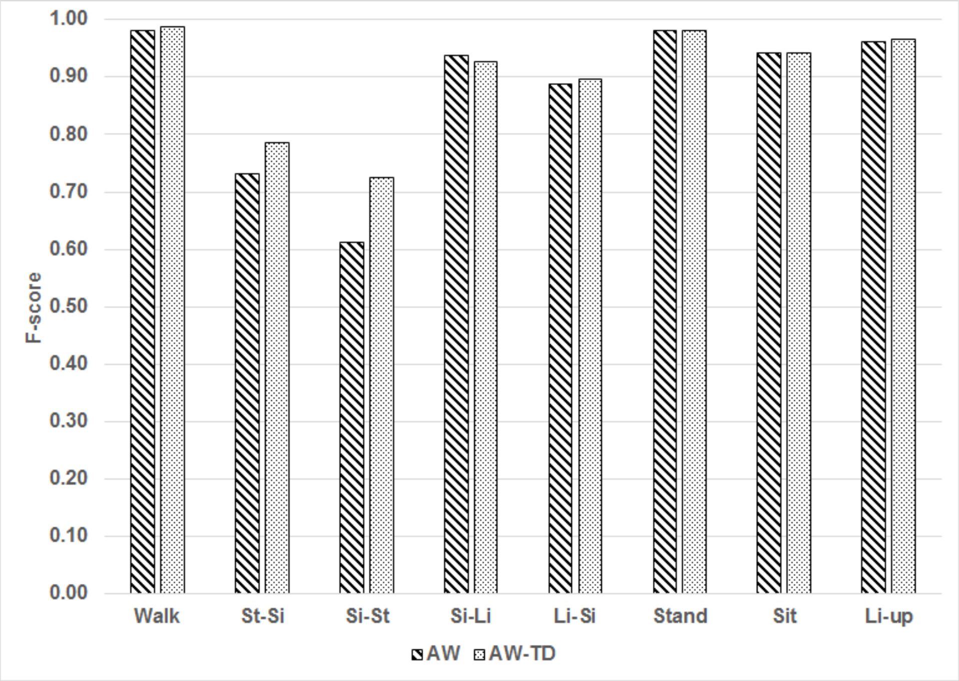
Table 4.6: Confusion matrix of AW-TD approach.

	Walk	St-Si	Si-St	Si-Li	Li-Si	Stand	Sit	Lie	Count	Recall
Walk	**3387**	40	41	0	0	1	0	0	**3469**	97.6%
St-Si[a]	1	**127**	0	1	0	1	0	0	**130**	97.7%
Si-St[b]	1	1	**94**	0	3	0	0	0	**99**	94.9%
Si-Li[c]	0	4	2	**126**	0	0	0	1	**133**	94.7%
Li-Si[d]	0	2	5	0	**103**	0	1	0	**111**	92.8%
Stand	4	3	3	0	0	**1000**	2	0	**1012**	98.8%
Sit	3	12	11	8	8	9	**685**	19	**755**	90.7%
Lie	1	4	4	4	5	16	10	**885**	**929**	95.3%

a. Stand-to-sit
b. Sit-to-stand
c. Sit-to-lie
d. Lie-to-Sit



(a)



(b)

Figure 4.5: Comparison of (a) precision and (b) F-score between AW and AW-TD for SBHAR dataset.

4.6 Conclusions

We propose an activity transition diagram to model the temporal dependence of the physical activities. The activity transition diagram is a part of the state validator of the activity recognition system. The state validator performs validation of activity transition for every window classification based on the proposed activity transition model and notifies the classification system to re-perform classification in the case an invalid transition is detected. In this study, we demonstrated the performance of the approach on two datasets in which one of them is a public dataset. The employed datasets were generated by different subjects with different styles and pace. The results showed that AW-TD achieved 95.4% overall accuracy, which is 2.4% better than AW approach. The standard deviations of the accuracy are 2.12% and 3.40% for AW-TD and AW respectively. AW-TD achieved an overall accuracy of 96.5%, which is 0.8% better than AW approach when tested on SBHAR dataset. The standard deviation of the accuracy are 2.71% and 3.87%

5

Ontological Reasoning with Uncertainty for Activity Recognition

5.1 Introduction

This chapter introduces the methodology of the ADL (later referred to as activity) recognition or dense sensing-based activity recognition. The approach is based on the aggregation of context information from diverse sources which contain some knowledge about the events in the environment. As such, it is assumed that activities can be recognized through the inference of user-object interaction and location of the user. Activity recognition in a smart environment still faces a number of challenges. Firstly, the activities performed by persons depend on their habits and lifestyle and hence they are carried out in different sequences and with different durations. Although there exist correlations among some activities, there is no strict pattern in a sequence of activities. Secondly, multimodal sensors embedded in smart home environment generate heterogeneous data that varies in terms of formats, sensing rates and semantics. Furthermore, a fusion of these sensor data is required to establish the context of the activity being carried out. Finally, uncertainties are always present in ambient intelligence environment [73]. For instance, sensor data are inherently noisy. This can be due to sensor errors (run out of batteries, imprecise outputs, missing activations etc.), communication failures and variability in human activities. These issues may significantly influence the accuracy of activity recognition.

Different approaches have been proposed by researchers for activity modeling and recognition. They can be classified into data-driven and knowledge-driven approaches [35], [50]. Data-driven approaches use learning-based techniques with robust activity models that extract specific features from sensor data. The main advantage of learning-based techniques is the ability to handle uncertainty and noise. Previous research works have shown that they are able to obtain high accuracy rate of activity recognition [56]–[58], [135]–[139]. Furthermore, learning-based techniques are applicable to different domains and achieve good results [32], [77]. However, data-driven approaches tend to suffer from the curse of dimensionality and require large amount of training data to train activity models. As users perform activities in various ways and orders, it is difficult to obtain sufficient and representative datasets [35], [50], [140], especially in smart home environment due to cost, privacy and ethical consideration. Moreover, collecting and manually annotating huge amount of sensor data is an extremely time-consuming task. Therefore, data-driven approaches suffer from scalability, applicability and adaptability [50], [72], [141].

Knowledge-driven approaches exploit prior knowledge to build semantic activity model by using knowledge engineering techniques (also called specification-based techniques), and then reason on it with input sensor data. The advantages of these approaches are interoperability and ability to adapt to different scenarios, which are essential for context-aware environment where the sensors are multimodal. Moreover, they provide a way to represent sensor data and contexts by a formal data structure with the aid of semantic descriptions, which makes them understandable to both human and machine. Consequently, they facilitate the development of semantic activity model and recognition process. Numerous knowledge-based techniques have been introduced for context modeling. Among the techniques, ontology-based models are preferable for managing and modeling context recently [12], [50]. Despite the advantages of ontology-based technique, there are still limitations that must be tackled: ontological reasoning is computationally expensive, support for modeling temporal information is minimal and they cannot deal with uncertainty.

In this thesis, we are focusing on the weakness of ontology-based techniques to deal with uncertainty, because it affects the accuracy of activity recognition [73]. There are three levels of uncertainty in decision making process: data uncertainty, comprehension uncertainty and projection uncertainty [74]. Data uncertainty is normally associated with errors in sensor's measurements, which arise due to incompleteness (missing sensor data), imprecise, inaccurate, timeliness and incongruent [50], [74], [142]. This study is focusing on incompleteness which is the most common in smart home environments because sensors operate with certain degree of reliability or loss of data during transmission. Existing ontology-based activity recognition systems can only infer an activity when all the contextual information that defines the activity is asserted. The contextual information is captured by the sensors embedded in the environment. If one of the sensor data is missing, ontology will not be able to infer the activity that is being carried out, which is indicated by a total ignorance about the current situation in the environment. In Dempster-Shafer (DS) [143], [144] theory, total ignorance can be assigned with a weightage (called belief) and combined with other evidences with a series of mathematical functions. Furthermore, DS theory can also resolve conflicting data by combining the evidences and arriving at a degree of belief [50], [145] to facilitate the activity recognition process.

In this thesis, we propose a novel reasoning algorithm which integrates ontological reasoning based on Description Logic (DL) [146] with DS theory. The proposed algorithm preserves the advantages of ontological reasoning and has the ability to manage data uncertainties that occur during the activity recognition process. An activity is modeled as a sequence of actions separated by elapsed time between two actions and may be used to represent the real-life activity. The reasoning algorithm assigns degree of beliefs to actions based on their states: active, inactive or uncertain which are determined by using the actions' temporal sequence and inference of the actions. Then, the algorithm aggregates the action contexts to produce a belief for the activity which supports the decision making of activity recognition process. In addition, we propose a four-layered activity ontology which systematically organizes the contextual information in accordance with the activity inference process. We also propose a methodology to incorporate the evidential parameters in the ontology for reasoning using DS theory. The new algorithm is applied on two datasets, one collected internally and one publicly available, and then compared with ontology-only based recognition approach and data-driven approach. It shows very good recognition accuracy compared with other approaches.

5.2 Related Works

5.2.1 Ontologies for Activity Recognition

A number of ontology-based systems have been developed for activity recognition. In [147], an DL-based reasoning engine is used to recognize coarse-grained and fine-grained activities. Bae [148] proposed a method for recognition of ADL. Ontology is used to model the activity while semantic reasoning and rule engines are used to recognize the activities. Okeyo et al. [149] proposed a novel sensor data segmentation approach for activity recognition. Activities are modeled using ontology and semantic reasoner is used to recognize the activities. Ye et al. [150] present a novel ontology-based approach for concurrent activity recognition. Semantic dissimilarity is used to segment a continuous sensor sequence into fragments, which corresponds to one ongoing activity. In [151], an ontology-based hybrid framework for activity recognition is proposed by combining the standard reasoning semantics of OWL 2 and the standard query language of the Semantic Web. The proposed framework allows the OWL 2 reasoning module to incorporate temporal correlations of complex activities which is essential in activity recognition. Khattak et al. [152] proposed an approach to improve the general health and life status of elderly peoples by monitoring the dietary intake and health activity information. The ontology is used to model the daily life activities and patient profile information, allowing the analysis of fine-grained situation for personalized service recommendations. Ahmadi-Karvigh et al. [153] proposed a novel ontology-based framework to allocate appliance-level electricity consumption to daily activities. In the framework, appliance usage data is separated into categories of activity events, which are next segmented into activity segments. Then, a classification model is used to classify the activity segments into activity classes. None of the aforementioned approaches address uncertainty in activity recognition scenarios. Riboni and Bettini [154] proposed a hybrid reasoning for activity recognition which combines data-driven and knowledge-driven approaches called COSAR. In COSAR, statistical classifier recognizes an activity which is then tested through consistency checking by ontological reasoning to verify the recognition. COSAR can deal with uncertainties since machine learning technique is used as the classifier. However, the approach suffers from data scarcity to train the activity model. In [155], a novel unsupervised approach that combines data-driven and knowledge-based techniques for mining activity recognition is proposed. The ontology is used to represent the domain knowledge for facilitating the unsupervised discovery of activity patterns. However, the system fall short in distinguishing semantically similar activities that are occurring close together. Furthermore, the system has limited ability to deal with uncertainty.

5.2.2 Reasoning under Uncertainty

A number of approaches have been proposed in the literature for reasoning with uncertainty. Probabilistic theory is a widely used method in dealing with uncertainty. It provides a mathematically sound representation for degrees of belief. Ranganathan et al. [156] use ontologies combined with probabilistic logic to infer on-going activity based on object and location contexts. Uncertainty is modeled by a confidence value specified to context predicates. Similar approach is found in [157] in which, probabilities are assigned to events to handle noisy and ambiguous observations. However, the approaches do not utilize ontological reasoning for inferring new context information. Helaoui et al. [158] combines log-linear models with DL to represent uncertainty in the ontology for activity recognition. However, log-linear DL do not support nominal and concrete domains to model concrete properties and values. Furthermore, the proposed approach does not support complex temporal modeling and reasoning. Several previous works deal with uncertainty by extending OWL through Bayesian Network. These approaches represent uncertain information by using probability and dependency annotation. For example, BayesOWL [159] extends OWL by a set of rules to transform the defined concepts in ontology into a Bayesian network. OntoBayes [160] improves BayesOWL by supporting

OWL properties and multi-valued random variables. However, it lacks OWL's class support and hence it is not possible to model relationship between concepts. Ausín [161] overcomes limitations of OntoBayes and BayesOWL and offers uncertainty information isolation to ease the reutilization of probabilistic ontologies. Although Bayesian model is capable of dealing with uncertainties due to inaccurate and contradicting sensor data, it is not capable of dealing with missing sensor data [50] which is the focus of this study. Using Bayesian theory a missing sensor data could be represented by a proposition of inactive sensor. However, such proposition is not always true because the system might not receive the data due to communication loss. Unlike Bayesian theory where each individual proposition is assigned a non-negative value (probability), DS theory distributes non-negative weights (masses) to any combination of propositions [144]. This means that the belief function can explicitly represent any ambiguity or ignorance about what has been observed such as missing sensor data.

DS theory has been used in activity recognition for handling uncertainty. Wu [162] proposed to combine sensor outputs using DS theory for context-aware computing. Hong et al. [163] introduced evidential-based activity model where DS theory is used for combining contextual information to infer activities. Zhang et al. [164] used similar model for activity recognition and presented evidence selection and conflict resolution techniques based on evidence theory. Directed acyclic graph-based activity model is introduced in [165], [166] for activity recognition. DS theory is extended by including temporal information when fusing contextual information to improve recognition accuracy. Sebbak et al. [167] proposed new conflict resolution and evidential mapping techniques to optimize decision making in activity recognition. Similar model found in [165] is used for modeling activity. Liao et al. [168] introduced new activity model in the form of three-layer lattice structure which allows historical data to be used as a priori knowledge, and DS theory is used to handle uncertainty derived from sensor errors. Finally, Chen et al. [169] proposed a framework to fuse activity classification obtained by processing signals from a depth camera and an accelerometer using DS theory. The framework resolved uncertainty due to differing modality sensors. Although all presented approaches address the uncertainty problem, ontology technique is not used, which we consider to be the most advantageous and convenient tool for activity modeling. Aloulou et al. [170] proposed an algorithm for handling uncertainty in sensor detection by modeling its reliability in terms of battery level, physical and operational behaviors. Ontology is used for representing uncertainty level in context information and DS theory is used to combine contexts acquired from multiple sources in order to obtain the consensual uncertainty value. Similar work is found in [171], in which a novel modeling approach based on DS theory is proposed to handle uncertainty in sensor data. The proposed approach handles uncertainty by modeling not only hardware characteristics of the sensor, but also the consensus of a group of sensors. However, the literature does not address the uncertainty due to missing sensor data in ontological reasoning which is the focus of this research. Uncertainty could be handled through a hybrid approach, in which the data-driven approach is used to enrich the activity ontology [140], [172]. However, incorrect activity definitions could be encoded and that may lead to incorrect classification. Furthermore, traditional ontological reasoning is used for activity recognition. Fuzzy logic has been used for handling uncertainty in ontology-based activity recognition [78]. However, fuzzy logic is dealing with the concept of vagueness in context information, not the occurrence of an activity. Dai et al. [173] proposed a missing data reconstruction approach based on similarity measure to improve human behavior prediction. However, the transfer learning algorithm requires a significant amount of data, in which it involves binary matrices of user behaviors where a matrix represents an activity data for a number of days.

Ontology offers several advantages over other specification-based techniques [35], [174]. Firstly, it is understandable, sharable and reusable by both human and machine, and hence allows non-technical users to encode domain knowledge. Secondly, it provides reasoning services to infer activity by fusing information through reasoning mechanism of DL such as subsumption, satisfiability and instance checking. Furthermore, ontological reasoning can detect possible inconsistencies in the definition of concepts and properties of an ontology. Consistency checking is crucial because it may lead to erroneous conclusions. Finally, user's activity preferences and styles can be encoded easily and hence facilitate personalized and adaptive modeling process. All the above features make ontology the preferred approach for activity modeling and recognition.

In this study, we propose a new reasoning algorithm for recognition of activities. It features reasoning mechanism of DL and uncertainty management due to missing sensor data while combining contexts from different sensors and provides a degree of belief of the activities, supporting the decision making process in order to improve the reasoning performance. The proposed algorithm is evaluated using two datasets, an internally collected and the other public one. In addition, we have implemented the HMM-based approach that can handle uncertainties to compare with the proposed approach. This work is, to the best of our knowledge, the first to propose an integration of DS theory with the OWL-DL based reasoning to deal with missing sensor data.

5.3 Ontological Reasoning with Uncertainty

An activity has diverse contextual information in terms of spatial, object and temporal contexts. Spatial context contains location and area information such as rooms, household furniture and appliances. Object contexts refer to human-object interactions such as opening a door, using a burner etc. Temporal contexts represent the time and duration. The contextual information is captured by embedded sensors in an environment, which provide clues about the activity being performed. By capturing and modeling this information, it is possible to infer the corresponding object interaction and location contexts from the activation of the sensors, which is generally referred to as human action context. For example, a magnetic contact switch installed on a door of a kitchen cabinet and PIR sensors installed in the kitchen can indicate the action of opening and closing the cabinet door. A series of human actions form an activity. By fusing human action contexts, it is possible to infer an activity being performed by a person in the inferred location with the inferred object interactions. In summary, activity is an aggregation of contexts from diverse sources which contain some knowledge about the events in the environment.

In this research, we propose an activity ontology organized into four layers of concepts, in which each layer of concept is explicitly defined in the ontology. The rationale is that by explicitly defining the concepts, annotation property can be used to annotate the evidential parameters which are required when reasoning using DS theory. The details are explained in Section 5.4.2. Figure 5.1 illustrates the four-layered activity ontology. The object interaction and location concepts are the atomic concepts which are used to describe the action concepts, while action concepts are used to describe activity concepts. An action might be associated with multiple activities. Activity inference is a process whereby a lower layer concept is semantically interpreted by the higher layer concept. The sensor layer models sensor states and provides contextual information on object interaction and location in the environment. From here, the process goes further up to the action layer, in which object interaction and location concepts are propagated and fused to infer an action. In the end, several actions are combined to form a conclusion on what activity is being performed. The modeling of the activity ontology is described in Section 5.4.1.

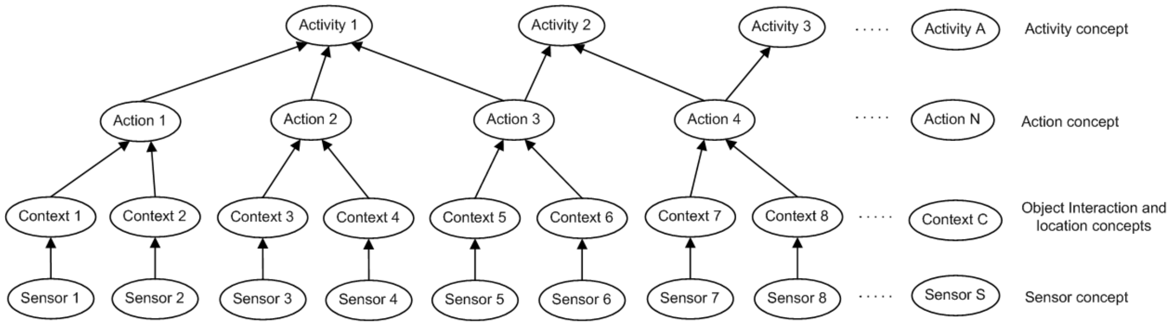


Figure 5.1: The generic conceptual activity ontology organized into four layers of concepts.

5.3.1 Modeling Uncertainty in Ontological Reasoning

From ontological engineering perspective, activity a is represented as a concept in an activity ontology O, and a concept is a specification that defines the aggregation of series of human action contexts. Such aggregation can be represented as conjunctive implication

$$\text{Activity}(d) = \text{Action}(x_n^{t_n}) \sqcap \left[\textstyle\bigcap_{i=n-1}^{1} \text{Action}(x_i^{t_i}) \sqcap \text{FartherThanT}_1 \sqcap \text{CloserThanT}_2\right] \quad (5.1)$$

where $\sqcap$ is the intersection operator as described in Table 5.1. $\text{Activity}(d)$ is the concept of activity d. $x_i^{t_i}$ is the ith contextual information at time t_i describing the activity $d \in O$, $i = (n-1), \dots, 2, 1$. Time is expressed with respect to an absolute time reference. $x_n^{t_n}$ is the most recent contextual information in the series at time t_n. The Action predicate is an atomic concept associated with $\text{Activity}(d)$ in O. FartherThanT_1 and CloserThanT_2 are optional temporal constraints which define the minimum, T_1 and maximum, T_2 elapsed time between contexts $x_i^{t_i}$ and $x_{i+1}^{t_{i+1}}$. Specifically, such definition requires context $x_{i+1}^{t_{i+1}}$ to have a predecessor (human action context) in the range of $t_{i+1} - \text{T}_2 < t_i < t_{i+1} - \text{T}_1$.

In ontology-based activity recognition system, an activity is recognized if every action concept associated with it is inferred. However, uncertainty due to missing data may arise during context reasoning process. If one of the action concepts is not inferred, then the activity will not be recognized. In order to accommodate such uncertainty, we propose a novel reasoning algorithm which integrates ontological reasoning with DS theory to support the reasoning process. The proposed algorithm assigns each context x with a degree of belief that is a function of uncertainty in the observations. It measures the strength of the context supporting an activity. Therefore, the higher the degree of belief of the context, the greater is the possibility that the activity is being performed. The concept of activity d could be represented with uncertainty component as follows.

$$\text{Activity}(d) = \left[\text{Action}\left(x_n^{t_n}\right) \text{ with } m\left(x_n^{t_n}\right)\right] \sqcap \left[\sqcap_{i=n-1}^{1}\left[\text{Action}\left(x_i^{t_i}\right) \text{ with } m\left(x_i^{t_i}\right)\right] \sqcap \text{FartherThanT}_1 \sqcap \text{CloserThanT}_2\right] \quad (5.2)$$

where $m\left(x_i^{t_i}\right)$ is the mass function of $x_i^{t_i}$. Then, Dempster's rule of combination can be used for fusing action contexts to calculate the degree of belief of activity d.

We propose the assignment of the degree of beliefs based on the states of the action contexts. We define three states to represent the occurrence of an action in the environment: active, inactive and uncertain. The active state represents that the action has occurred in the environment and inactive state represents the opposite case. The uncertain state represents the ignorance about the state of the action, either active or inactive. The action states are determined by the temporal sequence of the actions and inference of the actions. Temporal sequence of actions is the human action sequence associated with a timeline, in which the actions are performed one after another as illustrated in Figure 5.2. Assuming the context reasoning is performed at t_n and Action2 and Action3 are inferred while Action1 is not inferred due to missing sensor data, then, Action2 and Action3 are assigned the active state. Action4 is considered inactive since the action comes after Action3 in the sequence of actions while Action1 is assigned the uncertain state. The rationale is that Action1 might have been performed but the sensor data is lost, for example during transmission. Next, the sensors associated with the object interaction and location contexts which define the actions are not activated because the action is not actually performed in the environment. This uncertainty represents the ignorance about the environment being observed. The algorithm for determining the states of the action concepts in ontological reasoning is described in Section 5.5.2.

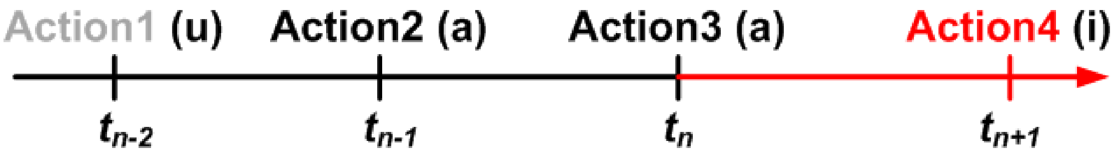


Figure 5.2: The determination of states of the actions in the action sequence timeline, (a) is active, (i) is inactive and (u) is uncertain.

In DS theory, the degree of belief, referred to as a mass, takes a value between 0 and 1. It represents the degree of belief of each subset of a frame of discernment denoted by Θ, where Θ is the universal set that comprises all possible states of an entity. Therefore, any subset of the frame of discernment can be assigned a mass, which means any ambiguity or uncertainty on what is being observed can be explicitly represented by the mass function. For instance, the elements representing the states (active, inactive and uncertain) of the action contexts are {action}, {¬action} and {action,¬action} respectively. The distribution of mass over the frame of discernment must satisfy the following two properties.

$$\sum_{A \subseteq \Theta} m(A) = 1 \quad m(\phi) = 0 \tag{5.3}$$

where A is a subset of Θ and ϕ is the empty set.

5.3.2 Evidential Operations

An action context with active state is inferred by semantically interpreting the lower-level contextual information. Therefore, the belief of an action is assigned by propagating the mass from the associated sensors. As shown in Figure 5.1, a sensor activation indicates a location context or an object interaction context. This relationship is defined by a compatibility relation [175], whereby the mass of a sensor can be translated to the corresponding context through compatibility mapping. The propagation of masses from location and object contexts to action context is defined by an evidential mapping [176] due to their uncertain relationship. An evidential mapping for the frame of discernment of a location context (evidence), Θ_E to the frame of discernment of an action context (hypothesis), Θ_H is given as follows

$$\Gamma^*(e_i) = \Big\{\big(H_{ij}, f(e_i \to H_{ij})\big), \dots, \big(H_{im}, f(e_i \to H_{im})\big)\Big\} \tag{5.4}$$

where $e_i \in \Theta_E, H_j \in \Theta_H, i = 1, \dots, n, j = 1, \dots, m$. Then, a mass function of H_j can be calculated if the evidence of Θ_E is propagated to Θ_H through an evidential mapping as follows.

$$m(H_j) = \sum_i m(e_i)\, f(e_i \to H_j) \tag{5.5}$$

Then, the propagated masses from the location and object contexts to action context are combined to produce a single mass representing the belief of the action context. Dempster's rule of combination solves the problem of combining evidence from two distinct sources as follows

$$(m_1 \oplus m_2)(C) = \frac{1}{1-k} \sum_{A \cap B = C} m_1(A)\, m_2(B) \tag{5.6}$$

where $\oplus$ denotes the combination operation and k is defined as follows.

$$k = \sum_{A \cap B = \emptyset} m_1(A) m_2(B) \tag{5.7}$$

k is the total conflicting mass between the two pieces of evidence. Similar methodology is used for propagating the action contexts to the activity context. Often, more than two action contexts are combined to infer an activity context. The problem of combining more than two evidences is described by

$$m_1 \oplus m_2 \oplus \dots \oplus m_n = \Big(\big((m_1 \oplus m_2) \oplus \dots\big) \oplus m_n\Big) \tag{5.8}$$

One major criticism of DS theory is its handling of conflicting mass which is called Zadeh paradox [177]. It gives counter-intuitive results in case of high conflict. Numerous combination rules have been proposed to resolve the paradox [178]. Murphy's rule overcomes the possibility of a single source dominating all other sources by averaging the sources of evidences as follows [179].

$$m_M(Z) = \frac{1}{n}\big(m_1(A) + m_2(B) + \cdots + m_n(N)\big) \tag{5.9}$$

After calculating the average of n sources, Dempster's rule of combination is applied $n-1$ times. The algorithm of propagating the masses and calculating the mass of the activity is described in Section 5.5.3.

5.4 Activity Model

We use Web Ontology Language (OWL), a DL-based markup language, to develop the context model. Ontology consists of three core elements: concepts, roles and individuals, which represent domain knowledge. A concept corresponds to a set of individuals, a role is a binary relationship between individuals and an individual is an instance of a given concept. The concepts can be organized in a hierarchical structure based on the "is-a" relationship to form superconcept and subconcept relations, whereby the subconcept inherits the properties of the superconcept. A logical statement relating concepts and/or roles is called axiom which can be formed by using the basic elements in Table 5.1. In DL, a knowledge base is comprised of Terminological Box (TBox) and Assertional Box (ABox).

The TBox contains the vocabularies of concepts and roles. Also, complex concept definitions can be built based on other concepts. For instance, it is possible to define

$$\text{Ambient\_Sensor} \equiv \text{Sensor} \sqcap \exists\text{hasLocation}.\text{Room} \tag{5.10}$$

In the expression, we associate the concept definition (right-hand side) with Ambient_Sensor. The concept definition consists of hasLocation role that define the binary relation between Sensor and Room concepts. The definition characterizes Ambient_Sensor as a sensor that is located in a room. The ABox contains the vocabularies of asserted individual and their relationships. For instance, by using the concepts above

$$\begin{gathered} \text{Sensor(SENSOR\_01)} \\ \text{Room(LOUNGE)} \\ \text{hasLocation(SENSOR\_01, LOUNGE)} \end{gathered} \tag{5.11}$$

assert individuals named SENSOR_01 and LOUNGE in the ABox. The first and second definitions, called concept assertions, state that "sensor 01" is a sensor and "lounge" is a room respectively. The third definition, called role assertions, one states that "sensor 01" is located in the "lounge".

A DL system not only stores the terminologies and assertions, but it can also reason about them, where implicit knowledge about concepts and individuals can be inferred from the knowledge that is explicitly contained in the knowledge base. Reasoning in the TBox includes satisfiability, equivalence and disjointness, which ultimately can be reduced to checking the subsumption of concepts [146]. A subsumption reasoning is to decide whether a concept C is more general than another concept D i.e. if D logically implies C. For instance, reasoning the following definition

$$\begin{gathered}\text{Location\_Sensor} \sqsubseteq \text{Sensor} \\ \text{Location\_Sensor} \equiv \exists \text{hasLocation}.\text{Room}\end{gathered} \tag{5.12}$$

will deduce that Location_Sensor is subsumed by (or "is a") Ambient_Sensor. The ABox reasoning includes consistency checking, instance checking, retrieval and realization. The consistency checking is to check whether the concept and role assertions are consistent with respect to the TBox. The instance checking is to check if an individual is an instance of a given concept. The retrieval problem is to find all individuals that are instances of a given concept and realization problem is to find the most specific concepts for a given individual, with respect to the subsumption ordering. The following example demonstrates the mechanism of instance checking since it is the main reasoning scheme in the proposed model. Using the concept definitions above, the following assertions are included in the ABox:

$$\begin{gathered}\text{Location\_Sensor}(\text{PIR\_01}) \\ \text{hasLocation}(\text{PIR\_01}, \text{LOUNGE})\end{gathered} \tag{5.13}$$

Then, it can be inferred that PIR_01 is an instance of Ambient_Sensor.

Table 5.1: Description logic notations.

Syntax	Description
A	Atomic concept
R	Atomic role
$\neg A$	Negation or complement of concepts
$C \sqcap D$	Intersection or conjuction of concepts
$C \sqcup D$	Union or disjunction of concepts
$\forall R.C$	Universal restriction (All values restriction)
$\exists R.C$	Existential restriction (Some values restriction)
$C \sqsubseteq D$	Concept inclusion
$C \equiv D$	Concept equivalence

Numerous methodologies of ontological modeling have been proposed. In this research work, we mostly followed a methodology introduced in [174]. Several reasons are identified as to why the methodology is chosen. Firstly, the methodology allows for context recognition through ontology-based reasoning without using rule-based system and external reasoning engine. Secondly, a set of temporal concepts (FartherThan and CloserThan) are introduced to define temporal constraints between two concepts. Lastly, the methodology introduces a role called hasPredecessor to specify temporal precedence relationship between concepts. The last two features are important because they allow activity concepts to be defined as a series of actions with specific durations.

5.4.1 Activity Ontology

The developed activity ontology consists of five main concepts: Sensor, Location, Object, Action and Activity. Sensor models the most recent state of a sensor. Embedded sensors in the environment can be divided into object sensors and location sensors. Therefore, two subconcepts are defined corresponding to those sensors in the ontology: Object_Sensor and Location_Sensor. Object_Sensor represents object sensors which monitor person's object interaction such as dispenser sensor, door sensor and water detector. Location_Sensor represents location sensors such as PIR sensors which monitor person's presence in an environment such as kitchen, living room and toilet. A sensor concept consists of subconcepts which represent its states. For example, a sensor to detect the presence of an item has two possible states: PRESENT and ABSENT. The two possible states of the sensor are defined as follows:

$$\begin{gathered}\text{Object\_Sensor} \sqsubseteq \text{Sensor}\\ \text{Item\_Sensor} \sqsubseteq \text{Object\_Sensor}\\ \text{Item\_Sensor\_PRESENT} \sqsubseteq \text{Item\_Sensor}\\ \text{Item\_Sensor\_ABSENT} \sqsubseteq \text{Item\_Sensor}\end{gathered} \tag{5.14}$$

Location concept models the location contexts that is provided by embedded sensors. Each location sensor that is included in the system corresponds to a different subconcept of Location concept in the ontology which represents a particular room or area. To define a location concept, restriction is applied by using concept equivalence operator through hasFluent role that specifies the relevant sensor in a given state as follows:

$$\begin{gathered}\text{In\_Kitchen} \sqsubseteq \text{Location}\\ \text{In\_Kitchen} \equiv \exists \text{hasFluent}.\,\text{PIR\_Kitchen\_ON}\end{gathered} \tag{5.15}$$

which describes a person is in the kitchen when the state of PIR sensor is ON. The hasFluent role establishes a relationship between individuals of the location concept and individuals of the sensor state concept. Object concept models the object interaction context which is provided by the object sensors. Object concept consists of subconcepts that represent the object interaction contexts in the environment. Similar modeling approach is used whereby concept equivalence operator is used to apply a restriction via hasFluent, for example

$$\begin{gathered}\text{Water\_Tap\_Is\_Open} \sqsubseteq \text{Object}\\ \text{Water\_Tap\_Is\_Open} \equiv \exists \text{hasFluent}.\,\text{Water\_Tap\_OPEN}\end{gathered} \tag{5.16}$$

The methodology introduced a concept called Interval which has the following definition [174].

$$\text{Interval} \equiv \text{hasFluent}.\,\text{Sensor} \sqcap \exists \text{hasBeginAt}.\,\text{Integer} \sqcap (\text{hasPredecessor} \circ \text{hasPredecessor} = \text{hasPredecessor}) \tag{5.17}$$

Interval concept models time intervals through hasBeginAt role which states the starting time of each time interval, and also the states of sensors that hold in each time interval through hasFluent role. The hasPredecessor role defines the precedence relationships between a time interval and the preceding time interval. Action or human action is a simple motion which typically lasts for a short time [32], [72], for example opening a water tap, turning on a burner, or closing a cabinet door. Using interval concept, an action concept is defined with additional constraints using object and location concepts. Considering the context of "a person is opening a water tap". We can describe the context as

$$\begin{gathered}\text{Opening\_Water\_Tap} \sqsubseteq \text{Action}\\ \text{Opening\_Water\_Tap} \equiv \text{Interval } \sqcap \text{In\_Kitchen} \sqcap \text{Water\_Tap\_Is\_Open}\end{gathered} \tag{5.18}$$

This definition requires the "water tap sensor" that monitors the water tap status and the PIR sensor that detects people in proximity in the state of OPEN and ON, respectively. An example of inferring an action, suppose at time t_6, the following individuals are asserted in the ABox:

$$\begin{gathered}\text{PIR\_Kitchen\_ON(PIR1)}\\ \text{Water\_Tap\_ON(WATER1)}\\ \text{Interval(E6)}\\ \text{hasBeginAt(E6, t6)}\\ \text{hasFluent(E6, PIR1)}\\ \text{hasFluent(E6, WATER1)}\end{gathered} \tag{5.19}$$

Then, reasoning individual interval E6 will return Opening_Water_Tap.

Activity concept models a series of actions by expressing temporal precedence relationships between action concepts using hasPredecessor and temporal constraints if required. Consider the following definition

$$\text{Cooking\_Meal} \sqsubseteq \text{Activity}$$

$$\text{Cooking\_Meal} \equiv \text{Using\_Burner} \sqcap \exists\text{hasPredecessor.}\big(\text{Opening\_Water\_Tap} \sqcap \exists\text{hasPredecessor.}(\text{Taking\_Pot} \sqcap \exists\text{hasPredecessor.Opening\_Cabinet})\big) \tag{5.20}$$

The definition is basically a list of action concepts, temporally ordered from the more recent to the older one. An example of activity recognition, suppose at time t_7, the following individuals have been asserted in the ABox that already contain (5.17).

$$\begin{array}{l} \mathrm{Interval(E3), hasPredecessor(E3, E2)} \\ \mathrm{Interval(E4), hasPredecessor(E4, E3)} \\ \mathrm{Interval(E5), hasPredecessor(E5, E4)} \\ \mathrm{Interval(E6), hasPredecessor(E6, E5)} \\ \mathrm{Interval(E7), hasPredecessor(E7, E6)} \end{array} \tag{5.21}$$

and the individuals have the following properties:

$$\begin{array}{c} \mathrm{Opening\_Cabinet\_Door(E3)} \\ \mathrm{Taking\_Pot(E4)} \\ \mathrm{Taking\_Pot(E5)} \\ \mathrm{Opening\_Water\_Tap(E6)} \\ \mathrm{Using\_Burner(E7)} \end{array} \tag{5.22}$$

Then, reasoning individual interval E7 will return Cooking_Meal.

5.4.2 Representation of Evidential Parameters

An individual in the ontology such as individual sensor state is a piece of evidence that supports a proposition. The strength of an evidence in favor of a proposition is called mass. To represent a mass in the ontology we propose a role called hasMass to store the values. The role has a datatype of double, which is implicitly in the range from 0 to 1. Listing 5.1 shows an example of an individual (PIR16_ON) of the PIR concept is asserted with hasMass to store its mass as defined by line 3.

Listing 5.1: Asserting an individual of a sensor state with hasMass role

```
1  <NamedIndividual rdf:about="#PIR16_ON">
2     <rdf:type rdf:resource="#PIR_Kitchen_ON"/>
3     <hasMass rdf:datatype="&xsd;double">1.0</hasMass>
4  </NamedIndividual>
```

When a context is propagated to another context, the relationship is represented by an evidential mapping, which reflects the strength of interrelationship between the propagated contexts and the inferred contexts. Figure 5.3 illustrates an example of the propagation of Opening_Cabinet, Opening_Water_Tap, Taking_Pot and Using_Burner contexts to Cooking_Meal context. The impact of the propagated contexts is represented by evidential mappings defined by ‘a’, ‘b’, ‘c’ and ‘d’. The evidential mappings are given in Table 5.2. For instance, the evidential mapping of “using burner” to “cooking meal” is 0.857. The evidential mappings can be estimated by examining for each context occurrence the number of times the corresponding sensors are activated. In the absence of training data, domain knowledge can be obtained from experts and users [165], [167]. We propose to use annotation property to represent evidential mappings in the ontology. Listing 5.2 shows an example of annotating evidential mappings of Cooking_Meal.

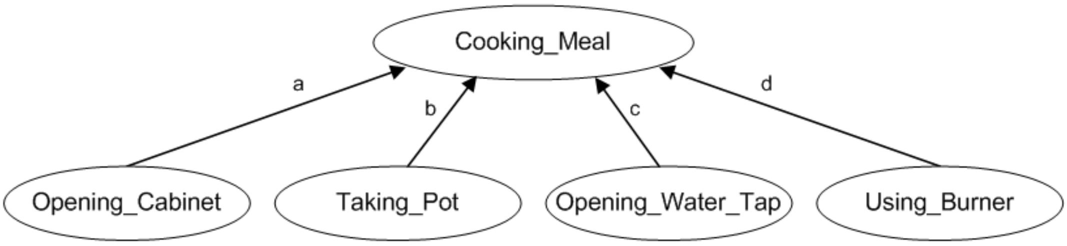


Figure 5.3: Propagating contexts to Cooking_Meal.

Table 5.2: Evidential mappings of Cooking_Meal.

Propagation	Evidential Mapping

Open_Cab → Cook_Meal	{Open_Cab} → {({Cook_Meal},0.625),
	({Cook_Meal,¬Cook_Meal},0.375)}
	{¬Open_Cab} → {({¬Cook_Meal}),1.0}
	{Open_Cab,¬Open_Cab} → {({Cook_Meal,¬Cook_Meal},1.0)}

Take_Pot → Cook_Meal	{Take_Pot} → {({Cook_Meal},0.609),
	({Cook_Meal,¬Cook_Meal},0.391)}
	{¬Take_Pot} → {({¬Cook_Meal}),1.0}
	{Take_Pot,¬Take_Pot} → {({Cook_Meal,¬Cook_Meal},1.0)}

Open_Wat_Tap → Cook_Meal	{Open_Wat_Tap} → {({Cook_Meal},0.518),
	({Cook_Meal,¬Cook_Meal},0.492)}
	{¬Open_Wat_Tap} → {({¬Cook_Meal}),1.0}
	{Open_Wat_Tap,¬Open_Wat_Tap} →
	{({Cook_Meal,¬Cook_Meal},1.0)}

Using_Burner → Cook_Meal	{Using_Burner} → {({Cook_Meal},0.857),
	({Cook_Meal,¬Cook_Meal},0.143)}
	{¬Using_Burner} → {({¬Cook_Meal }),1.0}
	{Using_Burner,¬Using_Burner} → {({Cook_Meal,¬Cook_Meal},1.0)}

Listing 5.2: Annotation of evidential mapping

```
 1 <AnnotationProperty rdf:about="#evidentialMapping"/>
 2 <Class rdf:about="#Cooking_Meal">
 3   <equivalentClass>
 4     <Class>
 5       ...
 6     </Class>
 7   </equivalentClass>
 8   <rdfs:subClassOf rdf:resource="#Activity"/>
 9   <evidentialMapping rdf:datatype="&xsd;string">
10     Opening_Cabinet:0.625,0,0.375;0,1,0;0,0,1
11   </evidentialMapping>
12   <evidentialMapping rdf:datatype="&xsd;string">
13     Taking_Pot:0.609,0,0.375;0,1,0;0,0,1
14   </evidentialMapping>
15   <evidentialMapping rdf:datatype="&xsd;string">
16     Opening_Water_Tap:0.518,0,0.492;0,1,0;0,0,1
17   </evidentialMapping>
18   <evidentialMapping rdf:datatype="&xsd;string">
19     Using_Burner:0.857,0,0.143;0,1,0;0,0,1
20   </evidentialMapping>
21 </Class>
```

Line 1 defines annotation property called evidentialMapping. Lines 3-7 define the concept definition of Cooking_Meal. Lines 9-20 define the usage of annotation property of evidentialMapping to annotate Cooking_Meal with evidential mappings of each propagated concept. Each annotation consists of the name of the action concept and its evidential mappings. The evidential mappings of each relationship are delimited by semicolons.

5.5 Activity Recognition Algorithm

5.5.1 Ontological Reasoning with Uncertainty

Activity recognition algorithm consists of two phases: knowledge acquisition and ontology reasoning. Knowledge acquisition phase is a repetitive process of acquiring sensor data and creating individuals of interval concept to reflect the current state of sensor activation. This is represented by line 1 in Listing 5.3, which shows the activity recognition algorithm. Before knowledge acquisition phase is executed, the ABox is initialized by adding a concept assertion for each sensor state which is deemed relevant to define activity concepts. The algorithm utilizes two lists called sensor list, $\mathbb{L}$ and interval list, $\mathbb{E}$. $\mathbb{L}$ is used to store the current sensor states which are used to define contexts in the ontology. The sensor states are represented by individuals of sensor concepts. Whenever a sensor changes its state, the individual of the sensor state is inserted into $\mathbb{L}$. Symmetrically, the individual of its previous state is removed from the list. In this way, $\mathbb{L}$ reflects the most recent states of the sensors.

Listing 5.3: Algorithm for ontological reasoning under uncertainty

$\mathbb{E}$ is a list of individual instances of Interval in the window, ordered chronologically with the most recently asserted individual at the head.

$\mathbb{D}$ is a list of activity concepts which contain X in their axioms.

```
populates ABox with individual instances of Interval
**if** enough time has passed **then**
    **for** all $e$ in $\mathbb{E}$ **do**
        infer location concept
        for all activity $D$ of which $e$ is an instance, assert $D(a)$ in ABox
        **for** all action $X$ of which $e$ is an instance **do**
            $\mathbb{D}$ = get all $D$ where $D \equiv C \sqcap ... \sqcap X \sqcap ...$ and $D \sqsubseteq$ Activity
            **for** all $D$ in $\mathbb{D}$ **do**
                determine the states of the action concepts (contexts) of $D$
                calculate the mass of activity $D$
            **end for**
        **end for**
    **end for**
**end if**
```

In each iteration, whenever there is a new sensor state and its corresponding individual is inserted into $\mathbb{L}$, the algorithm creates a new individual of interval concept and assert the individual with all individuals in $\mathbb{L}$ and its temporal precedence relationship with previous interval. Then, this individual is inserted into $\mathbb{E}$. $\mathbb{E}$ is a list of individuals of interval concept in the window, ordered chronologically with the most recently asserted individual at the head. Whenever a sensor changes its state, the previous individual interval which was added at the time the sensor changed to the previous state is removed from $\mathbb{E}$. Similarly, the individual of the sensor state is removed from $\mathbb{L}$. In this way, $\mathbb{E}$ represents a sequence of events happening in the environment, defined by the sensor activations. The most recent interval holds the current sensor states. Details of the algorithm can be referred to [174]. Periodically, the ontology reasoning phase is executed to infer activities from the ontology by performing reasoning on every intervals in $\mathbb{E}$ to infer action and activity concepts as defined by lines 2-5.

5.5.2 Determination of Action Concept States

Lines 6-12 define the proposed algorithm to support the reasoning under uncertainty. Lines 6-7 check if e is an instance of some action concept X corresponding to a context of some activity concept D. These action concepts are inserted into $\mathbb{D}$. For each activity concept in $\mathbb{D}$, the contexts of the activity are aggregated to calculate the mass of the activity. The aggregation can be divided into two operations: determining the states of the action concepts of the activity and calculating the mass of the activity as defined by lines 9 and 10 respectively. Listing 5.4 shows the algorithm to determine the states of the actions. The algorithm utilizes two lists, $\mathbb{X}$ and $\mathbb{Y}$ to store the action concepts which are assigned the active and inactive states respectively. The action concepts with uncertain state can be obtained by determining the concepts which are not in the lists. X is the (current) action concept which is inferred from e is inserted into $\mathbb{X}$ as defined by line 1. Line 2 determines the index number, n of X in the series of actions which is defined in the concept definition of D. Line 3 inserts all action concepts which come after X_n in the concept definition of D. Lines 4-5 acquire all individual intervals before e to determine the states of the remaining action concepts and get the first interval in the list. Then, line 6 decrements n to determine the state of the preceding action concept.

Lines 7-27 browse $\mathbb{V}$ which contains every intervals prior to e, to determine the states of the remaining action concepts of D, by searching for intervals which can satisfy the remaining parts of the axiom defined in concept definition of D. For example, consider the following definition:

$$\text{Activity1} \equiv \text{Action3} \sqcap \exists \text{hasPredecessor}.(\text{Action2} \sqcap \exists \text{hasPredecessor}.\text{Action1})$$

Assuming the interval being reasoned is an instance of Action2. Since Action3 comes after Action2 in the sequence of actions, Action3 is considered to be inactive. Then, the algorithm determines the states of actions that come before Action2 in the concept definition of Activity1, which in this case Action1. The state is determined by performing instance checking on the previous intervals if there is an interval which is an instance of Action1. The interval must also be a predecessor of Action2 in order to satisfy the axiom, including any temporal constraints if specified. If all conditions are satisfied, then the corresponding actions are considered to be active. Otherwise, the action is assigned the uncertain state.

Specifically, for every $\hat{e}$, lines 9-13 test the three conditions in order to determine if $\hat{e}$ is an instance of the action concept being checked. First, the condition is the equality of action concept, C which is inferred from $\hat{e}$ and the action concept, X_n associated with D. Second, the condition is if $\mathrm{hasPredecessor}(e, \hat{e})$ is asserted in the ABox. If it is true, there is a temporal precedence relationship between e and $\hat{e}$. The final condition is tested if temporal constraints are defined in the concept definition. If all the conditions are satisfied, it can be concluded $\hat{e}$ is a part of the axiom being evaluated, insert the action concept into $\mathbb{X}$ (line 23), and n is decreased to evaluate the preceding action concept as defined by line 24. The procedure is repeated and the iteration stops when there is no more interval or action concept to be checked. At the end of the iteration, $\mathbb{X}$ contains all action concepts with active state associated with D.

Listing 5.4: Determining the states of the action concepts

$\mathbb{X}$ is a list of action concepts with active state

$\mathbb{Y}$ is a list of action concepts with inactive state

D is an activity concept in $\mathbb{D}$

e is an individual of Interval being checked

```
insert $X(e)$ into $\mathbb{X}$
$n$ = determine the index number of $X$ in concept definition of $D$ where $D \equiv \ldots \sqcap X_{n+1} \sqcap X_n \sqcap X_{n-1} \sqcap \ldots$
insert all $X_k$ into $\mathbb{Y}$ where $D \equiv \ldots \sqcap X_k \sqcap X_n \sqcap \ldots, k = n+1, n+2, \ldots$
$\mathbb{V}$ = get individuals of Interval preceding $e$, ordered chronologically with the most recent individual at the head
get $\hat{e}$ at the head of $\mathbb{V}$
$n = n - 1$
**while** $\hat{e} \neq \emptyset$ and $i > 0$ **do**
    insert = False
    **if** $C = X_n$ where $\hat{e}$ is an instance of $C$ **then**
        **if** $\mathrm{hasPredecessor}(e, \hat{e}) \in \mathrm{ABox}$ **then**
            **if** FartherThanT1 or CloserThanT2 are defined where $D \equiv \ldots X_{n+1} \sqcap (X_n \sqcap \mathrm{FartherThanT_1} \sqcap \mathrm{CloserThanT_2})$ **then**
                get $\mathrm{hasBeginAt}(e, t)$ and $\mathrm{hasBeginAt}(\hat{e}, \hat{t})$
                **if** $(t - \mathrm{T_2}) < \hat{t} < (t - \mathrm{T_1})$ **then**
                    insert = True
                **end if**
            **else**
                insert = True
            **end if**
        **end if**
    **end if**
    **if** insert = True **then**
        $e = \hat{e}$
        insert $X_n(e)$ in $\mathbb{X}$
        $i = i - 1$
    **end if**
    get the next $\hat{e}$ in $\mathbb{V}$
**end while**
```

5.5.3 Propagation of Masses and Calculation of Belief

Subsequently, the mass of D is calculated by combining the action concepts to obtain the mass of the activity. The algorithm of calculating the mass is given in Listing 5.5. The algorithm assigns degree of belief to action concepts based on the states of the actions. All action concepts with inactive state are assigned value of 1.0 to the "not occurring action" element, to represent the action has not occurred in the environment. For all action concepts with uncertain state, a value of 1.0 is assigned to the uncertainty element, to represent the ignorance about the action being observed, either the action has occurred or not. As for the action concepts with active state, the degree of belief is assigned by propagating the mass from the associated sensor state concepts to the action concept.

Specifically, for each X in $\mathbb{X}$, the associated object and location concepts, C are acquired from the concept definition of X as defined by lines 1-2. Every object and location concepts are associated with sensor state concepts. For each C, lines 4-5 acquire the associated sensor state concept ($\mathsf{SensorState}$) and get all the individuals, s of $\mathsf{SensorState}$ which are asserted through $\mathsf{hasFluent}$ role in ABox. Lines 6-7 calculate the joint mass of the sensor states which in turn can be translated into the mass of C. Then, line 8 propagates C to X by multiplying the mass with evidential mapping, and line 10 computes the joint mass of X. Lines 12-14 assign a mass of 1.0 to the $\{\neg\bar{X}_j\}$ set element for each action concept X in $\mathbb{Y}$. Lines 15-17 assign a mass of 1.0 to the $\{\bar{X}_j, \neg\bar{X}_j\}$ set element to quantify the uncertainty. Then, lines 18-20 propagate all action concepts to D by evidential mapping. Finally, line 21 combines the masses to obtain the mass of the activity. An activity with the mass greater than m_t is inserted in $\mathbb{M}$ as defined by line 22. m_t is the threshold value which the mass must exceed to have the believe the activity has been performed. For every reasoning phase, the mass of an activity is updated whenever the calculated mass of current reasoning phase is higher than the previous one. Line 4 of Listing 5.3 infers user's location to determine the current location of the user. Whenever the user's location is changed, individuals of activity concepts in $\mathbb{M}$ are asserted in ABox.

Listing 5.5: Propagating masses and calculating the mass of the activity

```
for all $X_j(e)$ in $\mathbb{X}$ do
    $\mathbb{C}$ = get all $C$ where $X \equiv C_N \sqcap \ldots \sqcap C_i$ and $C_i \not\equiv \text{Interval}$
    for $C_i$ in $\mathbb{C}$ do
        SensorState = get SensorState where $C_i \equiv \exists\text{hasFluent.SensorState}$
        $S$ = get all $s$ where $s \in \text{SensorState}$ and $\text{hasFluent}(e, s) \in \text{ABox}$
        $m(S) = m^{\%}(s_1) \oplus m^{\%}(s_2) \oplus \ldots \oplus m^{\%}(s_N)$
        $m(C_i) = m(S)$
        $m_i(X_j) = m(C_i) \times m(C_i \rightarrow X_j)$ where $m(C_i \rightarrow X_j)$ is evidential mapping of
        propagating $C_i$ to $X_j$
    end for
    $m(X_j) = m_i(X_j) \oplus \ldots \oplus m_{i+N}(X_j)$  $m(\neg X_j) = 0.0$  $m(\{X_j, X_j\}) = 1.0 - m(X_j)$
end for
for all $X_j$ in $\mathbb{Y}$ do
    $m(X_j) = 0.0$   $m(\neg X_j) = 1.0$   $m(\{X_j, \neg X_j\}) = 0.0$
end for
for all $\bar{X}_j$ where $\bar{X}_j \notin \mathbb{X}, \mathbb{Y}$ and $D \equiv \ldots \sqcap \bar{X}_{n+1} \sqcap \ldots$ do
    $m(\bar{X}_j) = 0.0$   $m(\neg \bar{X}_j) = 0.0$   $m(\{\bar{X}_j, \neg \bar{X}_j\}) = 1.0$
end for
for all $X_j$ where $D \equiv X_N \sqcap \ldots \sqcap X_j, j = N, \ldots, 2, 1$ do
    $m_j(D) = m(X_j) \times m(X_j \rightarrow D)$ where $m(X_j \rightarrow D)$ is evidential mapping of
    propagating $X_j$ to $D$
end for
$m(D) = m_j(D) \oplus \ldots \oplus m_{j+N}(D)$
insert $D$ into $\mathbb{M}$ if $m(D) > m_t$
```

5.6 Scenario of Activity Recognition with Uncertainty

To better illustrate the behavior of the algorithm, consider the following simple scenario. A resident walk into the kitchen. In the first instance the kitchen PIR (M02) is active. Then, the system detects water tap sensor (W01) state and pot sensor (I04) state are OPEN and ABSENT respectively. Sensor M02, W01 and I04 are bi-state sensors. The temporally ordered sequence of sensor events is given in Listing 5.6. The ABox is initialized by adding the concept assertion for each sensor state that has been identified as relevant to recognize the activities in the scenario.

Listing 5.6: Sequence of sensor events of the scenario

Date	Time	Sensor	State
2015-11-05	10:59:49:0	M02	ON
2015-11-05	10:59:54:0	I04	ABSENT
2015-11-05	10:59:56:0	M02	OFF
2015-11-05	10:59:57:0	M02	ON
2015-11-05	11:00:13:0	W01	OPEN
2015-11-05	11:00:36:0	M02	OFF
2015-11-05	11:00:37:0	M02	ON

Table 5.3 shows the assertions of the intervals in the ABox as a sensor changes its state over time. The first column is the time of the sensor events in which each row corresponds to each line of the sensor events. The second column shows the role (hasFluent and hasPredecessor) assertions of the intervals. The third and fourth columns show the contents of interval list ($\mathbb{E}$) and sensor list ($\mathbb{L}$) at each time. Periodically, ontological reasoning is performed on all intervals in $\mathbb{E}$, which is indicated by *. The inference is given in the next row after each reasoning process. At time t_1, sensor M02 changes its state to "ON" that has been asserted in the ABox as $\mathrm{PIR\_Kitchen\_ON}(\mathrm{M02\_ON})$. Thus, the individual sensor state is inserted into $\mathbb{L}$ and a new interval (E1) is asserted in the ABox and added into $\mathbb{E}$. When the interval is asserted, its hasBeginAt role is asserted to specify its starting time and its hasFluent role is filled with all individual sensor state in $\mathbb{L}$. The same procedures are executed when sensor I04 changes its state to "ABSENT". Additionally, the precedence relationship between E2 and E1 is expressed by asserting $\mathrm{hasPredecessor}(\mathrm{E2}, \mathrm{E1})$ in the ABox. At time t_3, sensor M02 switches from "ON" to "OFF, which has not a corresponding individual sensor state in the ABox. Then, individual M2_ON is removed from $\mathbb{L}$ and the interval that was added at the time sensor M02 changed to "ON" is removed from $\mathbb{E}$ (i.e. interval E1). Subsequently, the system substitutes E2 with a new interval E3 which has the same role assertions with the exception of $\mathrm{hasFluent}(\mathrm{E3}, \mathrm{M02\_ON})$.

Table 5.3: Assertion of intervals in the ABox

Time	Assertion	$\mathbb{E}$	$\mathbb{L}$

t_1	hasBeginAt(E1,t_1)
	hasFluent(E1,M02_ON)
		E1	M02_ON

t_2*	hasBeginAt(E2,t_2)
	hasFluent(E2,M02_ON)
	hasFluent(E2,I04_ABSENT)
	hasPredecessor(E2,E1)
		E2,E1	M02_ON,
			I04_ABSENT

Taking_Pot(E2)			

t_3	hasBeginAt(E3,t_2)
	hasFluent(E3,I04_ABSENT)
	hasPredecessor(E3,E1)
		E3	I04_ABSENT

t_4	hasBeginAt(E4,t_4)
	hasFluent(E4,I04_ABSENT)
	hasFluent(E4,M02_ON)
	hasPredecessor(E4,E3)
		E4,E3	I04_ABSENT,
			M02_ON

t_5*	hasBeginAt(E5,t_5)
	hasFluent(E5,I04_ABSENT)
	hasFluent(E5,M02_ON)
	hasFluent(E5,W01_OPEN
	hasPredecessor(E5,E4)
		E5,E4,E3	I04_ABSENT,
			M02_ON,
			W01_OPEN

Taking_Pot(E4)
Taking_Pot(E5)
Opening_Water_Tap(E5)			

t_6	hasBeginAt(E6,t_6)
	hasFluent(E6,I04_ABSENT)
	hasFluent(E6,W01_OPEN)
	hasPredecessor(E6,E4)
		E6,E3	I04_ABSENT,
			W01_OPEN

t_7*	hasBeginAt(E7,t_7)
	hasFluent(E7,I04_ABSENT)
	hasFluent(E7,W01_OPEN)
	hasFluent(E7,M02_ON)
	hasPredecessor(E7,E6)
		E7,E6,E3	I04_ABSENT,
			W01_OPEN,
			M02_ON

Taking_Pot(E7)
Opening_Water_Tap(E7)			

When reasoning is performed at t_5, the inference yields Taking_Pot and Opening_Water_Tap actions since the interval is asserted with I04_ABSENT, M02_ON and W01_OPEN through hasFluent role. Subsequently, for each action concept, the activity concepts which are associated with the actions are added into $\mathbb{D}$. For instance, when $X =$ Opening_Water_Tap, then Cooking_Meal and Taking Medicine are added into $\mathbb{D}$. Next, for each activity concept in $\mathbb{D}$, the states of the action concepts of the activity are determined. When $D =$ Cooking_Meal, then $n = 3$ (X is the third action in the concept definition of D), X is added into $\mathbb{X}$ and Using_Burner (action concept after X) is added into $\mathbb{Y}$. Then, all intervals preceding E5 (E4 to E1) are acquired to determine the states of the remaining action concepts (Taking_Pot and Opening_Cabinet). The iteration is executed until there is no more intervals or the last action concept is evaluated: E4 is an instance of Taking_Pot and a predecessor of E5, while E3, E2 and E1 are not part of the axiom since they fail to satisfy the conditions. Thus, Taking_Pot is added into $\mathbb{X}$ and Opening_Cabinet is assigned the uncertain state. Subsequently, the beliefs of the actions are assigned according to the states that were determined previously. Action concepts in $\mathbb{Y}$ are assigned value of 1.0 to the "not occurring action" element while action concepts with uncertain state are assigned value of 1.0 to the uncertainty element.

For each action concept in $\mathbb{X}$, the beliefs are assigned by propagating the mass from the associated sensors. For instance, the belief of $X = \text{Opening\_Water\_Tap}$ is performed by propagating the mass of corresponding individual sensor states (W01_OPEN and M02_ON) as follows. The masses are acquired and translated to their corresponding object interaction and location concepts.

$$m(\{\text{Water\_Tap\_Is\_Open}\}) = m(\{\text{W01\_OPEN}\}) = 1.0$$

$$m(\{\text{In\_Kitchen}\}) = m(\{\text{M02\_ON}\}) = 1.0$$

An action concept is composed object interaction and location concepts. The mass propagation is defined by an evidential mapping as follows. Assuming $m(\{\text{Water\_Tap\_Is\_Open}\} \rightarrow \{\text{Opening\_Water\_Tap}\}) = 1$ and $m(\{\text{In\_Kitchen}\} \rightarrow \{\text{Opening\_Water\_Tap}\}) = 0.8$.

$$\begin{aligned} m_1(\{\text{Opening\_Water\_Tap}\}) \\ &= m(\{\text{Water\_Tap\_Is\_Open}\}) \\ &\times m(\{\text{Water\_Tap\_Is\_Open}\} \rightarrow \{\text{Opening\_Water\_Tap}\}) = 1.0 \end{aligned}$$

$$\begin{aligned} &m_2(\{\text{Opening\_Water\_Tap}\}) \\ &\qquad = m(\{\text{In\_Kitchen}\}) \times m(\{\text{In\_Kitchen}\} \rightarrow \{\text{Opening\_Water\_Tap}\}) = 0.80 \end{aligned}$$

Then, the joint mass is calculated to obtain the mass of Opening_Water_Tap as follows.

$$m(\{\text{Opening\_Water\_Tap}\}) = m_1(\{\text{Opening\_Water\_Tap}\}) \oplus m_2(\{\text{Opening\_Water\_Tap}\}) = 0.99$$
$$m(\{\text{Opening\_Water\_Tap}, \neg\text{Opening\_Water\_Tap}\}) = 0.01$$

Similar operation is performed for propagating the mass from action concepts to activity concepts. Assuming $m(\{\text{Taking\_Pot}\}) = 0.9694$, and $m(\{\text{Opening\_Cabinet}, \neg\text{Opening\_Cabinet}\}) = 1.0$ and $m(\{\neg\text{Using\_Burner}\}) = 1.0$ because the action concepts were assigned uncertain state and inactive state respectively, propagating the mass to $D = \text{Cooking\_Meal}$ is given as follows.

$$\begin{aligned} &m_1(\{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad = m(\{\text{Opening\_Cabinet}\}) \\ &\qquad \times m(\{\text{Opening\_Cabinet}\} \rightarrow \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad + m(\{\neg\text{Opening\_Cabinet}\}) \\ &\qquad \times m(\{\neg\text{Opening\_Cabinet}\} \rightarrow \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad + m(\{\text{Opening\_Cabinet}, \neg\text{Opening\_Cabinet}\}) \\ &\qquad \times m(\{\text{Opening\_Cabinet}, \neg\text{Opening\_Cabinet}\} \\ &\qquad \rightarrow \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) = 1.0 \end{aligned}$$

$$m_2(\{\text{Cooking\_Meal}\}) = m(\{\text{Taking\_Pot}\}) \times m(\{\text{Taking\_Pot}\} \rightarrow \{\text{Cooking\_Meal}\}) = 0.6059$$

$$\begin{aligned} &m_2(\{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad = m(\{\text{Taking\_Pot}\}) \times m(\{\text{Taking\_Pot}\} \rightarrow \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad + m(\{\neg\text{Taking\_Pot}\}) \\ &\qquad \times m(\{\neg\text{Taking\_Pot}\} \rightarrow \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad + m(\{\text{Taking\_Pot}, \neg\text{Taking\_Pot}\}) \\ &\qquad \times m(\{\text{Taking\_Pot}, \neg\text{Taking\_Pot}\} \rightarrow \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\ &\qquad = 0.3941 \end{aligned}$$

$$
\begin{aligned}
m_3(\{\text{Cooking\_Meal}\}) \\
&= m(\{\text{Opening\_Water\_Tap}\}) \\
&\times m(\{\text{Opening\_Water\_Tap}\} \to \{\text{Cooking\_Meal}\}) = 0.5128
\end{aligned}
$$

$$
\begin{aligned}
m_3(\{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\
&= m(\{\text{Opening\_Water\_Tap}\}) \\
&\times m(\{\text{Opening\_Water\_Tap}\} \to \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\
&+ m(\{\neg\text{Opening\_Water\_Tap}\}) \\
&\times m(\{\neg\text{Opening\_Water\_Tap}\} \to \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) \\
&+ m(\{\text{Opening\_Water\_Tap}, \neg\text{Opening\_Water\_Tap}\}) \\
&\times m(\{\text{Opening\_Water\_Tap}, \neg\text{Opening\_Water\_Tap}\} \\
&\to \{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) = 0.4872
\end{aligned}
$$

$$
\begin{aligned}
m_4(\{\text{Cooking\_Meal}\}) = m(\{\neg\text{Using\_Burner}\}) \times m(\{\neg\text{Using\_Burner}\} \to \{\neg\text{Cooking\_Meal}\}) \\
= 1.0
\end{aligned}
$$

The joint mass is calculated as follows.

$$
\begin{aligned}
m(\text{Cooking\_Meal}) \\
&= m_1(\text{Cooking\_Meal}) \oplus m_2(\text{Cooking\_Meal}) \oplus m_3(\text{Cooking\_Meal}) \\
&\oplus m_4(\text{Cooking\_Meal}) = 0.5249
\end{aligned}
$$

$$m(\neg\text{Cooking\_Meal}) = 0.4187$$

$$m(\{\text{Cooking\_Meal}, \neg\text{Cooking\_Meal}\}) = 0.0564$$

Then, the mass is compared with m_t to determine if the activity has reached the confidence level required to be recognized.

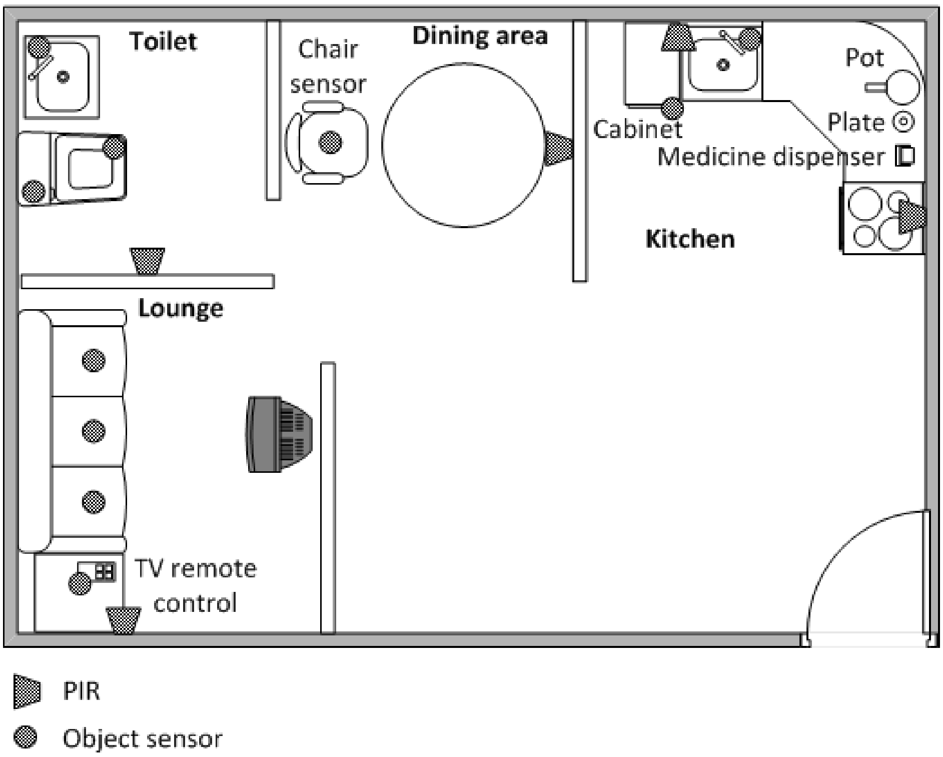


Figure 5.4: The sensor layout in the IELAB.

5.7 Experimental Setup for Activity Recognition

This section presents experiments performed with data recorded in two different scenarios and setups: the CASAS project [180] by Washington State University (WSU) and the intelligent environment laboratory (IELAB) at University of Auckland. The CASAS dataset contains sensor data recorded in a smart home, during which 20 undergraduate students were requested to perform a set of activities in a continuous manner. The activities are performed in following order: washing hand (60 s), cooking meal (420 s), taking medicine (30 s), having a meal (120 s) and washing dishes (240 s). The whole experiment yields more than five hours of data. The sensors are PIR sensors to detect human presence, “item sensors” to indicate the state of a given object and “door sensor” to indicate a door is opened or closed. The IELAB dataset is recorded in a laboratory which is partitioned into four areas: lounge, toilet, kitchen and dining as shown in Figure 5.4. Similarly, each area is supplied with PIR sensors to detect presence, “chair sensors" to indicate the occupancy of a chair, "item sensors" to indicate the state of a given object such as medicine dispenser and "door sensors" to indicate a door is opened or closed.

10 persons (age: 30 ± 3.3 years) were requested to perform a set of activities twice in a continuous manner as described in Table 5.4. The whole experiment yields more than two hour and a half of data. Different to the CASAS case, participants do not follow a specific order when performing the activities. In both datasets, we randomly chose 2 out of the 20 data to model the activity ontologies. The threshold m_t represents the amount of belief or the strength of the evidence that is required for an activity to be recognized, ranging from 0 (indicating no evidence) to 1 (denoting certainty). Therefore, we intuitively set m_t to 0.5, which we believe a reasonable degree of belief to recognize an activity has been carried out, since the value requires the system to have at least half of the evidence to conclude an activity has been carried out.

Table 5.4: Description of the experimental scenario.

Activity	Description	Duration, s
Cooking meal	The subject takes a pot from the cabinet, pours water into the pot and boils it.	180
Taking medicine	The subject takes the medicine container and pour water into a glass.	40
Having meal	The subject takes a plate, a glass of water and the medicine container to the dining area and eats the meal.	120
Washing dishes	The subject cleans all of the dishes and returns them to their places.	60
Toileting	The subject sits on the toilet seat, flush the toilet and washes his/her hands in the sink.	60
Watching TV	The subject turns on the television and rests on the sofa.	120

5.8 Results and Discussion

5.8.1 Comparison with Traditional Ontological Reasoning

The proposed approach (OT-DS) is implemented and compared with the traditional ontological reasoning (OT). We used a Java API called OWL API to access and manipulate the ontology and HermiT [181] as the reasoner. Table 5.5 and Table 5.6 show the recognition results. We evaluate the performance of the proposed approach in terms of accuracy (recall) or the number of correctly classified activities. In addition to the accuracy, we calculated the precision metric as shown in Figure 5.5 and Figure 5.6. Recall and precision are defined as:

$$Recall = (\mathrm{TP}/(TP + FN))\times 100, \quad Precision = (\mathrm{TP}/(TP + FP))\times 100 \tag{5.23}$$

where TP is the number of activities correctly recognized, FP is the number of activities incorrectly recognized and FN is the number of activities that have not been recognized. The rows correspond to different activities. The second column reports the number of contexts (action concepts) used when defining the corresponding activities. The third, fourth and fifth columns report, respectively, the results of the recognition of OT, OT-DS and HMM. Comparison with HMM is presented in Section 5.8.2. As shown in Table 5.5 and Table 5.6, OT approach performed generally well in all activities except "Washing dishes" from CASAS dataset with only 30% of them are detected, achieving accuracy rates of 76% (CASAS) and 70.8% (IELAB). Tracing back to the sensor dataset, failures of OT approach are due to missing sensor data. As for "Washing dishes", it is observed that the "item sensor" data associated with "returning bowl" context is missing from the remaining 14 data. As a result, the context which is required for the activity to be recognized is not inferred. OT-DS approach improves the results of OT in every activity considered, in which the recognition accuracy is significantly increased by 17% and 22.5% for CASAS and IELAB respectively. "Washing dishes" of CASAS dataset has seen the most improvement whereby the number of correctly classified activities is increased by 50%. It successfully aggregated the contexts, quantified the uncertainty in sensor data and concluded the activity is actually being performed. Overall, the proposed approach improves the recognition accuracy of OT approach by 19.75%. We evaluate the performance of the proposed approach in terms of accuracy (recall) or the number of correctly classified activities. In addition to the accuracy, we calculated the precision metric as shown in Figure 5.5 and Figure 5.6.

From the theoretical point of view, the mass of an activity being performed will be increased as more evidences are available, in which the evidences are the action concepts with active state. Analysis of the results is performed to determine the factors that affect the mass of the activities. In general, an activity has a reasonable degree of belief when half or more of its actions are in active state. Consider “Washing dishes” which is composed of “Opening Water Tap”, “Returning Pot” and “Returning Plate”. The activity has a mass of 0.670 when “Opening Water Tap” and “Returning Pot” are active and “Returning Plate” is inactive. Besides the number of evidences, evidential mappings of an action can also affect the belief value of an activity. For instance, “Watching TV” is composed of “Taking TV Remote Control” and “Sitting on Sofa”, in which the evidential mappings are 0.864 and 0.682 respectively. The mass of the activity is 0.561 if “Sitting on Sofa” is active while “Taking TV Remote Control” is uncertain. The mass is increased to 0.672 when “Taking TV Remote Control” is active and “Sitting on Sofa” is in uncertain state. We believe the calculated masses are sensible because “Taking TV Remote Control” is more indicative than “Sitting on Sofa” for recognizing “Watching TV”.

Table 5.5: Activity recognition with CASAS dataset.

	Contexts	Detected (OT), %	Detected (OT-DS), %	HMM, %
Washing hand	1	100.0	100.0	100.0
Cooking meal	4	70.0	100.0	100.0
Taking medicine	2	85.0	90.0	90.0
Having meal	2	95.0	95.0	90.0
Washing dishes	3	30.0	80.0	85.0
Accuracy		76.0	93.0	93.0

Table 5.6: Activity recognition with IELAB dataset.

	Contexts	Detected (OT)	Detected (OT-DS)	HMM
Cooking meal	4	65.0	95.0	90.0
Taking medicine	2	90.0	95.0	100.0
Having meal	2	75.0	90.0	100.0
Washing dishes	3	55.0	100.0	85.0
Toileting	3	65.0	85.0	95.0
Watching TV	2	75.0	95.0	90.0
Accuracy		70.8	93.3	93.3

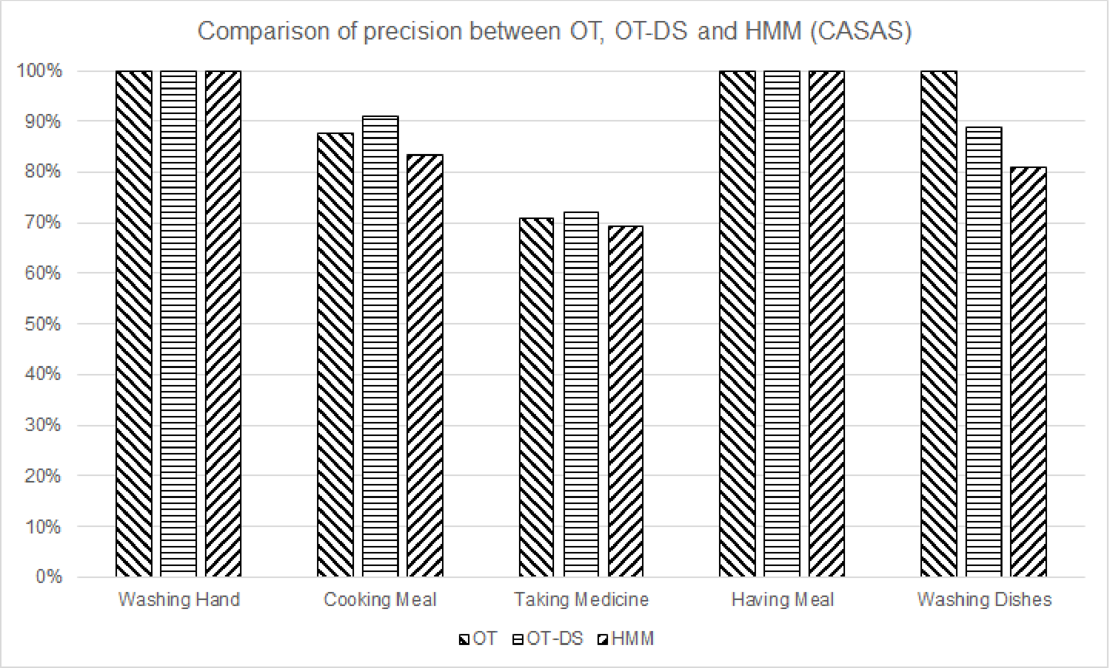


Figure 5.5: Comparison of precision between OT, OT-DS and HMM in recognizing activities using CASAS dataset.

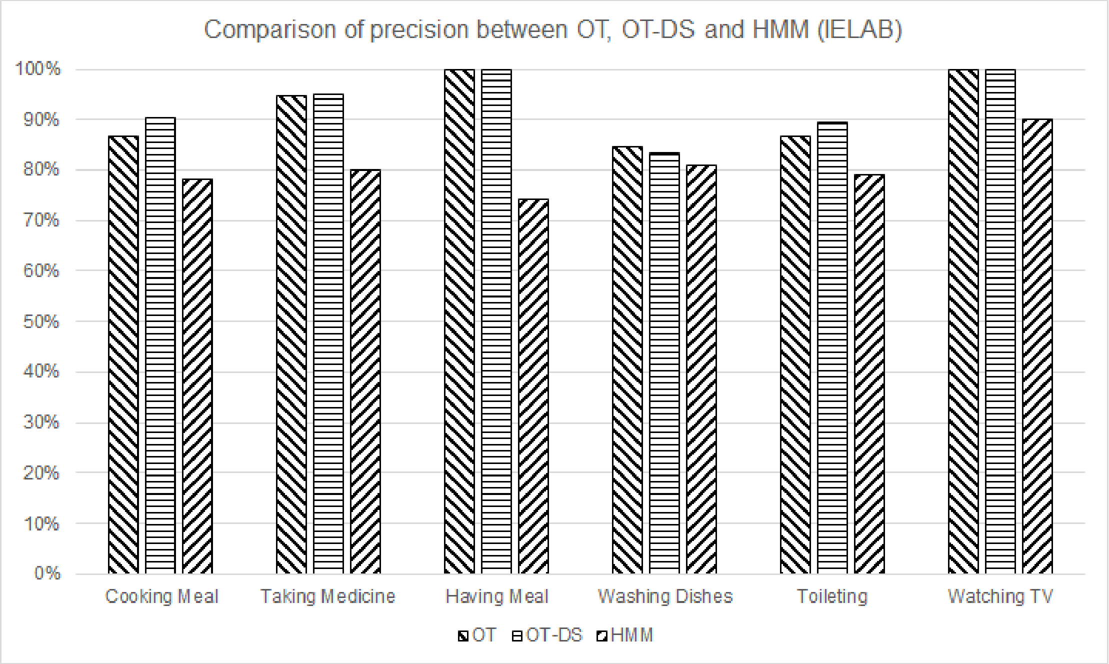


Figure 5.6: Comparison of precision between OT, OT-DS and HMM in recognizing activities using IELAB dataset.

Another factor that can affect the mass of an activity is when aggregating inactive actions. Consider "Cooking Meal" which is defined in (18). The mass of "Cooking Meal" is 0.813 when "Using burner" and "Opening water tap" are active while the other actions are uncertain. However, the aggregation yields lesser value (0.768) if "Opening water tap", "Taking Pot" and "Opening cabinet" are active and "Using burner" is inactive, although there is an extra evidence in the later aggregation. This is because, the first aggregation does not involve inactive actions while the second aggregation involves inactive state which would reduce the mass when combining the evidences. Next, "Using burner" which has high evidential mapping value is active in the first aggregation while in the second aggregation, "Using burner" is inactive.

Figure 5.7 and Figure 5.8 illustrate the average mass for each activity for CASAS and IELAB datasets, respectively. In Figure 5.7, average mass for "Washing hand" and "Having meal" are not available. This is because "Washing hand" is perfectly recognized by OT approach and no activity of "Having meal" is recognized by OT-DS. As shown in Figure 5.7 and Figure 5.8, average masses for all activities are over 0.6. In the experiments, 44 out of 59 activities are recognized through the integration of DS theory. Out of 44 activities, 40 activities are recognized when half or more of their actions are in active state. Therefore, a mass of 0.5 and above is a good threshold to recognize activities with a high degree of confidence. The mass threshold against the recognition accuracy for both datasets is given in Figure 5.9.

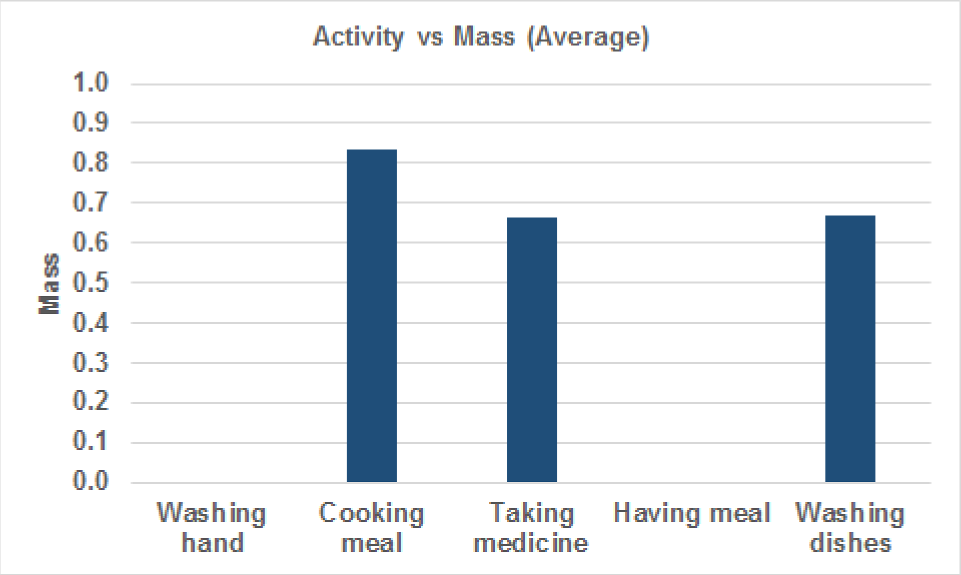


Figure 5.7: Average mass for each activity for CASAS dataset.

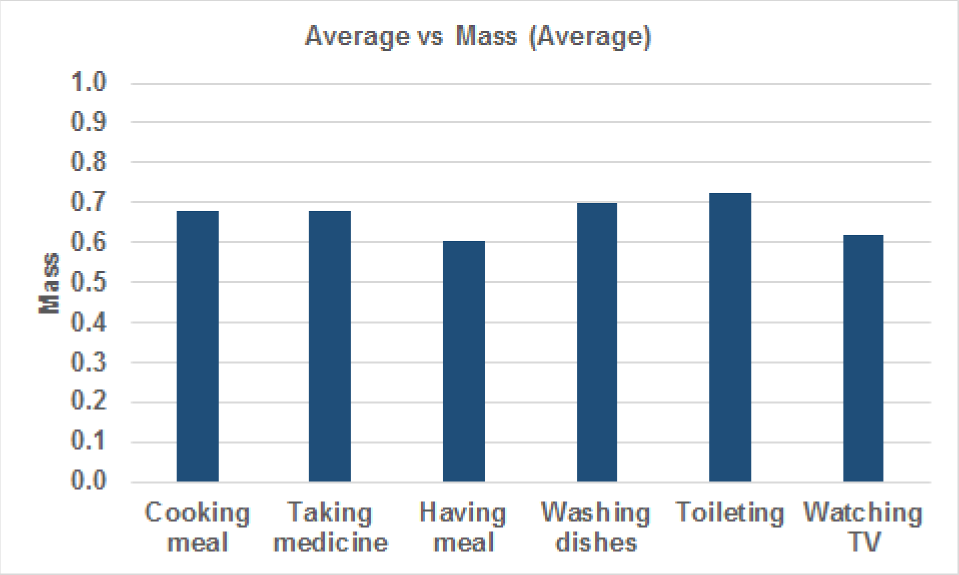


Figure 5.8: Average mass for each activity for IELAB dataset.

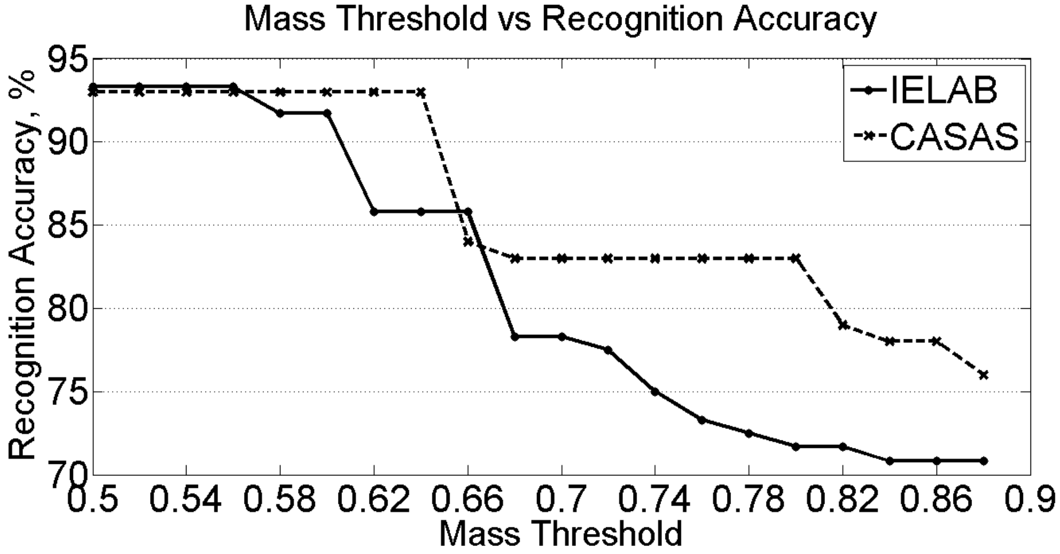


Figure 5.9: Recognition accuracy against increasing mass threshold.

5.8.2 Comparison with Data-driven Approach

We have also implemented the Markov Model (HMM), the approach used on CASAS dataset which is described in [56]. In both datasets, we randomly chose 10 out of the 20 data to train the model. The approach is implemented in MATLAB using Hidden Markov Model (HMM) toolbox. Sequences of sensor events for each activity from the datasets are created in MATLAB. Then, utilizing HMM toolbox, a HMM is created with random transition (A) and emission matrix (B). Initially, the model is created using 5 hidden states and is trained repeatedly using the sequences of sensor events of an activity until it terminates. The same process was repeated by increasing the number of hidden states. Afterwards, a random observation sequence of the activity is used to calculate the log likelihood of the sequence, $\log(O|A,B)$ for all the constructed HMM. The model with minimum number of hidden states with maximum log likelihood is considered as the optimized model for the activity. The procedure is repeated for all activities from both datasets.

From the results in Table 5.5 and Table 5.6, we can see that HMM approach can accurately recognize the activities with the accuracy above 85%. This is because a data-driven approach such as HMM is capable of handling missing sensor data in the datasets. It is shown in "Washing dishes" from CASAS dataset, whereby the number of correctly classified activities has improved by 55%. The HMM achieved the same average recognition accuracy as the OT-DS approach. Overall, the average of recognition accuracy for HMM is equal to OT-DS (93.15%), which is 19.75% higher than OT. In comparison with several state-of-the-art systems which are based on data-driven approaches, OT-DS appears promising in several regards. First, its performance compares favorably with the previously proposed systems. Kabir et al. [133] proposed a two-layer HMM that represents the mapping between sensor data and activity. The first layer predicts the activity class (group of activities according to location) using the sensor location information. Then, the corresponding second layer of the HMM is used to classify the activity using the sensor data. The proposed model is evaluated using three smart home datasets, achieving an average accuracy of 69.3%. In [182], an improved hierarchical HMM is proposed in the form of non-parametric hierarchical HMM, in which the number of hidden states is estimated from the data using Dirichlet Process. The classification is performed by capturing the relationships between actions and the activity labels using multinomial logistic regression. The proposed model achieved an average accuracy of 92.5% on a motion capture dataset.

In [183], an interval temporal Bayesian network is proposed to overcome the limitation of the traditional graphical models in modeling complex temporal relationships between events. The proposed model combines Bayesian network with interval algebra network (i.e. Allen's 13 temporal relations) to model multiple temporal dependencies between events. Two real video data are used to evaluate the proposed model and the results show it outperformed other competing models, achieving an average accuracy of 66.9%. Similar work could be found in [184], whereby a graphical model is proposed by combining Chinese Restaurant Process model with Allen's 13 temporal relations. The proposed model is evaluated using two real video data and a smart home dataset, achieving an average accuracy of 90%. An improved prediction model based on Bayesian network is proposed for activity recognition [185]. Unlike the traditional Bayesian network, the proposed model utilizes current features and next features to classify the next activity, whereby the next features are predicted using current activity. The experiments show the proposed model improved the traditional Bayesian model by 14.92%. Secondly, the aforementioned approaches require a sufficient amount of training datasets in order to obtain a reliable predictive model. In application with a larger number of activities and actions, it would be more difficult to obtain a sufficient dataset as users perform activities in a variety of ways and orders [35], [50], [140]. In our proposed approach, the activity model is based on ontological modeling which does not require a large amount of training data and domain knowledge can be obtained from domain experts or users to determine the evidential parameters. This brings two benefits. First, it allows non-technical users to encode knowledge about the domain and provide reasoning services. Second, it is technically applicable and scalable to real-world scenarios. In addition, the proposed approach can be applied to other applications beyond the domain of activity recognition. It only requires the domain knowledge and evidential parameters to be specified in the ontology.

5.9 Conclusions

In this thesis, we have introduced a novel reasoning algorithm to support ontological reasoning for activity recognition with Dempster-Shafer theory of evidence. The approach utilizes ontological reasoning mechanisms of description logic and overcomes the limitation of ontology based models in terms of their inability to handle uncertainty due to missing sensor data. In the proposed algorithm, the associated concepts are aggregated and the degree of belief is computed to make decision whether the corresponding activity has been performed or not. In addition, a four-layered activity ontology, which incorporates the representation of evidential parameters, is proposed. Experiments performed on an internally collected and a public smart home datasets have returned promising results in terms of recognition accuracy. The results show that the proposed algorithm is able to deal with imperfect observations and gives improved the accuracy over the ontological reasoning performance by 19.75%. It is also showed that the performance of the proposed algorithm is comparable to the data-driven approach without requiring a large amount of data. The activity ontology models contexts from embedded sensors. In future work, we plan to include context from wearable sensor which could also be used to deal with missing sensor data. Also, we plan to investigate the applicability of the algorithm for use in real-time scenarios.

6

Ontology-based Sensor Fusion Activity Recognition

6.1 Introduction

This chapter describes the hybrid approach to activity recognition using wearable and ambient sensors. The best of both sensing approaches is harnessed to achieve a robust and comprehensive context-aware activity recognition system by exploiting contextual information from the user and environment. We introduce the methodology of ontology-based sensor fusion that fuses user context with environmental contexts provided by the physical activity recognition system (Chapter 3 and 4) and sensors embedded in environment respectively as illustrated in Figure 6.1. The physical activity recognition system recognizes the physical activity of the user such as walking, standing, sitting and lying down while the ambient sensors provides the user-object interaction and location contexts within the environment. The contexts are mapped into the activity ontology, fused and then reasoned by the ontological reasoner to infer the activity being performed by the user. The ontological reasoner is enhanced with uncertainty handling by integrating the DS theory as described in Chapter 5. The focus of this approach is to highlight its advantages and the methodology of modeling the contexts using ontology. Firstly, the approach can resolve the uncertainty of imperfect observation due to sensor errors. For instance, the system would fail to recognize activity of having meal if the chair sensor is used alone and is not activated or the data is lost during transmission. In this case, sitting context provided by wearable sensor-based physical activity recognition system could be used to confirm the user is actually sitting on the chair. Secondly, the approach allows additional and more precise inference of information about activities because it fuses contextual information from both sensing approaches. Finally, the approach can infer activities which do not involve user-object interaction context such as wandering in the lounge, exercising etc.

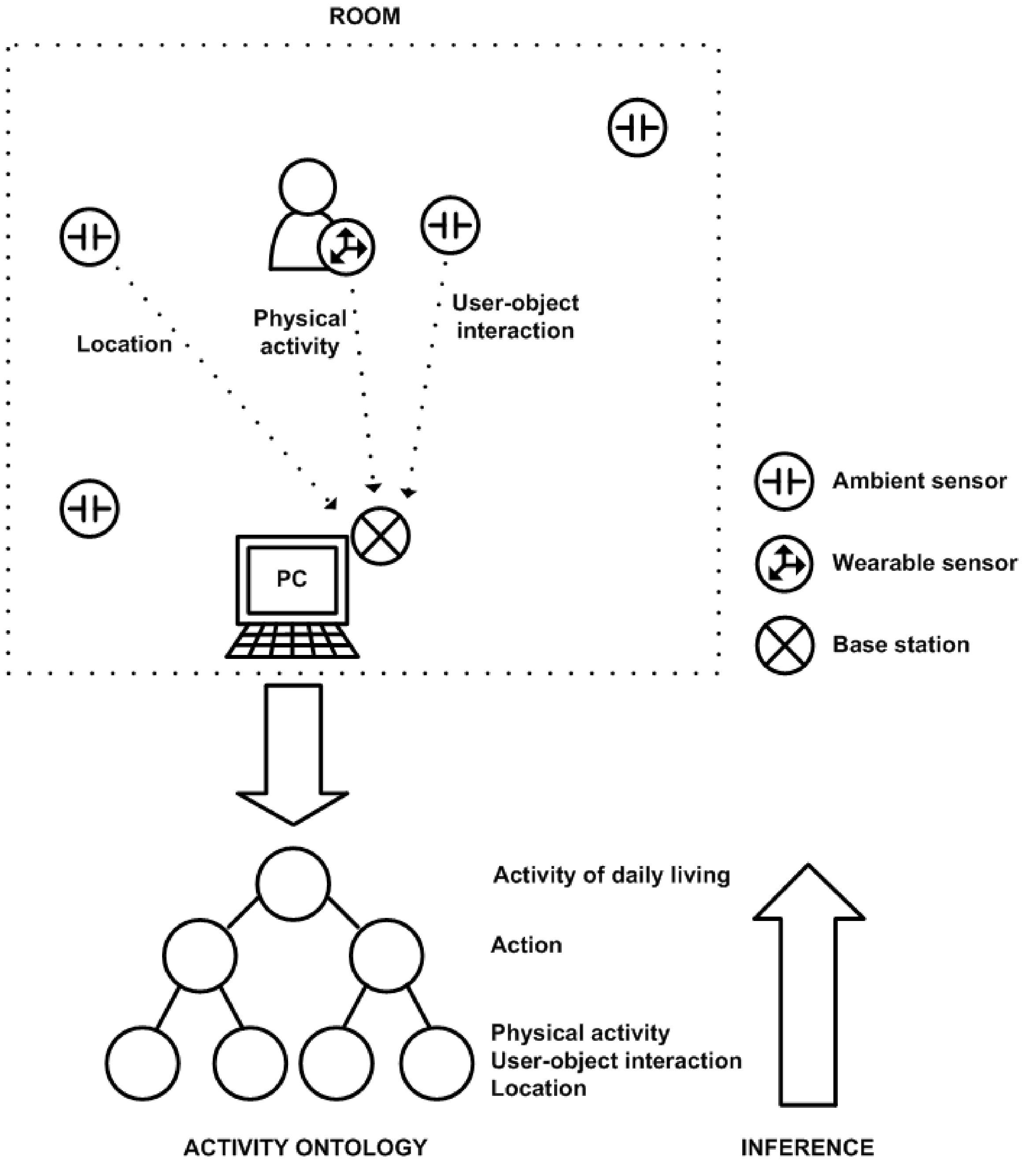


Figure 6.1: A hybrid sensing approach for context-aware activity recognition.

6.2 Related Works

Wearable sensor and dense sensing are complementary and can be used in combination to improve activity recognition results. Numerous RFID-based activity recognition systems have been proposed in which activities are recognized when tagged objects are detected by a wearable RFID readers [51], [186]–[189]. In [190], RFID has been used in combination with vision sensor for activity recognition. However, the systems are deemed to be less feasible because RFID sensors are prone to false reading [191]–[193]. Recently, accelerometers as the wearable sensors have been used in combination with RFID sensors for activity recognition. Gu et al. [194] fused sensor information from accelerometers and RFID sensors and recognized ongoing activities by using Emerging-Pattern-based technique. Three accelerometers are used which are worn on both hands and waist. A similar work has been done in [195] where three accelerometers and RFID sensors are used for activity recognition. In this work the inference is performed by fusing body postural orientation and hand activities, which indicate the activity being undertaken. In [196], four accelerometers and RFID sensors are used for nursing activity monitoring and the recognition is achieved by using support vector machine technique. All the aforementioned works used multiple accelerometers and RFID sensors system which is not feasible for long-term activity monitoring due to multiple sensor attachments and the drawbacks of RFID sensors. Accelerometers have also been used in combination with vision sensors for activity recognition [197]–[199]. In the literature, vision sensors are used to recognize activities that cannot be classified by wearable sensors only. For example, activities such as eating and reading can be classified by fusing accelerometer and vision sensors. However, the use of vision sensors in activity recognition is considered invasive and suffers from issues relating to privacy and ethics [200], [201].

Ambient sensors such as pressure sensors, contact switches and Passive InfraRed (PIR) sensors have been used in conjunction with wearable sensors to provide context information about the events occurring in the environments. Atallah et al. [75] proposed a real-time activity recognition system using a single accelerometer and ambient sensors such as door sensor, PIR sensor, pressure sensitive sensor etc. Activities are categorized into four classes according to their intensities. The wearable sensor-based system provides information on activity level classes and information from ambient sensors is used to indicate the type of activity that is being performed in a particular class. The classification is achieved by using Bayesian classifier and multivariate Gaussian model. De et al. [67] proposed an activity recognition system using wearable sensors (accelerometers, gyroscope etc.) attached to waist, back, leg and wrist and Bluetooth beacons are used as ambient sensors to indicate the user's location. The activity classification is selected from one of the sensor nodes, in which the selection is based on the relationship between the activity and the node's position. The classification is performed using conditional random field classifier. However, both systems do not consider contextual information such as current location, user-object interaction and physical activity that can be exploited to derive the activities. Ge and Xu [202] proposed an activity recognition system using accelerometers and gyroscopes as wearable sensors and wireless sensor networks as ambient sensors to provide location contexts based on received signal strength indicator (RSSI). The physical activity and location contexts are used to distinguish activities such as reading books and watching TV. The classification is performed using Markov model. Similar approach is found in [203], whereby a smartphone and infrared motions sensors are used as wearable and ambient sensors respectively and hidden Markov model is used to classify the activities. In both literatures, ontology is not used to model the contextual information from wearable and ambient sensors, which we consider to be the most advantageous and convenient tool for activity modeling.

Ontology has been used to construct activity models which can be processed by artificial intelligence-based inference. Riboni and Bettini [154] performed activity recognition using two accelerometers worn on the right wrist and waist to recognize user body and hand gestures. The ontologies represent and model the relations among contextual information such as activities, symbolic locations, objects, and time. Khattak et al. [199] proposed an ontology-based activity recognition by using accelerometer, gyroscope, vision and ambient sensors. However, object context is not specified in the ontology for activity inference. Wongpatikaseree et al. [204] proposed an ontology-based activity recognition system in which the activity model specifies human posture, location and object contexts. The human posture is modeled to solve the problem of several object sensors activated at the same time which is called "ambiguous activity problem". By modeling the activity with its corresponding human posture, the on-going activity can be ascertained during context reasoning. However, our proposed approach goes beyond the ones reported in [154], [199], [204]. The proposed approach is capable of handling not only uncertainty, but also provide more precise activity recognition, even for activities that do not involve interaction with objects. Furthermore, our work uses a single tri-axial accelerometer while accelerometer, gyroscope and vision sensor are used in [199], and ultrasonic sensor is used in [204], which has a minimum sensing distance and is prone to false triggers.

The proposed methodology is validated using an internal dataset which contains 10 ADL generated from experiments on a group of 20 subjects with an accelerometer on their right waist. In addition, we exploit the publicly available OPPORTUNITY dataset for benchmarking. The dataset is generated in a sensor rich environment from four subjects with sensor nodes (accelerometers, gyroscopes and magnetometer) attached to 17 different body parts. The wearable sensor nodes provide hand gestures contextual information in addition to physical activity contexts. This allow us to verify the efficacy of the proposed sensor fusion approach in different application domains.

6.3 Ontology-based Sensor Fusion

The activity ontology is organized into four layers of contexts as illustrated in Figure 6.2. As shown in Figure 6.2, the sensor layer consists of wearable and ambient sensors, and the contexts consists of physical activity, object interaction and location contexts. At this layer, the contexts are fused to describe action context. In addition to the five main concepts, Physical_Activity concept is implemented to model the physical activity context.

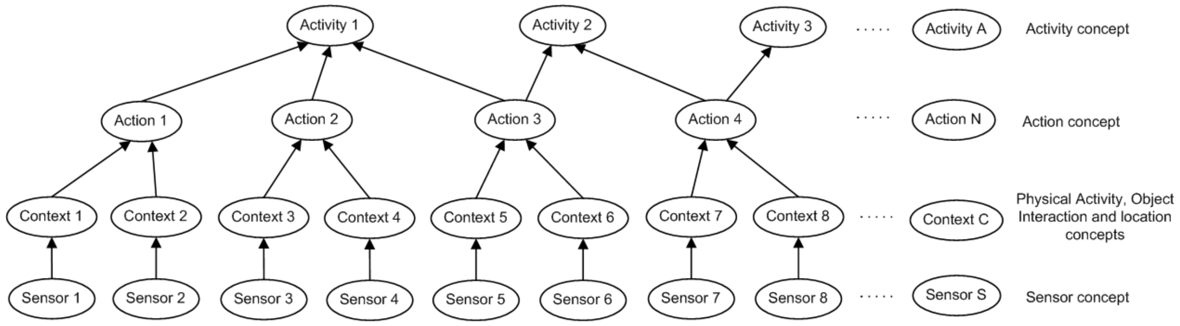


Figure 6.2: The generic conceptual activity ontology organized into four layers of concepts.

Similar approach of modelling ambient sensors is used to model the states of wearable sensor. The following definition defines walking state which is one of the states of wearable sensor-based physical activity recognition. A subconcept called $\mathrm{WS\_Body\_WALK}$ is defined to represent the walking state.

$$\mathrm{Wearable\_Sensor\_Body} \sqsubseteq \mathrm{Sensor}$$

$$\mathrm{WS\_Body\_WALK} \sqsubseteq \mathrm{Wearable\_Sensor\_Body} \tag{6.1}$$

$\mathrm{Physical\_Activity}$ models the physical activity context of the user. Each relevant physical activity in the system corresponds to a different subconcept of $\mathrm{Physical\_Activity}$ concept in the ontology that represents a physical activity. The subconcepts are defined through $\mathrm{hasFluent}$ role to specify that the wearable sensor must be in a given state for the physical activity to be inferred. For example, the concept of walking context is defined as follows:

$$\begin{gathered}\mathrm{Walking} \sqsubseteq \mathrm{Physical\_Activity} \\ \mathrm{Walking} \equiv \exists \mathrm{hasFluent.WS\_Body\_WALK}\end{gathered} \tag{6.2}$$

which describes the user is walking when the state of wearable sensor is walking. The $\mathrm{hasFluent}$ role is used to specify the state of the corresponding sensor for the concept to be inferred.

Missing sensor data would cause the activity recognition system fails to infer an activity. For instance, if the “chair sensor” is not activated due to sensor error or the sensor data is lost during transmission, $\mathrm{Sitting\_On\_Dining\_Chair}$ will not be inferred and as a result the corresponding activity is not recognized. Physical activity context can be used to support the context reasoning by including the physical activity concept in the concept definition of $\mathrm{Sitting\_On\_Dining\_Chair}$ as follows:

$$\mathrm{Sitting\_On\_Dining\_Chair} \equiv \mathrm{Interval} \sqcap \mathrm{In\_Dining\_Room} \sqcap (\mathrm{Dining\_Chair\_Is\_Seated} \sqcup \mathrm{Sitting}) \tag{6.3}$$

In this definition, $\mathrm{Sitting\_On\_Dining\_Chair}$ will be inferred even when “chair sensor” data is missing if the wearable sensor is in sitting state, which is modeled in the ontology as follows:

$$\begin{gathered}\mathrm{Sitting} \sqsubseteq \mathrm{Physical\_Activity} \\ \mathrm{Sitting} \equiv \exists \mathrm{hasFluent.WS\_Body\_SIT}\end{gathered} \tag{6.4}$$

It has to be noted that the concepts will not be inferred if there is uncertainty due to the missing sensor data associated with location concept.

Uncertainty might arise due to the fact that the activity can be performed in different ways. For example, consider Toileting1 definition as follows:

$$\mathrm{Toileting1} \equiv \mathrm{Opening\_Toilet\_Water\_Tap} \sqcap \exists \mathrm{hasPredecessor.}\big(\mathrm{Pressing\_Toilet\_Flush} \sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Sitting\_On\_Toilet\_Seat})\big) \tag{6.5}$$

where Sitting_On_Toilet_Seat is given as follows.

$$\mathrm{Sitting\_On\_Toilet\_Seat} \equiv \mathrm{Interval} \sqcap \mathrm{In\_Toilet} \sqcap (\mathrm{Toilet\_Seat\_Is\_Seated} \sqcup \mathrm{Sitting}) \tag{6.6 }$$

According to the above definition, toileting won't be inferred by the system if the activity is performed while standing because toilet seat sensor is not activated and wearable sensor is in the state of sitting. This uncertainty can be resolved by defining new concept definition of toileting as follows:

$$\mathrm{Toileting2} \equiv \mathrm{Opening\_Toilet\_Water\_Tap} \sqcap \exists \mathrm{hasPredecessor.}\big(\mathrm{Pressing\_Toilet\_Flush} \sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Standing\_At\_Toilet\_Seat})\big) \tag{6.7}$$

where Standing_At_Toilet_Seat is defined with standing context as a constraint given as follows:

$$\mathrm{Standing\_At\_Toilet\_Seat} \equiv \mathrm{Interval} \sqcap \mathrm{In\_Toilet} \sqcap \mathrm{Standing} \tag{6.8}$$

Some activities do not involve any object interaction such as wandering around in a room. Inferring these activities using contexts from ambient sensors only would be difficult and could raise uncertainty in recognition. However, with physical activity context, wandering around in a lounge can be described in activity ontology as follows:

$$\mathrm{Wandering\_In\_Lounge} \equiv \mathrm{Walking\_In\_Lounge} \sqcap \exists \mathrm{hasPredecessor.}(\mathrm{CloserThan3min} \sqcap \mathrm{Walking\_In\_Lounge}) \tag{6.9}$$

where Walking_In_Lounge is defined by fusing walking context and user in the lounge context as follows:

$$\text{Walking\_In\_Lounge} \sqsubseteq \text{Interval} \sqcap \text{In\_Lounge} \sqcap \text{Walking} \qquad (6.10)$$

Additional information about the activity being performed in the environment can be inferred by fusing contexts from wearable and ambient sensors. Consider the following simple scenario. A resident walk into the lounge. In the first instance the lounge PIR sensors are active. Then, the system detects sofa sensor state is PRESENT. According to the following concept description, the sensor activation indicates the resident is resting on the sofa.

$$\text{Resting\_On\_Sofa} \equiv \text{Sitting\_On\_Sofa} \sqcap \exists \text{hasPredecessor}.(\text{CloserThan5min} \sqcap \text{Sitting\_On\_Sofa}) \qquad (6.11)$$

where Sitting_On_Sofa is defined as follows:

$$\text{Sitting\_On\_Sofa} \equiv \text{Interval} \sqcap \text{In\_Lounge} \sqcap (\text{Sofa\_Is\_Seated} \sqcup \text{Sitting}) \qquad (6.12)$$

However, activation of the sofa sensor does not necessarily indicate the user is sitting on the sofa because the user might be lying down on the sofa, thus activating the sensor. In order to conclude the user is actually lying down on the sofa, the physical activity context provided by the wearable sensor is used in combination with other contexts to infer the action of lying on the sofa. Then, a more precise activity definition such as resting on sofa can be composed as follows.

$$\text{Taking\_A\_Nap} \equiv \text{Lying\_On\_Sofa} \sqcap \exists \text{hasPredecessor}.(\text{CloserThan5min} \sqcap \text{Lying\_On\_Sofa}) \qquad (6.13)$$

where Lying_On_Sofa is given as follows.

$$\text{Lying\_On\_Sofa} \equiv \text{Interval} \sqcap \text{In\_Lounge} \sqcap \text{LyingDown} \sqcap (\text{Sofa\_Is\_Seated} \sqcup \text{LyingDown}) \qquad (6.14)$$

6.4 Experimental Results

Two datasets, internally collected (IELAB) and OPPORTUNITY human activity dataset, are used to evaluate the performance of the proposed approach. The public dataset contains a set of complex naturalistic activities collected in a sensor rich environment [205]. In the experiments, we compared the accuracy of activity recognition systems without wearable sensor (AR) and with wearable sensor (ARW). In addition, we combine the ARW approach with OT-DS approach (later referred to as ARW-DS) described in Chapter 5. We evaluate the performance of the approaches in terms of accuracy (recall). In addition, we also calculated precision and F-score metrics.

6.4.1 IELAB: Intelligent Environment Lab Dataset

The IELAB dataset is recorded in laboratory described in Section 5.7. Wearable sensor is attached on the right waist to capture the physical activity of the subjects. 20 persons (age: 30.2 ± 3.2 years) were requested to perform a scenario in a continuously one after another with no specific order of the activities. A single experiment takes about 8 min. The activities are given in Table 6.1. The first column lists the activities and their duration. The second column lists the concept definition of the activities. The third column lists the actions that fuse physical activity context in their definitions. We randomly chose 2 out of the 20 data to model the activity ontology.

Table 6.2 shows the comparison of recognition accuracy of AR and ARW approaches. As can be seen in Table 6.2, AR approach performs generally well in recognizing the activities considered in the experiments, in which the number of correctly classified activities are above 75% except “Washing dishes” and “Watching TV”. An accuracy rate of 80.7% is reported. Tracing back to the sensor dataset, the failures are due to sensor observation errors such as PIR is not activated when it should be and missing sensor data which resulted in the associated action concepts to recognize the activities are not inferred. For instance, out of the seven “Watching TV” activity which are not recognized due to missing data, four of them are missing the “TV remote control sensor” data and four are missing the “sofa sensor” data, which the associated actions are “taking TV remote control” and “sitting on sofa” respectively.

From the results, we can see that ARW approach improves the recognition accuracy, in which the number of correctly classified for all activities are above 15. For instance, recognition of "Watching TV" is improved, whereby sitting context is fused to infer the action of "sitting on sofa", and a result the activity is recognized. In addition, ARW approach is also capable of dealing with missing data by inferring the synonym activity. This is shown in the results, whereby "Toileting2" activity is recognized when the activity is performed while standing instead of sitting. This shows that the system is able to better deal with uncertainty due to missing data by including physical activity contexts in the activity model. The experiments are also used to test and verify the capability of ARW approach in inferring additional information about the activities and recognizing activities which do not involve object contexts. In the experiments, ARW approach successfully infers additional information about "Resting on sofa" activity, whereby reasoning engine returns "Taking a nap" activity when lying down context is fused to infer the activity. ARW approach also successfully recognizes "Wandering in lounge" activity by fusing walking context. A success rate of 91.5% is reported (183 correctly classified activity out of 200). Failures are due to missing location sensor data which the proposed approach is not capable of dealing with since physical activity context does not provide information about the person's where about in the environment. The ARW approach also failed to recognize the activities if the missing sensor data involves sensors that capture hand gesture contexts such as "plate sensor" and "TV remote sensor".

As shown in Table 6.2, ARW-DS approach improves the accuracy of ARW significantly, in which all the activities are perfectly recognized. It successfully inferred the activities which are not recognized by ARW approach such as "Watching TV" which are missing the "TV remote sensor" data by quantifying the uncertainty while aggregating the contexts. However, recognition accuracies for "Taking a nap and "Wandering in lounge" are not improved in the experiments. This is because, the action contexts of the activities are not inferred due to the associated PIR sensor data is missing. The average mass of the activities recognized by ARW-DS approach is illustrated in Figure 6.3. Most of the activities have mass over 0.6 except "Watching TV", whereby the average mass is 0.5636. The activities are inferred when half or more of their action contexts are in active state. Overall, the ARW-DS improves the accuracy by 6.0% achieving a recognition rate of 97.5%. The comparison of precision and F-score of the approaches are illustrated in Figure 6.4 and Figure 6.5.

Table 6.1: The activities and their definitions.

Activity	Activity Concept Definition	Action Concept Definition
Cooking meal (60s)	Cooking_Meal ≡ (Using_Burner ⊔ Standing_Near_Burner) ⊓ ∃hasPredecessor. ((Opening_Kitchen_Water_Tap ⊔ Standing_Near_Burner) ⊓ ∃hasPredecessor. (Taking_Pot ⊓ ∃hasPredecessor. (Opening_Drawer ⊓)))	Standing_Near_Burner ≡ Interval ⊓ Near_Burner ⊓ Standing
Taking medicine (20s)	Taking_Medicine ≡ (Opening_Kitchen_Water_Tap ⊔ Standing_Near_Kitchen_Water_Tap) ⊓ ∃hasPredecessor. (Taking_Medicine_Dispenser)	Standing_Near_Kitchen_Water_Tap ≡ Interval ⊓ Near_Kitchen_Water_Tap ⊓ Standing
Having meal (60s)	Having_Meal ≡ Sitting_On_Dining_Chair ⊓ ∃hasPredecessor. Taking_Plate	Sitting_On_Dining_Chair ≡ Interval ⊓ In_Dining_Room ⊓ (Dining_Chair_Is_Seated ⊔ Sitting)
Washing dishes (60s)	Having_Meal ≡ Returning_Plate ⊓ ∃hasPredecessor. (Returning_Pot ⊓ ∃hasPredecessor. ((Opening_Kitchen_Water_Tap ⊔ Standing_Near_Kitchen_Water_Tap)))	Standing_Near_Kitchen_Water_Tap ≡ Interval ⊓ Near_Kitchen_Water_Tap ⊓ Standing
Toileting1 (30s)	Toileting1 ≡ Opening_Toilet_Water_Tap ⊓ ∃hasPredecessor. (Pressing_Toilet_Flush ⊓ ∃hasPredecessor. (Sitting_On_Toilet_Seat))	Sitting_On_Toilet_Seat ≡ Interval ⊓ In_Toilet ⊓ (Toilet_Seat_Is_Seated ⊔ Sitting)

Toileting2 (30s)	Toileting2 ≡ Opening_Toilet_Water_Tap ⊓ ∃hasPredecessor. (Pressing_Toilet_Flush ⊓ ∃hasPredecessor. (Standing_At_Toilet_Seat))	Standing_At_Toilet_Seat ≡ Interval ⊓ In_Toilet ⊓ Standing
Resting on sofa (60s)	Resting_On_Sofa ≡ Sitting_On_Sofa ⊓ ∃hasPredecessor. (CloserThan60sec ⊓ Sitting_On_Sofa)	Sitting_On_Sofa ≡ Interval ⊓ In_Lounge ⊓ (Sofa_Is_Seated ⊔ Sitting)
Watching TV (60s)	Watching_TV ≡ Sitting_On_Sofa ⊓ ∃hasPredecessor. Taking_TV_Remote	Sitting_On_Sofa ≡ Interval ⊓ In_Lounge ⊓ (Sofa_Is_Seated ⊔ Sitting)
Taking a nap (60s)	Taking_A_Nap ≡ Lying_On_Sofa ⊓ ∃hasPredecessor. (CloserThan60sec ⊓ Lying_On_Sofa)	Lying_On_Sofa ≡ Interval ⊓ In_Lounge ⊓ LyingDown ⊓ (Sofa_Is_Seated ⊔ LyingDown)
Wandering in lounge (60s)	Wandering_In_Lounge ≡ Walking_In_Lounge ⊓ ∃hasPredecessor. (CloserThan60sec ⊓ Walking_In_Lounge)	Walking_In_Lounge ⊑ Interval ⊓ In_Lounge ⊓ Walking

Table 6.2: Comparison of recognition accuracy for AR and ARW approaches.

Activity	AR, %	ARW, %	ARW-DS, %
Cooking meal	75.0	90.0	100.0
Taking medicine	100.0	100.0	100.0
Having meal	80.0	95.0	100.0
Washing dishes	70.0	75.0	100.0
Toileting1	75.0	95.0	100.0
Resting on sofa	100.0	100.0	100.0
Watching TV	65.0	85.0	100.0
Taking a nap	N.A.	85.0	85.0
Toileting2	N.A.	100.0	100.0
Wandering in lounge	N.A.	90.0	90.0
Overall accuracy	80.7	91.5	97.5

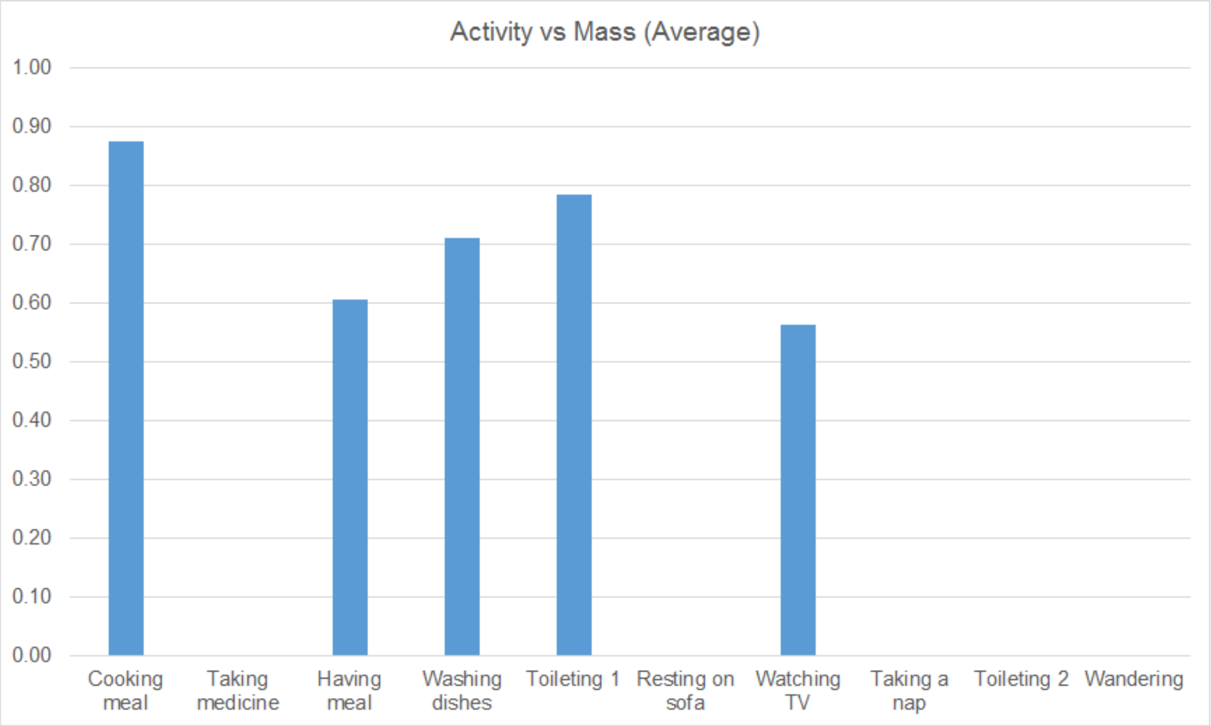


Figure 6.3: Average mass for each activity for IELAB dataset.

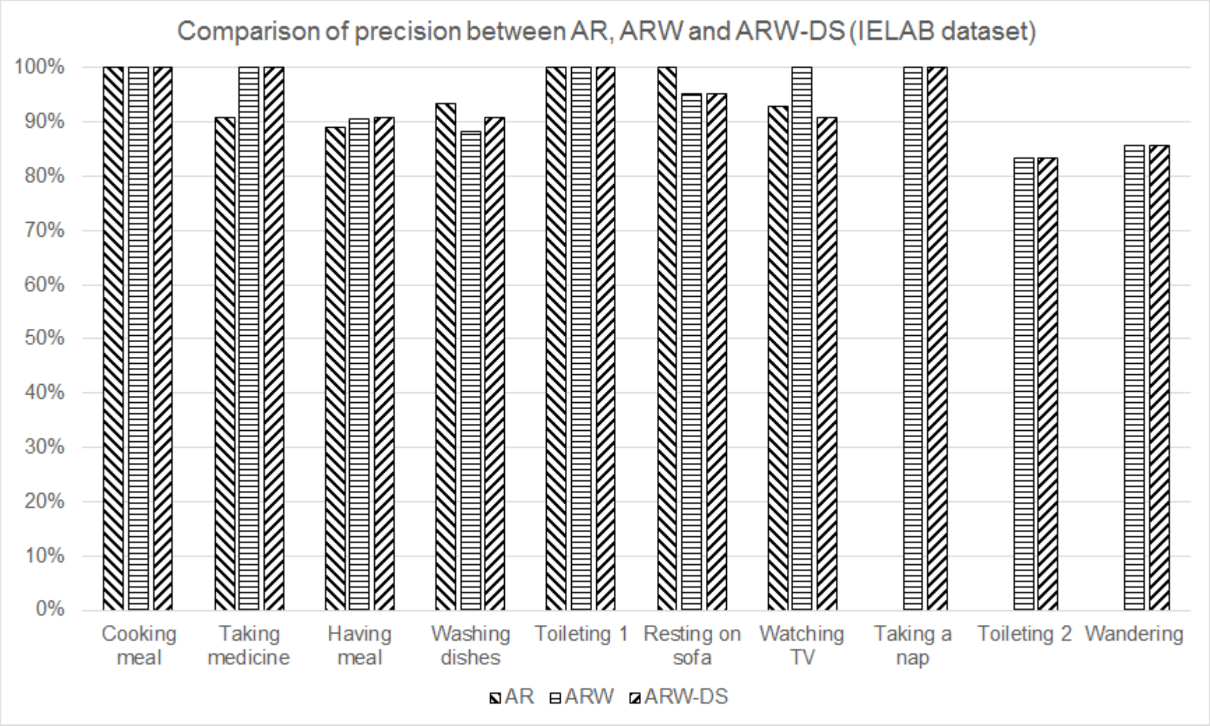


Figure 6.4: Comparison of precision between AR, ARW and ARW-DS for IELAB dataset.

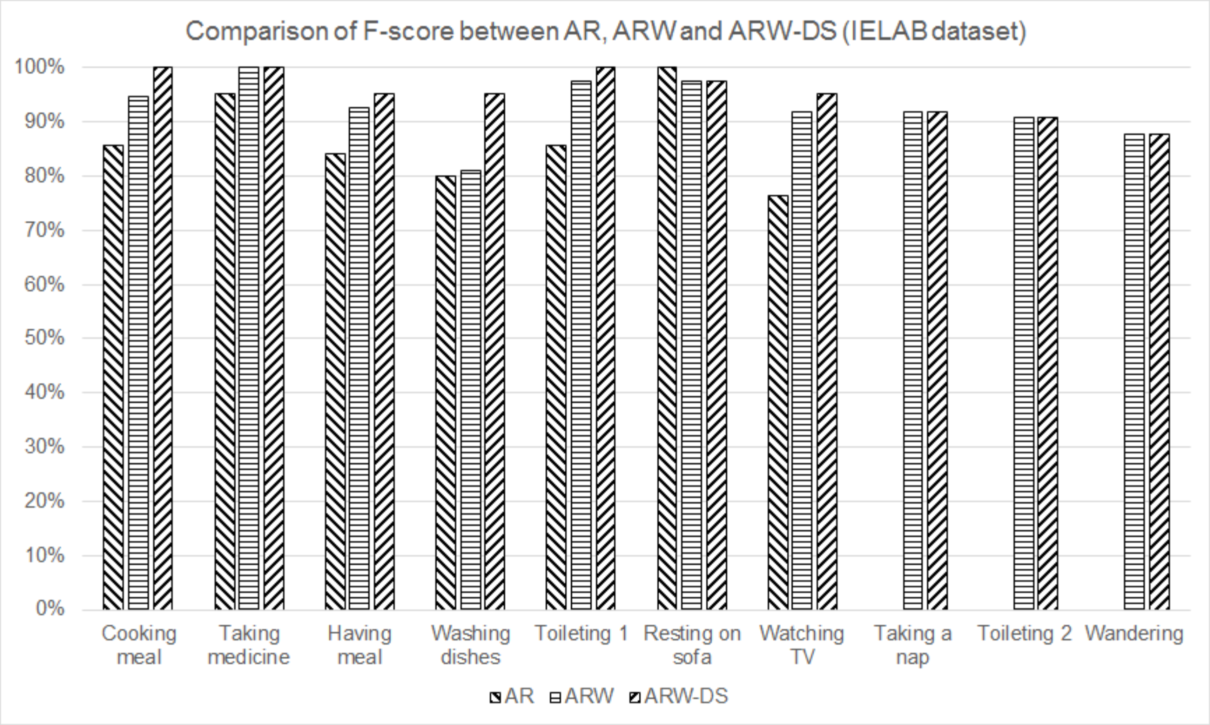


Figure 6.5: Comparison of F-score between AR, ARW and ARW-DS for IELAB dataset.

6.4.2 OPPORTUNITY Dataset

The OPPORTUNITY dataset contains wearable, object and location sensor data recorded in a simulated studio apartment. The room has a kitchen area with a fridge, dishwasher and 3-drawer, a living area with a deckchair and a dining area. Four subjects are involved in the experiments, in which each subject performs a scenario five times as follows. The scenario consists of five activities: relaxing (lying on the deckchair), grooming (walk around in the kitchen area checking the objects in 3-drawer, fridge and dishwasher door are closed properly), making coffee with milk and sugar, preparing sandwich and cleaning (returning the plate and cup in the dishwasher). The wearable sensors consist of accelerometers, gyroscopes and magnetometers, are attached to 17 body parts to sense physical activities and limb movements. Accelerometers are used to capture the user interaction with objects such as plate and cup. Reed switches are used to indicate the status of fridge and dishwasher doors and 3-drawer are, either open or closed. As for location sensors, four ultra-wideband location systems are placed at the corners of the room to estimate the position of the subject. The dataset is annotated, where the annotation not only contains information on the activity classes but also the physical activities (walking, standing, sitting and lying down), left and right hands gestures (opening or closing a door, reaching or releasing an object) and the objects being handled by the hands. For the purpose of this research, the states of the objects are defined as a value in the discrete set {PRESENT, ABSENT} and {OPEN, CLOSE} in order to easily integrate the sensor data in the ontology. We randomly chose 2 out of the 20 data to model the activity ontology.

Differently from the IELAB dataset, the OPPORTUNITY dataset contains hand gestures contextual information in addition to physical activity. In the experiment, we also modeled the contexts (opening and closing with right and left hands) in the ontology to resolve the uncertainty due to missing data. The concepts that represent the states of the wearable sensor and the concept definition of right hand gesture contexts are given as follows:

$$\text{Wearable\_Sensor\_RightHand} \sqsubseteq \text{Sensor}$$

$$\text{WS\_RightHand\_OPEN} \sqsubseteq \text{Wearable\_Sensor\_RightHand}$$

$$\text{WS\_RightHand\_CLOSE} \sqsubseteq \text{Wearable\_Sensor\_RightHand}$$

$$\mathrm{Opening\_With\_RightHand} \equiv \exists \mathrm{hasFluent.WS\_RightHand\_OPEN}$$

$$\mathrm{Closing\_With\_RightHand} \equiv \exists \mathrm{hasFluent.WS\_RightHand\_CLOSE} \qquad (6.15)$$

Similar approach is used for defining left hand gesture contexts. Then, these concepts are included in the corresponding action concepts such as follows:

$$\begin{aligned}&\mathrm{Closing\_Drawer1}\\ &\quad \equiv \mathrm{Interval} \sqcap \mathrm{At\_Drawer}\\ &\quad \sqcap (\mathrm{Drawer1\_Is\_Closed} \sqcup \mathrm{Closing\_With\_RightHand}\\ &\quad \sqcup \mathrm{Closing\_With\_LeftHand})\end{aligned}$$

$$\mathrm{Opening\_Drawer1} \equiv \mathrm{Interval} \sqcap \mathrm{At\_Drawer} \sqcap (\mathrm{Drawer1\_Is\_Closed} \sqcup \mathrm{Opening\_With\_RightHand} \sqcup \mathrm{Opening\_With\_LeftHand}) \qquad (6.16)$$

The concept definition for each activity is given below. Note that, only physical activity (lying down) and location (in lounge) contexts are used to infer “Relaxing” activity since the deckchair is not installed with a sensor.

$$\mathrm{Relaxing} \equiv \mathrm{Lying\_In\_Lounge} \sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Lying\_In\_Lounge} \sqcap \mathrm{CloserThan60sec})$$

$$\begin{aligned}\mathrm{Grooming} \equiv\ &\mathrm{Closing\_Fridge\_Door}\\ &\sqcap \exists \mathrm{hasPredecessor.}\Big(\mathrm{Closing\_Drawer3}\\ &\sqcap \exists \mathrm{hasPredecessor.}\big(\mathrm{Closing\_Drawer2}\\ &\sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Closing\_Drawer1})\big)\Big)\end{aligned}$$

$$\begin{aligned}&\mathrm{Making\_Coffee}\\ &\quad \equiv \mathrm{Taking\_Milk}\\ &\quad \sqcap \exists \mathrm{hasPredecessor.}\Big(\mathrm{Opening\_Fridge\_Door}\\ &\quad \sqcap \exists \mathrm{hasPredecessor.}\big(\mathrm{Taking\_Sugar} \sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Taking\_Cup})\big)\Big)\end{aligned}$$

$$\begin{aligned}\mathrm{Preparing\_Sandwich}\\ &\equiv \mathrm{Taking\_Bread}\\ &\sqcap \exists \mathrm{hasPredecessor.}\Big(\mathrm{Opening\_Drawer3}\\ &\sqcap \exists \mathrm{hasPredecessor.}\big(\mathrm{Taking\_Plate}\\ &\sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Opening\_Drawer2})\big)\Big)\end{aligned}$$

$$\mathrm{Cleaning} \equiv \mathrm{Closing\_Dishwasher\_Door} \sqcap \exists \mathrm{hasPredecessor.}\Big(\mathrm{Returning\_Cup} \sqcap \exists \mathrm{hasPredecessor.}\big(\mathrm{Returning\_Plate} \sqcap \exists \mathrm{hasPredecessor.}(\mathrm{Opening\_Dishwasher\_Door})\big)\Big) \qquad (6.17)$$

Table 6.3: Comparison of recognition accuracy for AR and ARW approaches.

Activity	AR, %	ARW, %	ARW-DS, %
Relaxing	N.A	90.0	90.0
Grooming	50.0	75.0	80.0
Making coffee	80.0	90.0	95.0
Preparing sandwich	85.0	100.0	100.0
Cleaning	95.0	95.0	95.0
Overall accuracy	77.5	90.0	92.0

Table 6.3 shows the comparison of recognition accuracy of AR and ARW approaches. AR performed generally well in recognizing the activities except “Grooming” whereby only 10 out of 20 activities are recognized. Tracing back to the dataset, failures are mainly due to missing sensor data, whereby 8 of the 10 data are missing the sensor data (3-drawer or fridge door) associated with the activity contexts. It is also that the sequence of actions performed by subjects when performing the activities do not perfectly match the sequence of actions modeled in the ontology. As shown in Table 6.3, ARW approach successfully recognizes “Relaxing” activity by fusing lying down context and the associated location context. Also, it is shown that ARW approach improves the number of correct classifications for all activities considered. The recognition accuracy is improved by 12.5%. “Grooming” activity has seen the most

improvement whereby the hand gesture contexts are fused to infer the associated actions and as a result the activity is recognized. There is an additional improvement to the overall recognition accuracy when ARW is integrated with DS theory. The recognition accuracies of "Grooming" and "Making coffee" are slightly improved by 5%. From the observation, the activities are recognized with a mass of 0.9468 and 0.8176. As shown in Table 6.3, the overall recognition accuracy of ARW-DS is 92% which is 2% higher than ARW. Comparison of precision and F-score between approaches are illustrated in Figure 6.6 and Figure 6.7.

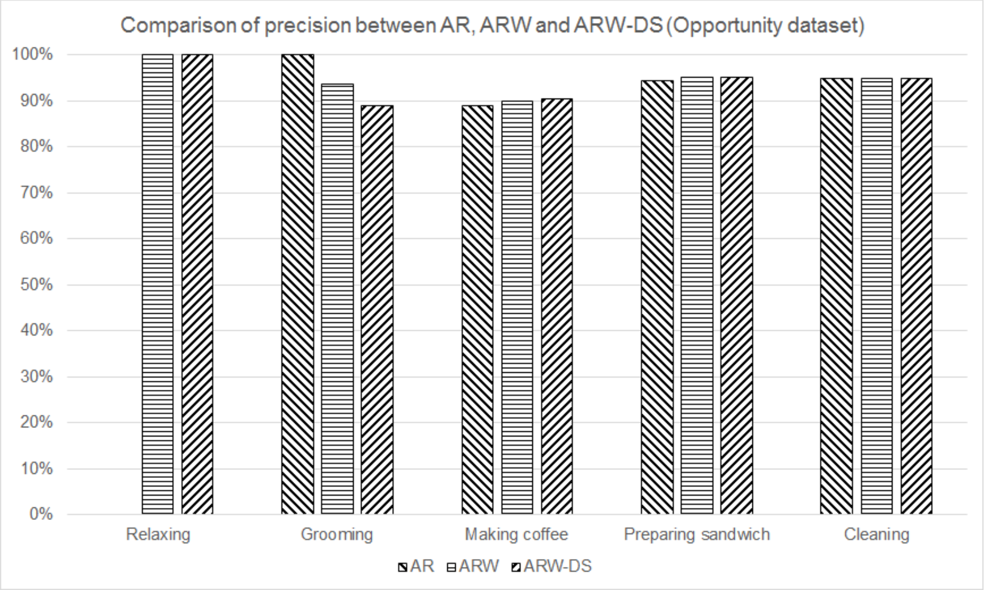


Figure 6.6: Comparison of precision between AR, ARW and ARW-DS for OPPORTUNITY dataset.

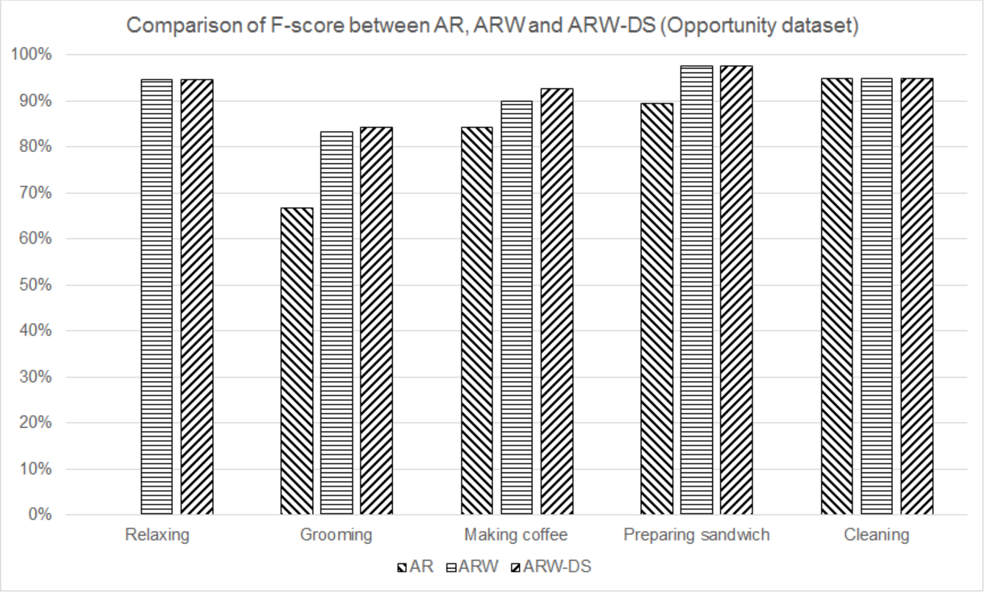
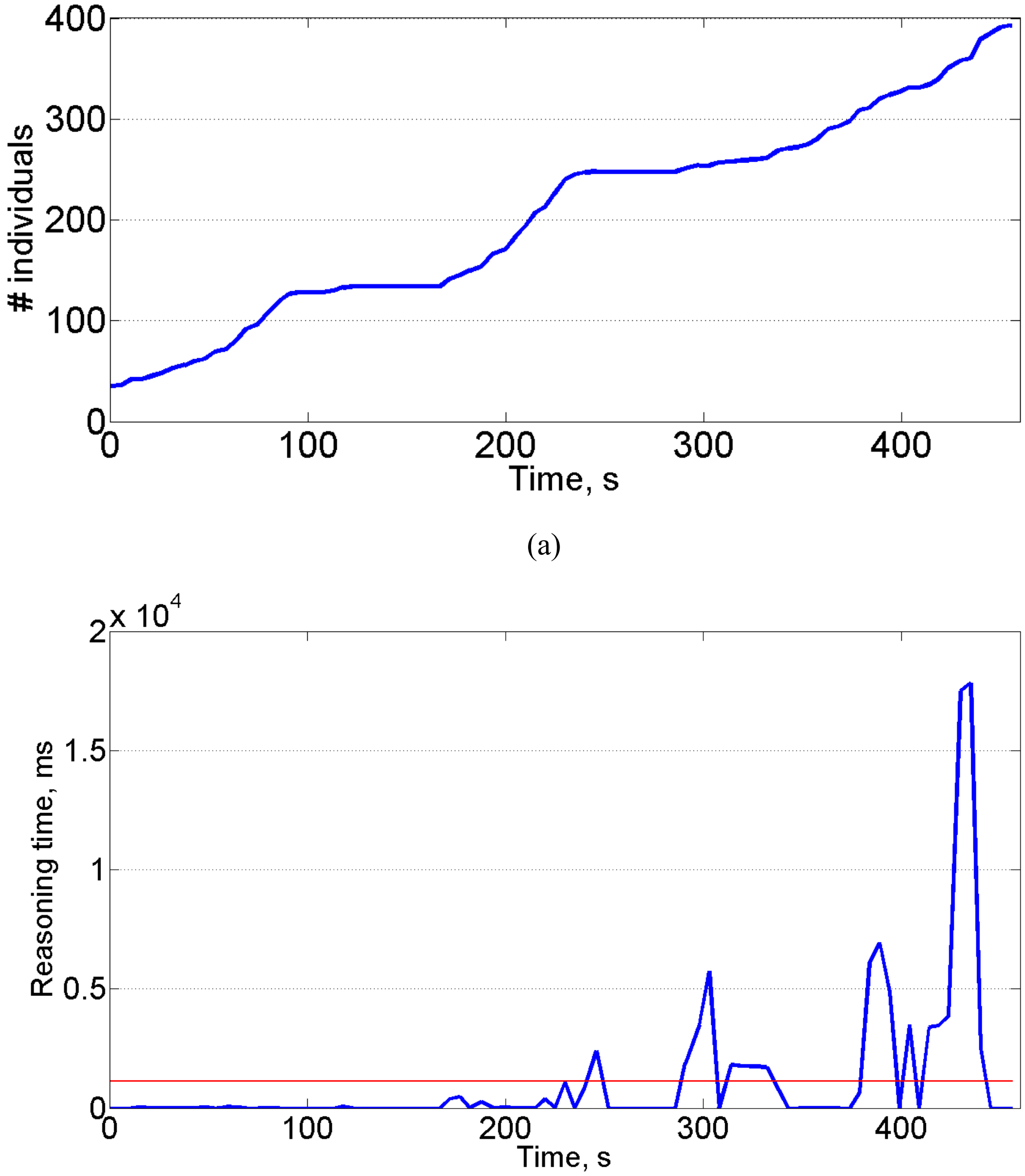


Figure 6.7: Comparison of F-score between AR, ARW and ARW-DS for OPPORTUNITY dataset.



(a)

(b)

Figure 6.8: IELAB dataset. (a) Number of individuals. (b) Reasoning time with Hermit.

The methodology described in [174] uses OWL 2 EL which is a subset of OWL 2. OWL 2 EL is an OWL 2 profile for which the reasoning tasks can be carried out in polynomial time. However, logical disjunction operator which is used in the proposed methodology is not supported by OWL 2 EL in order to guarantee the reasoning problems can be decided in polynomial time. Therefore, we investigate the proposed system performance to determine its feasibility. Figure 6.8 illustrates the behavior of the system during an experiment with the IELAB dataset. Figure 6.8(a) reports the number of individuals in the ontology (ABox) versus time. After 456s, the number of individuals (instances) is about 400. Figure 6.8(b) reports the corresponding reasoning time on a machine with Core i7 processor. The horizontal line indicates the average of reasoning time (1.1s). As shown in Figure 6.8(b), the reasoning time is significantly increased above 15s when the number of individuals reaches 350. Otherwise, the reasoning time always lower than 7s. It should also be noticed that similar behavior is observed in [174], in which the reasoning time ranges from 20 to 50s when the number of individuals is about 600. A solution is proposed by [174] to improve the reasoning time by removing the intervals which are not an instance of an activity concept after some time. This is assuming the intervals are not relevant anymore for future classifications. Reasoning performance can also be improved as described in [206], in which the approach specifically addresses the problem of disjunction operation. Parallel OWL reasoning approach is proposed in [207], enhancing the overall performance by a factor of 4.

6.5 Conclusions

In this chapter, we have proposed an ontology-based sensor fusion activity recognition system that fuses contexts from wearable and ambient sensors. It uses contextual information provided by both sensor approaches to infer activities of a subject (person) in the environment. In our study, the activities were performed by different subjects with no specific order. The subjects performed the activities with different styles and pace. In addition, a public dataset generated using different set of sensors containing different types of contexts is used. It was observed that the proposed approach can handle uncertainty due to missing object sensor data. The system can also infer activities more precisely and activities which do not involve object interaction. The overall recognition accuracy is 91.5% and 90% when tested on the internal and public datasets respectively. One limitation is that the proposed approach will not be able to deal with uncertainty due to missing location sensor data. This is because physical activity context provides some clues about the object interaction of the user, but not the user's whereabouts in the environment. One possible solution is indoor positioning/tracking system based on the use of wearable sensors such as accelerometers which could provide location of the user inside a building. In addition, the performance of ARW with uncertainty handling (ARW-DS) is evaluated. The results showed that ARW-DS further improves the recognition accuracy by 6% and 2% on the internal and public datasets respectively.

7

Conclusions

This thesis presents a number of robust and comprehensive context-aware activity recognition approaches and systems for elderly healthcare using wearable sensor and sensors embedded in environment. The systems are designed to harness the power of a wearable sensor (i.e. an accelerometer) that provides the physical activity context of the user, and then combines it with user-object interaction and location contexts provided by sensors in the environment. It achieves this through the development of a robust physical activity recognition methodology based on using a single wearable sensor (tri-axial accelerometer) which processes and classifies the body acceleration signals using machine learning techniques. A single wearable sensor is used to avoid attachment of multiple sensors to the body that can impede subject's daily activities. To achieve a robust physical activity recognition system, a novel adaptive sliding window approach is proposed to overcome the limitation of fixed-size sliding window approach. In addition, a physical activity transition model is proposed to model the temporal dependence of a sequence of physical activity to support the classification process. Given the advantages of ontology over other knowledge-based approaches, ontology-based modeling is used to encode domain knowledge by specifying and organizing the contextual information in a hierarchical structure (activity ontology). To overcome the limitation of ontology-based activity recognition in dealing with missing sensor data, an uncertainty handling technique using DS theory is integrated into the ontological reasoning process to enhance the performance of activity recognition. Furthermore, a sensor fusion methodology is proposed for activity recognition. The proposed methodology exploits contextual information from both user and environment to resolve uncertainty and achieve more precise inference of activities.

The proposed context-aware activity recognition system could be applied to various domains such as healthcare monitoring and home automation systems. In healthcare monitoring system, activity recognition can provide physicians with long-term information on human activities through continuous monitoring to assess and diagnose patients rather than data obtained from a single medical appointment. Chronic diseases which often the main reason of many deaths around the world could be prevented by changing certain behaviors related to physical activity and diet [208]. In this situation, activity recognition can be used to help physicians monitoring patients who are under exercise and diet programs. Home automation systems have to assist inhabitant especially elderly people in their daily living activities. An important aspect of home automation system is to monitor the behavior of the inhabitant in their ADL. This is to allow the home automation system to control the home appliances in order to facilitate the life of the inhabitants and also to optimize energy usage.

There are a number of challenges that must be considered when implementing the proposed system. In general, wearable devices are battery-powered and they are expected to last for a significant period of time. However, in wearable sensor-based activity recognition, the raw data is generated at a high rate and the processing unit is required to process and classify the data into different activities. Therefore, it is necessary to optimize the processing unit’s power consumption to prolong the life time of the system. Energy can be harvested from ambient sources such as vibration, radio frequency radiation and sound to prolong the battery life. Another challenge faced in implementing the proposed system is the signal propagation and attenuation between the sensors and base station. Besides walls and corridors, human body can be an obstacle itself for the signal propagation. In this case, approaches such as multiple input multiple output and novel transceiver architecture could be used to ensure data quality is not compromised. Form factor or the size of the wearable devices is also a critical factor in the feasibility of the proposed system. The wearable devices must be comfortable to wear by the user since it will be used for a long period of time. Lastly, the online execution of ontological reasoning is computationally expensive, especially when the reasoning tasks have high complexity. This is true in a real-life setup in which, a large number of activities and sensors are installed in the environment. In the following section, we summarize our achievements and results presented in the thesis, and discuss future directions for our research.

7.1 Achievements and Contributions

The following list highlights the major contributions of this thesis:

Chapter 3

- We developed a robust physical activity recognition system using a single tri-axial accelerometer. The approach includes a novel adaptive sliding window technique for segmentation of activity signal acquired from tri-axial accelerometer to overcome the limitations of fixed-size sliding window used in existing works. The technique adaptively adjusts the size of segmentation window according to the probability of the signal belongs to a particular activity. As a result, the window contains the right information when performing classification. In the experiments, we showed that the approach effectively segments activity signals resulting in better classification accuracy in a wide range of activities.

Chapter 4

- We proposed a transition model of physical activity to model the temporal dependence of the physical activities. The activity transition diagram is a part of the state validator of the activity recognition system. The state validator performs validation of activity transition for every window classification based on the proposed activity transition model and notifies the classification system to re-perform classification in the case an invalid transition is detected. In the experiments, we showed that the integration of state validator in the classification system improves the overall recognition accuracy.

Chapter 5

- We developed a robust ontology-based activity recognition capable of handling uncertainty. The approach includes a novel reasoning algorithm to support ontological reasoning with Dempster-Shafer theory of evidence. The algorithm utilizes ontological reasoning mechanisms of description logic and overcomes the limitation of ontology based models in terms of their inability to handle missing sensor data. The associated concepts are aggregated and the degree of belief is computed to make decision whether the corresponding activity has been performed or not. In addition, a four-layered activity ontology, which incorporates the representation of evidential parameters, is proposed. In the experiments, we showed that the reasoning algorithm is able to deal with missing sensor data and improves the recognition accuracy of traditional ontological reasoning. In addition, we showed that the performance is comparable to data-driven approach without requiring a large amount of data.

Chapter 6

- We proposed an ontology-based sensor fusion that fuses context information from wearable and ambient sensors for activity recognition. The approach comes with three advantages. First, it can resolve the uncertainty of imperfect observation due to sensor errors. For example, sitting on a chair can be recognized by considering the body posture of the user even if the chair sensor data is not available. Second, the approach allows additional and more precise inference of information about the activity being recognized because it fuses contextual information from both sensing approaches. Third, the approach can recognize activities which do not involve user-object interaction context. In the experiments, we showed the aforementioned advantages of the approach on internal and public datasets.

7.2 Future Work

There is still room for improvement in our work which can be addressed from two different perspectives, the physical activity recognition system and the ontology-based activity recognition system. First, there is the issue of fixed position of the accelerometer on the waist, the limitation of the adaptive sliding window and the implementation of the algorithm on a real-time platform. Second, the use of physiological sensors in activity recognition, the ideas on how to deal with concurrent and collective activity recognition and real-time performance issue. In this section, we focus on these aspects and propose them as future research directions:

- Although it has been shown that it is possible to achieve a robust physical activity recognition, it will be interesting to evaluate its performance if more physical activities such as walking up and downstairs and running are to be recognized. Also, the use of accelerometer only is limiting the information gathered from the user for activity classification especially when some activities would generate similar patterns of signals such as between walking and walking up or downstairs. The proposed physical activity recognition system is specific to be used with a waist-mounted accelerometer. Although this position allows some degree of variability it is important to explore the attachment of the accelerometer to different body parts. Wearable devices such as smartwatches with embedded inertial sensors are becoming increasingly popular, it is interesting to investigate the performance of the system if the accelerometer is attached to the wrist. Next, we plan to explore effects of additional mechanism in which the size of the window could be also reduced dynamically to capture short activity signals and further improve classification accuracy. We also plan to analyze the applicability of the adaptive sliding window for use in real-time scenarios. This will include the detailed analysis of computational complexity and their effect on real-time properties of the algorithms and the lifetime of the battery.
- Activity recognition in a smart environment often involves multiple users performing concurrent activities. We plan to investigate the segmentation mechanism to segment the sequence of sensor observations that generated by multiple users performing activities simultaneously. This also includes a single user performing interleaved

activities. In addition, multiple users may cooperate to perform activities. Therefore, it is important to model the user-user interaction context in order to recognize the collective activity. Next, we plan to fuse context information from physiological sensors to further improve the recognition accuracy. Although physiological sensors are mainly used to detect the health status of the users, they can also be used in the domain of activity recognition. For instance, heart rate measurement can be used to identify activities such as sleeping and exercising. We also plan to investigate the applicability of online execution of ontological reasoning in real-time scenarios. This includes distributed ontological reasoning and resource-aware scheduling algorithm.

References

[1] “WHO | Ageing and life-course,” *WHO*. [Online]. Available: http://www.who.int/ageing/en/. [Accessed: 20-Nov-2014].

[2] “Population ageing in New Zealand.” [Online]. Available: http://www.stats.govt.nz/browse_for_stats/people_and_communities/older_people/pop-ageing-in-nz.aspx. [Accessed: 28-May-2016].

[3] “Ageing characterises the demographic perspectives of the European societies - Issue number 72/2008 - Product - Eurostat.” [Online]. Available: http://ec.europa.eu/eurostat/en/web/products-statistics-in-focus/-/KS-SF-08-072. [Accessed: 10-Feb-2016].

[4] J. M. Ortman, V. A. Velkoff, H. Hogan, and others, “An aging nation: the older population in the United States,” *Wash. DC US Census Bur.*, pp. 25–1140, 2014.

[5] A. Marengoni, B. Winblad, A. Karp, and L. Fratiglioni, “Prevalence of Chronic Diseases and Multimorbidity Among the Elderly Population in Sweden,” *Am. J. Public Health*, vol. 98, no. 7, pp. 1198–1200, Jul. 2008.

[6] C. Naughton, K. Bennett, and J. Feely, “Prevalence of chronic disease in the elderly based on a national pharmacy claims database,” *Age Ageing*, vol. 35, no. 6, pp. 633–636, Nov. 2006.

[7] “Publicly funded hospital discharges - series,” *Ministry of Health NZ*. [Online]. Available: http://www.health.govt.nz/nz-health-statistics/health-statistics-and-data-sets/publicly-funded-hospital-discharges-series. [Accessed: 28-May-2016].

[8] “Public Health and Aging: Trends in Aging --- United States and Worldwide,” *Centers for Disease Control and Prevention*, 13-Feb-2003. [Online]. Available: http://www.cdc.gov/mmwr/preview/mmwrhtml/mm5206a2.htm. [Accessed: 28-May-2016].

[9] P. Laukkanen, E. Leskinen, M. Kauppinen, R. Sakari-Rantala, and E. Heikkinen, “Health and functional capacity as predictors of community dwelling among elderly people,” *J. Clin. Epidemiol.*, vol. 53, no. 3, pp. 257–265, Mar. 2000.

[10] Â. Costa, J. C. Castillo, P. Novais, A. Fernández-Caballero, and R. Simoes, “Sensor-driven agenda for intelligent home care of the elderly,” *Expert Syst. Appl.*, vol. 39, no. 15, pp. 12192–12204, Nov. 2012.

[11] A. Hough, “More people ‘fear losing independence in old age than death’, survey says,” 08-Feb-2010.

[12] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 1, pp. 414–454, First 2014.

[13] P. H. F. Peeters, “Design criteria for an automatic safety-alarm system for elderly,” *Technol. Health Care*, vol. 8, no. 2, pp. 81–91, Jan. 2000.

[14] S. Chaudhuri, H. Thompson, and G. Demiris, “Fall Detection Devices and their Use with Older Adults: A Systematic Review,” *J. Geriatr. Phys. Ther. 2001*, vol. 37, no. 4, pp. 178–196, 2014.

[15] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, “A Review of Smart Homes—Past, Present, and Future,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012.

[16] W. Allègre, T. Burger, J.-Y. Antoine, P. Berruet, and J.-P. Departe, “A non-intrusive context-aware system for ambient assisted living in smart home,” *Health Technol.*, vol. 3, no. 2, pp. 129–138, Jun. 2013.

[17] M. Chan, D. Estève, C. Escriba, and E. Campo, “A review of smart homes—Present state and future challenges,” *Comput. Methods Programs Biomed.*, vol. 91, no. 1, pp. 55–81, Jul. 2008.

[18] L. Liu, E. Stroulia, I. Nikolaidis, A. Miguel-Cruz, and A. Rios Rincon, “Smart homes and home health monitoring technologies for older adults: A systematic review,” *Int. J. Med. Inf.*, vol. 91, pp. 44–59, Jul. 2016.

[19] K. K. B. Peetoom, M. A. S. Lexis, M. Joore, C. D. Dirksen, and L. P. D. Witte, “Literature review on monitoring technologies and their outcomes in independently living elderly people,” *Disabil. Rehabil. Assist. Technol.*, vol. 10, no. 4, pp. 271–294, Jul. 2015.

[20] M. Amiribesheli, A. Benmansour, and A. Bouchachia, “A review of smart homes in healthcare,” *J. Ambient Intell. Humaniz. Comput.*, vol. 6, no. 4, pp. 495–517, Mar. 2015.

[21] W.-Y. Wang, C.-C. Chuang, Y.-S. Lai, and Y.-H. Wang, “A Context-Aware System for Smart Home Applications,” in *Embedded and Ubiquitous Computing – EUC 2005 Workshops*, T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, and L. T. Yang, Eds. Springer Berlin Heidelberg, 2005, pp. 298–305.

[22] S. Kurkovsky, “Pervasive computing: Past, present and future,” in *ITI 5th International Conference on Information and Communications Technology, 2007. ICICT 2007*, 2007, pp. 65–71.

[23] I. H. López-Nava and A. Muñoz-Meléndez, “Wearable Inertial Sensors for Human Motion Analysis: A Review,” *IEEE Sens. J.*, vol. 16, no. 22, pp. 7821–7834, Nov. 2016.

[24] O. D. Lara and M. A. Labrador, “A Survey on Human Activity Recognition using Wearable Sensors,” *IEEE Commun. Surv. Tutor.*, vol. 15, no. 3, pp. 1192–1209, Third 2013.

[25] M. F. Alhamid, M. Rawashdeh, H. Dong, M. A. Hossain, A. Alelaiwi, and A. E. Saddik, “RecAm: a collaborative context-aware framework for multimedia recommendations in an ambient intelligence environment,” *Multimed. Syst.*, vol. 22, no. 5, pp. 587–601, Oct. 2016.

[26] C. Bettini *et al.*, “A survey of context modelling and reasoning techniques,” *Pervasive Mob. Comput.*, vol. 6, no. 2, pp. 161–180, Apr. 2010.

[27] A. M. Khattak *et al.*, “Context Representation and Fusion: Advancements and Opportunities,” *Sensors*, vol. 14, no. 6, pp. 9628–9668, May 2014.

[28] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” in *Handheld and ubiquitous computing*, 1999, pp. 304–307.

[29] C. Anagnostopoulos and S. Hadjiefthymiades, “Advanced Inference in Situation-Aware Computing,” *IEEE Trans. Syst. Man Cybern. Part Syst. Hum.*, vol. 39, no. 5, pp. 1108–1115, Sep. 2009.

[30] J. Ye, L. Coyle, S. Dobson, and P. Nixon, “Using situation lattices to model and reason about context,” *Proc. MRC 2007 Coexist Context.*, pp. 1–12, 2007.

[31] O. Yurur, C. H. Liu, and W. Moreno, “A survey of context-aware middleware designs for human activity recognition,” *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 24–31, Jun. 2014.

[32] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, “Machine Recognition of Human Activities: A Survey,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[33] J. K. Aggarwal and M. S. Ryoo, “Human Activity Analysis: A Review,” *ACM Comput Surv*, vol. 43, no. 3, p. 16:1–16:43, Apr. 2011.

[34] N. C. Krishnan and D. J. Cook, “Activity recognition on streaming sensor data,” *Pervasive Mob. Comput.*, vol. 10, Part B, pp. 138–154, Feb. 2014.

[35] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-Based Activity Recognition,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[36] R. Jia and B. Liu, “Human daily activity recognition by fusing accelerometer and multi-lead ECG data,” in *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, 2013, pp. 1–4.

[37] K. Förster, A. Biasiucci, R. Chavarriaga, J. del R. Millán, D. Roggen, and G. Tröster, “On the Use of Brain Decoded Signals for Online User Adaptive Gesture Recognition Systems,” in *Pervasive Computing*, P. Floréen, A. Krüger, and M. Spasojevic, Eds. Springer Berlin Heidelberg, 2010, pp. 427–444.

[38] M. Sung, R. DeVaul, S. Jimenez, J. Gips, and A. Pentland, “Shiver motion and core body temperature classification for wearable soldier health monitoring systems,” in *Eighth International Symposium on Wearable Computers, 2004. ISWC 2004*, 2004, vol. 1, pp. 192–193.

[39] K. J. Heilman and S. W. Porges, “Accuracy of the LifeShirt® (Vivometrics) in the detection of cardiac rhythms,” *Biol. Psychol.*, vol. 75, no. 3, pp. 300–305, Jul. 2007.

[40] C. Metcalf *et al.*, “Fabric-based Strain Sensors for Measuring Movement in Wearable Telemonitoring Applications,” presented at the Assisted Living 2009, 2009, pp. 1–4.

[41] S. C. Mukhopadhyay, “Wearable Sensors for Human Activity Monitoring: A Review,” *IEEE Sens. J.*, vol. 15, no. 3, pp. 1321–1330, Mar. 2015.

[42] E. M. Tapia, S. S. Intille, and K. Larson, “Activity Recognition in the Home Using Simple and Ubiquitous Sensors,” in *Pervasive Computing*, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, pp. 158–175.

[43] D. H. Wilson and C. Atkeson, “Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors,” in *Pervasive Computing*, H.-W. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin Heidelberg, 2005, pp. 62–79.

[44] T. L. Kasteren, G. Englebienne, and B. J. Kröse, “An Activity Monitoring System for Elderly Care Using Generative and Discriminative Models,” *Pers. Ubiquitous Comput*, vol. 14, no. 6, pp. 489–498, Sep. 2010.

[45] D. Roggen *et al.*, “Collecting complex activity datasets in highly rich networked sensor environments,” in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, 2010, pp. 233–240.

[46] L. Bao and S. S. Intille, “Activity Recognition from User-Annotated Acceleration Data,” in *Pervasive Computing*, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, pp. 1–17.

[47] J.-H. Chiang, P.-C. Yang, and H. Tu, “Pattern analysis in daily physical activity data for personal health management,” *Pervasive Mob. Comput.*, vol. 13, pp. 13–25, Aug. 2014.

[48] P. Gupta and T. Dallas, “Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, Jun. 2014.

[49] L. Gao, A. K. Bourke, and J. Nelson, “Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems,” *Med. Eng. Phys.*, vol. 36, no. 6, pp. 779–785, Jun. 2014.

[50] J. Ye, S. Dobson, and S. McKeever, “Situation identification techniques in pervasive computing: A review,” *Pervasive Mob. Comput.*, vol. 8, no. 1, pp. 36–66, Feb. 2012.

[51] M. Philipose *et al.*, “Inferring activities from interactions with objects,” *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 50–57, Oct. 2004.

[52] T. van Kasteren and B. Krose, “Bayesian activity recognition in residence for elders,” in *3rd IET International Conference on Intelligent Environments, 2007. IE 07*, 2007, pp. 209–212.

[53] C. H. Lu and L. C. Fu, “Robust Location-Aware Activity Recognition Using Wireless Sensor Network in an Attentive Home,” *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 4, pp. 598–609, Oct. 2009.

[54] C. Zhu and W. Sheng, “Realtime Recognition of Complex Human Daily Activities Using Human Motion and Location Data,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 9, pp. 2422–2430, Sep. 2012.

[55] D. Sanchez, M. Tentori, and J. Favela, “Activity Recognition for the Smart Hospital,” *IEEE Intell. Syst.*, vol. 23, no. 2, pp. 50–57, Mar. 2008.

[56] D. J. Cook and M. Schmitter-Edgecombe, “Assessing the Quality of Activities in a Smart Environment,” *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, 2009.

[57] E. Kim, S. Helal, and D. Cook, “Human Activity Recognition and Pattern Discovery,” *IEEE Pervasive Comput.*, vol. 9, no. 1, pp. 48–53, Jan. 2010.

[58] D. J. Cook, N. C. Krishnan, and P. Rashidi, “Activity Discovery and Activity Recognition: A New Partnership,” *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 820–828, Jun. 2013.

[59] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, “Activity Recognition Using semi-Markov Models on Real World Smart Home Datasets,” *J Ambient Intell Smart Env.*, vol. 2, no. 3, pp. 311–325, Aug. 2010.

[60] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature Selection and Activity Recognition from Wearable Sensors," in *Ubiquitous Computing Systems*, H. Y. Youn, M. Kim, and H. Morikawa, Eds. Springer Berlin Heidelberg, 2006, pp. 516–527.

[61] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 119–128, Jan. 2006.

[62] Yi He and Ye Li, "Physical Activity Recognition Utilizing the Built-in Kinematic Sensors of a Smartphone," *Int. J. Distrib. Sens. Netw.*, pp. 1–10, Jan. 2013.

[63] R. Adaskevicius, "Method for Recognition of the Physical Activity of Human Being Using a Wearable Accelerometer," *Elektron. Ir Elektrotechnika*, vol. 20, no. 5, pp. 127–131, May 2014.

[64] A. T. Özdemir and B. Barshan, "Detecting Falls with Wearable Sensors Using Machine Learning Techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, Jun. 2014.

[65] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-Aware Human Activity Recognition Using Smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016.

[66] E. Garcia-Ceja, R. F. Brena, J. C. Carrasco-Jimenez, and L. Garrido, "Long-Term Activity Recognition from Wristwatch Accelerometer Data," *Sensors*, vol. 14, no. 12, pp. 22500–22524, Nov. 2014.

[67] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal Wearable Sensing for Fine-Grained Activity Recognition in Healthcare," *IEEE Internet Comput.*, vol. 19, no. 5, pp. 26–35, Sep. 2015.

[68] E. Nazerfard, B. Das, L. B. Holder, and D. J. Cook, "Conditional Random Fields for Activity Recognition in Smart Environments," in *Proceedings of the 1st ACM International Health Informatics Symposium*, New York, NY, USA, 2010, pp. 282–286.

[69] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate Activity Recognition in a Home Setting," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, New York, NY, USA, 2008, pp. 1–9.

[70] C. Bettini *et al.*, "A survey of context modelling and reasoning techniques," *Pervasive Mob. Comput.*, vol. 6, no. 2, pp. 161–180, Apr. 2010.

[71] G. KLYNE, "Composite Capability/Preference Profiles (CC/PP) : Structure and Vocabularies," *HttpwwwW3orgTR2001WD-CCPP-Struct-Vocab-20010315*, 2001.

[72] N. D. Rodríguez, M. P. Cuéllar, J. Lilius, and M. D. Calvo-Flores, “A Survey on Ontologies for Human Behavior Recognition,” *ACM Comput Surv*, vol. 46, no. 4, p. 43:1–43:33, Mar. 2014.

[73] C. Ramos, J. C. Augusto, and D. Shapiro, “Ambient Intelligence —the Next Step for Artificial Intelligence,” *IEEE Intell. Syst.*, vol. 23, no. 2, pp. 15–18, Mar. 2008.

[74] M. Endsley and D. Jones, “Confidence and Uncertainty in SA and Decision Making,” in *Designing for Situation Awareness*, 0 vols., CRC Press, 2011, pp. 113–130.

[75] L. Atallah, B. Lo, R. Ali, R. King, and G.-Z. Yang, “Real-Time Activity Classification Using Ambient and Wearable Sensors,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 1031–1039, Nov. 2009.

[76] K. Pelckmans, J. De Brabanter, J. A. K. Suykens, and B. De Moor, “Handling missing values in support vector machine classifiers,” *Neural Netw.*, vol. 18, no. 5–6, pp. 684–692, Jul. 2005.

[77] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[78] N. Díaz Rodríguez, M. P. Cuéllar, J. Lilius, and M. Delgado Calvo-Flores, “A fuzzy ontology for semantic modelling and recognition of human behaviour,” *Knowl.-Based Syst.*, vol. 66, pp. 46–60, Aug. 2014.

[79] W. Pedrycz and H. Camargo, “Fuzzy timed Petri nets,” *Fuzzy Sets Syst.*, vol. 140, no. 2, pp. 301–330, Dec. 2003.

[80] J. C. Hoffman and R. R. Murphy, “Comparison of Bayesian and Dempster-Shafer Theory for Sensing: A Practitioner’s Approach,” in *in SPIE Proc. on Neural and Stochastic Methods in Image and Signal Processing II*, 1993, pp. 266–279.

[81] M. A. Fiatarone *et al.*, “Exercise Training and Nutritional Supplementation for Physical Frailty in Very Elderly People,” *N. Engl. J. Med.*, vol. 330, no. 25, pp. 1769–1775, Jun. 1994.

[82] O. Eriksrud and R. W. Bohannon, “Relationship of Knee Extension Force to Independence in Sit-to-Stand Performance in Patients Receiving Acute Rehabilitation,” *Phys. Ther.*, vol. 83, no. 6, pp. 544–551, Jun. 2003.

[83] B. Unver, V. Karatosun, and S. Bakirhan, “Ability to rise independently from a chair during 6-month follow-up after unilateral and bilateral total knee replacement,” *J. Rehabil. Med.*, vol. 37, no. 6, pp. 385–387, Nov. 2005.

[84] M. M. Gross, P. J. Stevenson, S. L. Charette, G. Pyka, and R. Marcus, “Effect of muscle strength and movement speed on the biomechanics of rising from a chair in healthy elderly and young women,” *Gait Posture*, vol. 8, no. 3, pp. 175–185, 1998.

[85] S. Deandrea, F. Bravi, F. Turati, E. Lucenteforte, C. La Vecchia, and E. Negri, “Risk factors for falls in older people in nursing homes and hospitals. A systematic review and meta-analysis,” *Arch. Gerontol. Geriatr.*, vol. 56, no. 3, pp. 407–415, May 2013.

[86] B. Najafi, K. Aminian, F. Loew, Y. Blanc, and P. A. Robert, “Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly,” *IEEE Trans. Biomed. Eng.*, vol. 49, no. 8, pp. 843–851, Aug. 2002.

[87] L. Nyberg and Y. Gustafson, “Patient falls in stroke rehabilitation. A challenge to rehabilitation strategies,” *Stroke J. Cereb. Circ.*, vol. 26, no. 5, pp. 838–842, May 1995.

[88] K. Rapp, C. Becker, I. D. Cameron, H.-H. König, and G. Büchele, “Epidemiology of Falls in Residential Aged Care: Analysis of More Than 70,000 Falls From Residents of Bavarian Nursing Homes,” *J. Am. Med. Dir. Assoc.*, vol. 13, no. 2, p. 187.e1-187.e6, Feb. 2012.

[89] A. Zijlstra, M. Mancini, U. Lindemann, L. Chiari, and W. Zijlstra, “Sit-stand and stand-sit transitions in older adults and patients with Parkinson’s disease: event detection based on motion sensors versus force plates,” *J. Neuroengineering Rehabil.*, vol. 9, p. 75, 2012.

[90] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, “Activity identification using body-mounted sensors—a review of classification techniques,” *Physiol. Meas.*, vol. 30, no. 4, p. R1, Apr. 2009.

[91] S. D. Bersch, D. Azzi, R. Khusainov, I. E. Achumba, and J. Ries, “Sensor Data Acquisition and Processing Parameters for Human Activity Classification,” *Sensors*, vol. 14, no. 3, pp. 4239–4270, Mar. 2014.

[92] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, “Window Size Impact in Human Activity Recognition,” *Sensors*, vol. 14, no. 4, pp. 6474–6499, Apr. 2014.

[93] B. Fida, I. Bernabucci, D. Bibbo, S. Conforto, and M. Schmid, “Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer,” *Med. Eng. Phys.*, vol. 37, no. 7, pp. 705–711, Jul. 2015.

[94] O. Banos *et al.*, “Evaluating the effects of signal segmentation on activity recognition,” presented at the 2nd International Work-Conference on Bioinformatics and Biomedical Engineering, 2014.

[95] L. Santos, K. Khoshhal, and J. Dias, “Trajectory-based human action segmentation,” *Pattern Recognit.*, vol. 48, no. 2, pp. 568–579, Feb. 2015.

[96] S. Kozina, M. Lustrek, and M. Gams, “Dynamic signal segmentation for activity recognition,” in *Proceedings of International Joint Conference on Artificial Intelligence, Barcelona, Spain*, 2011, vol. 1622, p. 1522.

[97] A. Bifet and R. Gavalda, “Learning from Time-Changing Data with Adaptive Windowing.,” in *SDM*, 2007, vol. 7, p. 2007.

[98] M. Núñez, R. Fidalgo, and R. Morales, “Learning in environments with unknown dynamics: Towards more robust concept learners,” *J. Mach. Learn. Res.*, vol. 8, pp. 2595–2628, 2007.

[99] Z. Sheng, C. Hailong, J. Chuan, and Z. Shaojun, “An adaptive time window method for human activity recognition,” in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2015, pp. 1188–1192.

[100] M. Sekine, T. Tamura, T. Togawa, and Y. Fukui, “Classification of waist-acceleration signals in a continuous walking record,” *Med. Eng. Phys.*, vol. 22, no. 4, pp. 285–291, May 2000.

[101] M. N. Nyan, F. E. H. Tay, K. H. W. Seah, and Y. Y. Sitoh, “Classification of gait patterns in the time–frequency domain,” *J. Biomech.*, vol. 39, no. 14, pp. 2647–2656, 2006.

[102] M. Yoshizawa, W. Takasaki, and R. Ohmura, “Parameter Exploration for Response Time Reduction in Accelerometer-based Activity Recognition,” in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, New York, NY, USA, 2013, pp. 653–664.

[103] D. K. Aminian, K. Rezakhanlou, E. D. Andres, C. Fritsch, P.-F. Leyvraz, and P. Robert, “Temporal feature estimation during walking using miniature accelerometers: an analysis of gait improvement after hip arthroplasty,” *Med. Biol. Eng. Comput.*, vol. 37, no. 6, pp. 686–691, Nov. 1999.

[104] R. W. Selles, M. A. G. Formanoy, J. Bussmann, P. J. Janssens, and H. J. Stam, “Automated estimation of initial and terminal contact timing using accelerometers; development and validation in transtibial amputees and controls,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 1, pp. 81–88, Mar. 2005.

[105] K. Aminian, B. Najafi, C. Büla, P.-F. Leyvraz, and P. Robert, “Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes,” *J. Biomech.*, vol. 35, no. 5, pp. 689–699, May 2002.

[106] J. M. Jasiewicz *et al.*, “Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals,” *Gait Posture*, vol. 24, no. 4, pp. 502–509, Dec. 2006.

[107] M. Benocci, M. Bächlin, E. Farella, D. Roggen, L. Benini, and G. Troster, "Wearable assistant for load monitoring: recognition of on-body load placement from gait alterations," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*, 2010, pp. 1–8.

[108] A. Sant'Anna and N. Wickström, "A Symbol-Based Approach to Gait Analysis From Acceleration Signals: Identification and Detection of Gait Events and a New Measure of Gait Symmetry," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1180–1187, Sep. 2010.

[109] A. D. Ignatov and V. V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," *Multimed. Tools Appl.*, vol. 75, no. 12, pp. 7257–7270, May 2015.

[110] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 20–26, Jan. 2008.

[111] Z. Wang, D. Wu, J. Chen, A. Ghoneim, and M. A. Hossain, "A Triaxial Accelerometer-Based Human Activity Recognition via EEMD-Based Features and Game-Theory-Based Feature Selection," *IEEE Sens. J.*, vol. 16, no. 9, pp. 3198–3207, May 2016.

[112] L. Hu, Y. Chen, S. Wang, and Z. Chen, "b-COELM: A fast, lightweight and accurate activity recognition model for mini-wearable devices," *Pervasive Mob. Comput.*, vol. 15, pp. 200–214, Dec. 2014.

[113] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Appl. Soft Comput.*, vol. 37, pp. 1018–1022, Dec. 2015.

[114] W. Xiao, Y. Lu, W. Xiao, and Y. Lu, "Daily Human Physical Activity Recognition Based on Kernel Discriminant Analysis and Extreme Learning Machine, Daily Human Physical Activity Recognition Based on Kernel Discriminant Analysis and Extreme Learning Machine," *Math. Probl. Eng. Math. Probl. Eng.*, vol. 2015, 2015, p. e790412, May 2015.

[115] Z. He, "Activity recognition from accelerometer signals based on Wavelet-AR model," in *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*, 2010, vol. 1, pp. 499–502.

[116] A. Wang, G. Chen, J. Yang, S. Zhao, and C. Y. Chang, "A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone," *IEEE Sens. J.*, vol. 16, no. 11, pp. 4566–4578, Jun. 2016.

[117] Y. Zhang, S. Markovic, I. Sapir, R. C. Wagenaar, and T. D. C. Little, "Continuous functional activity monitoring based on wearable tri-axial accelerometer and gyroscope,"

in *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2011, pp. 370–373.

[118] V. Ahanathapillai, J. D. Amor, M. Tadeusiak, and C. J. James, “Wrist-Worn Accelerometer to Detect Postural Transitions and Walking Patterns,” in *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013*, L. M. R. Romero, Ed. Springer International Publishing, 2014, pp. 1515–1518.

[119] A. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, “A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1166–1172, Sep. 2010.

[120] A. C. Rencher and W. F. Christensen, *Methods of Multivariate Analysis*. John Wiley & Sons, 2012.

[121] S. G. Trost, Y. Zheng, and W.-K. Wong, “Machine learning for activity recognition: hip versus wrist data,” *Physiol. Meas.*, vol. 35, no. 11, p. 2183, Nov. 2014.

[122] A. Bayat, M. Pomplun, and D. A. Tran, “A Study on Human Activity Recognition Using Accelerometer Data from Smartphones,” *Procedia Comput. Sci.*, vol. 34, pp. 450–457, 2014.

[123] “Weka 3 - Data Mining with Open Source Machine Learning Software in Java.” [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 11-Aug-2015].

[124] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[125] L. Atallah, B. Lo, R. King, and G.-Z. Yang, “Sensor Positioning for Activity Recognition Using Wearable Accelerometers,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 4, pp. 320–329, Aug. 2011.

[126] W. Kerr, A. Tran, and P. Cohen, “Activity Recognition with Finite State Machines,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, Barcelona, Catalonia, Spain, 2011, pp. 1348–1353.

[127] H. Trinh, Q. Fan, P. Jiyan, P. Gabbur, S. Miyazawa, and S. Pankanti, “Detecting human activities in retail surveillance using hierarchical finite state machine,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1337–1340.

[128] N. Noorit and N. Suvonvorn, “Human Activity Recognition from Basic Actions Using Finite State Machine,” in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, T. Herawan, M. M. Deris, and J. Abawajy, Eds. Springer Singapore, 2014, pp. 379–386.

[129] C. Langensiepen, A. Lotfi, and S. Puteh, “Activities recognition and worker profiling in the intelligent office environment using a fuzzy finite state machine,” in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 873–880.

[130] J. R. Villar, S. González, J. Sedano, C. Chira, and J. M. Trejo-Gabriel-Galan, “Improving Human Activity Recognition and its Application in Early Stroke Diagnosis,” *Int. J. Neural Syst.*, vol. 25, no. 04, p. 1450036, Nov. 2014.

[131] W. Ding, K. Liu, X. Fu, and F. Cheng, “Profile HMMs for skeleton-based human action recognition,” *Signal Process. Image Commun.*, vol. 42, pp. 109–119, Mar. 2016.

[132] Y.-S. Lee and S.-B. Cho, “Layered hidden Markov models to recognize activity with built-in sensors on Android smartphone,” *Pattern Anal. Appl.*, pp. 1–13, Apr. 2016.

[133] M. H. Kabir *et al.*, “Two-Layer Hidden Markov Model for Human Activity Recognition in Home Environments, Two-Layer Hidden Markov Model for Human Activity Recognition in Home Environments,” *Int. J. Distrib. Sens. Netw. Int. J. Distrib. Sens. Netw.*, vol. 2016, 2016, p. e4560365, Jan. 2016.

[134] S. Kozina, H. Gjoreski, M. Gams, and M. Luštrek, “Three-layer activity recognition combining domain knowledge and meta-classification,” *J. Med. Biol. Eng.*, vol. 33, no. 4, pp. 406–414, 2013.

[135] K. I.-K. Wang, W. H. Abdulla, and Z. Salcic, “Multi-agent System with Hybrid Intelligence Using Neural Network and Fuzzy Inference Techniques,” in *New Trends in Applied Artificial Intelligence*, H. G. Okuno and M. Ali, Eds. Springer Berlin Heidelberg, 2007, pp. 473–482.

[136] O. Brdiczka, J. L. Crowley, and P. Reignier, “Learning Situation Models in a Smart Home,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 39, no. 1, pp. 56–63, Feb. 2009.

[137] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, “Discovering Activities to Recognize and Track in a Smart Environment,” *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 4, pp. 527–539, Apr. 2011.

[138] F. J. Ordóñez, J. A. Iglesias, P. de Toledo, A. Ledezma, and A. Sanchis, “Online activity recognition using evolving classifiers,” *Expert Syst. Appl.*, vol. 40, no. 4, pp. 1248–1255, Mar. 2013.

[139] L. Liu, Y. Peng, M. Liu, and Z. Huang, “Sensor-based human activity recognition system with a multilayered model using time series shapelets,” *Knowl.-Based Syst.*, vol. 90, pp. 138–152, Dec. 2015.

[140] L. Chen, C. Nugent, and G. Okeyo, “An Ontology-Based Hybrid Approach to Activity Modeling for Smart Homes,” *IEEE Trans. Hum.-Mach. Syst.*, vol. 44, no. 1, pp. 92–105, Feb. 2014.

[141] Liming Chen and Chris Nugent, “Ontology-based activity recognition in intelligent pervasive environments,” *Int. J. Web Inf. Syst.*, vol. 5, no. 4, pp. 410–430, Nov. 2009.

[142] K. Henricksen and J. Indulska, “Modelling and using imperfect context information,” in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004*, 2004, pp. 33–37.

[143] A. P. Dempster, “Upper and Lower Probabilities Induced by a Multivalued Mapping,” *Ann. Math. Stat.*, vol. 38, no. 2, pp. 325–339, Apr. 1967.

[144] G. Shafer and others, *A mathematical theory of evidence*, vol. 1. Princeton university press Princeton, 1976.

[145] R. R. Murphy, *Introduction to AI Robotics*, 1st ed. Cambridge, MA, USA: MIT Press, 2000.

[146] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*. New York, NY, USA: Cambridge University Press, 2003.

[147] L. Chen, C. D. Nugent, and H. Wang, “A Knowledge-Driven Approach to Activity Recognition in Smart Homes,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 961–974, Jun. 2012.

[148] I.-H. Bae, “An ontology-based approach to ADL recognition in smart homes,” *Future Gener. Comput. Syst.*, vol. 33, pp. 32–41, Apr. 2014.

[149] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, “Dynamic sensor data segmentation for real-time knowledge-driven activity recognition,” *Pervasive Mob. Comput.*, vol. 10, Part B, pp. 155–172, Feb. 2014.

[150] J. Ye, G. Stevenson, and S. Dobson, “KCAR: A knowledge-driven approach for concurrent activity recognition,” *Pervasive Mob. Comput.*, no. 0, p. , 2014.

[151] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, “MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns,” *Pervasive Mob. Comput.*, vol. 25, pp. 104–124, Jan. 2016.

[152] A. M. Khattak, W. A. Khan, Z. Pervez, F. Iqbal, and S. Lee, “Towards A Self Adaptive System for Social Wellness,” *Sensors*, vol. 16, no. 4, p. 531, Apr. 2016.

[153] S. Ahmadi-Karvigh, B. Becerik-Gerber, and L. Soibelman, “A framework for allocating personalized appliance-level disaggregated electricity consumption to daily activities,” *Energy Build.*, vol. 111, pp. 337–350, Jan. 2016.

[154] D. Riboni and C. Bettini, “COSAR: hybrid reasoning for context-aware activity recognition,” *Pers. Ubiquitous Comput.*, vol. 15, no. 3, pp. 271–289, Mar. 2011.

[155] J. Ye, G. Stevenson, and S. Dobson, “USMART: An Unsupervised Semantic Mining Activity Recognition Technique,” *ACM Trans Interact Intell Syst*, vol. 4, no. 4, p. 16:1–16:27, Nov. 2014.

[156] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, “Reasoning about uncertain contexts in pervasive computing environments,” *IEEE Pervasive Comput.*, vol. 3, no. 2, pp. 62–70, Apr. 2004.

[157] C. F. Crispim-Junior *et al.*, “Semantic Event Fusion of Different Visual Modality Concepts for Activity Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1598–1611, Aug. 2016.

[158] R. Helaoui, D. Riboni, and H. Stuckenschmidt, “A Probabilistic Ontological Framework for the Recognition of Multilevel Human Activities,” in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, New York, NY, USA, 2013, pp. 345–354.

[159] Z. Ding, Y. Peng, R. Pan, Z. Ding, Y. Peng, and R. Pan, “BayesOWL: Uncertainty modeling in Semantic Web ontologies,” in *Soft Computing in Ontologies and Semantic Web, volume 204 of Studies in Fuzziness and Soft Computing*, 2005, p. 27.

[160] Y. Yang and J. Calmet, “OntoBayes: An Ontology-Driven Uncertainty Model,” in *International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 2005, vol. 1, pp. 457–463.

[161] D. Ausín, D. López-de-Ipina, and F. Castanedo, “A Probabilistic OWL Reasoner for Intelligent Environments,” in *10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2014)*, 2014, p. 1.

[162] H. Wu, W. Grimm, R. B. Corporation, and C. H. Wu, “Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory,” 2003.

[163] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, and S. Devlin, “Evidential fusion of sensor data for activity recognition in smart homes,” *Pervasive Mob. Comput.*, vol. 5, no. 3, pp. 236–252, Jun. 2009.

[164] D. Zhang, M. Guo, J. Zhou, D. Kang, and J. Cao, “Context reasoning using extended evidence theory in pervasive computing environments,” *Future Gener. Comput. Syst.*, vol. 26, no. 2, pp. 207–216, Feb. 2010.

[165] S. McKeever, J. Ye, L. Coyle, S. Dobson, and C. Bleakley, “Activity Recognition using Temporal Evidence Theory,” *Articles*, Jul. 2010.

[166] A. Kushwah, S. Kumar, and R. M. Hegde, “Multi-sensor data fusion methods for indoor activity recognition using temporal evidence theory,” *Pervasive Mob. Comput.*, no. 0, p. , 2014.

[167] F. Sebbak, F. Benhammadi, A. Chibani, Y. Amirat, and A. Mokhtari, “Dempster–Shafer theory-based human activity recognition in smart home environments,” *Ann. Telecommun. - Ann. Télécommunications*, vol. 69, no. 3–4, pp. 171–184, Nov. 2013.

[168] J. Liao, Y. Bi, and C. Nugent, “Using the Dempster–Shafer Theory of Evidence With a Revised Lattice Structure for Activity Recognition,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 1, pp. 74–82, Jan. 2011.

[169] C. Chen, R. Jafari, and N. Kehtarnavaz, “Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors,” *IEEE Trans. Hum.-Mach. Syst.*, vol. 45, no. 1, pp. 51–61, Feb. 2015.

[170] H. Aloulou, M. Mokhtari, T. Tiberghien, R. Endelin, and J. Biswas, “Uncertainty handling in semantic reasoning for accurate context understanding,” *Knowl.-Based Syst.*, vol. 77, pp. 16–28, Mar. 2015.

[171] K. Yuan, F. Xiao, L. Fei, B. Kang, and Y. Deng, “Modeling Sensor Reliability in Fault Diagnosis Based on Evidence Theory,” *Sensors*, vol. 16, no. 1, p. 113, Jan. 2016.

[172] G. Azkune, A. Almeida, D. López-de-Ipiña, and L. Chen, “Extending knowledge-driven activity models through data-driven learning techniques,” *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3115–3128, Apr. 2015.

[173] P. Dai, S.-S. Ho, and F. Rudzicz, “Sequential behavior prediction based on hybrid similarity and cross-user activity transfer,” *Knowl.-Based Syst.*, vol. 77, pp. 29–39, Mar. 2015.

[174] A. Scalmato, A. Sgorbissa, and R. Zaccaria, “Describing and Recognizing Patterns of Events in Smart Environments With Description Logic,” *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1882–1897, Dec. 2013.

[175] J. D. Lowrance, T. D. Garvey, and T. M. Strat, “A Framework for Evidential-Reasoning Systems,” in *Classic Works of the Dempster-Shafer Theory of Belief Functions*, R. R. Yager and L. Liu, Eds. Springer Berlin Heidelberg, 2008, pp. 419–434.

[176] W. Liu, J. Hong, and M. F. McTear, “An extended framework for evidential reasoning systems,” in *Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, 1990*, 1990, pp. 731–737.

[177] L. A. Zadeh, “A Simple View of the Dempster-Shafer Theory of Evidence and Its Implication for the Rule of Combination,” *AI Mag*, vol. 7, no. 2, pp. 85–90, Jul. 1986.

[178] F. Smarandache and J. Dezert, “A simple proportional conflict redistribution rule,” *Multispace Multistructure Neutrosophic Transdiscipl. 100 Collect. Pap. Sci.*, vol. 4, p. 304, 2010.

[179] C. K. Murphy, “Combining belief functions when evidence conflicts,” *Decis. Support Syst.*, vol. 29, no. 1, pp. 1–9, Jul. 2000.

[180] P. Rashidi and D. J. Cook, “Keeping the Resident in the Loop: Adapting the Smart Home to the User,” *IEEE Trans. Syst. Man Cybern. Part Syst. Hum.*, vol. 39, no. 5, pp. 949–959, Sep. 2009.

[181] “HermiT Reasoner: Home.” [Online]. Available: http://www.hermit-reasoner.com/. [Accessed: 21-Mar-2016].

[182] N. Raman and S. J. Maybank, “Activity recognition using a supervised non-parametric hierarchical HMM,” *Neurocomputing*, vol. 199, pp. 163–177, Jul. 2016.

[183] Y. Zhang, Y. Zhang, E. Swears, N. Larios, Z. Wang, and Q. Ji, “Modeling Temporal Interactions with Interval Temporal Bayesian Networks for Complex Activity Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2468–2483, Oct. 2013.

[184] L. Liu, L. Cheng, Y. Liu, Y. Jia, and D. S. Rosenblum, “Recognizing Complex Activities by a Probabilistic Interval-Based Model,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[185] E. Nazerfard and D. J. Cook, “CRAFFT: an activity prediction model based on Bayesian networks,” *J. Ambient Intell. Humaniz. Comput.*, vol. 6, no. 2, pp. 193–205, 2015.

[186] K. P. Fishkin, M. Philipose, and A. D. Rea, “Hands-On RFID: Wireless Wearables for Detecting Use of Objects.,” in *ISWC*, 2005, vol. 5, pp. 38–43.

[187] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, “Fine-grained activity recognition by aggregating abstract object usage,” in *Ninth IEEE International Symposium on Wearable Computers, 2005. Proceedings*, 2005, pp. 44–51.

[188] M. R. Hodges and M. E. Pollack, “An ‘Object-Use Fingerprint’: The Use of Electronic Sensors for Human Identification,” in *UbiComp 2007: Ubiquitous Computing*, J. Krumm,

G. D. Abowd, A. Seneviratne, and T. Strang, Eds. Springer Berlin Heidelberg, 2007, pp. 289–303.

[189] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing Daily Activities with RFID-based Sensors," in *Proceedings of the 11th International Conference on Ubiquitous Computing*, New York, NY, USA, 2009, pp. 51–60.

[190] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A Scalable Approach to Activity Recognition based on Object Use," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[191] R. Krigslund, P. Popovski, G. F. Pedersen, and K. Olesen, "Interference Helps to Equalize the Read Range and Reduce False Positives of Passive RFID Tags," *IEEE Trans. Ind. Electron.*, vol. 59, no. 12, pp. 4821–4830, Dec. 2012.

[192] Y. J. Tu and S. Piramuthu, "A Decision-Support Model for Filtering RFID Read Data in Supply Chains," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 2, pp. 268–273, Mar. 2011.

[193] T. Keller, F. Thiesse, J. Kungl, and E. Fleisch, "Using low-level reader data to detect false-positive RFID tag reads," in *Internet of Things (IOT), 2010*, 2010, pp. 1–8.

[194] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A Pattern Mining Approach to Sensor-Based Human Activity Recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1359–1372, Sep. 2011.

[195] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simul. Model. Pract. Theory*, vol. 18, no. 4, pp. 446–455, Apr. 2010.

[196] Takebe Yoshihiro *et al.*, "Recognition of nursing activity with accelerometers and RFID," *Kybernetes*, vol. 42, no. 7, pp. 1059–1071, Sep. 2013.

[197] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. L. Lo, and G. Z. Yang, "Ambient and Wearable Sensor Fusion for Activity Recognition in Healthcare Monitoring Systems," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, P. D.-I. D. med S. Leonhardt, D.-I. T. Falck, and P. D. P. Mähönen, Eds. Springer Berlin Heidelberg, 2007, pp. 208–212.

[198] D. McIlwraith, J. Pansiot, and G.-Z. Yang, "Wearable and ambient sensor fusion for the characterisation of human motion," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5505–5510.

[199] A. M. Khattak *et al.*, "Towards Smart Homes Using Low Level Sensory Data," *Sensors*, vol. 11, no. 12, pp. 11581–11604, Dec. 2011.

[200] D. M. Gavrila, “The Visual Analysis of Human Movement: A Survey,” *Comput. Vis. Image Underst.*, vol. 73, no. 1, pp. 82–98, Jan. 1999.

[201] A. Yilmaz, O. Javed, and M. Shah, “Object Tracking: A Survey,” *ACM Comput Surv*, vol. 38, no. 4, Dec. 2006.

[202] Y. Ge and B. Xu, “Elderly personal intention recognition by activity and context recognition in smart home,” in *2014 9th International Conference on Computer Science Education (ICCSE)*, 2014, pp. 347–350.

[203] N. Roy, A. Misra, and D. Cook, “Ambient and smartphone sensor assisted ADL recognition in multi-inhabitant smart environments,” *J. Ambient Intell. Humaniz. Comput.*, vol. 7, no. 1, pp. 1–19, Jun. 2015.

[204] K. Wongpatikaseree, M. Ikeda, M. Buranarach, T. Supnithi, A. O. Lim, and Y. Tan, “Activity Recognition Using Context-Aware Infrastructure Ontology in Smart Home Domain,” in *2012 Seventh International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, 2012, pp. 50–57.

[205] “UCI Machine Learning Repository: OPPORTUNITY Activity Recognition Data Set.” [Online]. Available: http://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition. [Accessed: 21-Apr-2016].

[206] A. Steigmiller, B. Glimm, and T. Liebig, “Coupling Tableau Algorithms for Expressive Description Logics with Completion-Based Saturation Procedures,” in *Automated Reasoning*, S. Demri, D. Kapur, and C. Weidenbach, Eds. Springer International Publishing, 2014, pp. 449–463.

[207] K. Wu and V. Haarslev, “Parallel OWL Reasoning: Merge Classification,” in *Semantic Technology*, W. Kim, Y. Ding, and H.-G. Kim, Eds. Springer International Publishing, 2013, pp. 211–227.

[208] W. C. Willett, J. P. Koplan, R. Nugent, C. Dusenbury, P. Puska, and T. A. Gaziano, “Prevention of Chronic Disease by Means of Diet and Lifestyle Changes,” in *Disease Control Priorities in Developing Countries*, 2nd ed., D. T. Jamison, J. G. Breman, A. R. Measham, G. Alleyne, M. Claeson, D. B. Evans, P. Jha, A. Mills, and P. Musgrove, Eds. Washington (DC): World Bank, 2006.

A

Adaptive Sliding Window

```
classdef actRecProcCls
  properties (SetAccess = protected)
    w = 1   % integer for recognition process counter
    s = 1   % integer for start window
    e = 1   % integer for end window
    winCnt = 1  % integer for window being evaluated counter
    initState
    incorrectTransition = 0;
    gausClassifierFeat = 1;
    mvgParamNumber
    debug
  end
  methods

  function OBJ = actRecProcCls(debug)
    OBJ.debug = debug;
  end

  function run(OBJ, DAT, WIN, DTR, MVN, STD, RES, enable, publicDataset,
  mvgParamNumber)
    tstart = datestr(now,'HH:MM:SS:FFF');
    debug = OBJ.debug;
    OBJ.mvgParamNumber = mvgParamNumber;

    while OBJ.w ~= WIN.getTotalWin()
```

```
OBJ.e = OBJ.s + WIN.getWinSize() - 1;
if OBJ.e > DAT.getDataLength()
     break;
end
DAT.calcFeatures(OBJ.s, OBJ.e);
% evaluate window if it is transitional window
DTR.evalTransitionalWindow(DAT);
if DTR.getTrFlag == false
     % signals are moving, idling or small moving activities
     fprintf('Eval (Fix)\n');
     WIN.incrDynaWinCnt();
     DTR.evalDynamicStatic(DAT, STD);
     STD.setWinSzSeq(WIN.getDynaWinCnt());

     DTR.resetEvalClassifier();
elseif DTR.getTrFlag == true    % 1,0
     DTR.resetTrFlag();
     % signals are transitional activities
     fprintf('Eval (Dynamic)\n');
     WIN.incrDynaWinCnt();
     % local variable to count the loop until end window is detected
     tmpW = 0;
     isClassified = false;
     isMaxWin = false;
     st = OBJ.s;
     ed = OBJ.e;

     maxProb = log(0);
     bestState = 0;
     bestWinSize = 1;
     % expanding windows
     preEvalState = 0;
     while isMaxWin == false
        tmpS = st - (WIN.getMagOverlap * tmpW);
        tmpE = ed;

        DAT.calcFeatures(tmpS, tmpE);% evaluate (new) dynamic window
        DTR.resetEvalState();
        DTR.evalMobilityV3(DAT, STD);
        evalState = DTR.getEvalState();
        if preEvalState == 0 && evalState > 0
          preEvalState = evalState;
          maxWin = STD.getMaxWindow(preEvalState);
          fprintf('Assign win classification: %u\n', preEvalState);
        elseif preEvalState == 0 && evalState == 0
          isMaxWin = true;
        end
        fprintf('%u) DTR: %u (PE: %u) | ', WIN.getDynaWinCnt(), evalState,
        preEvalState);
        if evalState > 0 && evalState == preEvalState
          [probN,~]         =         MVN.computePdf(DAT,        evalState,
          OBJ.gausClassifierFeat, OBJ.mvgParamNumber);
          fprintf('Prob: %e\n', probN);
          if probN > maxProb
          maxProb = probN;
          bestState = evalState;
          bestWinSize = WIN.getDynaWinCnt();
          isClassified = true;
          else
          fprintf('End Window (Prob)\n');
          STD.setWinSzSeq(WIN.getDynaWinCnt()-1);
          isMaxWin = true;
          end
        else
          fprintf('Classification is changed!\n');
           [probN,~]        =        MVN.computePdf(DAT,        evalState,
          OBJ.gausClassifierFeat, OBJ.mvgParamNumber);

          fprintf('Prob: %e\n', probN);
          STD.setWinSzSeq(WIN.getDynaWinCnt()-1);
          isMaxWin = true;
        end
        fprintf('\n');
        st = st + WIN.getMagExpansion();
        ed = st + WIN.getWinSize() - 1;
        if isMaxWin == false
          WIN.incrDynaWinCnt();
        end
        tmpW = tmpW + 1;
        if WIN.getDynaWinCnt() > maxWin
          fprintf('End Window (Limit)\n');
          STD.setWinSzSeq(WIN.getDynaWinCnt()-1);
          isMaxWin = true;
        end
     end     % end while loop

     if isClassified == true
        fprintf('State: %u, WinSize: %u (%u-%u)\n',...
        bestState, bestWinSize, OBJ.s, (OBJ.e * WIN.getDynaWinCnt()));
        DTR.setEvalState(bestState);
        WIN.setDynaWinCnt(bestWinSize);
     end
end
evalState = DTR.getEvalState();
if evalState > 0
     fprintf('Classify!\n');
     if WIN.getDynaWinCnt() == 1
        RES.setResult(evalState, OBJ.w);
        STD.setStateSeq(evalState);
        STD.setWinSeq(OBJ.w);
        % save the features for later use by state diagram (if error
        classification)
        DAT.calcFeatures(OBJ.s, OBJ.e);
```

```
                  STD.setFeatSeq(DAT);
               else
                  for i = 1:WIN.getDynaWinCnt()
                     RES.setResult(evalState, OBJ.w+(i-1));
                  end
                  STD.setStateSeq(evalState);
                  STD.setWinSeq(OBJ.w+(WIN.getDynaWinCnt()-1));
                  % save the features for later use by state diagram (if error
                  classification)
                  DAT.calcFeatures(OBJ.s, (OBJ.s + (WIN.getMagOverlap() *
                  WIN.getDynaWinCnt())));
                  STD.setFeatSeq(DAT);
               end
               DTR.resetEvalState();
         end % end if evalState >0

         stSeq = STD.getStateSeq();
         fprintf('%u %u %u\n', stSeq(1), stSeq(2), stSeq(3));
         error = STD.checkStateSeq();
         if error == true && enable == true
               % Correction of classification %
               evalState = STD.classifyV1(OBJ.s, DAT, WIN,
         end
         if WIN.getDynaWinCnt() == 0
               OBJ.s = OBJ.s + WIN.getMagOverlap();
               OBJ.w = OBJ.w + 1;
         else
               OBJ.s = OBJ.s + (WIN.getMagOverlap() * WIN.getDynaWinCnt());
               OBJ.w = OBJ.w + WIN.getDynaWinCnt();
               WIN.resetDynaWinCnt();
         end
      end % end while OBJ.w ~= WIN.getTotalWin()
      tend = datestr(now,'HH:MM:SS:FFF');
   end % function run
   end % method
end % class
```

B

Ontological Reasoning with Uncertainty Handling

```
contextReasoner.createReasoner(ontologyHelper.getOntology());
currWindowIntervals.addWindowList(ontologyHelper.getCurrentWindowIntervalsList());

List<Interval> windowList = currWindowIntervals.getWindowList();
Collections.reverse(windowList);
listOfAllIntervals.addAll(0, windowList);

currWindowIntervals.clearWindowList();
ontologyHelper.clearCurrentWindowIntervalsList();

hasLocation = false;
isAssertPossibleADL = false;
String inferredLocation = null;

currentInferredADLs = new ArrayList<OWLClass>();
List<OWLNamedIndividual> intervalList = ontologyHelper.getIntervalList();

for(OWLNamedIndividual e : intervalList) {
   if(!hasLocation) {
      hasLocation = contextReasoner.locationClassifier(e);
      Set<OWLClass> locationClass = contextReasoner.getLocLeafClass();
```

```
     if(!locationClass.isEmpty()) {
        for(OWLClass loc : locationClass) {
             inferredLocation = loc.getIRI().getFragment();
        }
     }
   }


   Set<OWLClassExpression> eTypes = e.getTypes(ontologyHelper.getOntology());
   eTypes.remove(ontologyHelper.getClass("Interval"));
   if(!eTypes.isEmpty()) {
     if(contextReasoner.adlClassifier(e)) {
        for(OWLClass inferredADL : contextReasoner.getADLLeafClass()) {
             currentInferredADLs.add(inferredADL);
             assertADL(e, inferredADL, 1.0, timestamp);
        }
     }
     findPossibleADLs(e, beginAtValue, timestamp);
   }
}      // end of for loop intervalList


if(currentLocation.equals("")) {
   if(inferredLocation != null) {
     currentLocation = inferredLocation;
   }
} else {
   if(inferredLocation != null) {
     if(!currentLocation.equals(inferredLocation)) {
        // keep previous location
        previousLocation = currentLocation;
        // update current location
        currentLocation = inferredLocation;
        isAssertPossibleADL = true;
     }
   }
}


System.out.println("isAssertPossibleADL: " + isAssertPossibleADL);


if(isAssertPossibleADL) {
   List<PossibleADLModel> toBeInferredADLs = getToBeInferredPossibleADLs();
   if(!toBeInferredADLs.isEmpty()) {
       assertPossibleADL(toBeInferredADLs, timestamp);
   }
}


if(!evidenceReasoner.getPossibleADLs().isEmpty()) {
   System.out.println(evidenceReasoner.getPossibleADLs().size() + " possible
   activities of daily living detected.");
   for(PossibleADLModel possibleADLModel : evidenceReasoner.getPossibleADLs()) {
      System.out.print(possibleADLModel.toString()        +        "("        +
      possibleADLModel.getBeliefValue() + "), ");
   }
}




// function to find possible ADLs
private  void  findPossibleADLs(OWLNamedIndividual  e,  int  beginAtValue,  String
timestamp) {

   …
   for(OWLClassExpression equivalent : possibleADLEquivalentClasses) {
      if(!equivalent.getClassesInSignature().contains(actionClass)) {
         // Definition does not contain the action. Skip...
         continue;
      }
      List<OWLClass> inferredActionsOfPossibleADL = new ArrayList<OWLClass>();
      Map<Interval,OWLClass>     inferredIntervalActionsOfPossibleADL     =     new
      LinkedHashMap<Interval, OWLClass>();
      List<ContextModel>                       ADLContexts                        =
      ontologyHelper.getPossibleADLContexts(equivalent);
      // if it is not last index, action class is not the last context, get the
      following contexts
      int actionClsIndex = 0;
      for( ; actionClsIndex < ADLContexts.size(); actionClsIndex++) {
         if(ADLContexts.get(actionClsIndex).getAction().equals(actionClass)) {
               break;
         }
      }
      System.out.print(actionClsIndex);
      System.out.println();

      List<ContextModel> negPropContexts = new ArrayList<ContextModel>();
      for( int i = 0; i < actionClsIndex; i++) {
         negPropContexts.add(ADLContexts.get(i));
      }


      for(Interval itv : listOfAllIntervals) {
         Set<OWLClassExpression>                    itvTypes                      =
         itv.getInterval().getTypes(ontologyHelper.getOntology());
         itvTypes.remove(ontologyHelper.getClass("Interval"));
         if(itvTypes.contains(ADLContexts.get(actionClsIndex).getAction())) {
         // check if current interval's type is the same as possible ADL contexts
               boolean add = true;
               ContextModel ctxMod = ADLContexts.get(actionClsIndex);
               Set<OWLLiteral> itvBeginAtLits =
               itv.getInterval().getDataPropertyValues(ontologyHelper.getDataPropert
               y("hasBeginAt"), ontologyHelper.getOntology());
         OWLLiteral itvBeginAtLit = itvBeginAtLits.iterator().next();
         int itvBeginAt = Integer.parseInt(itvBeginAtLit.getLiteral());
         if(ctxMod.getCloserThan() != null) {
            int ctxModCloserThanVal = ctxMod.getCloserThanVal();
            int closerThan = beginAtValue - ctxModCloserThanVal;
            // out of temporal context
            if(itvBeginAt <= closerThan) add = false;
         }
         if(ctxMod.getFartherThan() != null) {
            int ctxModFartherThanVal = ctxMod.getFartherThanVal();
            int fartherThan = beginAtValue - ctxModFartherThanVal;
            // out of temporal context
            if(itvBeginAt >= fartherThan) add = false;
         }
         // check if current interval is predecessor of the successor interval

         if(successorItv != null) {
            Set<OWLIndividual> successorItvPred =
            successorItv.getInterval().getObjectPropertyValues(ontologyHelper.g
            etObjProperty("hasPredecessor"), ontologyHelper.getOntology());
            if(!successorItvPred.contains(itv.getInterval())) {
               add = false;
            }
         }
         if(add) {
            inferredActionsOfPossibleADL.add(ADLContexts.get(actionClsIndex).ge
            tAction());
            inferredIntervalActionsOfPossibleADL.put(itv,
            ADLContexts.get(actionClsIndex).getAction());
            // all contexts left in ADLContexts are uninferred actions, which
            will be used for adding uninferred masses
            contextsToBeRemoved.add(ADLContexts.get(actionClsIndex));
            if(actionClsIndex != (sizeOfADLContexts-1)) {
               actionClsIndex++;
            } else {
               break;
            }
         }
   }
   successorItv = itv;
}

…
for(Interval itv : inferredIntervalActionsOfPossibleADL.keySet()) {
   // assign mass to action with active state
   …
}

for(ContextModel ctx : contextsToBeRemoved) {
   ADLContexts.remove(ctx);       // remove active context
}
```

```
    for(ContextModel ctx : negPropContexts) {
      ADLContexts.remove(ctx);        // remove inactive contexts
    }

    // set masses to calculate possible ADL mass - add masses for inactive actions
    for(ContextModel ctxModel : negPropContexts) {
      evidenceReasoner.addMassOfInactiveContext(ctxModel.toString());
    }

    // set masses to calculate possible ADL mass - add masses from uncertain actions
    for(ContextModel ctxModel : ADLContexts) {
      evidenceReasoner.addMassOfUncertainContext(ctxModel.toString());
    }

    evidenceReasoner.combineMass(possibleADL);

  }//end for(OWLClassExpression equivalent : possibleADLEquivalentClasses)
}
```

Office of the Vice-Chancellor
Finance, Ehtics and Compliance

The University of Auckland
Private Bag 92019
Auckland, New Zealand

Level 10, 49 Symonds Street
Telephone: 64 9 373 7599
Extension: 87830 / 83761
Facsimile: 64 9 373 7432

UNIVERSITY OF AUCKLAND HUMAN PARTICIPANTS ETHICS COMMITTEE (UAHPEC)

24-Nov-2016

MEMORANDUM TO:

Prof Zoran Salcic
Electrical & Computer Engineer

Re: Application for Ethics Approval (Our Ref. 018288): Approved with comment

The Committee considered your application for ethics approval for your project entitled Activity recognition using wearable and sensors embedded in environment .

Ethics approval was given for a period of three years with the following comment(s):

Please remove the following:
Your Head of Department has given their assurance that your participation, or lack thereof, ...
And replace with:
Your Head of Department has given assurance that your participation or non-participation...

2. Email invitation
Please remove the following 'Thank you for your participation' as the participant has not agreed to participate at this stage.

The expiry date for this approval is 24-Nov-2019.

If the project changes significantly you are required to resubmit a new application to UAHPEC for further consideration.

In order that an up-to-date record can be maintained, you are requested to notify UAHPEC once your project is completed.

The Chair and the members of UAHPEC would be happy to discuss general matters relating to ethics approvals if you wish to do so. Contact should be made through the UAHPEC Ethics Administrators at ro-ethics@auckland.ac.nz in the first instance.

All communication with the UAHPEC regarding this application should include this reference number: 018288.

(This is a computer generated letter. No signature required.)

Secretary
University of Auckland Human Participants Ethics Committee

c.c. Head of Department / School, Electrical & Computer Engineer
Dr Kevin I-Kai Wang
Mohd Halim Mohd Noor

Additional information:

1. Should you need to make any changes to the project, write to the Committee giving full details including revised documentation.

2. Should you require an extension, write to the Committee before the expiry date giving full details along with revised documentation. An extension can be granted for up to three years, after which time you must make a new application.

3. At the end of three years, or if the project is completed before the expiry, you are requested to advise the Committee of its completion.

4. Do not forget to fill in the 'approval wording' on the Participant Information Sheets and Consent Forms, giving the dates of approval and the reference number, before you send them out to your participants.

5. Send a copy of this approval letter to the Awards Team at the, Research Office if you have obtained funding other than from UniServices. For UniServices contract, send a copy of the approval letter to: Contract Manager, UniServices.

6. Please note that the Committee may from time to time conduct audits of approved projects to ensure that the research has been carried out according to the approval that was given.