

CONTEXT-BASED REASONING IN AMBIENT INTELLIGENCE

Hristijan Gjoreski

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia, October, 2014

Evaluation Board:

Prof. Dr. Marko Bohanec, Chairman, Jožef Stefan Institute, Ljubljana, Slovenia

Dr. Juan Carlos Augusto, Member, Middlesex University, London, United Kingdom

Prof. Dr. Roman Trobec, Member, Jožef Stefan Institute, Ljubljana, Slovenia

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL



Hristijan Gjoreski

CONTEXT-BASED REASONING IN AMBIENT INTELLIGENCE

Doctoral Dissertation

KONTEKSTNO SKLEPANJE V AMBIENTALNI INTELIGENCI

Doktorska disertacija

Supervisor: Prof. Dr. Matjaž Gams

Co-Supervisor: Dr. Mitja Luštrek

Ljubljana, Slovenia, October 2014

To my family, past, present, and future

На мојата фамилија, мината, сегашна и идна

Abstract

The availability of small, wearable, low-cost, power-efficient sensors, combined with advanced signal processing and information extraction, is driving the revolution in the ambient intelligence (AmI) domain. This revolution has enabled novel approaches and technologies for accurate measurements in the area of healthcare, enhanced sports and fitness training, and life-style monitoring.

Early AmI systems included a single type of sensors that has made it possible to develop the first proof-of-concept applications. As the field has matured, these systems have gained additional sensors, resulting in the development of advanced and more accurate multi-sensor techniques and applications. However, combining multiple sources of information from multiple sensors is a challenging task. The first issue is that each sensor has its own technical configuration (for example, the data sampling rate) and requires different data-processing techniques in order to first align the different sensor data, and later to extract useful information. The second issue is that even if the multi-source data is aligned, it can be challenging to find an intelligent way to combine this multi-source information in order to reason about the user or the environment. While several approaches for combining multiple sources of information and knowledge have been developed (such as Kalman filters, ensemble learning, and co-training), these approaches have not been specialized for AmI tasks.

This thesis addresses the problem of combining multiple sources of information extracted from sensor data by proposing a novel context-based approach called CoReAmI (Context-based Reasoning in Ambient Intelligence). The CoReAmI approach consists of three phases: context extraction, context modeling, and context aggregation. In the first phase, multiple contexts are extracted from the sensor data. In the second phase, the problem is modeled using the already extracted contexts. In the third phase, when evaluating a data sample, the models that correspond to the current context are invoked, and their outputs are aggregated in the final decision.

The feasibility of this approach is shown in the three domains that have emerged as essential building blocks in AmI: activity recognition, energy-expenditure estimation, and fall detection. For each of these domains, the thesis offers an appropriate description of the domain, its relevance, and its most relevant related work. The application of the CoReAmI approach to each problem domain is then described, followed by a thorough evaluation of the approach. The results show that CoReAmI significantly outperforms the competing approaches in each of the domains. This is mainly due to the fact that, by extracting multiple sources of information and combining them by using each source of information as a context, a multi-view perspective is created, which leads to better performance than with conventional approaches.

Povzetek

Dostopnost majhnih nosljivih senzorjev z nizko porabo energije in nizko ceno ter napredne metode za procesiranje signalov in luščenje informacij so omogočile razcvet ambientalne inteligence. Pojavila se je vrsta novih metod in tehnologij za natančno merjenje na področju zdravstva, izboljšano športno vadbo in spremljanje življenjskega sloga.

Zgodnji sistemi ambientalne inteligence so uporabljali po eno vrsto senzorjev, kar je zadostovalo za prvo potrditev uporabnosti ambientalne inteligence. Z razvojem področja pa so se začeli pojavljati sistemi, ki združujejo več vrst senzorjev, kar je spodbudilo razvoj naprednejših in natančnejših metod za obdelavo večsenzorskih podatkov ter razvoj novih aplikacij. Vendar je kombiniranje več virov informacij iz različnih senzorjev zahtevna naloga. Vsak senzor ima namreč svoje tehnične lastnosti (npr. frekvenco vzorčenja) in zahteva uporabo prilagojenih tehnik za predprocesiranje podatkov, ki omogočajo usklajevanje različnih senzorjev in luščenje uporabnih informacij. A tudi če ta problem rešimo, ostaja izziv na inteligenten način kombinirati informacije iz različnih virov, da lahko pravilno sklepamo o uporabniku ali okolju. Sicer je znanih več metod, ki kombinirajo podatke in znanje iz različnih virov (Kalmanovi filtri, strojno učenje z ansambli, co-training ipd.), niso pa prilagojene za naloge ambientalne inteligence.

V disertaciji opišemo nov postopek za kombiniranje več virov informacij, pridobljenih z nosljivimi senzorji, imenovan CoReAmI (Context-based Reasoning in Ambient Intelligence). Sestavljajo ga tri faze: luščenje, modeliranje in združevanje kontekstov. V prvi fazi kontekste izluščimo iz senzorskih podatkov. V drugi fazi problem modeliramo s pomočjo izluščenih kontekstov. Ko vrednotimo nov primer iz problemske domene, pa v tretji fazi uporabimo modele, ki pripadajo kontekstom tega primera, in na koncu združimo njihove izhode.

Učinkovitost opisanega postopka smo pokazali na treh domenah, ki so se izkazale za temeljne gradnike ambientalne inteligence: prepoznavanju aktivnosti, ocenjevanju porabe človeške energije in zaznavanju padcev. Vsako domeno opišemo, pojasnimo njen pomen in predstavimo glavno sorodno delo. Sledi razlaga, kako se v domeni uporabi postopek CoReAmI. Zatem ovrednotimo uspešnost postopka in podamo rezultate poizkusov. Rezultati CoReAmI se izkažejo za bistveno boljše od rezultatov metod, s katerimi se primerjamo. Glavni razlog za to je, da uporaba več virov informacij in njihovo kombiniranje na način, da vsakega uporabimo kot kontekst, omogoča pogled na problem z več perspektiv, kar pripelje do boljšega delovanja kot pri običajnih metodah.

Contents

Abstract	VII
Povzetek	VIII
Abbreviations	XI
Symbols.....	XIII
1 Introduction.....	1
1.1 Challenges	2
1.2 Approach and Hypothesis	3
1.3 Scientific Contributions.....	3
1.4 Impact and Publications	3
1.5 Thesis Overview.....	5
2 Background	7
2.1 Ambient Intelligence	7
2.1.1 Sensing	9
2.1.2 Reasoning	10
2.2 Data Fusion	12
2.3 Context and Context-based Approaches in AmI.....	13
3 Context-based Reasoning in Ambient Intelligence – CoReAmI Approach	17
3.1 Multi-view Machine Learning Approaches	17
3.2 CoReAmI Approach.....	20
3.3 Context Extraction.....	22
3.4 Context Modeling.....	27
3.4.1 Classification	32
3.4.2 Regression	33
3.4.3 Expert Rules	35
3.5 Context Aggregation	36
3.6 CoReAmI Time Complexity Analysis	40
3.7 Summary and Discussion	42
4 Activity Recognition Domain	45
4.1 Related Work.....	46
4.2 Activity Recognition with CoReAmI.....	48
4.2.1 Sensors.....	50
4.2.2 Baseline Activity Recognition.....	51
4.2.3 Context-based Activity Recognition	53
4.3 Experimental Setup	58

4.3.1 The Experimental Scenario	58
4.3.2 Method Evaluation	59
4.4 Results.....	59
4.5 Summary and Discussion.....	62
5 Energy Expenditure Estimation Domain	65
5.1 Related Work	65
5.2 Energy Expenditure Estimation with CoReAmI	66
5.2.1 Sensors	69
5.2.2 Context Extraction	70
5.2.3 Context-based Energy Expenditure Estimation	71
5.3 Experimental Setup.....	72
5.3.1 Experimental Activity Scenario	72
5.3.2 Method Evaluation.....	72
5.4 Results.....	74
5.5 Summary and Discussion.....	78
6 Fall Detection Domain.....	81
6.1 Related Work	81
6.2 Fall Detection with CoReAmI	83
6.2.1 Sensors	83
6.2.2 Context Extraction	85
6.2.3 Context-based Fall Detection.....	87
6.3 Experimental Setup.....	89
6.3.1 The Experimental Scenario	89
6.3.2 Method Evaluation.....	91
6.4 Results.....	91
6.5 Summary and Discussion.....	96
7 Conclusions and Future Directions.....	99
7.1 Scientific Contributions	100
7.2 Future Directions	101
8 Acknowledgements	103
9 References	105
Index of Figures.....	115
Index of Tables	117
Index of Algorithms	119
Appendix A: Extracted Features for the Baseline Activity Recognition Approach.....	121
Appendix B: Extracted Features for the Activity Recognition Performed for the Fall Detection Approach.....	123
Appendix C: Bibliography.....	125
Appendix D: Biography	129

Abbreviations

A	...	Activity
AAL	...	Ambient Assisted Living
ADL	...	Activities of Daily Living
AFP	...	Acceleration Fall Pattern
AI	...	Artificial Intelligence
AmI	...	Ambient Intelligence
ANN	...	Artificial Neural Network
ANN-Acc	...	Artificial Neural Network using Acceleration only
AR	...	Activity Recognition
ARS	...	Activity Recognition System
AVC	...	Acceleration Vector Changes
BR	...	Breath Rate
BM	...	Body Movement
BSN	...	Body Sensor Network
CA	...	Current Activity
CoReAmI	...	Context-based Reasoning in Ambient Intelligence
ECG	...	Electrocardiography
EE	...	Energy Expenditure
EEG	...	Electroencephalography
EMG	...	Electromyography
F	...	F-measure
FD	...	Fall Detection
GPR	...	Gaussian Processes Regression
GPS	...	Global Positioning System
GPS-INS	...	Global Positioning System-Inertial Navigation System
GSR	...	Galvanic Skin Response
HCI	...	Human-Computer Interaction
HMM	...	Hidden Markov Model
HR	...	Heart Rate
HSN	...	Home Sensor Network
IMU	...	Inertial Measurement Unit
kNN	...	k-Nearest-Neighbors
L	...	Location
LT	...	Last Transition
M5P	...	Model tree, M5 Prime
MAE	...	Mean Absolute Error
MEMS	...	Microelectromechanical Systems
MET	...	Metabolic Equivalent of Task
ML	...	Machine Learning
MLR	...	Multiple Linear Regression
OWL	...	Web Ontology Language
PA	...	Previous Activity
p.p.	...	percentage points

RF	...	Random Forest
RFID	...	Radio-frequency Identification
RMSE	...	Root Mean Squared Error
RMS	...	Root Mean Square
RTLS	...	Real Time Location System
S/S	...	Sitting/Standing Activity
SCI	...	Science Citation Index
STD	...	Standard Deviation
SVM	...	Support Vector Machine
SVR	...	Support Vector Machine for Regression
UWB	...	Ultra-wideband
WEKA	...	Waikato Environment for Knowledge Analysis

Symbols

\mathbf{x}_t	...	observation vector at a time point t
\mathbf{X}_T	...	observation sequence consisting of T observation measurements
X	...	set of observation sequences
w	...	sliding window
$dIns$...	data instance feature vector
c	...	context
C	...	set of contexts
v^c	...	context value of the context c
V^c	...	set of context values of the context c
f^c	...	function that extracts context value for the context c
Rv^c	...	reasoning data for the context value v^c for the context c
m^c	...	context model for the context c
d	...	the output (decision) of a context model
a	...	function that aggregates the outputs of each context model
b	...	function that models the reasoning data
y	...	final output (decision)
$dArray$...	array of decisions
c_j	...	class label
n	...	number of all contexts
g	...	number of all context values
p	...	number of data samples in one window
h	...	number of the training data instances

1 Introduction

Ambient intelligence (AmI) is a scientific field that refers to environments consisting of smart devices (sensors and actuators) that can sense and respond to the presence of people [1][2][3]. An AmI system should work in a way that supports people's everyday life activities, tasks, and rituals in an easy, natural way using information and intelligence that is hidden in the data provided by the sensors.

The development of AmI systems that support elderly life has attracted significant attention in recent years. The European Commission introduced a project framework called Ambient Assisted Living (AAL) [4] in response to the rapid ageing of the world's population, which threatens to overwhelm society's capacity to take care of its elderly members. The percentage of persons aged 65 and over in developed countries is projected to rise from 7.5 percent in 2009 to 16 percent in 2050 [5]. This has driven the development of innovative AmI systems to help the elderly live independently for longer and with minimal support from the working-age population [6].

Technological advances in the miniaturization of sensors and microprocessors have enabled significant developments in AmI. In recent years the commercial market for consumer devices has presented a number of small, wearable, low-cost, power-efficient sensors. For example, most modern smartphones include several sensors, which provide data that can be used for numerous AmI tasks. The sensors provide information about the user, including body accelerations, physiological data (heart rate, electrocardiography (ECG), breath rate, and similar), and the user's location. The multiple sensor data, combined with advanced signal processing and information extraction, are driving the revolution in the AmI domain. This has enabled novel approaches and technologies for accurate measurements in healthcare systems, enhanced sports and fitness training, and life-style monitoring.

A key aspect in AmI is the intelligence; that is, enabling intelligent environments to make decisions based on perceived sensor data. In this way, a link is established between the sensor data and the real world in which the AmI system operates. The algorithms that transform the sensor data into a decision are called reasoning algorithms and are usually borrowed from the well-established artificial intelligence (AI) field. The technologies included in the reasoning step are the key in the process of providing the environment with intelligence. These algorithms usually include techniques that process the sensor data, analyze it, and provide a decision, which may concern the user's behavior, activities, and health.

Context is also another important aspect in AmI. In general, context is any information that characterizes the circumstances in which an event occurs [7]. The more context-aware the AmI system is about the user, the better the decision and the reasoning should be. Consider the example of an elderly user whose vital signals (ECG, heart rate, breath rate, and similar) and activities are monitored by an AmI system. At a particular moment, the system monitors that the user is sitting and has relatively high heart and breath rates. This could have been an alarming situation, but not if the user was exercising a few moments previously. Therefore, an

AmI system that is aware of the context – that is, the previous activity – should reason better than one that reasons without context.

This thesis addresses the problem of reasoning about the user by combining multiple sources of information (sensor data) and using a context-based approach. We propose and develop a novel general approach called CoReAmI (*Context-based Reasoning in Ambient Intelligence*), which uses multiple sources of information and context to reason about the user. In particular, a multiple view perspective is created, in which each source of information is used as a context separately. The proposed approach is thoroughly evaluated on three tasks that have emerged as essential building blocks in AmI: activity recognition, energy expenditure estimation, and fall detection.

The proposed CoReAmI approach is a general and modular framework that can be adapted to a range of tasks in AmI. It consists of three phases: context extraction, context modeling, and context aggregation. Each phase is generally defined and can use different techniques from the related literature, as long as the phase's purpose is achieved. In other words, multiple contexts should be extracted in the context extraction phase, context models should be constructed in the context modeling phase, and the outputs of the context models should be aggregated in the context aggregation phase.

Please note that even though in this thesis the user is the main subject to reason about, the method can be adapted to reason about the user's environment also.

1.1 Challenges

The widespread availability of sensors forms the technological layer for the realization of AmI. As the sensor technology improves, so do the techniques that analyze the sensor data. The initial approaches were based only on one type of sensor and the first proof-of-concept applications were developed. As the field matured, multiple sensors were included in the data analysis. This enabled the development of advanced techniques that combine multiple sensor data. However, combining multiple sources of information from multiple sensors is a challenging task. First, each sensor has its own technical configuration (for example, data sampling rate, sampling range, sampling accuracy, power requirements) and requires different data processing techniques in order to extract useful information. This task is addressed in the context extraction phase (Section 3.3) of the CoReAmI approach. The phase deals with the process of extracting context information from multiple raw sensor data.

Another challenge is to develop an approach that combines multiple sources of information together in order to reason about the user (for example, the activity of the user). Suitable candidates for implementation of a reasoning algorithm are machine learning (ML) approaches such as classification and regression learning. However, each ML approach should be adjusted to the particular AmI problem domain; unlike ML, the AmI approaches are highly dependent on the problem domain. This necessitates an advanced approach that combines multiple sources of information and reasons about the user. In this thesis, we deal with this challenge with the context modeling and context aggregation phases of the CoReAmI approach. In the former, each context is modeled using the data from the other's sources of information. In this way, multiple reasoning models are constructed that reason about the user from different points of view. In the context aggregation phase, the decisions of each model are aggregated and the final decision is provided.

1.2 Approach and Hypothesis

In CoReAmI, multiple contexts are extracted from multiple sensor data and the reasoning is then performed by combining the data using the context information. Using multiple contexts to reason about the user creates a multi-view perspective, which leads to better performance than with conventional approaches. The proposed approach is thoroughly evaluated and compared to the state of the art on three tasks in the AmI domain: activity recognition, fall detection, and energy-expenditure estimation.

The main *hypothesis* that is proposed, investigated, and confirmed in this thesis as follows:

Extracting and combining multiple sources of information by using a context-based approach (that is, using each source of information as a context) can lead to better reasoning performance compared to conventional approaches in an AmI domain.

1.3 Scientific Contributions

This thesis generated the following original contributions:

1. A novel, general, context-based reasoning approach in AmI called CoReAmI, which extracts and combines multiple sources of information by using each as a context. The approach reasons about the user using multiple models constructed for each of the contexts individually. CoReAmI includes the development of methods for context extraction, modeling, and aggregation. It is also supported by problem definition and theoretical analysis of context-based reasoning in AmI.
2. Applying the CoReAmI approach on three AmI problem domains, which resulted in:
 - 2.1. A new context-based approach for recognizing human activities using single wearable accelerometer, which outperformed the competing approaches.
 - 2.2. A new context-based approach for estimating human energy expenditure using multiple wearable sensors, which outperformed the competing approaches.
 - 2.3. A new context-based approach for detecting human falls using inertial and location wearable sensors, which outperformed the competing approaches.
3. A novel method for partitioning a ML dataset into multiple subsets and this way creating multiple views on the data by using each feature as a context.
4. Preparing several AmI datasets for human activity recognition, energy-expenditure estimation and fall detection (some of these are already available at: <http://dis.ijs.si/ami-repository/>).

1.4 Impact and Publications

The work in this thesis resulted in the development of AmI systems that are widely used in three European projects: Confidence [8], Chiron [9] and Commodity12 [10]. In the first of these projects, activity-recognition and fall-detection modules are used to detect alarming situations and daily behavior change of an elderly person. In the second, activity recognition is used in order to estimate the energy expenditure of users who have heart-related problems. The third project is an ongoing project, in which methods that use smartphone sensors and an

accelerometer-equipped heart rate monitor are used to recognize the activity and estimate the energy expenditure of diabetes patients.

We used the experience gained in the projects and created the real-time activity-recognition and fall-detection RaReFall system [11]. RaReFall was evaluated as the best-performing system at EvAAL 2013, the international competition in activity recognition [12]. It achieved the best overall results, including those achieved by the competitors in the previous year [13]. The competition's set-up is unique and requires each competing team to bring its own activity-recognition system at the competing living lab. The system also received significant media coverage and appeared in several national papers and TV news in Slovenia and Macedonia [14][15].

A number of previous publications underlie this thesis: four journal articles (three of which have already been published with science citation index (SCI) and one which has been accepted for publication with SCI), 17 conference papers, a book chapter, and two patent applications.

We started our research by developing **activity-recognition** approaches. Our initial studies focused on developing accelerometer data processing techniques and applying ML algorithms in order to recognize the activities of users [16][17][18][19]. The next study, [20], combined the activity-recognition and fall-detection problems and analyzed the way in which the sensor placements affect the performance of the activity-recognition approach. The research on activity recognition was then underlined with a master's thesis [21] in which inertial and location sensors were combined to recognize the activity of a user. A follow-up study was published in the Journal of Medical and Biological Engineering [22]. Encouraged by the satisfactory results achieved in that study, we created the RaReFall activity-recognition and fall-detection system, which won the EvAAL 2013 activity-recognition competition [23]. The RaReFall system was also presented as a live demo at the IEEE International Conference on Pervasive Computing and Communications (PERCOM 2014), [24]. A complete description of the RaReFall system, with its practical capabilities and performance, was published at the Jožef Stefan International Postgraduate School Students' Conference (IPSSC 2014), where it received the best paper award. Furthermore, a study of the unique setting of the EvAAL competition, including a detailed description of the RaReFall system and the experience gained at the competition, was accepted for publication in the IEEE Pervasive Computing Journal, [13]. In the most recent study, published at the European Conference on Artificial Intelligence – ECAI 2014 [24], we significantly improved the performance of the RaReFall system by applying the context-based reasoning approach described in this thesis.

Almost in parallel to the activity recognition, we started our research on fall detection. In the first study, we dealt with the detection of a fall event as a simple activity and recognized it by using only acceleration data and detecting large accelerations [16]. This approach resulted in numerous false detections during non-fall events. Consequently, we enhanced the fall-detection approach with the recognized activity of the user, [20]. We also analyzed the different sensor placements and their effect on the fall-detection performance. In the next study, [25], we introduced the concept of a context in the fall-detection domain for the first time. We also included location sensors, which significantly improved the performance of the approach. The next study was a demo presentation at the *European Conference on Artificial Intelligence*, ECAI 2012, [26]. A follow-up study was published in the *Journal on Artificial Intelligence Tools* [27]. The context-based reasoning for fall detection (which is also described in this thesis) by analyzing inertial and location sensors was first published at the *International*

Joint Conference on Ambient Intelligence (AmI-12) [28], followed by an extended version published in the *Journal of Ambient Intelligence and Smart Environments* [29].

The study on the **energy-expenditure estimation** is the most recent of the three. The most relevant work was published at the *ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp 2013), [30].

While developing the activity-recognition approach for the Chiron project, we developed a firmware for the sensors that makes it possible to use them in real-life scenarios. This work was published as a **chapter** in a book entitled *System design for remote healthcare* [31].

Finally, two **patent applications** were submitted [32][33]. The first one proposed a method and system for context-based activity recognition [32]. The second application proposed a method and system for detecting a person driving a vehicle while using a phone [33].

A comprehensive list of related publications is presented in Appendix C.

1.5 Thesis Overview

The remainder of the thesis is organized as follows.

Chapter 2 describes the background of the thesis. The chapter starts by describing the AmI field with an emphasis on two of its most important technologies: sensing and reasoning. We also present an overview of multiple sensor data fusion approaches. Finally, we describe relevant context-based approaches in AmI and in general.

Chapter 3 presents the CoReAmI approach, including sections explaining each of the phases of the approach: context extraction, context modeling, and context aggregation. At the end of the chapter a summarization and discussion is provided. This chapter represents the core of the thesis and explains the basic principles of the approach.

Chapters 4, 5 and 6 deal with the application of the CoReAmI approach to three problem domains – activity recognition, energy expenditure estimation and fall detection – respectively.

Each of the chapters includes: (i) a description of the particular problem domain and its challenges; (ii) a description of the related work and state-of-the-art approaches in the problem domain; (iii) a description of our approach – that is, applying the CoReAmI approach to the specific problem domain; (iv) a description of the experimental setup; (v) a thorough evaluation of the approach by providing results and comparison to competitive approaches; and (vi) a summary and discussion of the study and future directions.

Finally, **Chapter 7** concludes the thesis and provides directions for future work.

2 Background

In this chapter we describe the field of AmI and two key technologies that enable it to develop into a fast-growing, relevant scientific field: sensing and reasoning. Additionally, we give an overview of a scientific field that deals with extraction of information and knowledge from multiple sensor data, called data fusion. Finally, we conclude the chapter with the description of approaches that use the context(s) extracted from sensor data and analyze the user's situation from context-based perspective.

2.1 Ambient Intelligence

In recent years, people are surrounded by technology which tries to increase their quality of life and facilitate the daily activities. However, sometimes technology is difficult to handle or people have a lack of knowledge to use it. AmI is an emerging discipline that brings intelligence to our everyday environments and makes those environments sensitive to us [34]. The vision of AmI is a global intelligent environment, which is aware of the people and their state, and thus provides intelligent and intuitive interfaces embedded in the everyday objects around them. These interfaces should respond to the presence and behavior of individuals and should assist to each of them in their everyday activities and life. The basic idea behind AmI is that by enriching an environment with technology (mainly sensors and devices interconnected through a network), a system can be built to take decisions to benefit the users of that environment based on real-time information gathered and historical data accumulated.

AmI is a fast-growing multi-disciplinary area which builds upon the advances in multiple well-established areas in computer science. Augusto and McCullagh [35] propose a relation between AmI and the following five scientific areas (shown in Figure 2-1): artificial intelligence (AI), sensors, networks, pervasive/ubiquitous computing, and human computer interfaces (HCI). These are all relevant and interrelated areas but none of them conceptually covers the full scope of AmI, so AmI should not be confused with any of these areas. AmI puts together the resources from each of them to provide flexible and intelligent services to users acting in their environments. An AmI system gives more importance to the user and the intelligence needed to allow the system to anticipate and respond to needs of the user [36]. For example, AmI is not specialized for development of new ML algorithms, but mainly for their application and usage in practical situations. The research in the sensors field is oriented in development of new sensor technologies or improvement over the already existing. Again, AmI applies these sensors in practical situations for building an intelligent sensor environment. The relation with networks research is that AmI uses sensors connected in networks (wired and wireless). Pervasive and ubiquitous computing have similar paradigm as AmI, connecting numerous devices and computing using their data. However, these areas are more technically oriented and the focus is more on distributed systems and computing, while in AmI the user is in the main focus. Finally, HCI involves planning, design and uses of the interaction between human and computers. AmI also applies this concept for the interaction between the user and the AmI system.

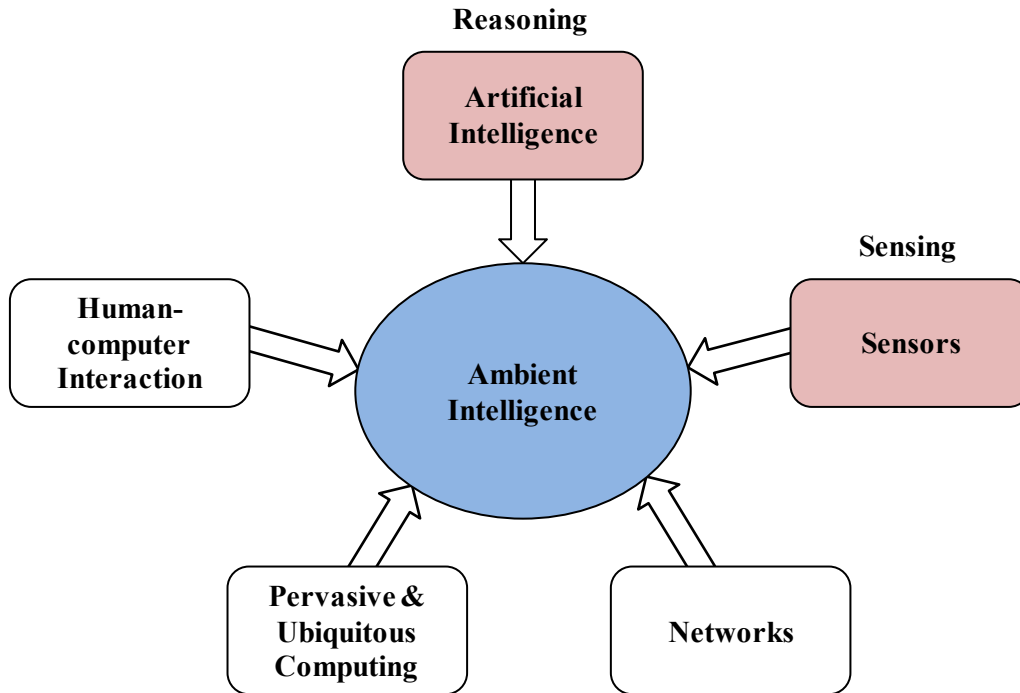


Figure 2-1. Relation between AmI and other areas in computer science [35].

AmI combines the state-of-the-art in each of these areas in order to create better autonomous systems to support and improve everyday life. As each of the areas advances, also the AmI strengthens, expands and advances. For example, as the sensors become smaller and less obtrusive, more and more people will use them, thus richer sensor data for an AmI system to advance. Additionally, better AI algorithms would result in better reasoning and decision making in AmI systems. With this thesis we mainly contribute in AmI from AI perspective, i.e., we introduce a novel context-based reasoning approach that uses AI techniques (classification, regression and expert rules) applied on different types of sensor data.

AmI vision is also the subject of criticism [37]. Because AmI is related to people and data collected from them or their environments, privacy is a great issue that requires special attention. Additionally, AmI shares the same skepticism from the society with AI, i.e., is it possible to create a system that would reason with intelligence and how much intelligent a system is. Additionally, in order for an AmI system to become commercial and widely accepted it has to persuade people in its practical abilities, not only from the technological point of view. Therefore, we strongly believe that the technological and privacy issues would get solved in time (once the practical usability and potential of the system is shown, e.g., prototype AmI systems); and the main issue that requires attention at this moment is the intelligence part and its reasoning abilities in an AmI system. Therefore, the intelligence of an AmI system is in the main focus of this thesis by proposing a novel context-based reasoning mechanism (intelligence) in AmI.

A key factor in the AmI system is intelligence. In an AmI report, provided by the IST Advisory Group [38], a list of the key technologies that are required to make AmI environments a reality is given [39]. Among all the technologies they list, the intelligence process is highly influenced by two technologies oriented to provide the environment with intelligence: sensing, and reasoning. Sensing systems allow perceiving the state of the user and the environment by means of sensors, then, reasoning systems use that data to decide how to act upon the environment to get the intended goals.

2.1.1 Sensing

The first step in the process of providing an AmI system with intelligence is to know the state of the user or users being supported by the system as well as the state of the environment itself. In order to achieve this automatically without interaction by the user, sensors are employed. Sensors are a key technology that allows linking the real world to the reasoning algorithms. There are many different types of sensors.

Recent advances in sensors technology, especially with the introduction of the MEMS (Micro-Electro-Mechanical Systems) technology, significantly increased the usage of sensors in everyday life. This technology enables sensors to be mass produced at low cost. MEMS sensors are small, light and can handle much greater shocks than conventional mechanical designs. Just as an example, today's regular smartphone includes dozens of sensors: accelerometer, gyroscope, magnetometer, microphone, GPS sensor, light sensor, proximity sensor, and similar. Depending on the type of data they provide, the sensors can be divided in three categories [40]:

- Physiological sensors, which measure ambulatory blood pressure, core body temperature, blood oxygen, and signals related to respiratory inductive plethysmography, electrocardiography (ECG), electroencephalography (EEG), and electromyography (EMG).
- Biokinetic sensors, which measure acceleration and angular rate of rotation derived from human movement. Typical representatives in this category are inertial sensors: accelerometer, gyroscope and magnetometer.
- Ambient sensors, which measure environmental phenomena, such as humidity, light, sound, pressure level and temperature.

The first two types of sensors are also known as wearable or body-worn sensors. This type of sensors is of particular interest for our research, because they give unique and important information about the observed user.

Commercial sensors exhibit a wide range of power supply requirements, calibration parameters, output interfaces, and data rates. Figure 2-2 shows the power consumption and data rate across a sampling of commercial systems for continuous, ambulatory monitoring [41].

Depending on the type of communication they use, the sensors can be wireless (Bluetooth, ZigBee, WiFi) or connected with wires (e.g., Ethernet cables). Even though the wireless sensors are more suitable for the unobtrusive AmI system, they are prone to security issues and require implementation of additional data protection techniques in order to protect the user-sensitive data.

Once multiple sensors are present in a system, a sensor network is established. If the sensors are worn by the user, the network is called body sensor network (BSN). This type of network became popular in the recent years with the increased popularity of smartphones, which enabled users to connect multiple (or a single) sensor and monitor their activities and fitness level.

Sensors usually come with unique features and challenge conventional data analysis techniques in order to make sense of the sensor data. If the sensors are imprecise, the data can be noisy, and if a sensor fails there may be missing values. Sensor data often needs to be handled as they are acquired (on the fly) or as streaming data. Additionally, the data may have a spatial or temporal component to it.

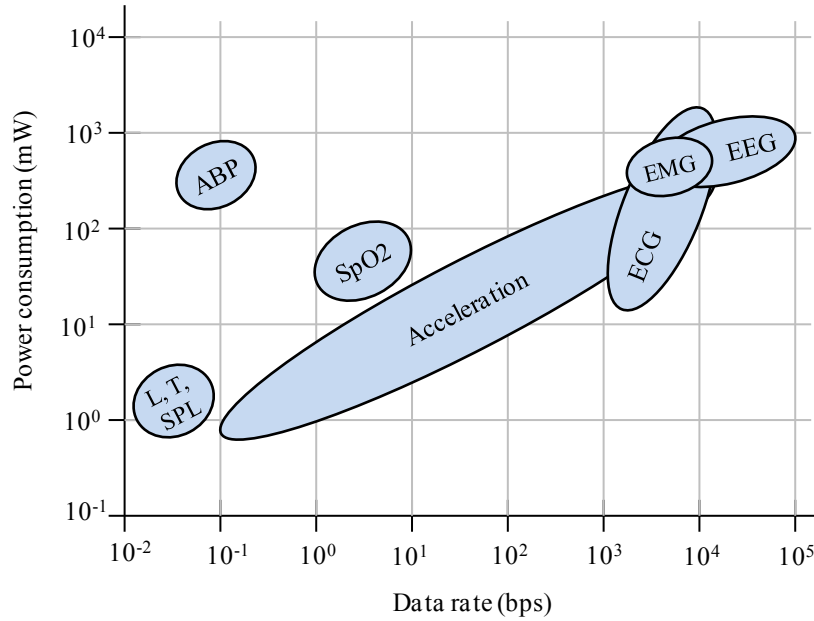


Figure 2-2. Average power consumption of continuous ambulatory monitoring applications [41].
 ABP: ambulatory blood pressure; L, T, SPL: light, temperature, sound pressure level; SpO2: pulse oximetry; RIP: respiratory inductive plethysmography; ECG: electrocardiography; EMG: electromyography; EEG: electroencephalography.

In our CoReAmI approach, the focus is on the sensor data representation as a context and the reasoning. Therefore the approach is relatively independent on the types of sensors used, as long as the sensor data can be used to extract a context. In our implementations for the three problem domains, we mainly used wearable sensors connected in a BSN. Some of them provided the data through Bluetooth communication (e.g., Xsens and Shimmer accelerometers), and some of them through Ethernet cables (e.g., the Ubisense's location data).

2.1.2 Reasoning

One of the main goals in an AmI system is to be "intelligent", i.e., to be able to make decisions, or reason, based on the sensor data. The autonomy expected from AmI system can be achieved by exploiting its reasoning capabilities, rather than by focusing on implementation issues or the available technology [42]. Therefore, in this thesis we start from the premise that reasoning is one of the most important aspects and should be tackled at the beginning of the creation of an AmI system. Other specific requirements and technicalities (sensor requirements, network requirements, privacy issues, etc.) should follow afterwards once the intelligence is established. In this section we explain several reasoning techniques that are well-established in the literature [43].

Reasoning with *expert rules* is an often used technique in the literature. In this approach the domain specific information is offered to applications by means of rules, which improves the reusability of the application. The rules are a translation of the knowledge of a domain expert and typically have a "IF condition THEN action" representation. Rule-based reasoning requires that general knowledge about a certain domain is available and can be expressed by rules. Examples of well-known rule-based reasoners are Jess [44] and Drools [45].

Ontology-based approaches are also described in the literature. These approaches usually represent the data with ontologies (hierarchy of classes, objects and relations) and then reason

about it using rules, e.g., SWRL [43][46]. In [47], logical inference on ontologies is used to generate a preoperative assessment report for patients in hospitals. In particular, the authors developed a system that uses modular ontologies developed in the OWL (Web Ontology Language) and an automated logic reasoner. In general, ontology-based approaches are especially interesting if general knowledge about the domain is known or can be derived. The concept is similar to expert-rule reasoning; however ontology-based approaches require the data to be represented by a taxonomic hierarchy of classes and relations between objects.

In recent years also *agent-based architectures* are used to reason and represent an AmI system. These approaches are usually proposed with a multi-agent architecture and represent the data and the reasoning using agents. The agents can include and simulate different behaviors, e.g., sensing, learning, acting, etc. Wang et al. [48] presented an agent-based AmI platform, developed in the JADE agent environment that facilitates fast integration of new control algorithms, device networks, and user interfaces. We also developed a multi-agent AmI system [27] that uses agent-based architecture in order to detect human falls. In that study, the reasoning about the fall event was performed by reasoning agents which were using expert rules.

Machine Learning (ML) approaches can also be used to reason in AmI. In order to perform ML-based reasoning, the reasoning task and the sensor data should be represented in a feature space, i.e., each reasoning example should be represented by set of features, i.e., feature vector. This way, a ML approach uses the previous feature vectors in order to learn a (statistical) model to reason about the situation. Typical examples are the numerous activity-recognition approaches that use classification models trained on accelerometer data in order to recognize the activity of the user [20]. Beside the supervised techniques such as classification and regression learning, unsupervised techniques such as clustering can also be used. For example, Siirtola et al. [49] presented a combination of clustering and classification in order to recognize activities of the user by using a wrist-worn accelerometer.

There are several other reasoning approaches in the literature that are worth mentioning, because each of them presents a new, specific way of reasoning and deals with different aspect of the reasoning paradigm. *Case-Based reasoning* is based on the idea that problems tend to repeat, meaning that new problems are often similar to previous ones and thus past solutions may be of use in the current situation [50]. Case-based reasoning is applicable to problems when the domain is not understood well enough for expert-based modeling [51]. *Diagrammatic reasoning* uses visual representations (charts, graphs, maps) for the reasoning. This kind of reasoning requires that the problem domain can be represented using a diagram [52]. *Qualitative reasoning* systems reason about the behavior of physical systems, without having precise quantitative information. *Fuzzy reasoning* techniques take into account that not all information is known at every moment in time or that variables might have more than one value [53]. Fuzzy reasoning is interesting as sensor systems often contain imprecise or missing information. In [54], fuzzy representations of user interests, and content meaning are combined with ontologies to improve the accuracy and reliability of personalized information retrieval. *Logic reasoning* uses techniques from logic to abstract a problem and reasons about the problem using the rules of a formal language. Afterward, the results need to be interpreted. Semantic reasoners such as Pellet [55], FaCT++ [56], and RacerPro [57] use description logics to perform reasoning.

Some of these reasoning approaches are also used for context-based reasoning, such as ontology-based, ML, and expert rules. A discussion about the application of these approaches by including context to the AmI domain is provided in Subsection 2.3.

Our CoReAmI approach is relatively independent on the types of the reasoning techniques that are applied. It can include an arbitrary technique from the reasoning techniques explained above. In particular, in this thesis we applied CoReAmI on three problem domains, in which we used: ML approaches (classification) for the activity recognition, ML approach (regression) for energy-expenditure estimation, and expert rules for the fall detection. We plan to test other reasoning approaches for the future implementations of the CoReAmI approach.

2.2 Data Fusion

Data fusion is another paradigm that deals with sensor data, their combination and to some extent with reasoning. Sensor data fusion can be defined as the process of collecting information from multiple and possibly heterogeneous sources and combining them to obtain a more descriptive, intuitive, and meaningful result. With the intelligence (reasoning) expectation increasing, using multiple sensors is one of the ways to obtain the required breadth of information, and fusing the outputs from multiple sensors is the way to obtain the required depth of information when a single sensing modality is inadequate [58]. However, different sensors may use different physical principles, cover different information space, generate data in different formats at different updating rates, and the sensor-generated information may have different resolution, accuracy, and reliability properties.

To date, this paradigm has been widely considered in fields such as defense [58], air-traffic control or robotics [59]. Multi-sensor data fusion has suggested more than 30 fusion architectures [60] that represent the different phases of the fusion procedure (see [61] for a survey). One widely used application of sensor fusion is GPS-INS. There, the GPS (Global Positioning System) and INS (Inertial Navigation System) data is fused together using Extended Kalman Filter [59]. This is useful, for example, in determining the altitude of an aircraft using low-cost sensors.

When analyzing sensor data, data-fusion systems may employ a centralized or distributed model [62]. Sensors in the centralized model transmit data to a central server, which fuses and analyzes the data it receives. In the distributed model, each sensor has onboard processing capabilities and performs local computation before communicating partial results to other nodes in the sensor network. The choice of model directly affects the computational architecture and type of sensor that is used for the task. In both cases, sensor data is collected from disparate sources and later combined to produce information that is more accurate, more complete, or more insightful than the individual pieces. Kalman filters are a common technique for performing sensor data fusion [63]. Probabilistic approaches [64] have also been effective for modeling sensors and combining information from disparate sources.

The general data fusion model proposed by JDL Data Fusion Group initially included four differentiating process levels. Levels 1 and 2 are generally concerned with numerical information and numerical fusion methods (such as probability theory – Bayesian approach, or Kalman filtering [65]). Levels 3 and 4 are concerned with the extraction of knowledge or reasoning to some extent. Levels 3 and 4 are thus concerned with the extraction of high-level knowledge (situation awareness for example) from low level fusion processes, the incorporation of human judgment and the formulation of decisions and actions. Numerous sensor fusion algorithms exist in the literature such as: sensor agreement (e. g., voting, sensor selection [66]), fault-tolerant abstract sensors [67], and decision methods (e. g., Bayes inference, Dempster-Shafer reasoning, Fuzzy logic inference [68]).

When it comes to AmI, not many studies are focused on the level 3 and 4 data fusion. Usually the studies are focused on level 1 data fusion and how to aggregate multi-sensor data from technical aspect [41] (i.e. communication protocols, saving data into databases, providing services etc.). Hongwei *et al.* [40] presented an AmI sensor fusion architecture, which mainly deals with levels 1 and 2; however ideas and concepts for the other two levels are introduced. The proposed architecture consists of 4 parts: body sensors network – BSN (provides body sensing data of the users), home sensor network (consists of both the body-worn sensors and the ambient sensors installed in the user's home), access devices (gateways), and a central server (used to store the profiles information of the users, the sensory data collected, the detection results, and report and alarm logs).

In this thesis we focus on the 3th and 4th level, i.e., extracting information and knowledge from sensors data. Therefore, the CoReAmI approach assumes that the sensors are already connected, and the data is reliably and constantly received. CoReAmI mainly focuses on the data processing part and extracting knowledge by using context-based reasoning techniques.

2.3 Context and Context-based Approaches in AmI

When two people are communicating a great deal of information is conveyed without explicit communication. For example a waiter extensively and implicitly uses situational or context information in order to provide better service, e.g., if the customers sit on a table and have a conversation, the waiter (in order to be polite) may wait for them to finish and then approach the table and provide the service. The context helps to facilitate grounding between participants in an interaction. In general, the context is any information that characterizes the circumstances in which an event occurs.

However, in human-computer interaction, there is little shared context between the human and the computer. Usually the computer is aware only of the specific task that is given and no context information is present. To overcome this problem, the concept of context-aware computing is developed. The idea of “context-aware computing” is to have computers understand the real world so that human-computer interactions can happen at a much higher abstraction level [69], hence to make the interactions much more user friendly or transparent to human users. Applications that use context to provide task-relevant information and/or services to a user are called context-aware applications.

The term “context-aware” was first introduced by Schilit *et al.* [70][71] in 1994 to address the ubiquitous computing mode where a mobile user's applications can discover and react to changes in the environment they are situated. While the basic idea of context-aware computing may be easily understood using some examples like the previously described ones, Dey and Abowd [72] tried to formally define it as: a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. They further suggested formally classifying context-aware computing into three categories: (i) presentation of information and services to a user; (ii) automatic execution of services for a user; and (iii) tagging of context to information for later retrieval.

Typical application of a context-aware computing is the smartphone/tablet switching the orientation of the screen depending on the direction to which the user is holding the device (e.g., horizontal or vertical). This way, the content of the device is adjusted to the user's orientation. Another successful context-aware function is turning off the screen of the smartphone when the user is talking on the phone and is holding the phone to the ear. Turning the screen off when not needed, results in prolonging the battery life of the device.

The context information is very often associated with location. However, Schimdt et al. [73] showed that there is more to context than location. Furthermore, the context is not necessarily associated to an event. Dey et al. [7], defined the context as any information used to characterize the situation of an entity, which can be a person, a place or an object. Thus, the context includes both the users and the environment information. Dey aggregated the context information using four categories: location (where), identity (who), activity (what) and time (when). Ferreira et al. [74], later extended this list by adding the fifth dimension: trigger (why). In general, the context information may consist of many different parameters such as location, status of the environment (e.g. temperature or light), vital signs (e.g. heart rate or blood pressure), and many others. Korkea-aho [75] summarized the following list of contexts:

- Identity of the user
- Spatial information: locations, orientation, speed, acceleration, object relationship in physical space
- Temporal information: time of the day, date, season of the year
- Environmental information: temperature, humidity, air quality, light, noise level
- Social situation: whom the user is with, the nearby people, family relationships
- Nearby resources: accessible devices, hosts, other facilities
- Resource usability: battery capacity, display resolution, network connectivity, communication bandwidth and cost
- Physiological measurements: blood pressure, heart rate, respiration rate, muscle activities, tone of voice
- User's physical activity: talking, reading, walking, running, driving a car
- User's emotional status: preferences, mood, focus of attention
- Schedules and agendas, conventional rules, policies

In general, the decision of which information should be considered as a context and which not, depends from case to case. For example, if the situation is that two people are talking, context can be: the location, the hand gestures, heart rate, etc. However, if the situation is the heart rate of the user, the context can be conversation, the location, etc.

One key aspect of an AmI system is the use of context information. Therefore, a correct information representation and management is vital. It is not enough to gather information about the context, but that information must be represented and processed in the right way.

Ontology-based approaches have shown to be successful in representation of context information. The web-ontology-based languages (e.g., OWL) are common formalisms for context representation. Recently, several reasoning languages and tools have been adjusted to reason and work with context information (Pellet [55], FaCT++ [56], and RacerPro [57]). Their goal is to retrieve relevant information, check the consistency of the available data, and derive implicit ontological knowledge. Turhan et al. [14] presented a case study from the AmI domain, i.e., a context-aware door-lock. They presented an ontology-based approach for context representation and reasoning using description logic (DL). In particular, they have built an OWL schema to represent the contexts, and tested three DL reasoners (RACER, its commercial successor RacerPro, and Pellet). However, their scenario is rather too simple to evaluate the performance of these reasoners in much broader context-aware applications.

In general, ontology-based approaches have the advantage of representing the context information and the relations between contexts in a structured way (e.g., context taxonomy). However, the reasoning capabilities are limited and thus cannot serve as a standalone solution for the needs of ambient context-aware applications.

In recent years, learning and reasoning using context proved to be effective and to outperform the competing (non-context) approaches in the AmI domain.

In the human energy-expenditure domain, researchers realized that single-regression approaches cannot accurately predict physical activity intensity across a range of activities and that different activities require different energy-expenditure regression models. Therefore, they used the activity of the user as a context. Crouter et al. [76] used the acceleration counts in order to divide the activities into three categories and assigned appropriate energy-expenditure estimation equations. Lester et al. [77] used a Naive Bayes classification model to first recognize three activities (rest, walking and running) out of the accelerometer data, and then to apply the appropriate regression equations in order to estimate the energy expenditure. Vyas et al., [78] proposed a method that uses an activity-recognition model that recognizes dozens of activities which are used as context, and then it combines multiple regression models according to the probabilities for the recognized activities.

Context-based approaches also proved successful in the fall-detection AmI domain. A context-based approach to fall detection is presented in the study by Li et al. [79]. In particular, they used five wearable accelerometers and two environmental sensors that monitored the vibration of the furniture. They combined the user's posture information extracted from the accelerometers, and the context information extracted from the environmental sensors, in order to detect the fall situations. They successfully tested their approach on events such as slow falls and fall-like events using three test subjects.

Context-based approaches have been used also for activity recognition; in particular for group activity recognition (recognizing activities for a group of people). Lan et al. [80] presented a context-based ML approach to group activity recognition by using cameras. They improved their basic activity recognition by including contexts, in their case called action context descriptors. These descriptors included information not only for the activity of the particular person, but also the context, which was represented by the behavior of the other people nearby.

By integrating more context information and more complex contexts, the need for modularization and standardization of the context-based approaches appeared. Many context-aware computing research projects address or include this topic in the course of providing system architecture support ([81][82][83]).

The Context Toolkit system (<http://www.cs.cmu.edu/~anind/context.html>) developed in the Georgia Institute of Technology GVU (Graphics, Visualization & Usability) Center is regarded quite successful in supporting modularizing system components [84]. It effectively separates concerns of context from its usage. However, the main focus in the toolkit are the technical aspects of how to represent context out of sensor data (communication protocols – HTTP, SMTP, TCP/IP etc., markup languages, client-server architecture, classes, objects, providing services, etc.) and not the reasoning with the context information. In this thesis we focus mainly on the context-based reasoning, i.e., once the context information is extracted how to combine it in order to make a better decision in an AmI system.

Shaofeng et al. [85] presented a context-aware architecture for AmI task – elderly healthcare. They propose a context-based data fusion module that combines the information from previously extracted contexts and reasons about the user's health. In particular, they

propose a Bayesian network that combines four sensor data components to assess the patient's heart status: activity, temperature, heart rate, ECG waveform morphology and heart status.

Approaches that use single user context in order to reason about the user usually perform better than the ones that do not use the context information [77][78][79]. However, reasoning about the user by using multiple contexts is a challenge, which is addressed in this thesis.

In this thesis we also propose a context-based reasoning approach that uses multiple contexts extracted from the sensor data. The sensor data is modeled using each of the contexts individually. The result is multiple viewpoints about the same situation, each view corresponding to a particular context of the user.

3 Context-based Reasoning in Ambient Intelligence – CoReAmI Approach

In this thesis we propose a novel *Context-based Reasoning* approach in *Ambient Intelligence* called *CoReAmI*. It is based on two principles: (i) using context and (ii) using multiple points of view on the same situation.

In order to explain the first principle, consider an example of a user whose heart rate and activities are monitored by an AmI system. Suppose that the system monitors that the user is sitting and has relatively high heart. This could have been an alarming situation, but not if the user was exercising a few moments previously. Therefore, a system that is aware of the context – that is, the previous activity – should reason better than one that reasons without context.

The second principle is related to using multiple views in order to reason about a user or environment in general. An intuitive example of this concept could be sensing food, a process of forming a decision ("complete picture") about the food that we eat. When we eat, multiple senses contribute to forming the "complete picture". First, we use the sight to collect the information about the appearance of food. Then, we usually smell the food and finally we taste it. Each of the senses gives unique information about the food and when all three inputs are combined, the "complete picture" of the food is formed. However, the three inputs are combined in an intelligent way, not independently. They are combined in such a way that if some sense is missing it influences also the other two. A typical example is when we have a cold and most of the food that we eat has the same taste, and that is only because we cannot smell right.

Using these two principles, we developed CoReAmI, which reasons about the situation from different points of view created by using each source of information as a context individually.

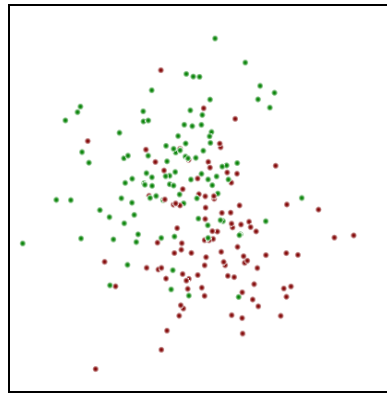
3.1 Multi-view Machine Learning Approaches

The idea of reasoning by using multiple views of the same problem, in AI and ML relates to the *principle of multiple knowledge*. This principle was formally defined by Gams [86], who stated that "*in order to obtain better performance, it is generally better to construct and combine multiple models than to use one model alone*". In this case, one or multiple models corresponds to a view on the problem (data) [87][88]. Numerous empirical studies in AI confirmed this principle, i.e., by constructing and combining multiple views for the problem one should expect better performance compared to the construction of a single, general, model over the whole decision space [89][90][91].

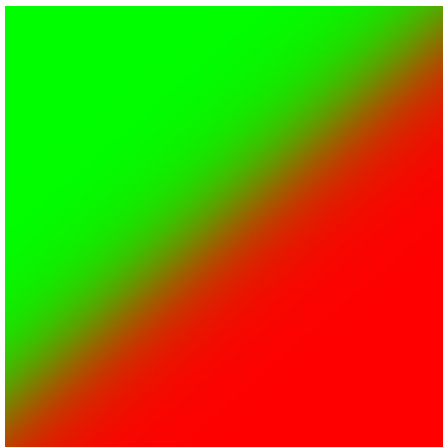
The multi-view principle is extensively studied in ML by ensemble learning approaches. The main idea behind ensemble learning is to train multiple learners (base learners) to solve the problem and then to combine their outputs to provide the final output. This way, an ensemble exploits the complementarity of multiple models and makes a better decision compared to single-model approaches. Dietterich [92] studied the process of combining

(aggregation) of the decisions provided by multiple models. He showed that it is better to find a good aggregation function instead of choosing the best single model. This is also empirically shown by numerous successful applications. Namely, ensemble-based approaches have shown to be successful and are state-of-the-art in numerous ML domains. For example, the well-known ML competition, KDD Cup [93], was won by ensemble methods most of the times.

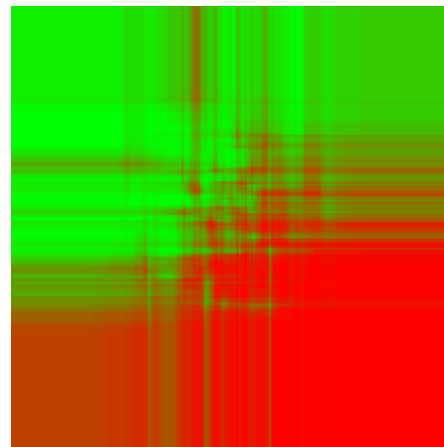
For an intuitive comparison between single models and multiple ones (ensemble-based learning) consider the dataset shown in Figure 3-1 (a) [94]. It consists of a number of data instances (examples): green and red dots. The problem is to find a model (function) that will split the dataset in the best way (to separate the green from the red dots). A linear classification model (shown in Figure 3-1 (b)) splits the data with a single line. The model is quite simple and the error (portion of the green being on the red side) is substantial. Now, consider a Random Forest (RF) model [96], which consists of multiple Decision Trees [95] (shown in Figure 3-1 (c)). The purity of the color indicates the portion of the trees that provided the same decision. The purity of the split is better with the RF, because it creates multiple Decision Trees (in this case 50) constructed on subsets of the whole dataset and therefore reducing the error compared to the single model. In general, single models find a single function to reason about n features. The result is a hyperplane in n dimensions, which covers all the parameter space. Overall, the model function might be too general and flattens out some areas in the parameter space, while by having multiple models one might be able to model these areas more precisely.



(a) Training dataset (binary classification)



(b) Single linear classification model



(c) Random Forest model (ensemble)

Figure 3-1. Visualization of the decision space divided by a single regression model and multiple regression models (ensemble-learning approach) [94].

The example shown in Figure 3-1 (c) is a RF, which is an ensemble approach. In particular, it is a special case of Bagging (Bootstrap aggregating) approaches [97]. Bagging approaches, in general, are based on training multiple models on different subsets of the whole training dataset, constructed by sampling the whole dataset with replacement, and then combining the outputs from each model by averaging (regression) or voting (classification). The RF method is a specific case that for each subset learns a Random Decision Tree – a modified Decision Tree that chooses a random subset of features on each splitting step.

Another well-established ensemble learning approach is the Random Subspace [98]. It is an ensemble method, proposed by Ho [98], which also modifies the training dataset; however, this modification is performed in the feature space. That is, a pre-defined number of features is selected randomly from whole feature set. This procedure is repeated multiple times, creating a different training set for each selection. Then, for each training set, a regression model is built. Similar to Bagging, the final output is provided by aggregating the outputs from each model by averaging (regression) or voting (classification).

In accordance with the ensemble approaches and the multi-view paradigm, we propose the CoReAmI approach, which also creates multiple models in order to better reason about the user or the environment. It is a general approach, which can be applied to any AmI domain where the data can be represented by multiple contexts.

CoReAmI first extracts multiple contexts from multiple sources of information (sensor data). This way, a dataset containing all the contexts and their values is created. Then, it partitions the dataset into multiple subsets according to the values of the extracted contexts i.e., *context-based data partitioning*. An example dataset with the context-based data partitioning is shown in Figure 3-2. The dataset consists of three contexts: A, B and C, and a class (decision). In this particular example in Figure 3-2, B is chosen to be a context, and the dataset is divided into three subsets, each corresponding to a value of B, i.e., B1, B2 and B3. Each row in the dataset is called data instance (a vector that contains the extracted context values and the class value). Therefore, the subset for B1 contains only those data instances (examples) with B1 value. The same procedure is performed for each of the contexts individually, resulting in multiple views, i.e., *context-based views*, of the dataset. In the next step, for each of the subsets a model is constructed that reasons about the user. Finally, when evaluating a data instance, the decisions provided by each model are aggregated together (by an aggregation function) and the final decision is provided.

Even though the CoReAmI approach is generally defined, when ML methods are used to model the subsets, it becomes an ensemble-learning technique. In particular, multiple ML models are trained on multiple subsets of the data. However, when creating the subsets, CoReAmI do not use conventional ML techniques, such as: sampling with replacement (Bagging) or choosing random features (Random Subspace), but it uses semantics about domain, i.e., each feature is used as a context and the dataset is partitioned according its values. This way, CoReAmI not only exploits the complementarity of multiple models like conventional ensemble approaches, but also contains models that tend to be more accurate for a particular context than those trained on the whole training set. The reason for that is that each model is trained on a subset of the training set that is more homogeneous than the whole set, and used in the context of this subset, i.e., to reason about data instances (examples) similar to the ones in the subset.

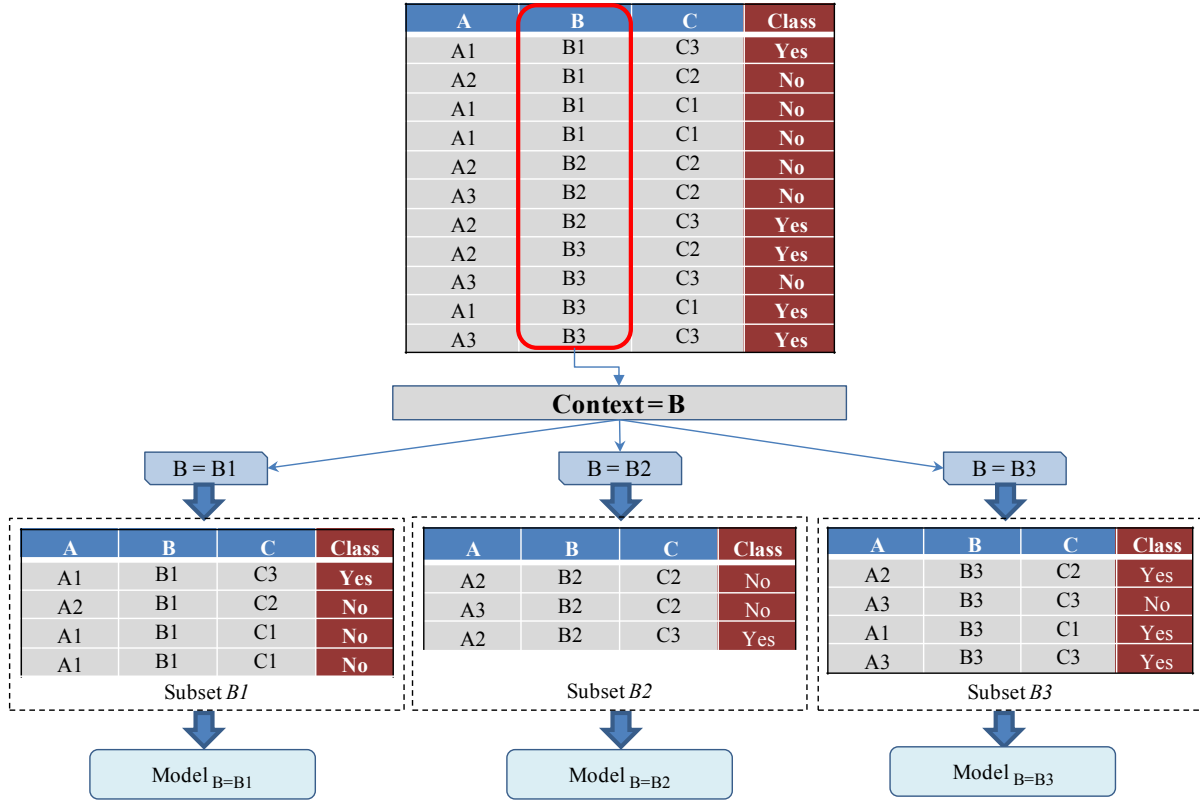


Figure 3-2. An example of a context-based view of the data. Context-based data partitioning is performed using the feature B as a context.

In CoReAmI the problem is modeled with context models, which are constructed for each context value, resulting in a single-level tree with each leaf being a model (see Figure 3-2). This is to some extent similar and can be seen as a generalization of the Model Tree approach used for regression problems [99]. In the Model Tree approach a regression tree is constructed (usually the tree has several levels and multiple features are used as a splitting criteria in each node) and in each leaf of the tree a linear regression model is learned. Therefore, if CoReAmI is applied to solve a regression problem, it becomes an ensemble of "special" Model Trees, where an arbitrary method can be used in the leaves of the tree. When constructing the tree, CoReAmI uses each feature as a context, resulting in one level tree constructed for each feature individually. On the other hand, Model Tree uses all of the features in order to construct multi-level tree.

3.2 CoReAmI Approach

The CoReAmI is a general approach for context-based reasoning in AmI. The reasoning flowchart is shown in Figure 3-3. At the top are the sensors $\{s_1, \dots, s_m\}$, which provide the raw data. The multiple sensors data is usually represented by multivariate time-series with mixed sampling rates, which are input to CoReAmI. The CoReAmI consists of three phases: (A) context extraction, (B) context modeling and (C) context aggregation. Here, we provide a brief description for each of the phases. Detailed explanation is given in Section 3.3, 3.4 and 3.5

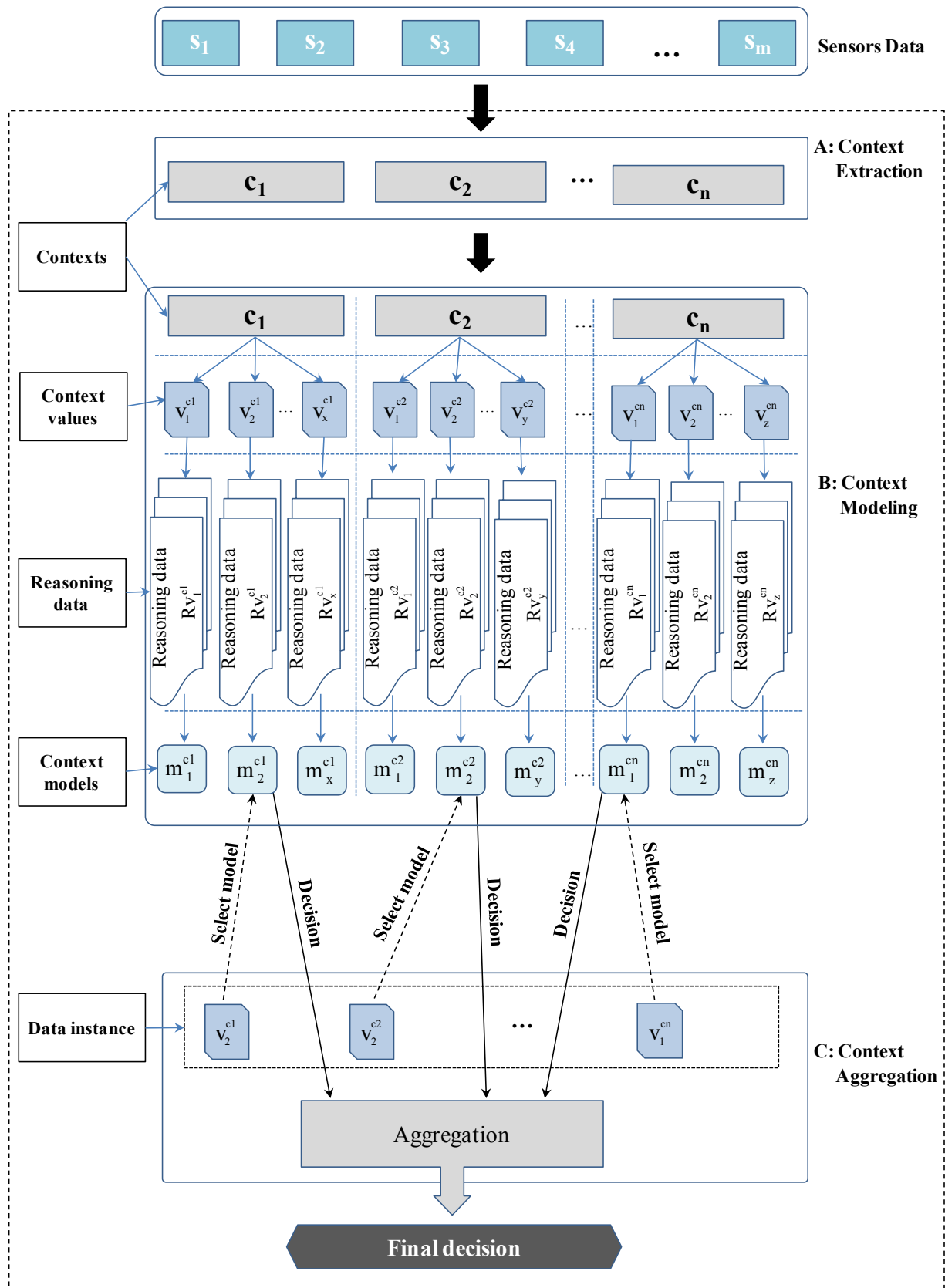


Figure 3-3. CoReAmI reasoning flowchart.

In the first phase (A) the raw sensor data are acquired and the multiple contexts are extracted $\{c_1, \dots, c_n\}$ using different types of techniques: data-preprocessing techniques, data synchronization, data segmentation, etc. In our approach, context represents information about the user which is extracted from the sensor data, e.g., user's activity extracted from wearable accelerometer data. This phase should include domain experts, who should try to define multiple contexts that are relevant for the particular problem. This phase is similar to the feature extraction problem in ML, except that in our approach the contexts are usually (but not necessarily) defined manually by the domain experts. Actually, the contexts in CoReAmI are features that represent context information. Thus, each context should describe and include unique information about the problem. This characteristic of the feature diversity is also used in the popular co-training ML approach [100], where multiple views of the data are used and the more different the data between different views are, the better the performance. Similar to features, each context has values (v^c), which can be numerical or categorical (e.g., "sitting" for the "activity" context).

In the phase B, the context modeling about the problem (activity, fall, energy expenditure, etc.) is performed using the contexts defined in the previous phase. First the context-based partitioning of the dataset is performed, i.e., the dataset is partitioned according to each context and its values. Therefore, for each context value a reasoning model (m^c) is constructed using its reasoning data – the reasoning data is a subset of the whole dataset that has that particular context value (Rv^c). For example, the reasoning data for the "sitting" model will be constructed using the data instances that contain the value "sitting" for the activity context. This way, the approach considers multiple views on the data using each of the features as a context.

In accordance with the Dietterich's conclusions [92] about the advantage of having an aggregation process, CoReAmI also uses aggregation techniques in the final phase C. For a given testing data instance, a decision is made for each context individually and all the context decisions are aggregated and the final decision is made. When the modeling is performed by classification or regression learning algorithms, the invoked models represent an ensemble of classification or regression models, respectively.

The approach is general and can be applied to a different problem in the AmI domain, although the techniques in each phase should be adapted to the specific problem. In the following three sections (3.3, 3.4 and 3.5) each of the phases is described in details. In Section 3.6 analysis about the time complexity of CoReAmI is provided, and the last section summarizes the CoReAmI approach and discusses some aspects of it.

3.3 Context Extraction

The main goal of the context extraction phase is to extract useful contexts from the sensor data, so they can later be used to reason about the user. This phase includes techniques that perform calculation over the raw sensor data, so that higher level information, i.e., context, can be extracted. Please note that when a ML method is applied in the modeling phase, the term context corresponds to the term feature, therefore the phase context extraction corresponds to feature extraction.

The context extraction phase depends on the particular problem domain and should incorporate the expert knowledge in order to define the contexts which are the most relevant for the particular domain. For example, for the fall-detection domain it is important to know what the user is doing (activity), where is the user (location) and whether the user is moving or not at a particular moment (body movements). Therefore, our context-based fall-detection

method (explained in Chapter 6) extracts those three contexts from the raw sensor data in order to detect a fall. Each of the problem domains described in this thesis uses context extraction techniques that are adapted for the particular domain.

The following two definitions define the terms *context* and *context value*.

Definition 1. *Context* c is a variable that characterizes the situation of the user. The set C of n contexts is defined as:

$$C = \{c_1, \dots, c_n\} \quad (1)$$

Each context has a set of possible values, therefore a context value is defined as:

Definition 2. Let $c \in C$. The values that correspond to the context c are called *context values*. The set that contains all the possible values of the context c is defined as V^c . The context values can be numerical or categorical (discrete) and are marked with v^c , where $v^c \in V^c$.

Figure 3-4 shows an overview of the context extraction phase in CoReAmI. Let us consider a definite number of sensors, m , which provide data about the user. Each of these sensors provides a time-series data, which may have different data sampling rates (the other technical characteristics, such as sampling range, power requirements are not part of this analysis, but in general should be taken in consideration). For instance, an accelerometer may sense at 50Hz, heart rate sensor at 1Hz, etc. All these time-series are processed in the context extraction phase. Depending on the number and type of sensors, different processing techniques may be applied. For example, if multiple time-series are received, a synchronization technique may be applied in order to align the time-series according to their timestamps. Some sensors provide noisy data; therefore appropriate filtering techniques should be applied.

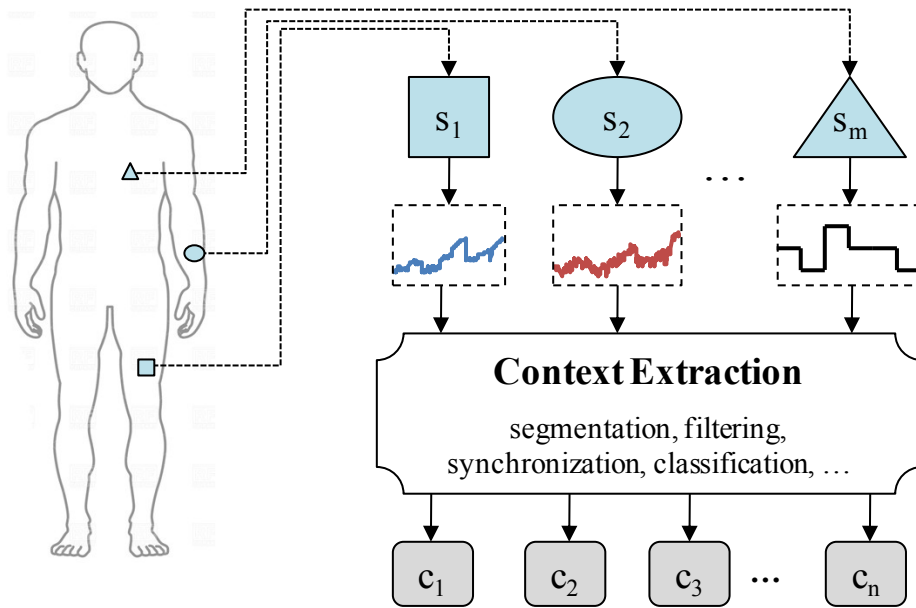


Figure 3-4. Context extraction phase in the CoReAmI approach.

In order to define the context extraction phase, let us consider a user or environment where the state is observed with several sensors providing measurements at each time step t . Sensors usually provide continuous stream of data samples. However, when a reasoning algorithm reasons about the user, it is performed over an observation sequence defined in a time interval. The techniques that segment the data into time intervals are called *segmentation techniques*.

The most commonly used segmentation technique is the sliding-window, which combines a set of data samples in one window. The time interval for which the data is collected is called *window size*. The window with fixed size (number of data samples or time interval) moves across the stream of data and segments it. Afterwards, these segments can be used in order to reason about the user. If the *windows* have some data samples as intersection, then this technique is named overlapping sliding windows. This is useful in applications when it is difficult to define strict borders between consecutive segments. A typical example is the elementary activity-recognition task, where the transitions between activities happen suddenly and it is difficult to segment the data so each sample contains only one activity. There are also more advanced techniques to segment the sensor data, which pre-analyze the data in order to find significant changes which are then considered as boundaries for the sample [101].

When multiple sensors on multiple devices are present in an AmI system, they need to be *synchronized* so their data can be analyzed together. This is a challenging task because usually commercial sensors exhibit a wide range of data sampling rates (as shown in Figure 2-2). The most convenient way to synchronize different sensor data is to label each of the data samples with a timestamp (the moment when the data sample is sensed by the sensor), and later use these timestamps in order to combine and analyze the data. Usually a Network Time Protocol (NTP) server is used to adjust the same absolute time on different devices. In our implementations, we used this approach together with the sliding window segmentation approach and we were able to combine multiple types of sensors. In particular, for each sliding window segment for each sensor the data samples that correspond (according to the timestamp) to the segment are retrieved from the database and analyzed. An example is shown in Figure 3-5, where multiple sensor data are aligned for the task of energy-expenditure estimation.

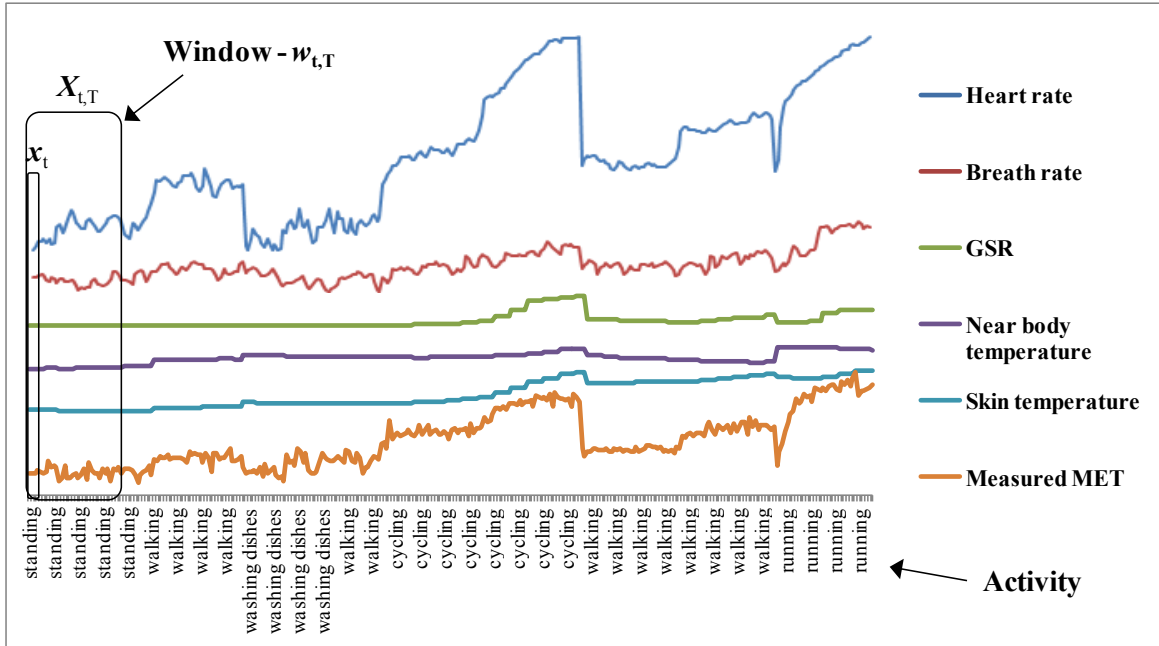


Figure 3-5. Synchronized multiple sensor data (approximately five minutes of data).

Definition 3. *Observation vector* x_t is a multi-dimensional signal vector containing values from each sensor associated to a time point t . It is assumed that it is possible to construct an observation from all the sensors regardless of the frequency with which a particular sensor provides measurements.

Definition 4. *Observation sequence* $X_{t,T}$ with length T , is defined as a vector of observation vectors, i.e., a matrix, $X_{t,T} = \{x_t, x_{t+1} \dots, x_{t+T}\}$. The observation sequence consists of the measurements that correspond to the particular time interval (window), $w_{t,T}$.

For each observation sequence $X_{t,T}$, multiple contexts are extracted. The process of transforming an observation sequence into a set of context values is called *context extraction*.

Definition 5. Let X be a set of observation sequences, i.e., $X_{t,T} \in X$. Let V^c be a set of context values for a particular context c . *Context extraction* is a process of applying a function f^c , which takes as input an observation sequence $X_{t,T}$ and gives as output a context value v^c , where $v^c \in V^c$:

$$f^c: X \rightarrow V^c \quad (2)$$

In order to extract the context value, the function f^c may include different data processing techniques. These techniques depend on the type of the sensor data and differ from type to type. For example, if the sensor provides a single value inside the window, that value can be directly used as a context value. However, if the sensor provides multiple values, an aggregated value should be computed. The aggregated value can be computed by calculating a statistical value (e.g., average, minimum, maximum, variation, etc.) or it may even be an output of a computational model that takes as input the raw sensor data and outputs an aggregated value. Additionally, some sensors provide noisy data, which requires processing the raw data with noise filtering techniques before they can be further used.

The pseudo code of the context extraction phase in CoReAmI is given with Algorithm 1:

Algorithm 1. Context extraction phase in CoReAmI.

#Phase A: Context Extraction

Input: observation sequence $X_{t,T}$, set of context C ,

Output: data instance $dIns$

#observation sequence

$X_{t,T} = \{x_t, x_{t+1} \dots, x_{t+T}\}$

#contexts

$C = \{c_1, c_2, \dots, c_n\}$

#empty data instance

$dIns = ()$

FOR each c in C **DO**

| #extract the context value for each context

| $v^c = f^c(X_{t,T})$

| #add the context value to the data instance

| $dIns.add(v^c)$

END

RETURN $dIns$

The following subsection shows an example of extraction of a commonly used context in the literature – activity of the user. The extraction of the user's activity from accelerometer data is a common task and it includes almost all of the previously mentioned data processing techniques: synchronization, segmentation, filtering, construction of ML classification model, and extracting context – the activity of the user.

Activity Extraction

In order to successfully reason about the user, an AmI system should be aware of the user's activity. Therefore, automatically extracting the activity of the user using sensor data is valuable information, i.e., a valuable context. In this subsection we present an approach to extraction of the user's activity. As an example we are using multiple wearable accelerometers. This technique of mapping sensor data output to an activity is known in the literature under the phase activity recognition [22].

The approach presented here is well-known in the AR literature. It is a ML approach where a classification model is trained to recognize an activity by using numerous features computed from the acceleration data. Once the model is trained it can be used to extract the activity using the raw acceleration data as input. An overview of the procedure is shown in Figure 3-6, and includes the following modules: data synchronization, data segmentation, data filtering, feature computation, and applying a classification model.

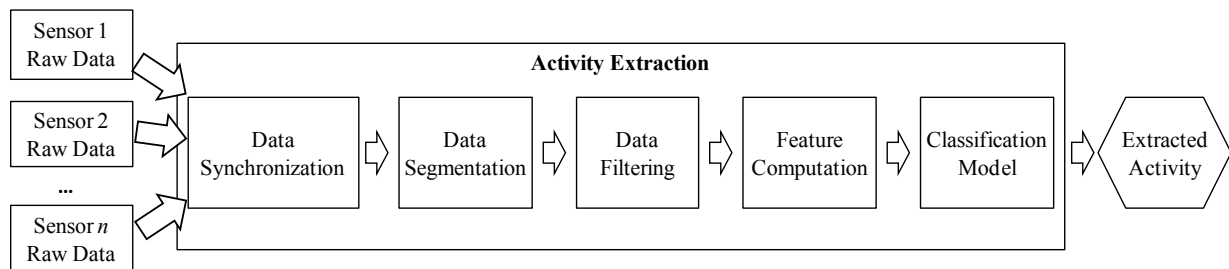


Figure 3-6. Activity extraction procedure.

Because multiple sensor data is used to extract the activity of the user, first they need to be *synchronized*. To achieve this, the data samples of the multiple sensors are aligned according to their timestamps. In the next step, the data is *segmented* using the defined window size, i.e., in this case an overlapping sliding window is used.

Once the data samples are synchronized and segmented, further processing is performed using band-pass and low-pass *filters* [102]. The acceleration is the sum of the acceleration due to the gravity and the acceleration due to the movement of the sensor (and the person wearing it). The band-pass filter thus has two goals: (1) to eliminate the low-frequency acceleration (gravity) that captures information about the orientation of the sensor with respect to the ground and (2) to eliminate the high-frequency signal components generated by noise, thus preserving the medium-frequency signal components generated by dynamic human motion. The band-pass-filtered data is used for the extraction of features relevant for dynamic activities, such as walking and cycling. The low-pass filter is used to eliminate most of the signals generated by dynamic human motion, preserving the low-frequency component, i.e., gravity [103]. The low-pass-filtered data thus contains sensor orientation information, which is highly relevant for the recognition of static activities (postures), such as lying, sitting and standing.

In the *feature computation* module, the relevant features are extracted using the preprocessed sensor data in each data window. The following features are among the most commonly used in the literature and therefore computed for each of the sensors:

- The mean value of the total acceleration vector and the acceleration along x, y and z axis.
- Standard deviation of the acceleration vector and the acceleration along x, y and z axis.

Once the features are computed, a feature vector is formed, and is fed into a *classification model*, which recognizes the activity of the user. The classification model is previously trained on feature vectors computed over training data. The classification can be performed using an arbitrary algorithm for training a classification model, e.g., Decision Tree [95], RF [96], Naive Bayes (NB) [104], Support Vector machine (SVM) [105], and similar.

To summarize, the activity extraction procedure is just an example of a relatively complex context extraction. We applied a ML approach on acceleration data in order to extract the activity of the user. However, some sensors provide data which can be used without further processing. For example, most of the commercial heart rate sensors provide the already processed heart rate values which can be used as they are. In our energy-expenditure estimation approach (Chapter 5), we showed how the heart rate values can be used as a context.

3.4 Context Modeling

Modeling is a process of constructing a model which uses the contexts to make a decision. CoReAmI exploits each of the extracted contexts individually, and the data from the other contexts are used as a reasoning data. In other words, *context-based data partitioning* is performed, i.e., the context values segment the whole data set into subsets for which a context model is constructed. Therefore, the *context model* for a particular context value v^c of a particular context c can be defined as a function, which transforms the *reasoning data* into a decision. The following three definitions define the terms: reasoning data, context-based data partitioning and context model:

Definition 6. Let $c \in C$. Let v^c be a context value of the context c . *Reasoning data* Rv^c is defined as a subset of the whole dataset, which contains only the data instances that have that particular value v^c for the context c .

Definition 7. Let $c \in C$. Let v^c be a context value of the context c . Let Rv^c be the reasoning data for the v^c . *Context-based data partitioning* is the process of creation of the reasoning data Rv^c for the context value v^c . An example is given in Figure 3-8.

Definition 8. Let $c \in C$ and $v^c \in V^c$. *Context model* m_v^c is defined over the reasoning data Rv^c for the context value v^c . The set of all context models (created for all context values for all contexts) is defined as M and $m^c \in M$. The size of the set M is defined as g , where $|V^c|$ is the size of the set of values for the context c :

$$g = \sum_c^n |V^c| \quad (3)$$

The pseudo code of the context modeling phase in CoReAmI is given with Algorithm 2.

Algorithm 2. Context modeling phase in CoReAmI. The `constructModel()` function constructs a model given a dataset. The `context_based_partitioning()` function creates a subset (reasoning) dataset, given a dataset and context value.

#Phase B: Context Modeling

Input: set of contexts C , dataset dt ,

Output: set of context models M

#contexts

$C = \{c_1, c_2, \dots, c_n\}$

#initialize empty set of context models

$M = \{\}$

FOR each c in C **DO**

FOR each context value v^c of c

 #sample the data according to the context value

$Rv^c = \text{context_based_partitioning}(dt, v^c)$

 #construct the model using the reasoning data

$m_v^c = \text{constructModel}(Rv^c)$

$M.\text{add}(m_v^c)$

END

END

RETURN M

Figure 3-7 illustrates a simplified example of the general idea of the context-based reasoning about the user's health using three contexts: activity, heart rate and breath rate. There are three possible views of the user's situation. First, the context is defined by the activity of the user. The user's heart rate and breath rate are used to reason about the health. Please note that the reasoning data may include additional data from the sensors, not just the extracted context data. For example, when using heart rate values for modeling the heart rate context, instead of using discrete values (such as low, medium and high), one may use the numeric values as provided by the sensor (e.g., heart rate: 92 min^{-1}). In the next view, the context is the heart rate and the reasoning is performed using the other two: activity and breath rate. And third, the context is the breath rate and the reasoning is performed using the user's heart rate and the activity. For each view a context model is created using the appropriate reasoning data. The models provide decisions individually, which are finally combined using an additional mathematical model. This way, a multi-view perspective is provided which leads to better reasoning about the user's health.

A conventional approach would be to reason about the user's state using the whole available information at once by creating a single reasoning model. The main difference in our approach is that we create multiple reasoning models using different contexts. Instead of directly finding a model function over all sensor data, we first consider a set of context models. These models are constructed using subsets of the whole reasoning dataset. Each subset corresponds to a

single context value. Therefore, the model constructed for the activity *sitting* uses only the data instances that contain that activity. This way, each context separates the dataset into several subsets (reasoning data) that correspond to the context values. An example is illustrated with Figure 3-8, which contains example dataset for three contexts and a decision column and a graphical illustration of dividing the dataset into subsets according to the appropriate context values. This way, we define a model for each context value of each context. In the example given, breath rate is taken as a context and for each context value the reasoning data is shown. Therefore, the reasoning data for the low breath rate contains only those data instances that have a low value for the breath rate context.

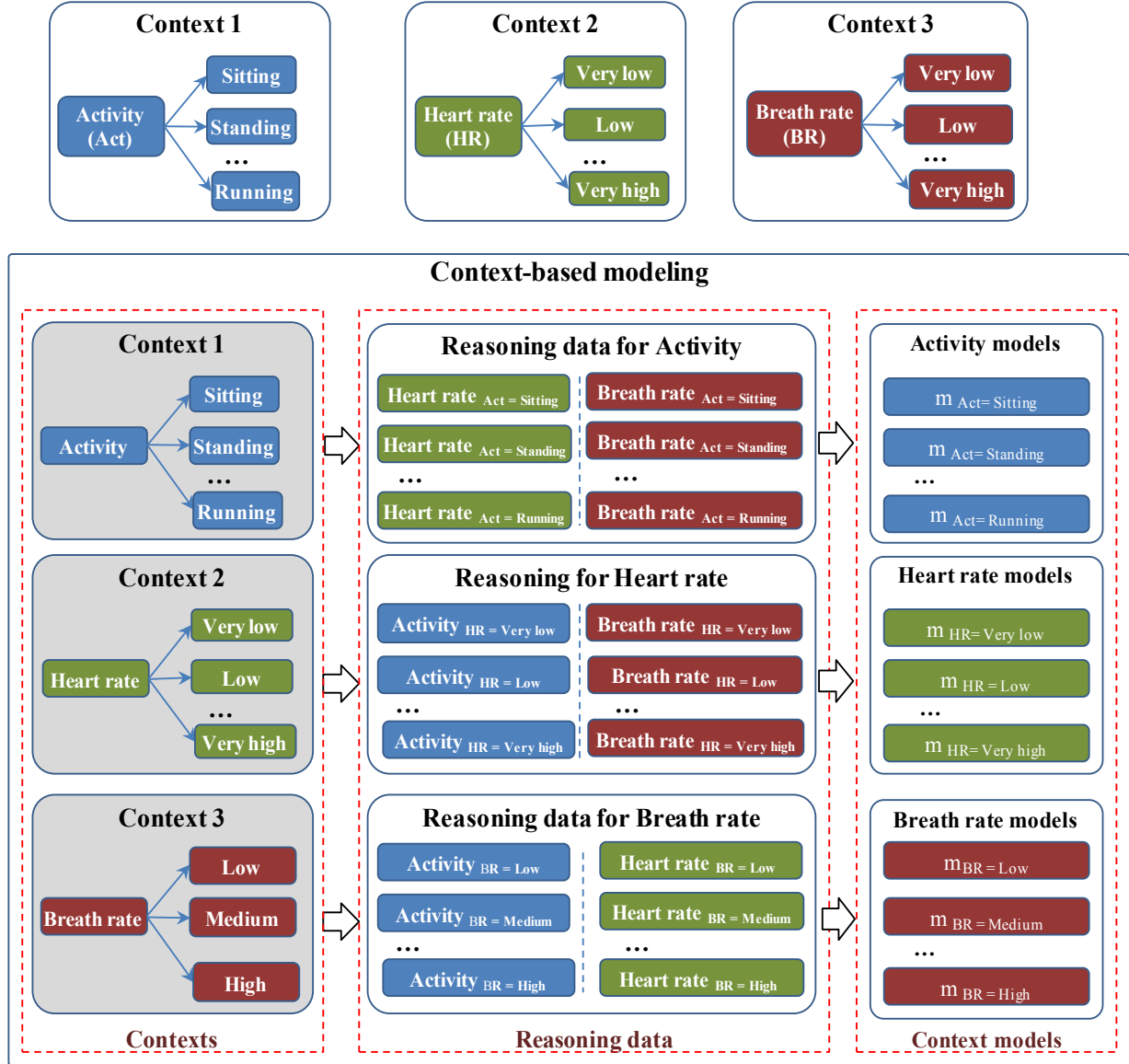


Figure 3-7. Context-based modeling phase in CoReAmI.

The modeling function can be in the form of any mathematical model. In general two types of approaches are considered in the literature [108]. In data-driven approaches, the models are constructed from pre-existing datasets using existing ML techniques. The decision is then performed against the learnt models whenever sensor data is obtained. In knowledge-driven

approaches, knowledge engineers and domain experts specify models using a knowledge engineering process. The models capture common sense and domain knowledge about the target situation. Please note that in this expert rules case, the modeling phrase refers to the process of construction of the rules. The experts may chose or not to consider the reasoning data for each of the contexts.

In this thesis, practical applications using two data-driven approaches are shown: classification (Subsection 3.4.1) and regression (Subsection 3.4.2); and one knowledge-driven approach by using expert rules (Subsection 3.4.3). In principle, other reasoning techniques can also be applied, e.g., spatial-temporal reasoning, agent reasoning, ontology-based reasoning; however they need to be adapted for a specific task.

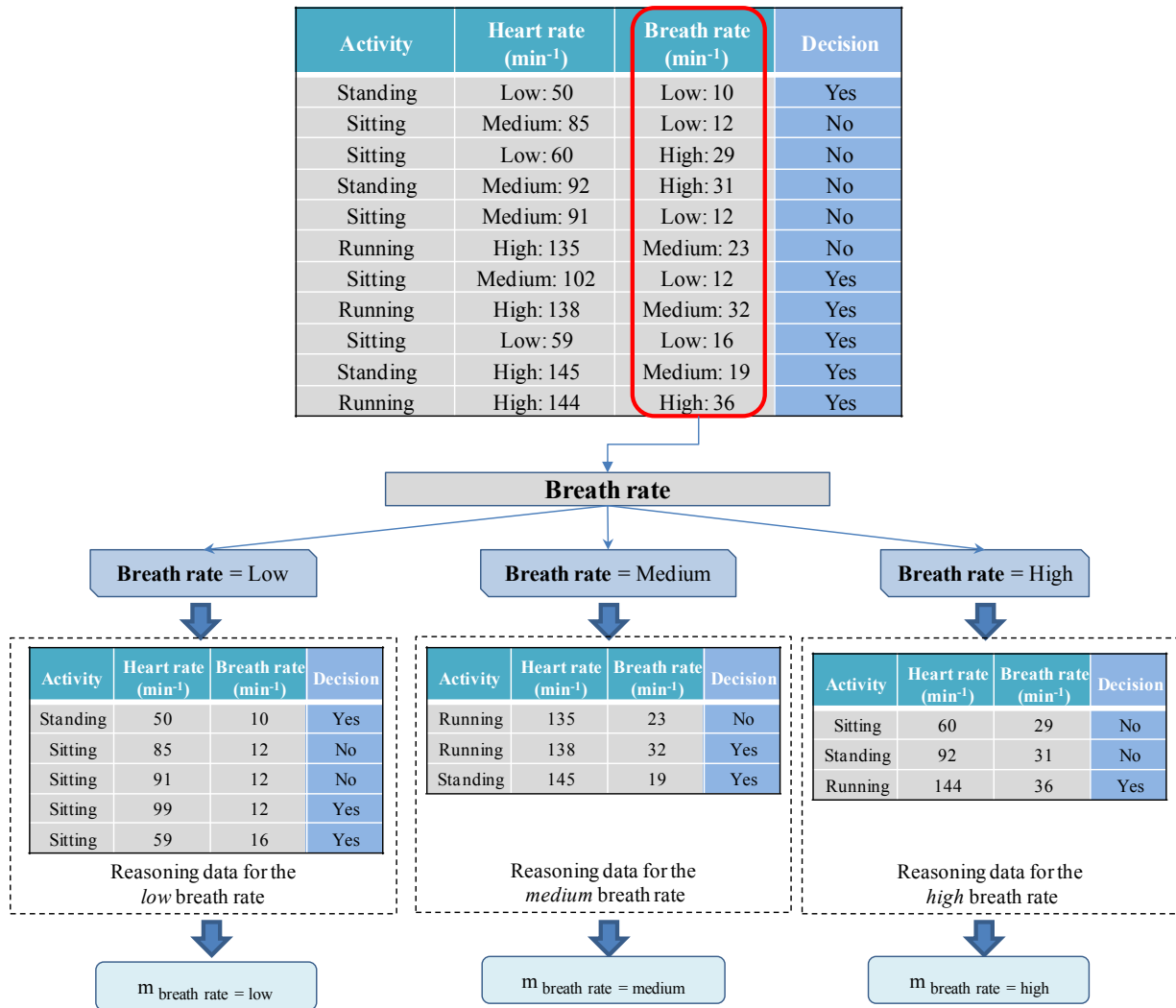


Figure 3-8. Graphical illustration of an example dataset and dividing the reasoning data according to the context values, i.e., context-based data partitioning. The breath-rate feature is chosen as a context.

Before continuing to the modeling approaches, let us consider another aspect that emerged as important in the context modeling phase, i.e., data discretization. Figure 3-8 shows that the reasoning data may contain different data than the one used to define the context. In particular, the values for heart rate and breath rate are numeric instead of the discrete values used to represent the context. In order to achieve this, CoReAmI considers different discretization techniques.

Discretization

Data-driven approaches require sufficient data in order to learn an accurate model. That is, sufficient number of data instances should be present in the reasoning data. However, this is usually an issue if the context information is represented in a numeric format, i.e., numeric context values (e.g., heart rate context). If each of the numeric values is used as a context value, that would leave only one or few data instances that have that particular heart rate value. Because of this, sometimes it is useful to represent the context values with discrete values, i.e., intervals. In the heart rate example that would be the already discussed values of: low, medium, high heart rate. Each discrete value is represented by an interval, and the data instances that have heart rate values inside a particular interval are taken as the reasoning data. This process of transforming numerical values into intervals is called data discretization.

There are two basic approaches to the problem of data discretization in the literature. One is to discretize in the absence of any knowledge of the target class (decision) of the data instances in the training set – the so-called *unsupervised discretization*. The other is to take the target class into account when discretizing – *supervised discretization*.

The simplest unsupervised discretization technique is *equal-width binning*. This technique is fixed in advance and is data-independent, i.e., divides the range of the values into a predetermined number of equal intervals. This is frequently done at the time when data is collected. But, like any unsupervised discretization method, it has a risk of destroying differences that may be useful in the learning process. Equal-width binning often distributes data instances unevenly: some bins contain many data instances while others contain none. This can seriously impair the ability of the context value to help build good models.

Another approach allows the intervals to be of different sizes, i.e., it chooses them so that the same number of training examples falls into each one. This method, called *equal-frequency binning*, divides the values range into a predetermined number of bins based on the distribution of examples along that axis – sometimes called histogram equalization because if you take a histogram of the contents of the resulting bins it will be completely flat.

Equal-frequency binning is still oblivious to the target classes, and this can cause bad boundaries. For example, if all data instances in a bin have one class, and all data instances in the next higher bin have another except for the first, which has the original class, surely it makes sense to respect the class divisions and include that first data instance in the previous bin, sacrificing the equal-frequency property for the sake of homogeneity. In order to tackle this problem supervised discretization is used. The most commonly used supervised discretization technique in ML is the Decision Tree splitting criteria [95]. It finds such splits for a given feature that the information gain of the class value is maximized. This technique is mainly used for classification tasks; however it was also adapted to regression tasks by Yong et al. [99]. Their adaptation is used in the well-known M5P regression tree algorithm [99]. Instead of information gain, it uses the standard deviation of the class value, which should be minimized. The standard deviation reduction (SDR) achieved by a given split is calculated by the formula:

$$SDR = sd(T) - \left[\frac{|T_1|}{|T|} \times sd(T_1) + \frac{|T_2|}{|T|} \times sd(T_2) \right] \quad (4)$$

where T is the set of data instances before the split, T_1 and T_2 are the sets that result from the binary split, $|T|$ is the number of data instances in the set T , and $sd(T)$ is the standard deviation of the class value. The discretization procedure tests all the possible splits and selects the one

with the highest SDR. This division process can be repeated multiple times until some criterion is met (predefined minimum number of data instances in a bin, minimum value for SDR, etc.).

3.4.1 Classification

When the decision in the reasoning problem is represented with discrete values, e.g., recognizing the activity of the user: standing, sitting, walking, etc., a classification approach can be used in order to model the reasoning data for each context. This will result in a classification model created for each context value, i.e., ensemble of classification models.

In order to present the approach we will use the activity-recognition task as an example (Figure 3-9). The goal of activity recognition is to determine the actions or the states of one or more people through the analysis of sensor data. The detailed description of the whole context-based activity recognition is presented in Chapter 4. There should be noted that there is a difference in the activity extraction approach presented in Subsection 3.3 and the approach presented here. In the previous approach, the goal was only to extract the current activity using only the data in the current window of data. In this approach, the goal is to improve upon the previous approach by including multiple contexts.

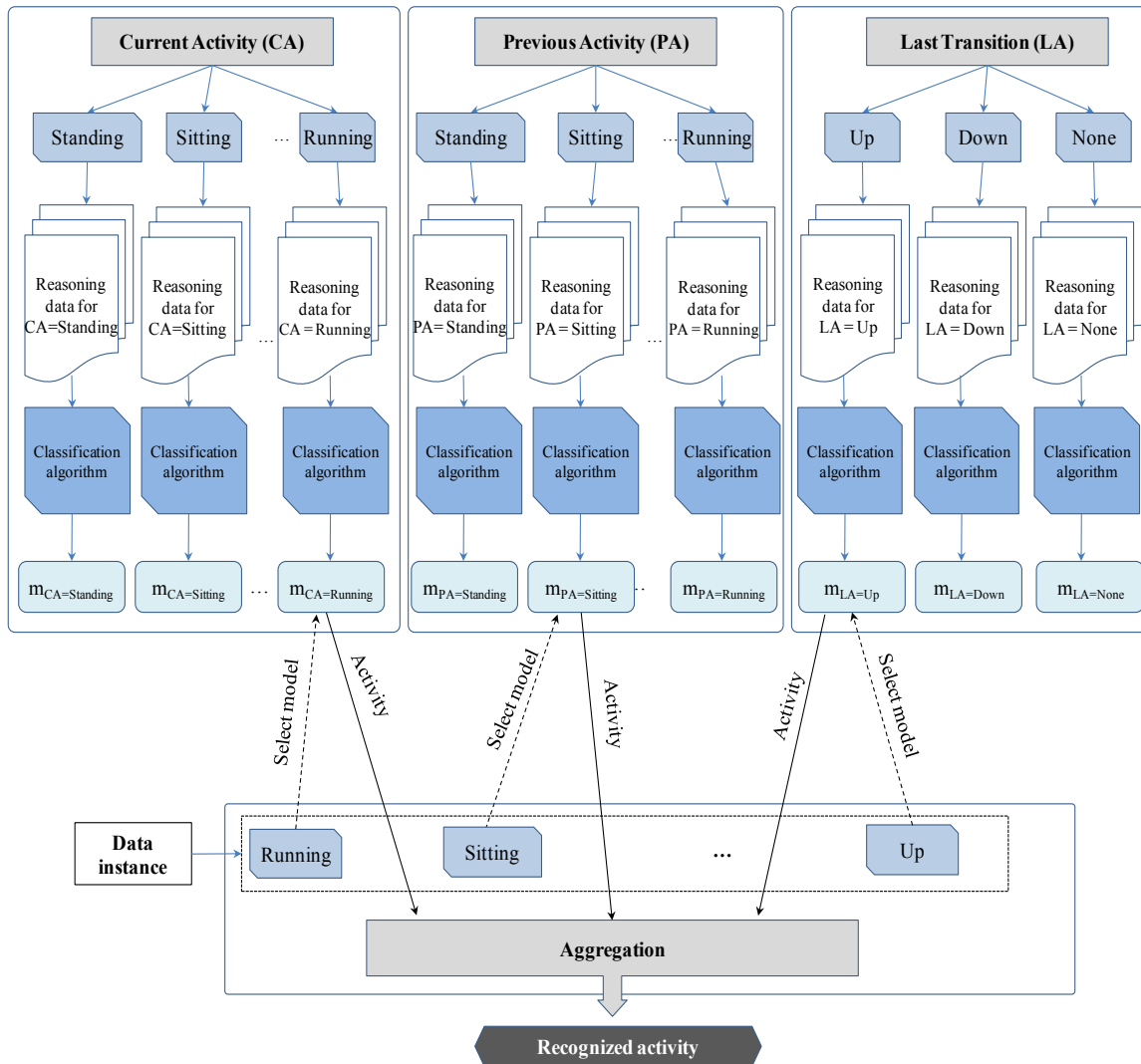


Figure 3-9. Context-based classification for activity recognition.

Figure 3-9 shows a simplified example of CoReAmI applied for activity recognition. CoReAmI consists of three contexts: the current recognized activity, the previous recognized activity and the last recognized transition. In order to simplify the explanation, let us assume that each of the recognitions is performed by a ML approach as the activity extraction explained in Section 3.3. The idea here is to use the context (previous activity and last transition) of the current activity in order to improve the recognition of the current activity. For example, if we know that the previous activity of the user is walking, the probability of the next activity being lying is smaller than the probability of being standing.

In the modeling phase of CoReAmI, for each context value (e.g., "Standing") of each context (e.g., "Current activity"), a classification model (e.g., $m_{CA=Standing}$) is trained. The training dataset for the model is a subset of the whole training dataset. It contains only the data instances which have the corresponding value of the context; thus, the model for the first context for the first value, i.e., $m_{CA=Standing}$, is trained only on the data instances that contain the value "standing" for the "Current activity" context. Once the subsets of the training data are defined, the models are trained using an arbitrary classification method, e.g., Decision Tree [95], RF [96], Naive Bayes [104], SVM [105], and similar.

For a given test data instance, a custom ensemble of classification models is constructed. Figure 3-9 shows an example of a data instance containing "Running" as the current activity, "Sitting" as the previous activity, and "Up" as the last transition. Thus, those three classification models ($m_{CA=Running}$, $m_{PA=Sitting}$ and $m_{LA=Up}$) are included in the ensemble. This way, the testing data instance is evaluated by the multiple models that should be more accurate than those trained on the whole training set. The reason is that each classifier is trained on a subset of the training set that is more homogeneous than the whole set, and used in the context of this subset, i.e., to classify examples similar to the ones in the subset. Once a data instance is classified by the appropriate models, the final decision about the recognized activity is made using a method for aggregating the outputs of the classification models: majority voting, plurality voting, etc.

3.4.2 Regression

The regression learning approach is similar to the classification approach; the only difference is that the reasoning problem is defined with numerical values instead of discrete ones. Therefore, regression learning algorithms are used in order to learn the models for each context. An example of a numeric reasoning problem is the energy-expenditure estimation using wearable sensor data. The main goal in energy-expenditure estimation is to map the sensor data into estimations (e.g., calories). We will use this task as an example to present the approach, a more detailed description of the whole context-based energy-expenditure estimation is presented in Chapter 5.

Similar to the activity-recognition approach, a simplified example of context-based reasoning for energy-expenditure estimation is presented in Figure 3-10. It consists of three contexts extracted from multiple sensor data: user's activity, heart rate, and breath rate. Suppose that the heart rate and breath rate are provided by a sensor and the activity context is extracted from accelerometer data by a ML approach as explained in Section 3.3. Detailed explanation of context extraction phase for energy expenditure estimation is provided in Chapter 5.

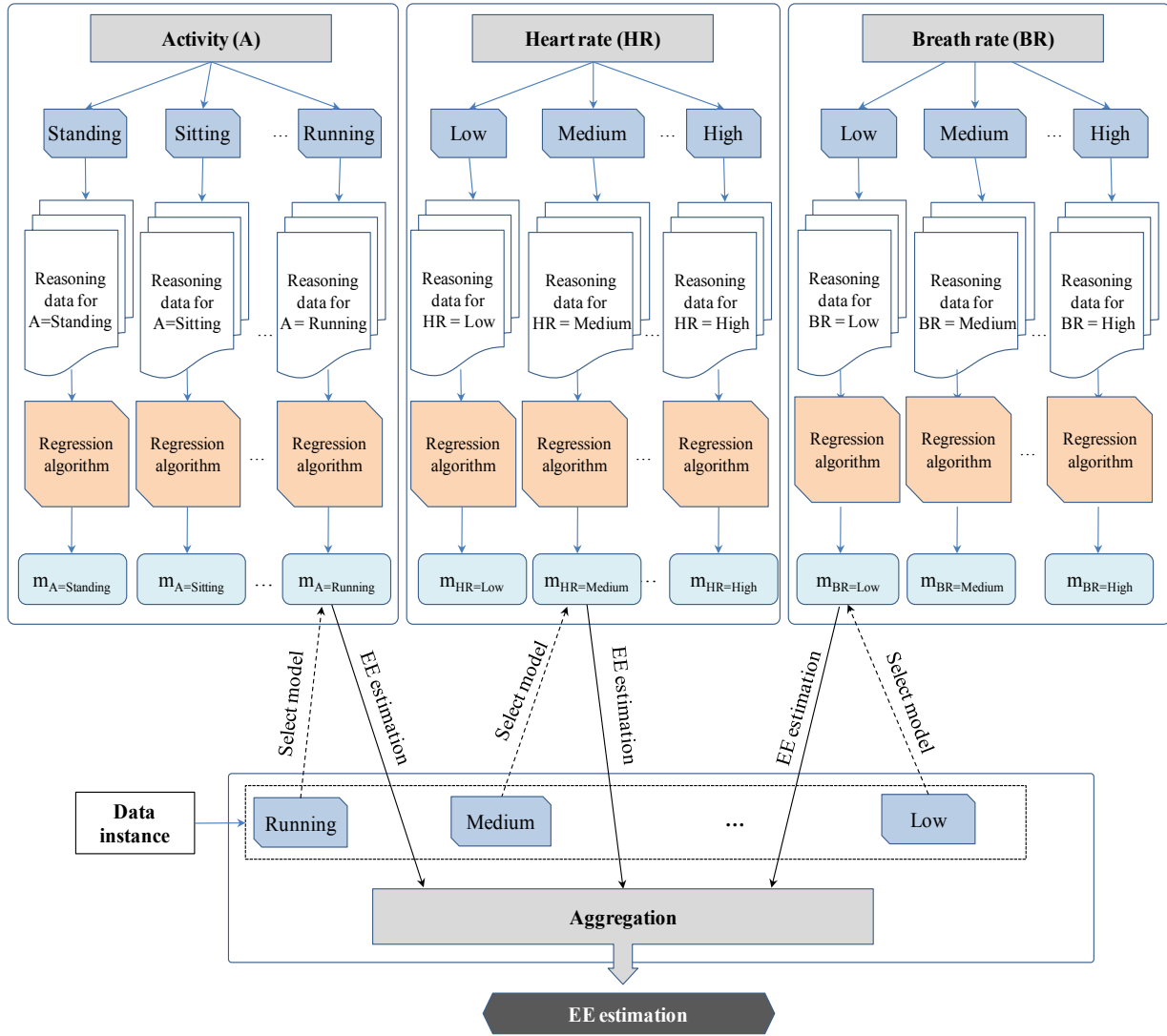


Figure 3-10. Context-based regression for energy expenditure.

In the modeling phase, for each discrete value of each context a regression model is trained using the data of the other contexts. The training dataset for the model is the subset of the whole training dataset that contains only the data instances which have the corresponding feature value; thus, the model for the first activity $m_{A=Running}$ is trained only on the data instances that contain the activity "Running". Once the training data is selected, the model can be trained using an arbitrary regression method, such as: Linear Regression [106], Support Vector Machine for Regression (SVR) [107] and Model Trees (M5P) [99].

In the testing phase, for a given test data instance, a custom ensemble of regression models is assembled. Figure 3-10 shows an example of a data instance containing "Running" as the activity, "Medium" for the heart rate, and "Low" for the breath rate. Thus, those three regression models ($m_{A=Running}$, $m_{HR=Medium}$ and $m_{BR=Low}$) are included in the ensemble. Once a data instance is evaluated by the appropriate models, the final decision about the estimated energy expenditure is made using a method for aggregating the outputs of the classification models: averaging, choosing the median etc.

3.4.3 Expert Rules

The expert-rules approach is preferred when the ML approaches are not suitable (e.g., not enough labeled data to train ML models) and the domain knowledge can be encoded with expert rules. An example is describing a fall situation with rules: "the user is lying on the floor and is not moving". We will use the fall-detection domain in order to explain the modeling with expert rules.

Fall detection is a process of detecting a fall situation using sensor data. The context-based reasoning for fall detection is presented in Figure 3-11. First, three contexts are extracted: user's activity, location of the user and the movement of the user's body. Using these contexts the expert constructs the rules that should detect a fall. Because the expert constructs the rules using domain-specific knowledge, he may or may not use the reasoning data as a help in the construction process (the dotted lines in Figure 3-11).

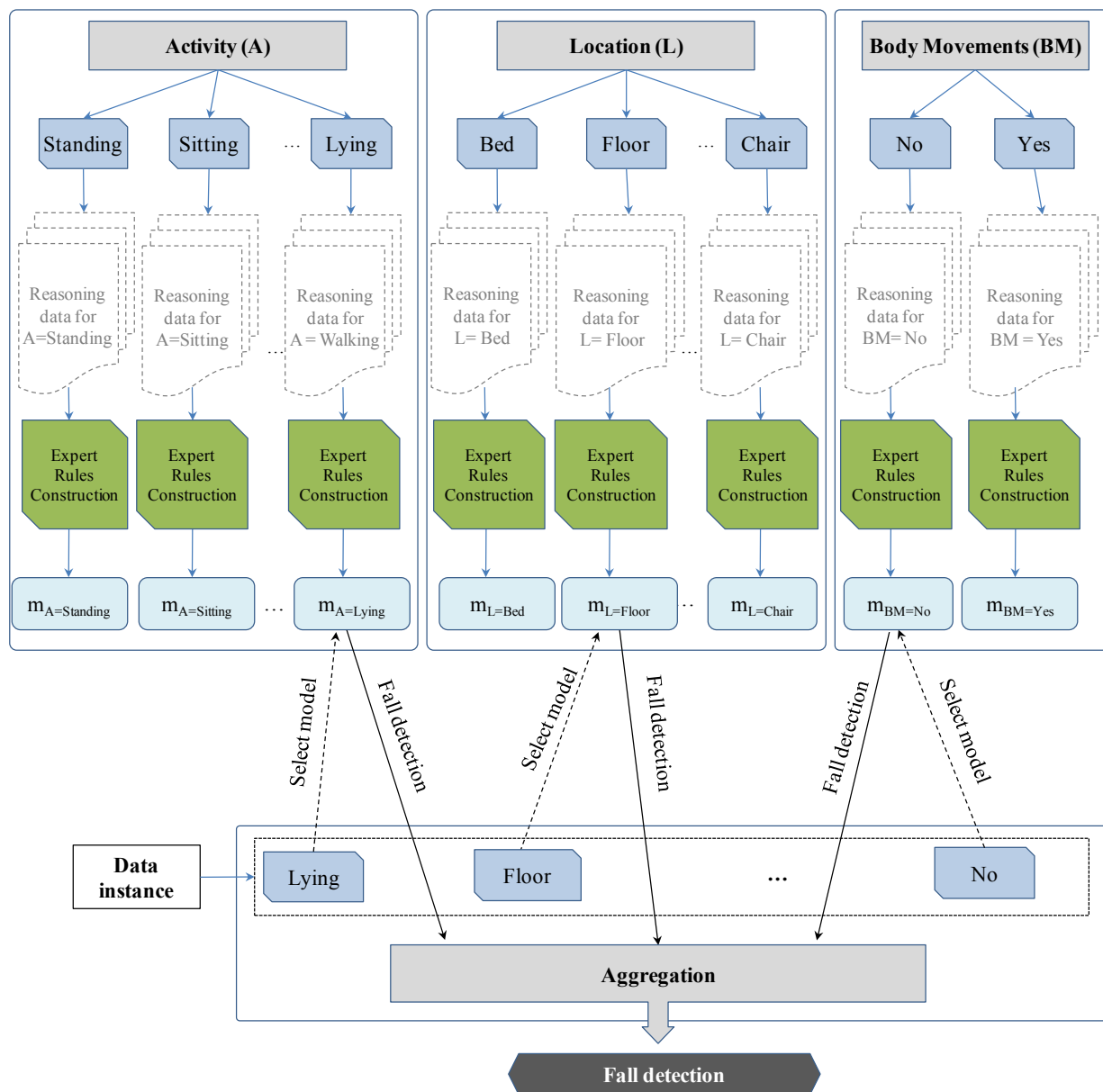


Figure 3-11. Context-based reasoning for fall detection.

To explain the basic principle of the context-based reasoning, let us consider the following example in which a user is lying on a bed, i.e., a non-fall situation. First, the activity of the user is considered as context, i.e., "Lying". That means the rules are created using the other two sources of information, i.e., body movement and location. The expert considers each combination of context values and defines a rule for detection of a fall situation. Example rules for non-fall and fall event, when the activity is used as context are the following:

$$\text{IF } (body\ movement = "no" \wedge location = "floor") \text{ THEN "fall"} \quad (5)$$

$$\text{IF } (body\ movement = "no" \wedge location = "bed") \text{ THEN "non-fall"} \quad (6)$$

The first rule states that in order to detect a fall the user should not move ("No") and the location should be the "Floor". However, in our case, the location is the "Bed", therefore the second rule will be used to make the decision, i.e., non-fall. The same situation is considered by using the location and the body movements as a context. The final decision is presented by aggregating the decisions given by each of the models. This was an example when the rules are simple and the context values are uniquely defined over the whole reasoning interval, i.e., single value for the interval. However, if this is not the case, one can also apply more advanced rule-based reasoning techniques such as fuzzy (soft) rules [110], temporal-reasoning approaches [111], e.g., event calculus [112]. For example, if multiple activities are recognized in the reasoning interval, the rule can be more complex and use thresholds for the portion of the recognized activity in order to reason about the user. In our case, the rule can state that: the user has to lie without movement for 80% of the reasoning interval in order to recognize a fall. Additionally, one can also include a temporal reasoning approach. That is, to use the time when a particular context value is recognized. For example, if acceleration fall pattern is recognized before "Lying" activity, then a fall should be detected. However, if the acceleration fall pattern is recognized after the "Lying" activity, no fall should be detected.

3.5 Context Aggregation

Each of the context models provides an output, i.e., a decision about the reasoning task. In the context aggregation phase, the outputs for each context are combined, i.e., aggregated into a final decision. In particular, for given test instance, $tInst$, only the models that contain the appropriate context values will be invoked, resulting in an ensemble of n context models (single model for each context). For example, the output of the context c for a given data instance $tInst$ is defined as follows:

$$d_c = m^c(tInst) \quad (7)$$

Definition 9. Context aggregation is a process of applying a function a that aggregates the outputs of each context model and provides the final decision about the reasoning task, i.e., $y = a(d_1, \dots, d_n)$.

Consider the energy-expenditure estimation shown in Figure 3-10: a user is running, has medium heart rate and low breath rate. The data instance will thus be evaluated by the three models that correspond to those values: $m_{A=Running}$, $m_{HR=Medium}$ and $m_{BR=Low}$. Then, their outputs should be aggregated to estimate the final energy expenditure. The pseudo code for context aggregation phase is given with Algorithm 3.

Algorithm 3. Context aggregation phase in CoReAmI.

The function `findCorrespondingModel ()` returns a model from the set of models, which corresponds to the particular context value – give as input.

#Phase C: Context Aggregation

Input: set of context C , set of context models M , test data instance $tInst$

Output: final decision y

```

#contexts
 $C = \{c_1, c_2, \dots, c_n\}$ 
#test data instance
 $tInst = (v_1^{c_1}, v_2^{c_2}, \dots, v_k^{c_n})$ 
#initialize empty array of decisions
 $dArray = \{\}$ 
#initialize the final decision
 $y = \text{"None"}$ 

-----

#evaluate the test data instance
FOR each context value  $v$  in  $tInst$ 
|    $m^c = \text{findCorrespondingModel}(v)$ 
|   #the output (decision) of the evaluation
|    $d = m^c(tInst)$ 
|   #add the decision to the decision set
|    $dArray.add(d)$ 
END
#aggregate the decisions into a final decision
 $y = a(dArray)$ 

RETURN  $y$ 

```

The aggregation process is important in approaches where multiple models are provided. The experience with ensemble approaches shows that it is better to find a good aggregation function instead of choosing the best single model and that this way a stronger generalization is achieved [86]. Dietterich [92] attributed the benefit from aggregation to the following three reasons:

- *Statistical issue.* It happens when the space for searching the models is too large to explore for limited training data. Additionally, there may be several models giving the same accuracy on the training data. If the learning algorithm chooses one of these models, there is a risk that a mistakenly chosen model could not predict the future data well. However, by aggregating the models, the risk of choosing a wrong model can be reduced.
- *Computational issue.* ML algorithms usually perform local search that may get stuck in local optima. Even if there are enough training data, it may still be difficult to find the best model. By running the local search from many different starting points, the aggregation

may provide a better approximation to the true unknown hypothesis. By aggregating (combining) the models, the risk of choosing a wrong local minimum can be reduced.

- *Representational issue.* In many ML tasks, the true (unknown) model could not be represented by any model in the model space. By combining the models, it may be possible to expand the space of representable functions, and thus the learning algorithm may be able to form a more accurate approximation to the true unknown model.

These are probably the most important factors for which the traditional learning approaches fail in some domains. Through aggregation, the variance as well as the bias of learning algorithms may be reduced; this has been confirmed by many empirical studies [113][114]. In recent years three types of techniques emerged as successful in the literature: averaging, voting, median, and aggregating with learning – stacking. Some of them are more suitable for aggregating numeric outputs of the models (regression) and some of them for discrete output (classification and expert rules).

Averaging

Averaging is the most popular and fundamental aggregation method for numeric outputs. In order to explain how averaging works a regression example is taken. Consider a set of n individual models $\{m_1, \dots, m_n\}$ and the output of m_i for a given data instance is d_i , the task is to combine all decisions to attain the final decision d . There are two types of averaging functions: simple and weighted averaging.

The simple averaging obtains the aggregated output by averaging the outputs of individual models directly. Specifically, simple averaging gives the combined output d as:

$$y = \frac{1}{n} \sum_{i=1}^n d_i \quad (8)$$

Weighted averaging obtains the aggregated output by averaging the outputs with different weights implying different importance for different models. Specifically, weighted averaging gives the aggregated output as:

$$y = \sum_{i=1}^n w_i d_i \quad (9)$$

where w_i is the weight for d_i and the weights w_i are usually assumed to be constrained by:

$$w_i \geq 0 \text{ and } \sum_{i=1}^n w_i = 1 \quad (10)$$

Voting

Voting is the most popular aggregation method for nominal outputs, e.g., classification models. Consider a set of n individual classification models $\{m_1, \dots, m_n\}$, the output of m_i for a given data instance is d_i , and the task is to aggregate the outputs from each classifier in order to predict the class label k_j from a set of l possible labels $\{k_1, \dots, k_l\}$. Therefore, d_i^j is the output of the m_i for the class label k_j . The d_i^j is a binary number, i.e., $d_i^j \in \{0, 1\}$, which takes value of 1 if m_i classifies k_j as the class label and zero otherwise.

There are several popular voting methods, such as: majority voting, plurality voting, and weighted voting. The majority voting is the most popular one. Each classifier votes for one class label, and the final output class label is the one that receives more than half of the votes; if none of the class labels receive more than half of the votes, a rejection option is given and the aggregated classifier makes no prediction.

$$y = \begin{cases} k_j, & \text{if } \sum_{i=1}^n d_i^j > \frac{1}{2} \sum_{k=1}^l \sum_{i=1}^n d_i^k \\ \text{rejection}, & \text{otherwise} \end{cases} \quad (11)$$

where d_i^j is the output of the classifier m_i for the class k_j , which can be 0 or 1.

In contrast to majority voting which requires the final decision to take at least half of the votes, plurality voting takes the class label which receives the largest number of votes.

$$y = k_{\arg \max_j \sum_{i=0}^n d_i^j} \quad (12)$$

If the classifiers are with unequal performance, a weighted voting scheme is preferred. The weighted voting scheme gives more weight to the vote of the classifiers with better accuracy.

$$y = k_{\arg \max_j \sum_{i=0}^n w_i * d_i^j} \quad (13)$$

where w_i is the weight assigned to the classifier m_i for the class k_j .

Median

Another popular aggregation technique is the one that chooses the median value. This technique is defined when numerical outputs are provided by the models. This technique first sorts the values according to their value, and then the value that is in the middle is chosen. In the case of even number of models, the average of the two in the middle can be calculated.

Stacking

Stacking is a general technique where a ML model is trained to combine the individual learners [115][116]. The individual models are called the first level model, while the aggregation model is called the second level model, or meta-learner. The basic idea is to train the first-level models using the original training dataset, and then to generate a new dataset for training the second-level model, where the outputs of the first-level models are regarded as input features while the original labels are still regarded as labels of the new training data. The first-level models are often generated by applying different learning algorithms, and so, stacked ensembles are often heterogeneous, though it is also possible to construct homogeneous stacked ensembles. Stacking is a general framework which can be viewed as a generalization of many ensemble learning methods. But it can also be viewed as a specific aggregation method which combines by learning.

3.6 CoReAmI Time Complexity Analysis

In this section we present the analysis of the time complexity of CoReAmI. Because CoReAmI is a general framework, it is relatively difficult to estimate the overall complexity. Additionally, the time complexity depends on the particular problem: sensors data specifications (e.g. sensor sampling rate), techniques chosen for context modeling and techniques chosen for context aggregation. Therefore, we perform the analysis for each of the phases individually and summarize it at the end of the section.

Context Extraction

The time complexity in the context extraction phase depends on the sensors and their raw data. That is, if multiple sensors are available, they may provide data with different sampling rates. For the purpose of simplicity, let us assume that there is a single sensor that provides data with sampling rate of sr Hz. Let us also assume that the sliding window with size T seconds is used for segmenting the data, i.e., an observation sequence $(X_{t,T})$ with size T is formed. Therefore, the number of data samples in each window is $p = T \cdot sr$. That means when extracting a context value for a particular window, p data samples are considered.

The further analysis depends on the chosen extraction function. This function may be of different types, however usually it represents calculation of a statistical value, such as: average, variation, maximum and minimum. Table 3-1 shows the time complexity of the four functions.

Table 3-1. Context extraction time complexity, where x_i is a sensor data sample value, and p is the number of data samples in one window.

Context extraction function	Mathematical definition of the context extraction function	Time complexity
Average	$\bar{x} = \frac{1}{p} \cdot \sum_{i=1}^p x_i$	$O(p)$
Variation	$\delta_x^2 = \frac{\sum_{i=1}^p (x_i - \bar{x})^2}{p}$	$O(p)$
Maximum	$Max = \text{maximum}\{x_i i = 1 \dots p\}$	$O(p)$
Minimum	$Min = \text{minimum}\{x_i i = 1 \dots p\}$	$O(p)$

In general, the context extraction function can be different for different contexts. Therefore, the upper bound of the time complexity of the whole context extraction phase can be generally defined as $n \cdot O(f(p))$, where n is the number of contexts and $O(f(p))$ is the time complexity of the most "computationally expensive" context extraction function. Because n is a constant, the time complexity of the context extraction phase can be simplified only to $O(f(p))$.

Context Modeling

The time complexity in the context modeling phase depends on the techniques chosen for modeling the contexts, e.g., ML or expert rules.

In the case of a ML-based context modeling, the complexity is defined as the time needed to learn the model (e.g., a Decision Tree). It usually depends on the number of training instances h , number of features, and similar [117]. Because the analysis of the time complexity of different ML algorithms is not in the scope of this thesis, let us assume that the chosen algorithm uses a modeling function $b(h)$, thus its complexity is defined as $O(b(h))$. Because the CoReAmI learns a model for each context value of each context the time complexity can be defined as $g \cdot O(b(h))$, where g is the number of all context values. Similar to context extraction phase, g is a constant, therefore the time complexity of the context modeling phase can be simplified only to $O(b(h))$.

Please note that if additional discretization of the context values is needed, the time complexity of the discretization procedure should be added. For example, if the Decision Tree's splitting criterion is applied for discretization, that requires sorting of the numerical values. The efficient sorting algorithms have time complexity of $O(h \log(h))$, where h is the number of the training data instances. Additionally, it requires calculating the information gain formula for each of the possible splits. However, usually the time complexity of the ML algorithm is much bigger than the discretization process, therefore the upper bound of the whole ML-based context modeling process can be defined as $O(b(h))$, where $b(h)$ is the modeling function of the ML algorithm (e.g., Decision Tree, RF, SVM).

In the case where the context modeling is performed by construction of expert rules, the time complexity depends on the process of constructing the expert rules. If an automatic rule-learning is used, it will depend on the rule-learning algorithm and the amount of data (similar to ML). If the experts do not use the sensors data and simply define the rules according to their knowledge (IF THEN rule), the complexity is a constant, i.e. the time spent while defining the rules. The executing time is linear with number of conjunctions in the rule. Please note that the time complexity of the computation of the context values (the variables in the rule) is already computed in the context extraction phase.

Context Aggregation

Similar to the previous two phases, the time complexity in the context aggregation phase depends on the techniques chosen for aggregating the decisions provided by the context models. Therefore, if the averaging aggregation function is chosen, the time complexity is $O(n)$, where n is the number of contexts (please note that each context provides only one decision). The same analysis stands for the other simple techniques such as: weighted voting, plurality voting and similar. In the case of the median technique, the upper bound of the complexity is defined by the sorting algorithm, which is generally defined as $O(n \log(n))$. In the case of more complex aggregation techniques such as stacking, the analysis is similar to the ML-based context modeling in the previous subsection, i.e., the upper bound is defined by complexity of the chosen ML algorithm. In general, the complexity of the aggregation function can be defined as $O(a(n))$, where n is the number of context.

One can also add the time complexity of the evaluation of the context models, i.e., the time needed for the context model to evaluate a data instance. The evaluation complexity can be generally defined as $O(e(m))$, where $e(m)$ is the function that evaluates the data instance and depends on the chosen ML algorithm in the second phase, e.g., if a Decision Tree is chosen, the evaluation depends on the depth of the tree; if kNN is chosen, the evaluation depends on the number of training instances, which should be used to find the k most similar ones.

Summary

To summarize, the approach is general and allows usage of different techniques in each of the three phases, therefore it is relatively difficult to define the overall time complexity. However, the following factors mainly influence the time complexity:

- $O(f(p))$ – Computational complexity of the most "computationally expensive" context extraction function $f(p)$, where p is the number of data samples in an window
- $O(b(h))$ – Computational complexity of the context modeling function $b(h)$ (e.g., the chosen ML algorithm), where h is the number of training instances.
- $O(e(m))$ – Computational complexity of the evaluation function $e(m)$, where m depends on the ML algorithm chosen that influences the evaluation of a data instance, e.g., depth of the tree in Decision Tree.
- $O(a(n))$ – Computational complexity of the context aggregation function, where n is the number of contexts.

One can also analyze the CoReAmI construction and execution time complexities separately. That is, the construction time is the time needed to construct the models (e.g., extracting features and training a classifier) and the execution time is the time needed to evaluate an instance (e.g., using the in real-world). The construction and execution time complexities are summarized with the following formulas:

$$\text{construction: } O(f(p)) + O(b(h)) \quad (14)$$

$$\text{execution: } O(f(p)) + O(e(m)) + O(a(n)) \quad (15)$$

As the equations show, the construction time complexity depends on the context extraction function and the context modeling function. In particular it depends on p (the number of data samples in the window) and h (the number of training instances). On the other hand, the execution time complexity depends on the complexity of the context extraction function, the evaluation function and the aggregation function.

In general, the construction time complexity is much higher than the execution complexity. The reason for this is that the evaluation functions and the aggregation functions usually have lower complexity compared to the modeling functions $b(h)$. This was also confirmed with our practical examples, when applying CoReAmI on the three problem domains. For the aggregation functions, in our experiments we used linear functions (e.g., average, weighted average), or functions with complexity $O(n \log(n))$ (choosing the median). Also, the evaluation function for most of the ML algorithms is also quick and is calculated in milliseconds. Therefore, in general case the modeling function (e.g., ML algorithm) chosen in the second phase of CoReAmI will influence the most on the overall time complexity of CoReAmI. Of course, one should not forget that also the context extraction functions can be as complex as learning a ML model; this was the case when we used an AR classification model to calculate the activity of the user, which was later used as a context. In similar manner, the aggregation function can also be as complex as learning a ML model, e.g., Stacking [115].

3.7 Summary and Discussion

We proposed a novel approach to context-based reasoning in AmI called CoReAmI. A summarized version of the execution of the CoReAmI algorithm is given in Algorithm 4.

Algorithm 4. Execution of CoReAmI – evaluating an observation sequence. The function `findCorrespondingModel ()` returns a model from the set of models, which corresponds to the particular context value – given as input.

Input: observation sequence $X_{t,T}$, set of contexts C , set of models M

Output: final decision y

```

#observation sequence
 $X_{t,T} = \{x_t, x_{t+1} \dots, x_{t+T}\}$ 
#contexts
 $C = \{c_1, c_2, \dots, c_n\}$ 
#empty data instance
 $dIns = ()$ 
#initialize empty set of decisions
 $dArray = \{\}$ 
#initialize the final decision
 $y = \text{"None"}$ 
-----
FOR each  $c_i$  in  $C$  DO
|   #extract the context value for each context
|    $v^{ci} = f^{ci}(X_{t,T})$ 
|   #add the context value to the data instance
|    $dIns.add(v^{ci})$ 
END
#evaluate the data instance
FOR each context value  $v$  in  $dIns$ 
|    $m_v = \text{findCorrespondingModel}(v)$ 
|   #the output (decision) of the evaluation
|    $d = m_v(tInst)$ 
|   #add the decision to the decision set
|    $dArray.add(d)$ 
END
#aggregate the decisions into a final decision
 $y = a(dArray)$ 

RETURN  $y$ 

```

The CoReAmI approach consists of three phases: context extraction, context modeling and context aggregation. The approach is general and can be applied to a different problem in the AmI domain, however appropriate adaptation should be performed on the techniques used in each phase.

The context extraction phase is similar to the feature extraction problem described in ML. Similarly to the feature extraction process the more different the contexts are, the better. In other words, each context should describe and include unique information about the problem. This characteristic of the feature diversity is also used in the popular co-training ML approach [100], where multiple views of the data are used and the more different the data between different views is, the better is the performance.

In the next phase, the context models are constructed using the contexts defined and extracted in the previous phase. The models are constructed in such a way that for each context value a model is constructed using the information (data) from the other contexts. When a data instance is evaluated, it is done with multiple models for each context in which the user is at the moment of reasoning. When the modeling is performed by classification or regression learning algorithms, the result is an ensemble of classification or regression models, respectively. In recent years, ensemble-based approaches have shown to be successful and are state-of-the-art in numerous fields in the ML and pattern recognition domains. The main idea behind ensemble learning is to train multiple base learners to solve the same problem. In our case, each base learner is a model selected from a set of models, one for each possible context value (see Figure 3-7). Our approach not only exploits the complementarity of multiple models like most other ensemble approaches, but also contains models that tend to be more accurate for a particular context than those trained on the whole training set. The reason for that is that each model is trained on a subset of the training set that is more homogeneous than the whole set, and used in the context of this subset, i.e., to reason about samples similar to the ones in the subset. In other words, CoReAmI semantically splits the domain (dataset) into meaningful viewpoints and not on some statistics about the data (as most of the conventional ensemble-based algorithms).

In the final phase the outputs of the context models are aggregated and the final decision is provided. The aggregation process brings additional value in such approaches, where multiple models are provided. The experience with ensemble approaches shows that it is better to find a good aggregation function instead of choosing the best single model. This way a stronger generalization is achieved. Through aggregation, the variance as well as the bias of learning algorithms may be reduced; this has been confirmed by many empirical studies [113][114].

Because of its generality, the approach can be applied to different tasks in the AmI domain, where multiple-source context information is available. Probably the biggest limitation is that considerable human effort is needed to present the context information appropriately (context extraction phase). However, if the contexts are already defined (which is often the case in ML tasks, for example by features extracted from the sensors data), one can continue with CoReAmI by adapting the context modeling and aggregation phases and significantly reducing the adaptation time.

We evaluated the approach on three problem domains: activity recognition, energy-expenditure estimation and fall detection; which are explained in Chapters 4, 5, 6.

4 Activity Recognition Domain

Activity Recognition (AR) is the first problem domain on which we apply the CoReAmI approach. AR can generally be defined as a process of recognizing activities [118][119]. The goal of AR is to determine the actions or the states of one or more people through the analysis of sensor data from ambient sensors or wearable sensors [120].

The most recent literature in AR field shows that wearable accelerometers are among the most suitable sensors for unobtrusive single-person AR. For the sake of the user's convenience, AR applications are often limited to a single accelerometer. Numerous studies have shown that the performance of an Activity Recognition System (ARS) strongly depends on the accelerometer placement (e.g., chest, abdomen, waist, thigh, ankle) and that some placements are more suitable (in terms of AR performance) for particular activities [121][20][122]. Therefore, the chosen placement usually depends on the application built on top of the ARS, which may be elderly care, sports tracking etc.

One of the best accelerometer placements for AR [20] – and even more for the applications built on top of AR, such as fall detection [29] and human energy-expenditure estimation [123] – is on the torso (chest or abdomen). However, two very common activities – sitting and standing (S/S) – are difficult to recognize with such a placement and are usually mutually misrecognized and almost impossible to distinguish using standard AR approaches. The reason is that they do not differ in the torso orientation, so distinguishing them with a single accelerometer placed there is challenging. Additional motivation to distinguish these activities is that together with the lying, they constitute around 90% of all the activities during a normal day of an average person [124][125]. Furthermore, the American Medical Association (AMA) agrees that sitting for extended periods of time can be bad for personal health: sitting disease (sedentary lifestyle), cardiovascular disease, etc. Conversely, standing is the complete opposite and is a healthy activity: it increases energy expenditure, tones muscles, improves posture, increases blood flow and ramps up metabolism [124]. Therefore, having an ARS that recognizes these two activities is essential and may improve a person's health, e.g., by suggesting standing after a long period of sitting.

In this chapter we present an approach to AR that recognizes the activities of the user using single accelerometer placed on the torso. In particular, the approach is a combination of a ML model (baseline approach) and the CoReAmI approach, which is used to distinguish the standing and sitting activities.

The chapter is organized as follows. First, we present the related work in AR. Then, we present the AR approach, which consists of: (i) a baseline AR approach, which uses standard ML to recognize the activity of the user; and (ii) the CoReAmI approach applied to distinguish sitting from the standing activity. In the next two sections the experimental setup and the results are presented. In the final section, a discussion and directions for future work are provided.

4.1 Related Work

AR approaches can be divided into those using ambient (non-wearable) sensors and those using the wearable type. Ambient sensors are particularly useful for monitoring environments in which multiple people are present (cameras, humidity sensors, temperature sensors, RFID readers, and similar). On the other hand wearable (body-worn) sensors (accelerometers, gyroscopes magnetometers, location sensors, pedometers, ECG, EEG, and similar) can be used to monitor one person and recognize his or her activities.

The most exploited non-wearable approach is based on cameras [126]. Although this approach is physically less obtrusive for the user compared to the wearable sensors, it suffers from problems such as low image resolution, target occlusion and time-consuming processing. However, often the biggest issue is user privacy: the user has to accept the fact that a camera will record him/her.

Currently the most exploited and probably the most mature approach to AR is using wearable accelerometers, which are both inexpensive and effective [127][128][129]. There are two common types of wearable-sensor approaches to AR that have shown to be successful: using domain knowledge encoded with rules, and using ML.

A common approach to accelerometer AR is based on manually created expert rules (domain knowledge). These rules are usually based on features that are calculated from sensor orientations and accelerations. Wu et al. [128] presented an approach in which decision rules are used to recognize activities. Even though the rules are only one of several components in their approach, they showed that rules can successfully contribute to the AR. Another implementation of such rules is presented by Lai et al. [129]. The authors used six accelerometers placed on the neck, waist, left wrist, right wrist, left thigh, and right thigh. The reported accuracy was almost perfect, i.e., 99.5%, but the number of sensors is excessive for everyday use. In our approach we try to achieve satisfactory performance using a single accelerometer, which makes the system less obtrusive and more acceptable for everyday usage.

The most traditional AR approach is based on ML. This approach usually implements widely used classification methods, such as Decision Tree, SVM, kNN and Naive Bayes. Depending on the number of models trained, these approaches can be single-model or ensemble-based.

Single-model examples include Kwapisz et al. [127], who used an accelerometer placed on the thigh and tested their single-model approach by comparing the results of three classification methods on dynamic activities such as walking, running and jogging. However, static activities (body postures) are also of great importance in some situations, e.g., these activities are the main indicators of any degradation in elderly health (e.g., an increase in the time spent lying down). Ravi et al. [130] used an accelerometer on a mobile phone and tested their approach with five classification methods. The results showed that when the same person's data was used for training and testing, the accuracy was 90%, but when a different person's data was used for the testing, the accuracy dropped to 65%. In our approach the evaluation is performed on the data from different people, as the developed model is intended for use by people who were not involved in the training of the model.

Ensemble-based approaches have been a popular research topic in the recent years. Banos et al. [131] used an ensemble of classification models and multiple accelerometer sensors to recognize the activity of the user. They showed that an ensemble-based approach outperforms the traditional single-model approaches. In their approach they used multiple accelerometers placed on different body locations and each classification model was constructed for an

individual accelerometer. In our approach, we also use ensemble of classification models. In particular, we use Random Forest (RF) classification model as a baseline approach to recognize activities others than sitting and standing. Additionally, in the CoReAmI approach, context-based ensemble of classification models is constructed in order to distinguish between standing and sitting activities.

Human activities have certain natural regularities and temporal dependence (smoothness), e.g., people do not abruptly switch back and forth between lying and cycling, approaches that use the previous activities can help in recognizing the current activity. A common way to address this problem, and consequently reduce spurious activity transitions, is by using hidden Markov models (HMMs) [132]. Lester et al. [133] showed that incorporating HMMs improves the recognition of activities. We also performed experiments by applying HMMs in order to smooth the spurious transitions between activities, and the results showed slight improvement of 1-3 percentage points depending on the sensor placement [22]. In this thesis we incorporated this concept by analyzing the before activity segment (see Figure 4-5) and including it as a context in the CoReAmI approach.

Context-based approaches have been also used also for AR; in particular for group activity recognition (recognizing activities for a group of people). Lan et al. [80] presented a context-based ML approach to group activity recognition by using cameras. They improved their basic activity recognition by including contexts, in their case called action context descriptors. These descriptors included information not only for the activity of the particular person, but also the context, which was represented by the behavior of the other people nearby. In our case, we focus on single-user AR using wearable sensors, i.e., single accelerometer. The whole context information is provided by the single accelerometer.

To summarize, the research on AR is fairly extensive, but the issue of distinguishing standing from sitting with a single accelerometer on the torso is largely sidestepped. Because of the mutual misclassification some researchers simply do not include both standing and sitting among the activities to recognize: Lara et al. [134], for example, omitted standing, while Lee et al. [135] and Khan et al. [136] omitted sitting. Others solve the problem by merging these two classes [137][138][139]. And finally, some do report it [140][141][142], however the results show that the accuracy achieved for these two activities is significantly lower than the accuracy achieved for the other activities, such as walking, lying and running. We also observed this problem in our previous work [22], where we showed the lower accuracy is due to these two activities being mutually misclassified. There are also some examples of successful recognition of these two activities. Khan et al. [143] managed to achieve above 95% accuracy for the both activities using a hierarchical AR scheme that uses features and sensor tilt angles extracted for the current activity data segment. However, their approach uses a strong assumption that the accelerometer data and the accelerometer tilt angles are different during the sitting and standing activities. Although this may be true for that particular dataset, it is not reliable for real-life situations. On the contrary, sitting with upright torso – as recommended by experts – has very similar torso orientation (therefore sensor tilt angle) as standing. This was also justified in their additional tests; when they tested the same algorithm on another, more realistic AR scenario, the accuracy significantly dropped for 20 percentage points for each of the two activities. Given the acceleration characteristics of both activities (static with similar tilt angles), we are quite confident that additional context information should be analyzed in order to distinguish these two activities.

4.2 Activity Recognition with CoReAmI

Our approach to AR with CoReAmI is based on previous research experience, which is shortly presented here.

As shown in the previous section, the AR approaches that use accelerometer data are the most common in the literature and also achieve substantial accuracy. Therefore, we also used accelerometers for the research performed for the two European projects: Confidence [8] and CHIRON [9]. Even though AR was not the main goal in the projects, it eventually emerged as one of the most important components, being the foundation for further reasoning in the main tasks – the detection of falls, the detection of unusual behavior, the estimation of human energy expenditure and others. Using the experience gained in these projects, we developed the RaReFall (Real-time Activity-Recognition and Fall-Detection) system [11], which uses accelerometer data in order to recognize the activities of a user (an overview of the system is shown in Figure 4-1). The system consists of two accelerometers placed on the abdomen and the right thigh. The AR is performed on a laptop using the raw sensor data acquired through Bluetooth.

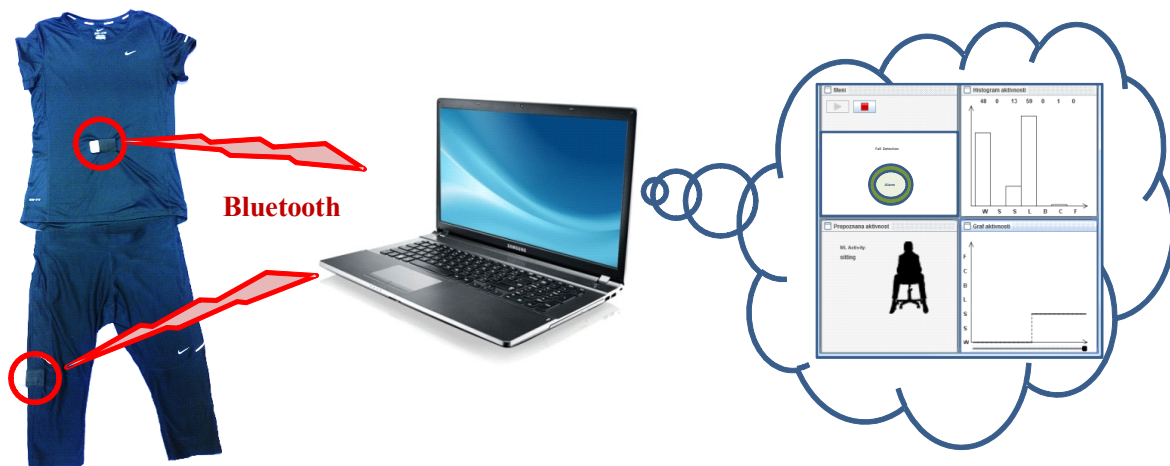


Figure 4-1. The RaReFall system.

The RaReFall system was evaluated as the best-performing at the international competition in AR – EvAAL-AR 2013 [12]. EvAAL is a competition with an objective to evaluate AR systems intended to be used by the elderly in real life. Therefore, the performance of each competing ARS is evaluated live in a living lab. The evaluation is performed on an activity scenario that included activities of daily living (watching TV, working in the kitchen, bathroom activities, sleeping). The competition requires each competing team to bring their own ARS in order to recognize the activities of an actor performing the activity scenario. The evaluation is thorough and is performed using several criteria: recognition performance, user-acceptance, recognition delay, system installation complexity and interoperability with other systems.

Table 4-1 shows the scores on the scale of 0–10 for the 2012 and 2013 competition editions. The RaReFall system obtained the highest final score for both years, by achieving not only high accuracy, but also scoring well on the other criteria [13]. More details about the other competitors can be found in [13].

Table 4-1. EvAAL-AR '12 and '13 teams and results (score: from 0 to 10)

	Team	Accuracy	Delay	Installation complexity	User Acceptance	Interoperability	Overall score
EvAAL-AR '13	RaReFall (Slovenia)	6.94	10	10	8.55	7.2	8.36
	CNR (Italy)	4.04	10	10	7.04	6.15	6.94
	Seville'13 (Spain)	4.68	9	10	6.99	5.54	6.89
	Chiba'13 (Japan)	4.43	10	0	5.44	2.24	4.86
EvAAL-AR '12	Seville'12 (Spain)	4.33	9	10	7.47	7.63	7.07
	CMU&Utah (USA)	7.17	9	0	7.93	6.15	6.51
	Chiba'12 (Japan)	1.44	5	0	5.6	5.09	3.13
	Dublin (Ireland)	0	0	10	5.2	1.25	2.67

The scoring of the ARS was overseen by an evaluation committee (EC) according to the following five criteria:

- Recognition accuracy – how accurately the system recognizes the target activities.
- Recognition delay – the elapsed time between the time at which the user begins an activity and the time at which the system recognizes it. The maximum allowed delay was 20 seconds, after that the system was evaluated with 0 points. In order to get the maximum score, the competing system was required to have a delay no more than 2 seconds.
- Installation complexity – how much effort is required to install the ARS in the living lab. It was measured in minutes of work per person needed to complete the installation. The maximum allowed installation time was 60 minutes, after that the system was evaluated with 0 points. In order to get the maximum score, the competing system was required to have no more than 10 minutes of installation time. This and the following two parameters were evaluated by the EC.
- User acceptance – how invasive the ARS is in the user's daily life. Evaluated by the EC using a questionnaire available at: <http://evaal.aaloa.org/2013/quest>.
- Interoperability with AAL systems – the metrics used are: the use of open-source solutions, availability of libraries for development, integration with standard protocols. Evaluated by the EC using a questionnaire available at: <http://evaal.aaloa.org/2013/quest>.

Despite the satisfactory performance achieved by our RaReFall system, the thorough evaluation setup at the competition revealed several aspects that would benefit from improvement. In order to make the system more user-acceptable and to minimize the obtrusiveness of the system, we decided to reduce the number of sensors to one – placed on the torso. This change caused a mutual misrecognition of the sitting and standing activities (which can also be noticed in the confusion matrix shown in Table 4-3). Because of this, we applied the CoReAmI approach only for these two activities, which resulted in an ARS as shown in Figure 4-2. The system consists of a torso-placed accelerometer, a baseline AR approach (similar to the one used in the RaReFall system) which recognizes the activities of the user, and finally enhancing this approach with the CoReAmI approach in order to distinguish the

sitting and the standing activities. First, the accelerometer data is used to recognize the activities with the baseline AR approach. If the recognized activity is sitting or standing, it is further processed by the CoReAmI approach. Otherwise, it is used as the final recognized activity. The CoReAmI approach for AR extracts multiple contexts using the accelerometer data (e.g., current activity, last transition, etc.), and trains multiple classification models in order to make the final decision about the recognized activity.

In the following subsections the sensor and its data are briefly introduced, then the baseline approach is presented, and finally the application of the CoReAmI approach is presented.

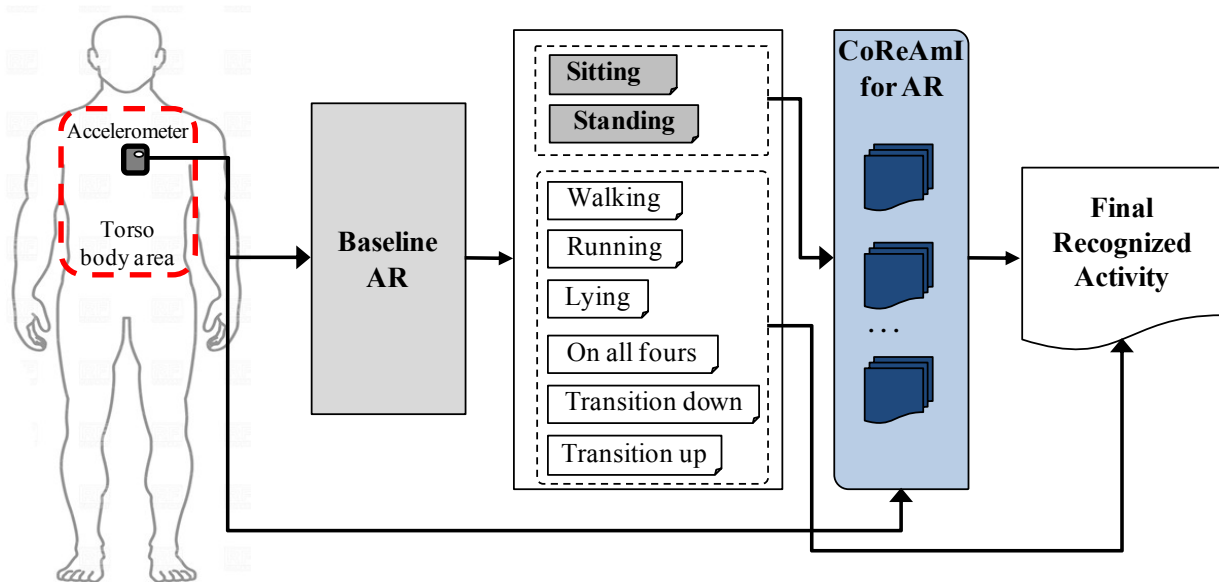


Figure 4-2. Overview of the context-based activity-recognition process. The baseline approach combined with the CoReAmI approach for the sitting and the standing activity.

4.2.1 Sensors

The sensor equipment consists of a Shimmer sensor platform [144], shown in Figure 4-3. The platform, beside the accelerometer, has an on-board MSP430 microcontroller, wireless communication via Bluetooth or 802.15.4 low power radio, and the option of local storage to a 2GB micro SD card. It additionally provides digital (I2C, SPI) communication for new potential sensors. The 450mAh Li-ion battery lasts 8-10 hours when the sensor sends the data in real time through Bluetooth. To record the users' acceleration, 50 Hz of data sampling frequency was used and acceleration sensitivity level of 15 m/s^2 . This sensitivity level and sampling frequency were chosen using empirical analysis of the human movements in our previous tests. It was shown that it is a reasonable tradeoff between the quality of the data on one side and the sensors battery consumption on the other side.

The sensor placement was chosen to be the torso. It was chosen as a trade-off between the obtrusiveness with respect to the user and the achieved AR performance on preliminary tests [18]. The accelerometers can be attached to the user's torso in several ways, making the system more user-acceptable and also adjustable to the occasion (e.g., worn indoors or outdoors) – elastic strap, pockets sewn into garment, etc. In our experiments we used a custom-made elastic strap with Velcro at the end, which meant it could be adapted to different types of users.

A 3-axis accelerometer measures accelerations in three directions (axes). Because of the Earth's gravity, all objects experience a gravitational pull towards the Earth's centre. Therefore, the sensor's acceleration vector consists of the acceleration due to the gravity and acceleration due to the movement of the sensor. The information about the gravity enables computing the sensors orientation (e.g. vertical, horizontal), which is of particular importance for the AR task, especially for static activities or postures, such as sitting, lying and standing. On the other hand, the acceleration measured due to the user's movement enables recognition of dynamic activities such as walking, running, cycling, and similar.

A custom PC application was created to record, preprocess the sensor data. We used a single accelerometer, therefore no synchronization was required. During the recordings, the accelerometer data was acquired on a laptop in real-time using Bluetooth. Additionally, the data were manually labeled with the corresponding activity, which was later used for the training of the activity-recognition classification model. Finally, the data was saved into the database for further processing.



Figure 4-3. Shimmer sensor platform [144].

4.2.2 Baseline Activity Recognition

Using our experience in AR, we developed a ML approach which was used as a baseline to initially recognize the activities of the user. It is an approach that trains a classification model in order to classify the current activity of the user, which can be: standing, sitting, walking, running, lying, on all fours, transition up, or transition down. It is a similar approach to the one explained in Subsection 3.3, excluding the phase of data synchronization (no need of synchronization because a single accelerometer is used).

The process of the baseline AR is shown in Figure 4-4. It is divided into four phases: first, the continuous sensor data stream was segmented, next the data is filtered, then for each segment features were computed, and finally the trained classification model recognized the user's activity.

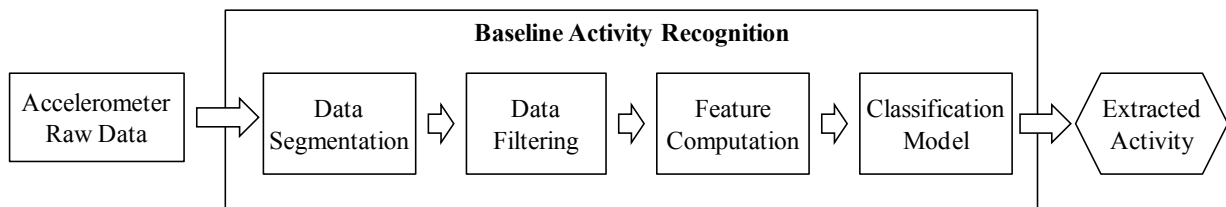


Figure 4-4. Baseline activity recognition data flow.

The first phase is the *data segmentation*, which uses an overlapping sliding-window technique, dividing the continuous sensor-stream data into data segments – windows. A window of a fixed size (width) moved across the stream of data, advancing by half its length in each step. Preliminary tests showed that a two-second window size and one-second overlapping was a reasonable trade-off between the duration of the activities and the recognition delay.

Once the sensor measurements are segmented, further pre-processing is performed using two simple *filters*: low-pass and band-pass. The band-pass filter is a combination of the low and high-pass filters and has two goals: (1) to eliminate the low-frequency acceleration (gravity) that captures information about the orientation of the sensor with respect to the ground and (2) to eliminate the high-frequency signal components generated by non-human motion and high-frequency noise, thus preserving the medium-frequency signal components generated by dynamic human motion. The band-pass filtered data is used for the computation of features relevant for dynamic activities, such as walking, running and cycling. The low-pass filter has the opposite purpose: to eliminate most of the signal generated by dynamic human motion and preserve the low-frequency component, i.e., gravity. In this way the low-pass filtered data contains the sensor-orientation information, which is relevant for the recognition of the static activities (postures), such as lying, sitting, standing and kneeling.

Algorithm 5 shows the low-pass filter. It uses a low-value filtering factor to generate a value that uses 20% of the unfiltered acceleration data and 80% of the previously filtered value. This factor was chosen empirically. High values for this parameter were tested (i.e. 0.8, 0.9), because we were more interested in the low-passed values (gravity component). As shown with the sample code below, the previous values are stored in the *pDS* 3D vector and the current values are stored in the *cDS* 3D vector. Because the acceleration data comes in regularly, these values settle out quickly and respond slowly to sudden but short-lived changes in the motion.

Algorithm 5. Low-pass filter.

Input: current data sample *cDS*, previous data sample *pDS*

Output: low-pass filtered vector *lowP*

```

#initialize the low-pass filtered vector
lowP = ()
#define the  $\alpha$  smoothing factor
 $\alpha = 0.8$ 

BEGIN
    lowP.X =  $\alpha \cdot pDS.X + (1 - \alpha) \cdot cDS.X$ 
    lowP.Y =  $\alpha \cdot pDS.Y + (1 - \alpha) \cdot cDS.Y$ 
    lowP.Z =  $\alpha \cdot pDS.Z + (1 - \alpha) \cdot cDS.Z$ 
END

RETURN    lowP

```

Algorithm 6 shows the version of a high-pass filter that was used in our research. It uses the previously calculated low-pass value (gravity component) and simply extracts the gravity component out of the current value. The result is saved in the *highP* 3D vector.

Algorithm 6. High-pass filter.

Input: current data sample *cDS*, low-pass filtered data sample *lowP*

Output: high-pass filtered vector *highP*

BEGIN

$highP.X = cDS.X - lowP.X$

$highP.Y = cDS.Y - lowP.Y$

$highP.Z = cDS.Z - lowP.Z$

END

RETURN *highP*

In the next phase, all the relevant *features* are computed. The choice of which features to compute was done after studying the literature in the AR field, especially the approaches that use accelerometer data. Afterwards we performed empirical analysis of the data and preliminary tests, and finally we computed 28 features (e.g., mean acceleration values along each of the acceleration axes, standard deviation for each of the axes, and similar). A complete list of the features is provided in the Appendix A, it contains features computed for each of the axes of the acceleration vector (x, y and z) and also for the length of the acceleration vector. The features can be roughly divided into two groups: ones that represent the dynamic activities and are computed using the band-pass filtered data, such as standard deviation of the acceleration values; and ones that represent the static activities (lying, on all fours, sitting, standing, etc.) and are computed using the low-pass filtered data, such as: accelerometer's tilt (inclination) angles.

Once the feature vector is formed, it is fed into the *classification model*, which recognizes the activity of the user. The classification is performed using the API of the software toolkit WEKA [109]. Among the several methods tested (Decision Tree, Naive Bayes, kNN, SVM and RF), RF yielded the best results in preliminary tests [20][21]. RF is an ensemble of Decision Trees in which the final decision is formed by a majority vote of the tree models [96].

4.2.3 Context-based Activity Recognition

The preliminary results showed that the baseline approach achieves relatively high accuracy (above 80%); however, two very common activities were mutually misrecognized: sitting and standing. The recognition accuracy for these two activities was not sufficient for such basic and common activities and moreover they were mutually misrecognized. An example confusion matrix is shown in Table 4-3, which shows that from all sitting activities only 43.6% are recognized as sitting and most of them were misrecognized as standing. Because of this, we further analyzed only these two activities with the CoReAmI approach.

In order to recognize the current activity, the CoReAmI approach uses not only the data segment for that particular (current) activity, but also the context of the current activity, i.e., the

data segments before and after the current activity. An example that shows 3-axial acceleration (Acc) data for a sitting activity is given in Figure 4-5. The sitting is preceded by walking and transition down, and followed by a transition up and standing. The data segments for the sitting and the standing are similar, so a method that analyzes only these segments would have problems distinguishing them. However, if the method also analyzes the transitions and activities before and after the sitting, recognizing the sitting is easier. In the example in Figure 4-5, such a method would recognize the S/S activity as the current one, and since the previous activity was walking followed by a transition that can only be down (it is impossible to transition up from walking), the current activity should be sitting. For each of these segments, contexts are extracted and then the CoReAmI is applied to make the final decision about the recognized activity.

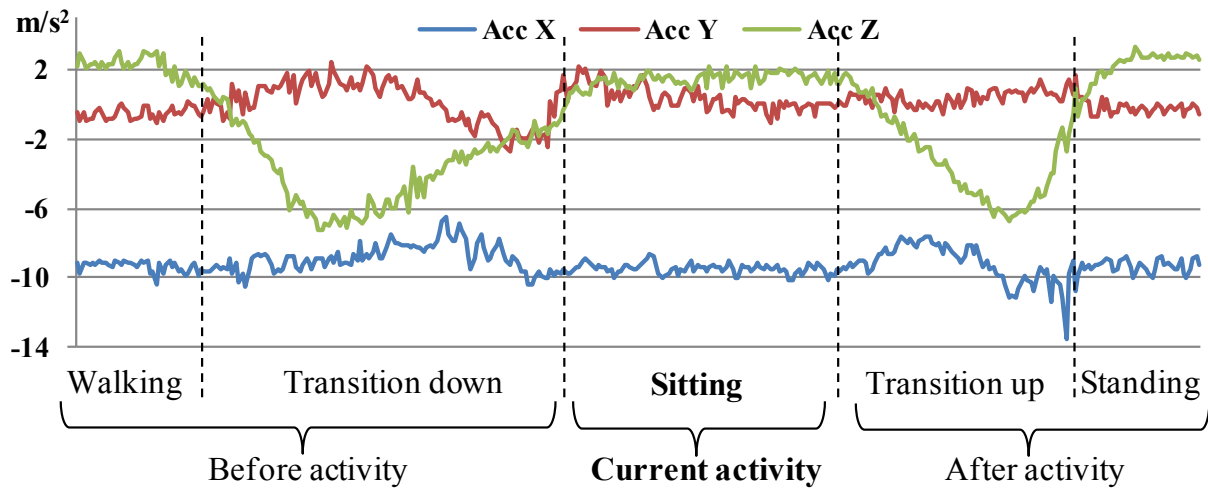


Figure 4-5. 3D Acc data for a sitting activity, including the activity segments before and after.

The whole procedure of AR is as follows. First, the accelerometer data is used to recognize the activities with the baseline AR approach. If the recognized activity is S/S it triggers the CoReAmI. Then, CoReAmI collects the necessary data in order to reason about the current activity, i.e., it collects data for: before, current and after activity segments. First the CoReAmI requests the acceleration data before the S/S activity (before activity data segment). Next, while the baseline recognizes S/S activity, the data is fed to the CoReAmI as current activity segment. Once an activity that is different than S/S is recognized, the current activity segment ends. Then, the CoReAmI also checks the data after the S/S segment (after activity segment). Once the three segments are completed the CoReAmI reasoning is applied and the final activities are recognized for the current S/S segment. Also a mechanism that deals with noise in the current activity segment is implemented. If an S/S segment is interrupted by a single activity other than S/S it is considered as noise and is converted into an S/S activity.

After empirical analysis of the data by domain experts, 12 contexts were extracted (shown in Table 4-2).

Context no.1 represents the previous activity of the user. The intuition for this context is that if the method is aware of the previous activity, it can better reason about the current activity. Next context is about the last transition of the user. The importance of this context is shown with the example given in Figure 4-5. Because the accuracy of the baseline approach for the transitions is low (around 40%), we included an additional ML model (context no. 3) trained to recognize only transitions (up or down).

Table 4-2. The contexts extracted for the context-based AR.

No.	Context	Description
1	Previous activity	The previous activity as classified by the baseline (excluding transitions).
2	Last transition	The last transition (if any) as classified by the baseline.
3	Last transition 2	The last transition (if any) as classified by another ML model (baseline-trans) trained only to distinguish transition up from transition down.
4	Current activity	The current activity as classified by the baseline.
5	Tilt angle x-axis	The tilt (inclination) angle of the sensor along the x-axis (forward, backward) during a sit or stand activity.
6	Standard deviation of the tilt angle along the x-axis	The standard deviation of the tilt angle along the x-axis during a sit or stand activity. This context represents the variations of the tilt angle along x-axis while the user is sitting or standing.
7	Standard deviation of the tilt angle along the y-axis	The standard deviation of the tilt angle along the y-axis during a sit or stand activity. This context represents the variations of the tilt angle along y-axis while the user is sitting or standing.
8	Standard deviation of the tilt angle along the z-axis	The standard deviation of the tilt angle along the z-axis during a sit or stand activity. This context represents the variations of the tilt angle along z-axis while the user is sitting or standing.
9	Standard deviation of the x-axis	The standard deviation of the x-axis acceleration during a sit or stand activity. This context represents the movements of the torso (up, down) while the user is sitting or standing.
10	Standard deviation of the y-axis	The standard deviation of the y-axis acceleration during a sit or stand activity. This context represents the movements of the torso (left, right) while the user is sitting or standing.
11	Standard deviation of the z-axis	The standard deviation of the z-axis acceleration during a sit or stand activity. This context represents the movements of the torso (forward, backward) while the user is sitting or standing.
12	Next transition and activity	The next transition and activity as recognized by the baseline-trans and baseline, respectively.

Context no.5 represents the tilt of the sensor and therefore the tilt (inclination) of the user's torso (forward, backward). The tilt angle is calculated as the angle between the actual acceleration (e.g. the Earth's gravity for static activities) and the x-axis (the first example in Figure 4-6). For instance, the angle φ_x between the acceleration vector and the x axis is computed as follows (where the values a_x , a_y and a_z represent the actual acceleration vector):

$$\varphi_x = \arccos\left(\frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right) \quad (16)$$

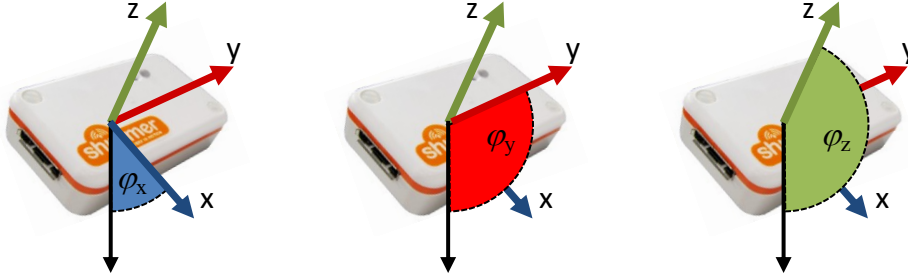


Figure 4-6. Graphical representation of the sensor tile (inclination) angles.

The next three contexts (no. 6, 7 and 8) are computed using the standard deviation of the tilt angles along each of the axes during the sitting or standing activities. They represent the variations of the tilt angle along each of the axes while the user is sitting or standing. The intuition for this context is that these variations should be different during these two activities.

The next three contexts (no. 9, 10 and 11) represent the movements of the torso (up, down, left, right, forward and backward) while the user is sitting or standing. The standard deviation is computed for the acceleration values of each of the axes. The final context represents the combination of the transition and the recognized activity for the after activity segment.

The CoReAmI approach for AR is presented in Figure 4-7. At the top, the 3-axis accelerometer data is used to extract the 12 contexts. Each context has context values. Most of the values are in numerical format, e.g., standard deviation, tilt angles, etc. In order to train a reasonable number of models and to maintain a reasonable amount of data for training, a discretization procedure was performed. Each numerical feature was discretized into a number of intervals (e.g., very low, low, medium etc.) using the Decision Tree's split criterion as implemented in WEKA [109]. It is the most commonly used supervised discretization technique in the ML community, which finds such splits for a given feature that the information gain of the class (activity) value is maximized. This is repeated as long as at least 10% of the data instances remain in each interval. Finally, each of the intervals is denoted by one discrete context value (low, medium, high, etc.).

Once the context values are defined, for each discrete value (e.g., "Standing") of each context (e.g., "Current activity"), a classification model (e.g., $m_{CA=Standing}$) is trained. The training dataset for the model is a subset of the whole training dataset. It contains only the data instances which have the corresponding value of the context thus, the model for the first context feature for the first value, i.e., $m_{CA=standing}$, is trained only on the data instances that contain the value "Standing" for the "Current activity" context. Once the subsets of the training data are defined, the models are trained using an arbitrary classification method. In this study the J48 (Decision Tree) [95] classification method was chosen because of the reasonably high understandability of the learned models, and also because it outperformed the other tested algorithms (RF, Naive Bayes and SVM) on preliminary tests.

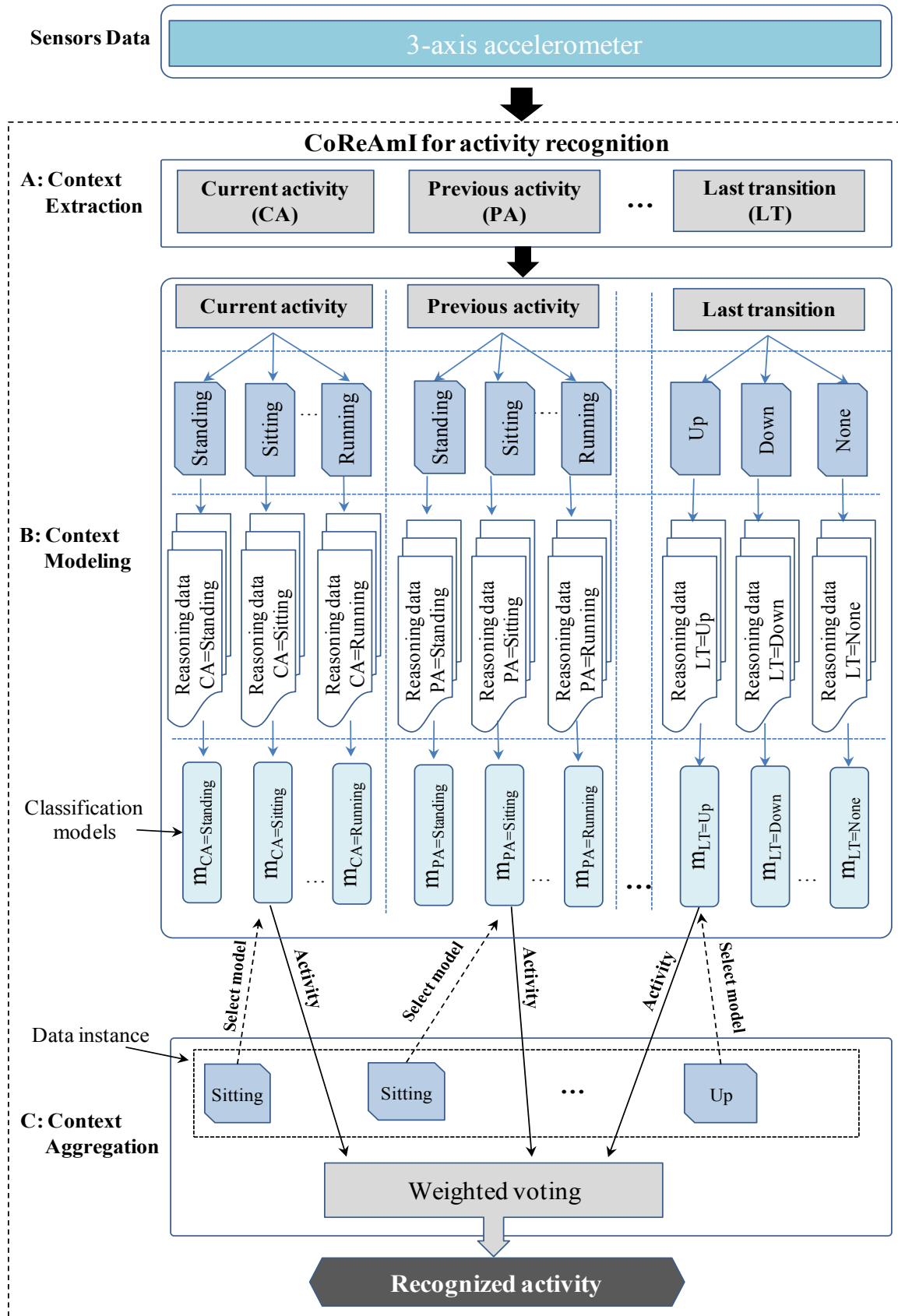


Figure 4-7. CoReAmI for activity recognition.

For a given test data instance, a custom ensemble is assembled from the models, i.e., the models that contain the context values are invoked. Figure 4-7 shows an example of a data instance containing "Sitting" as the "Current activity", "Sitting" as the "Previous activity", and "Up" as the "Last transition". Thus, those three models ($m_{CA=Sitting}$, $m_{PA=Sitting}$ and $m_{LT=Up}$) are included in the ensemble.

Once a data instance is classified by the appropriate models, the final decision about the recognized activity is made using a method for combining the outputs of the models. After testing several commonly used methods for combining classifiers outputs, such as majority vote, plurality vote and weighted voting [109], the last one proved to perform best. The weighted voting scheme gives more weight to the vote of the classifiers with better accuracy. Because in our case only two activities are possible (sitting and standing) we applied separate weights for each of the two activities. Therefore, each classifier has a weight for each of the two activities: w^{sit} and w^{stand} . The weights are also normalized to the interval $[0, 1]$ for both activities individually. The output activity (decision d) of a sample x , is calculated by:

$$d = k_{\arg \max_j \sum_{i=0}^n w_i^j * d_i^j} \quad (17)$$

where n is the number of the classifiers, k_j is the recognized activity (sit or stand), w_i^j is the weight assigned to each classifier for the activity k_j , d_i^j is the output of each classifier for the class k_j , which can be 0 or 1.

4.3 Experimental Setup

4.3.1 The Experimental Scenario

A complex, 90-minute, test scenario was designed in cooperation with a medical expert to capture the real-life conditions of a person's behavior, although it was recorded in a laboratory. The scenario was performed by ten volunteers and included eight elementary activities (the percentage of instances per class): standing – Stand (13%), sitting – Sit (11%), lying – L (32%), on all fours – A4 (8%), walking – W (23%), running – R (9%), transition down – TD (2%) and transition up – TU (2%). These activities were selected because they are the most common elementary, everyday-life activities. They were grouped into three groups. The first group was exercising activities: walking on a treadmill with a one-percent inclination at 4 km/h and 6 km/h, running on a treadmill with a one-percent inclination at 8 km/h and cycling on a stationary bicycle with 65 RPM with the difficulty set to 80 watts for the first six minutes and 160 watts for the other six minutes. In the second group, elementary activities and transitions between the activities were recorded. The sequence of activities performed in these sub-scenarios was predefined and volunteers were asked to follow them. In the third group, everyday-life activities were recorded. The sequence of activities was not predefined and the volunteers were asked to mimic their normal, everyday-life behavior when executing activities such as cooking, reading, typing, washing dishes, scrubbing the floor, etc.

Altogether, ten sub-scenarios were recorded, resulting in 140 recordings, as some sub-scenarios were repeated multiple times, yielding a total of approximately 1,000,000 raw-data samples per volunteer. These raw-data samples were transformed into approximately 5,000 data instances per volunteer.

4.3.2 Method Evaluation

For the evaluation of the AR, the leave-one-person-out cross-validation technique was used. This means the model was trained on the data recorded for nine people and tested on the remaining person. This procedure was repeated for each person (ten times). This evaluation approach is more reliable than using the same persons' data for training and testing. Using the same person's data would give overly optimistic results if the intended use of the model is to classify the activities of previously unseen people. Four, commonly used in AR, evaluation metrics were analyzed: the recall, precision, accuracy and F-measure (F1). The following formulas define each of the metrics, where Q can be any type of activity (sitting, standing, etc.):

$$recall = \frac{\text{Number of correctly recognized activities of type Q}}{\text{Number of all the activities labeled as Q}}, \quad (18)$$

$$precision = \frac{\text{Number of correctly recognized activities of type Q}}{\text{Number of all the activities recognized as Q}}, \quad (19)$$

$$accuracy = \frac{\text{Number of correctly recognized activities of all types}}{\text{Number of all the activities}}, \quad (20)$$

$$F1 = \frac{2 * recall * precision}{recall + precision}, \quad (21)$$

The first three formulas are explanatory enough and explain the meaning of the metric. The last one, F-measure (F1), is a harmonic mean between the recall and the precision; it weights the recall and the precision in a balanced way.

We compared the results achieved by the CoReAmI approach and the baseline approach. Additionally, we compared the accuracy achieved by a ML method that uses the context features, but without the context-based modeling phase. We also compared the accuracy achieved by each of the context models used individually (without the aggregation phase). Finally, we compared the accuracy achieved by the baseline, CoReAmI and ML approaches (Decision Tree and Random Forest ensemble) that use the same 12 context features (without the context modeling and context aggregation phases).

For each comparison, tests to confirm the statistical significance of the results were performed using paired Student's T-test with a significance level of 5%.

4.4 Results

Figure 4-8 shows the recall and precision for the sitting (left) and the standing (right) activities achieved by the baseline approach and CoReAmI. The improvements of CoReAmI over the baseline approach are significant, i.e., 16 percentage points (p. p.) and 31 p. p. for the recall, and 17 p. p. and 22 p. p. for the precision, respectively.

Table 4-3 and Table 4-4 show the detailed confusion matrices, recall and precision values for the baseline AR and CoReAmI. Further analysis of only these two activities (the 2x2 confusion sub-matrix marked with red rectangle in Table 4-3 and Table 4-4) shows that the accuracy on them is improved by 24 percentage points, from 62% to 86% (also shown in Figure 4-9 with the red and blue bars, respectively).

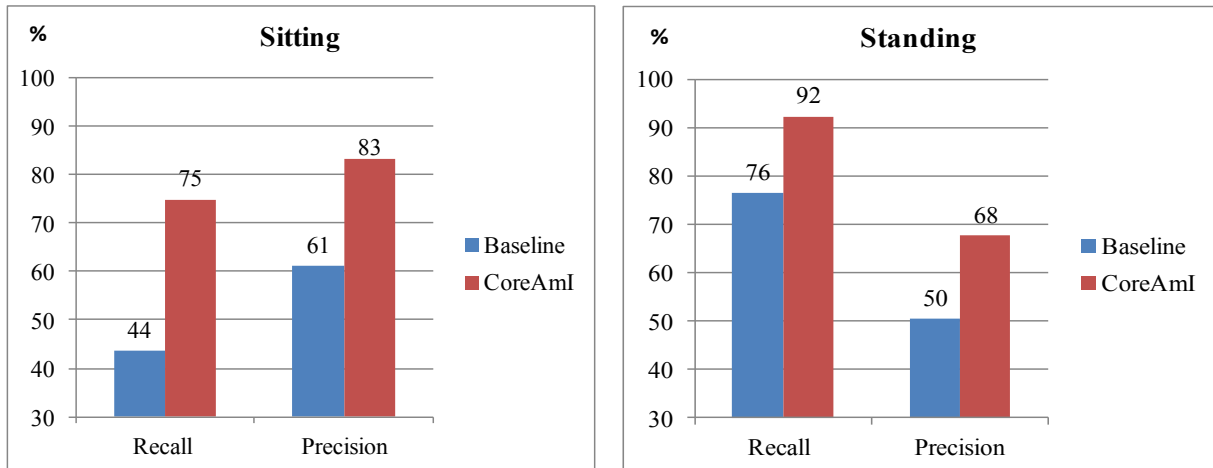


Figure 4-8. Recall and precision for the sitting (left) and the standing (right) activities achieved by the baseline approach and CoReAmI.

Table 4-3. Confusion matrix for the baseline AR.

Baseline	W	Stand	Sit	TD	TU	A4	R	L	Recall
W	9098	304	5	53	109	8	40	0	94.6%
Stand	103	4265	1086	9	23	3	0	0	77.7%
Sit	55	2687	1793	22	21	2	0	0	39.1%
TD	211	237	23	254	95	57	0	35	27.9%
TU	55	140	24	40	264	75	2	56	40.2%
A4	2	511	0	20	11	3287	0	77	84.1%
R	82	8	2	0	1	0	3522	0	97.4%
L	3	306	0	38	52	241	10	12816	95.2%
Precision	94.7%	50.4%	61.1%	58.3%	45.8%	89.5%	98.5%	98.7%	

Table 4-4. Confusion matrix for the CoReAmI AR.

CoReAmI	W	Stand	Sit	TD	TU	A4	R	L	Recall
W	9023	249	141	52	104	8	40	0	93.8%
Stand	71	5068	321	9	18	2	0	0	92.3%
Sit	44	1074	3432	13	16	1	0	0	74.9%
TD	196	203	103	235	85	55	0	35	25.8%
TU	51	90	101	38	245	73	2	56	37.3%
A4	2	508	8	19	11	3283	0	77	84.0%
R	79	5	8	0	1	0	3522	0	97.4%
L	3	305	3	37	52	241	10	12815	95.2%
Precision	95.3%	67.6%	83.4%	58.3%	46.1%	89.6%	98.5%	98.7%	

Since CoReAmI consist of 12 contexts, each of which can be used individually, we compared their accuracy for the standing and sitting activities (blue bars in Figure 4-9) with the accuracy of the complete CoReAmI (red bar). The results show that CoReAmI outperforms each of the contexts used individually (only the context models learned for the particular context). This shows that by combining the models using a weighted average of their outputs, the ensemble outperformed the individual models.

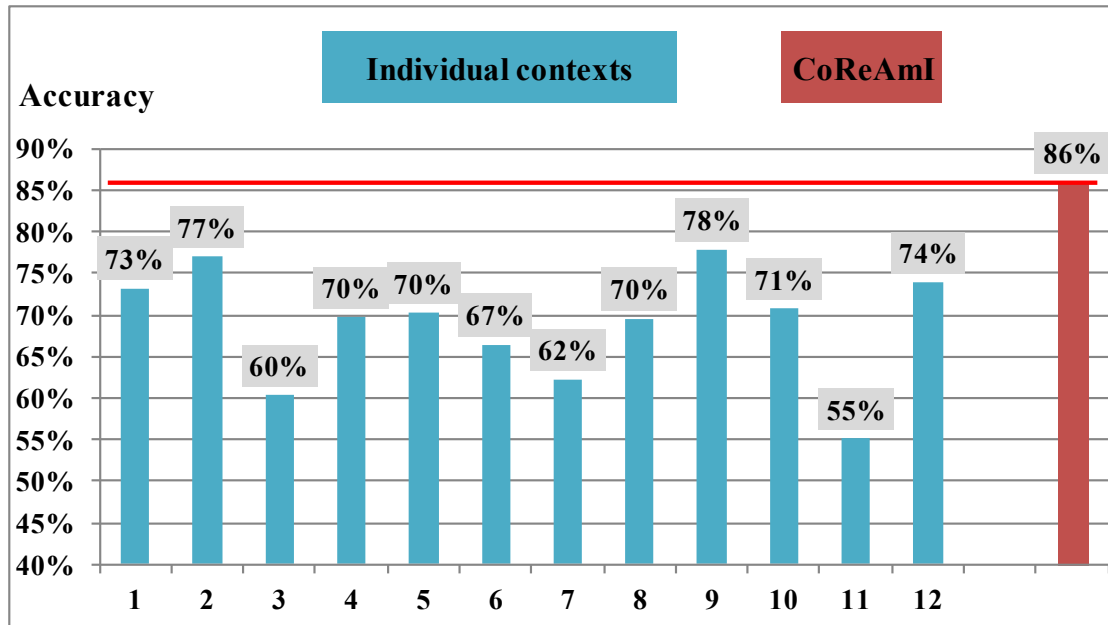


Figure 4-9. Comparison of the accuracy achieved for the standing and sitting activities using: each of the context models individually and CoReAmI.

We additionally compared the accuracy and the F-measure for the standing and sitting activities achieved by the baseline, CoReAmI and ML approaches that use the same 12 context features, but simply combine them into a feature vector without CoReAmI (see Figure 4-10). For training the ML algorithm we used: (i) the same J48 algorithm (J48-ML) as used for training the context models in CoReAmI, (ii) the RF (RF-ML) ensemble algorithm.

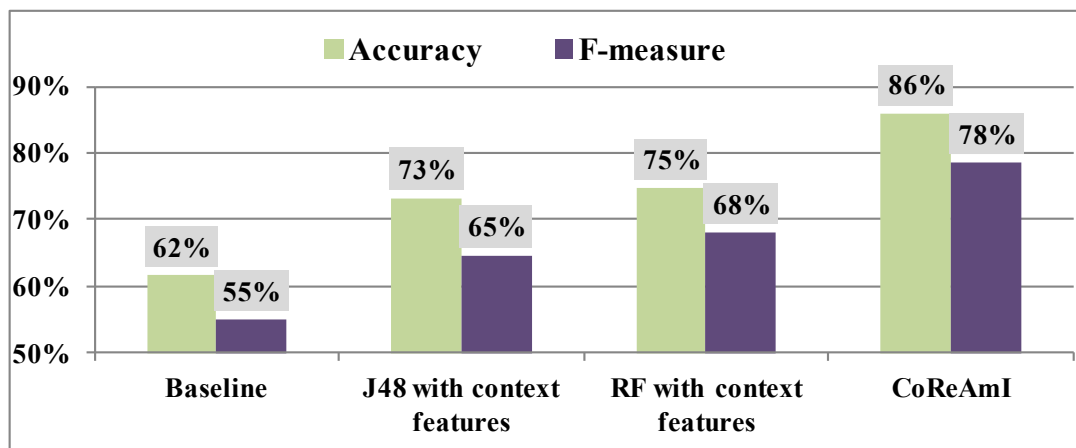


Figure 4-10. Comparison of the accuracy and the F-measure for the standing and sitting activities achieved by the baseline, J48 trained with the context features (J48-ML), Random Forest trained with the context features (RF-ML), and CoReAmI.

Figure 4-10 shows that CoReAmI significantly outperforms the baseline and the two ML approaches in both the accuracy and F-measure. The results achieved by the J48-ML approach show that the additional context features improve the performance compared to the baseline approach (11 p.p. accuracy and 10 p.p. F-measure). However, when the context-based modeling (CoReAmI) is applied, this improvement is significantly higher, i.e., 24 p.p. accuracy and 23 p.p. F-measure. The results achieved by the RF-ML approach are similar to the J48-ML and are significantly lower than the CoReAmI approach. This shows the advantage of using a context-based ensemble (CoReAmI) compared to the RF – a commonly used ensemble in the AR literature.

4.5 Summary and Discussion

Distinguishing standing from sitting in real-life circumstances has so far been too demanding to perform accurately with one accelerometer attached to the torso. But several "impossible tasks" get solved in time as sophisticated methods are introduced. The trick in our case was to represent the available information from multiple viewpoints and to intelligently integrate these viewpoints. The proposed CoReAmI approach significantly improved the recognition performance for the sitting and standing activities, i.e., 24 p.p. accuracy and 23 p.p. F-measure. The improvement was achieved in two steps.

First, we introduced additional contexts (features) designed specifically to distinguish standing from sitting. The results in Figure 4-10 showed that by introducing these contexts and applying a regular ML (we used J48 Decision Tree), the accuracy and the F-measure for the two activities improved by 11 p.p., and 10 p.p. respectively. This improvement shows that the chosen contexts contain additional information that helps ML perform better.

In the second step, we introduced the context-based modeling and aggregation phases in the CoReAmI approach – which resulted in the creation of context-based ensemble of classifiers. The CoReAmI improved upon the J48-ML by another 13 p. p., for both the accuracy and F-measure (shown in Figure 4-10). Additionally, we compared the results to RF, which also created ensemble of Decision Trees. Our CoReAmI achieved significantly higher accuracy and F-measure compared to the RF-ML, i.e., improvement of 11 p.p and 10 p.p. for accuracy and F-measure, respectively. This improvement shows the advantage of the context-based reasoning in CoReAmI, which exploits the complementarity of context-based multiple classifiers. Each context classification model is trained on a subset of the training set that is more homogeneous than the whole set, and used in the context of this subset, i.e., to classify examples similar to the ones in the subset. In other words, in our approach we semantically split the dataset (by using contexts) and not on some statistics about the data (as is the case with the RF). This improvement confirms the hypothesis of the thesis, i.e., combining multiple sources of information by using context-based approach, i.e., using each source of information as a context, improves the performance of the reasoning.

Even though the proposed approach significantly improved the accuracy for the sitting and standing activities in general, 23% of the sitting samples were still not recognized as sitting. We further analyzed this problem and realized that in most of these cases, the transition segment before the sitting was not recognized by the appropriate contexts. In some of the cases data samples were missing (due to sensor malfunctions), and in some of the cases the transition was performed too quickly. In the future we plan to use more reliable hardware and more thoroughly analyze the transitions, especially the quick ones.

We also plan to apply CoReAmI to data from the accelerometers placed on other parts of the body, e.g., the thigh, ankle etc. This will require some modifications: for instance, initial results show that for the thigh placement, the problematic activities are sitting and lying, so they are the ones to which CoReAmI should be applied. It should also be possible to apply CoReAmI to all the activities.

Even though CoReAmI brings additional processing and computational complexity, the proposed method still executes almost in real-time. The only significant delay is introduced by the analysis of the after-activity segment. However, this delay is in the range of seconds and should not be a problem for some time-sensitive applications. For example, a few seconds delay in the recognition of a user lying on the floor and raising an alarm is still acceptable.

The problem discussed in this section may appear narrow, focusing only on the recognition of the standing and sitting activities. However, since AR with accelerometers is fairly mature, it seems time to tackle the remaining weaknesses such as this one. Also, since these two activities are significantly different from the health perspective, distinguishing them is important for health-promotion applications whose popularity is on the rise.

For future work we also consider analysis of high level activities, such as going to work, eating, exercising and sleeping. This requires increasing the abstraction level and the reasoning interval to minutes or even hours. Additional sensor information is also considered for this type of activities, such as: GPS, WiFi signals and microphone. Because these types of sensors are present in almost all smartphones, a smartphone implementation is also considered for future work. Two options are considered: (i) the smartphone sends the sensors data on a server and the CoReAmI implementation is on the server; (ii) light version of CoReAmI is implemented on the smartphone itself, and the reasoning is performed locally.

5 Energy Expenditure Estimation Domain

The human Energy-Expenditure (EE) estimation is the second problem domain on which we applied the CoReAmI approach. EE is the process of expending energy while performing everyday activities. It directly reflects the level of physical activity which makes it important for sports training, weight control, management of metabolic disorders (e.g., diabetes), and other health goals. There are different approaches that can reliably estimate the EE. Direct calorimetry measures the total heat output of a person in an accurate way, but is only usable in laboratory conditions. The slightly less accurate indirect calorimetry analyzes the respiratory gases and requires wearing a breathing mask, making it non-practical for everyday usage. Doubly labeled water is both accurate and convenient, but can measure only long-term EE. Finally, self-reporting is highly unreliable. Therefore, if both accuracy and convenience are required, a different approach is needed.

With the increasing accessibility and miniaturization of sensors and microprocessors, ubiquitous monitoring systems are becoming a practical solution for measuring EE. Such systems primarily measure the physical activity with accelerometers, but can include additional sensors that indirectly measure the metabolic activity, such as a heart rate monitor or thermometer. The main challenge is how to estimate the EE from wearable sensor outputs accurately, irrespectively of the user's activity, ambient conditions and other circumstances, i.e., contexts.

In this chapter we present the application of the CoReAmI approach to the task of EE estimation. The chapter is organized as follows. First, we present the related work in EE estimation. Then, we present the approach itself, including the sensor equipment, context extraction phase and the context-based EE estimation. In the next two sections the experimental setup and the results are presented. In the final section, a discussion and directions for future work are provided.

5.1 Related Work

The first methods for EE estimation with wearable sensors used linear regression equations to map a single accelerometer output to EE [145][146][147][148]. The accelerometer output was often expressed in "counts", an aggregate acceleration measure reported by devices such as ActiGraph. To estimate the EE, investigators have used these "counts" to develop linear regression models. Although numerous studies have shown reasonably good correlation between the counts and the EE [145][149], the estimation accuracy of accelerometer count-based linear regression has been shown to contain systematic errors and vary with the type of activities, resulting in overestimations for the walking activity and underestimations during the moderate intensity lifestyle activities [150]. This limitation is probably due to the insufficient information provided by the counts and the simplicity of the linear model. Efforts have been made for improving the estimation accuracy by using a richer representation of the accelerometer output consisting of multiple features [151][152], as well as non-linear

regression methods such as artificial Neural networks [153][154][155] or Support vector machine for regression (SVR) [156][157]. These approaches were experimentally shown to substantially improve upon the earlier work [158].

Researchers soon realized that single-regression approaches cannot accurately predict physical activity intensity across a range of activities and that different activities require different energy-expenditure equations. Crouter et al. [76] used the acceleration counts in order to divide the activities into three categories and assigned the following EE estimations: one metabolic equivalent of task (MET) to inactivity and two regression equations for light and intense activity, thus achieving a better estimate than previous single-regression methods. The advances in the accelerometer-based activity type recognition allowed finer-grained activities as the context for EE estimation [159][123][103]. Lester et al. [77], used a Naive Bayes classification model to first recognize three activities (rest, walking and running) out of the accelerometer data, and then to apply the appropriate regression equations in order to estimate the EE. They also considered GPS and barometer information to estimate the slope of walking/running, and showed that additional sensor information improves the EE estimation. However, even with these three types of sensors (accelerometer, GPS, and barometer) they still encountered two problems: (i) EE underestimation of activities that are not characterized with acceleration, but are still energy demanding, e.g., carrying a box and (ii) EE underestimation of activities that follow an intense activity, i.e., the "cool-down" effect (sitting after intense running). Both problems can be solved by sensing other physiological parameters such as heart rate and breathing rate. This may seem as an additional burden to the user, because it requires additional sensors attached to him/her. However, today's commercial wearable devices already provide multiple sensors packed in a single enclosure, e.g., BodyMedia, Zephyr BioHarness, Basis, Empatica wristband, etc.

The BodyMedia armband sensor uses both multiple sensors and multiple regression models. Vyas et al., [78], the research team of the BodyMedia, proposed a method that uses an activity-recognition model that recognizes dozens of activities which are used as context, and then it combines multiple regression models according to the probabilities for the recognized activities. They showed that by using multiple sensors: an accelerometer, two thermometers, galvanic-skin-response and heat-flux sensors, the estimation of EE significantly improves. Additionally, a recent review showed that it is the most accurate consumer EE estimation device [160].

The aforementioned studies showed that: (i) using multiple regression models for different user's activities (i.e., context) outperforms single-regression approaches, and (ii) using multiple features extracted from multiple sensor data provides more accurate EE than using only acceleration data (even when multiple acceleration features are extracted). In this work we advance upon these findings and propose a method that: uses multiple sensor data and uses not only the activity as the context, but multiple contexts, so that each measurement can be placed in multiple contexts simultaneously (e.g., activity = running, heart rate = high, breath rate = moderate, etc.). This way, a context-based reasoning is performed, which provides the benefit of combining multiple "viewpoints" when estimating the EE, resulting in further improved accuracy compared to the previous approaches.

5.2 Energy Expenditure Estimation with CoReAmI

The EE estimation is usually expressed in METs, where one MET is defined as the energy expended at rest. MET values usually range from 0.9 (sleeping) to over 20 in extreme exertion.

EE estimation is a process that transforms sensor data into EE estimation (e.g., METs). In our approach the user is wearing multiple sensors and our algorithm estimates the EE on some predefined time interval (e.g., every 10 seconds). As shown in the previous section, advanced approaches in EE estimation already use the user's activity as a context and apply regression models for each activity in order to estimate the user's EE [159][77]. An example of such approach is shown in Figure 5-1.

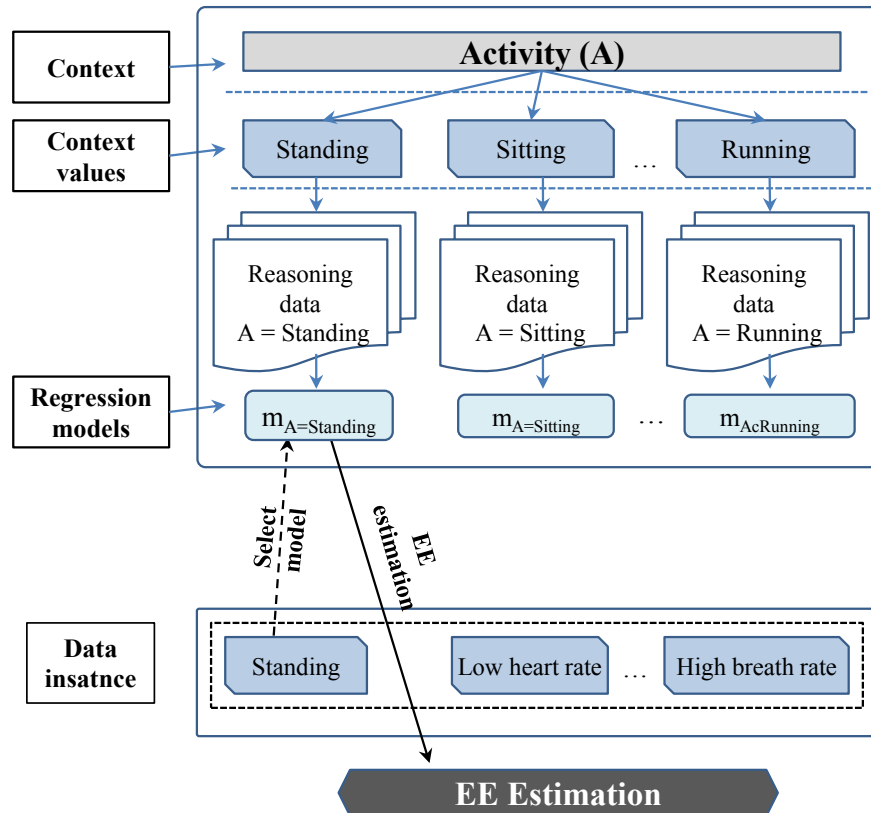


Figure 5-1. EE estimation using activity as a context.

In our approach, we go one step further and extract multiple contexts from the sensor data. Therefore, we applied the CoReAmI approach adapted to EE estimation task (shown in Figure 5-2). In the context extraction phase, multiple contexts are extracted from the sensor data, such as activity, heart rate and breath rate. Then, in the context modeling phase, for each context value a regression model is constructed. In the final phase, a data instance is evaluated by multiple context models, which are invoked according to the context values. The estimations of each model are then aggregated by choosing the median value as the final EE estimation value.

The main idea of using context-based approach for EE can be explained through the following example. Consider that three contexts are extracted: activity, heart rate and breath rate. There are three possible views of the user's situation. First, the context is defined by the activity of the user and the other two contexts are used to model the EE depending on the context: heart rate and breath rate. Next, the context is the heart rate and the modeling is performed using the other two contexts: activity and breath rate. And third, the context is the breath rate and the modeling is performed using the user's heart rate and the activity. This way, a multi-view perspective is provided which should lead to better EE estimation.

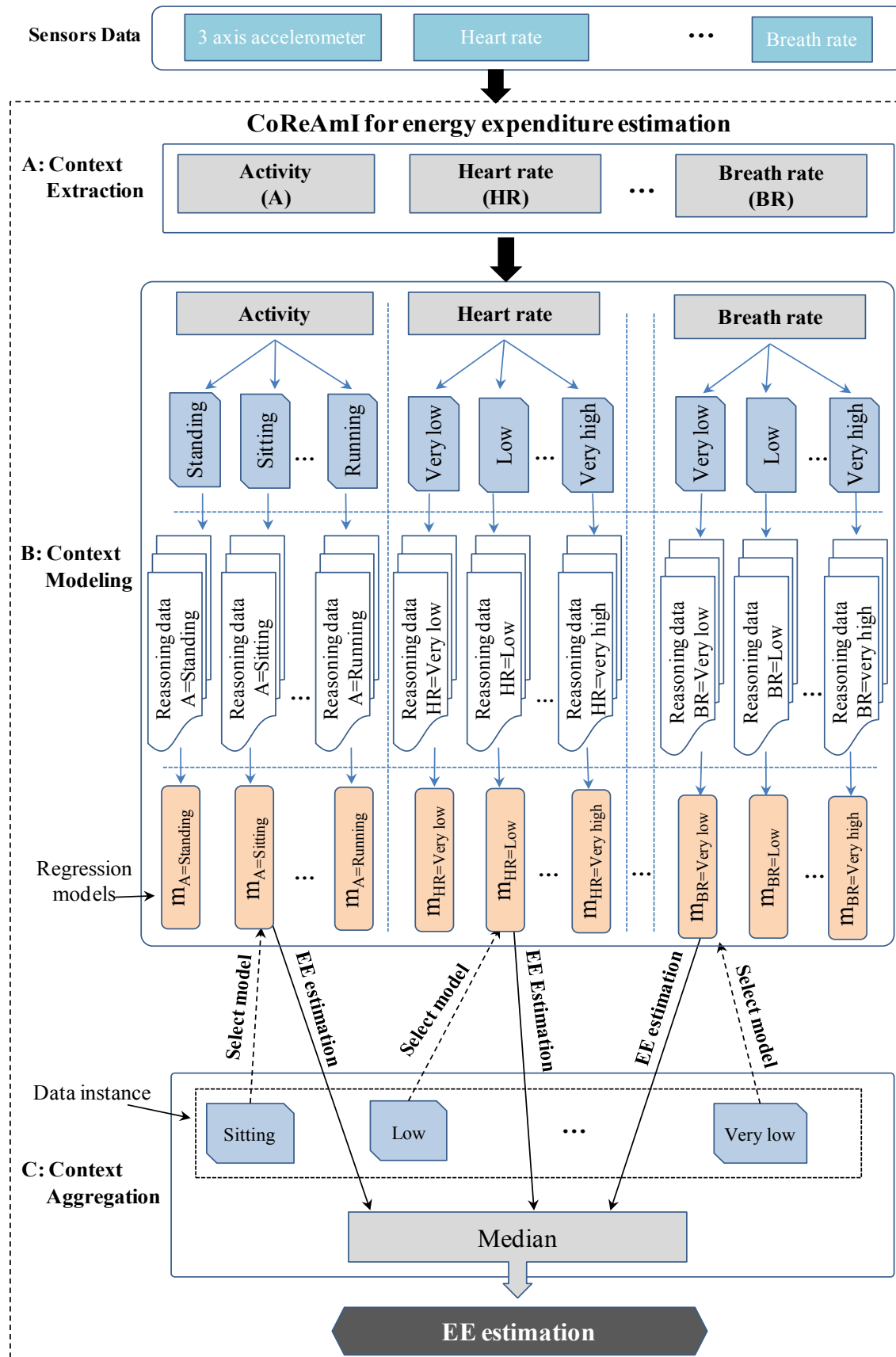


Figure 5-2. CoReAmI for EE estimation.

5.2.1 Sensors

A person wearing the sensor equipment and walking on a treadmill is shown in Figure 5-3. It consists of the following wearable sensors: two 3-axis accelerometers, a Zephyr sensor, and a BodyMedia sensor. Each of the sensors used in this study provided different information about the user's EE. Because the BodyMedia sensor is the state of the art EE estimation sensor, its MET output was used for comparison. Additionally, a Cosmed indirect calorimeter device was used to provide the reference output for the EE estimation.

3-axis accelerometers – The same as for the AR, Shimmer sensor platform was used to measure the accelerations of the user. Each user wore two accelerometers while performing the activities. One accelerometer was worn on the chest in order to measure the body-trunk movements and orientation, and the other one was worn on the right thigh in order to measure the leg movements and orientation. They were attached to the users' chest and thigh using elastic Velcro straps. The placement was chosen as a trade-off between the physical obtrusiveness and the performance achieved for the activity recognition in the preliminary tests [20][21].



Figure 5-3. A person wearing the sensor equipment for EE estimation.

Zephyr BioHarness sensor – The Zephyr BioHarness sensor is a commercial sports strap worn on the chest. It measured user's heart rate, breath rate, and chest skin temperature, which were also used as context in our context-based reasoning method.

BodyMedia sensor – The BodyMedia sensor is a state-of-the-art commercial sensor for EE estimation, which was worn on the left upper arm as suggested by the manufacturer. It served as a benchmark for comparison of the EE estimation, and additionally it provided the user's

galvanic skin response, ambient temperature and arm skin temperature, which were used as contexts in our context-based reasoning method.

Cosmed indirect calorimetry – Oxygen uptake (VO_2) during each activity was measured breath-by-breath and averaged every ten seconds using the Cosmed K4b², a light-weight portable indirect calorimetry system. A flexible Hans-Rudolph face mask held in place by a head harness covered each user's nose and mouth and measured the volume of inspired and expired air. A sample tube running from the mask to the analyzer unit delivered expired air for the determination of O_2 and CO_2 content. Prior to each test, the Cosmed unit was calibrated according to the manufacturer's guidelines. Flow control and gas calibration were performed using the Cosmed automated calibration system. This sensor's data were used as a ground truth for the regression learning algorithms and for the evaluation of the method performance.

5.2.2 Context Extraction

A custom PC application was created to record, preprocess and synchronize the multiple sensor data. During the recordings, the accelerometers' data were acquired on a laptop in real-time using Bluetooth. Additionally, the data were manually labeled with the corresponding activity, which was later used for the training of the activity-recognition classification model. The data provided from the other sensors was labeled with the appropriate timestamp and saved locally in the sensor's internal memory. Afterwards, together with the accelerometer data and the activity labels, they were transferred into a database for offline analysis. Once the multiple sensors' data were saved into the database, they were synchronized (offline) using the unique timestamp for each data sample. In order to perform synchronization between sensors (devices), almost all of the devices (PC, Zephyr, BodyMedia, Cosmed) were adjusted to the same absolute time before the recordings, i.e., the same NTP server was used to fix the absolute time. Because the Shimmer sensors does not have that option, their data was streamed in real-time, thus each data sample was labeled with unique timestamp from the PC. In the next step, they the data samples were segmented using a non-overlapping sliding window of ten seconds. In each data window, eight contexts were extracted: activity, acceleration peak counts, heart rate, breath rate, chest body temperature, arm body temperature, arm galvanic skin response (GSR), and near-body temperature. The true EE was calculated by averaging the MET values recorded by the Cosmed sensor over the ten-second time segments.

The following eight contexts were extracted: activity, heart rate, breath rate, acceleration peaks count, chest skin temperature, GSR, arm skin temperature and near-body ambient temperature. Table 5-1 shows which contexts were extracted from which sensor.

Table 5-1. Contexts extracted from each sensor.

Sensor	Context
Shimmer accelerometers	Activity, acceleration peaks count
Zephyr	Heart rate, breath rate, chest skin temperature
BodyMedia	GSR, arm skin temperature, ambient temperature
Cosmed	Reference METs

Except for the activity and the acceleration peak counts, all other contexts are provided directly by the sensors (Zephyr BioHarness, BodyMedia or Cosmed) and are computed by

averaging the raw sensor data in the ten-second intervals. The physiological signals provided by the Zephyr BioHarness and BodyMedia (heart rate, breath rate, etc.) differ from user to user and were additionally normalized. After empirical analysis of the data, we used the 15-minutes lying activity data recorded at the beginning of the activity trials in order to calculate the average value for each sensor data, which was subtracted from each sensor value.

To extract the acceleration peak counts and the activity of the user from the acceleration data, they were first filtered using a band-pass filter [161]. The acceleration peaks count is the number of times the length of the acceleration vector stops increasing and starts decreasing or vice versa in the 10-second interval. For the activity recognition we used a previously developed classification method based on ML [22]. The method uses the data from the two accelerometers (chest and thigh), extracts 128 features and applies a Random Forest (RF) classification model to recognize the atomic activities of the user: lying, sitting, standing, walking, running, cycling, bending, on all fours and kneeling. It achieved 93% accuracy on a one-second recognition interval. For the EE, the majority activity value was chosen for each ten-second window interval. A similar implementation of the activity-recognition method achieved the best recognition performance at the EvAAL-2013 competition [12][23].

5.2.3 Context-based Energy Expenditure Estimation

Once the context values are defined, for each discrete value of each context, a regression model is trained. The training dataset for the model is a subset of the whole training dataset. It contains only the data instances which have the corresponding value of the context; thus, the model for the first context for the first value, i.e., $m_{A=Standing}$, is trained only on the data instances that contain the value *Standing* for the *Activity* context. Once the training data is selected, the model can be trained using an arbitrary regression method. We used and later compared the results achieved by five linear and non-linear regression learning methods (ANN, SVR, etc.) as implemented in the WEKA ML toolkit [109].

Most of the context information is provided in numerical format, i.e. numbers that represent the user's heart rate, temperature, etc. In order to train a reasonable number of models for different context values, similar to the AR approach, a discretization procedure was performed. Each numerical context was discretized using the split criterion proposed by Yong et al. [99]. The mathematical definition is given in Section 3.4. The discretization procedure was repeated as long as at least 10% of the data instances remain in each interval. This resulted in 46 discrete context values.

By applying this discretization technique, 46 regression models were trained in total (for each context value of each context). As shown in Figure 5-2, the EE of a data instance is estimated by an ensemble consisting of a subset of the whole set of models. The models included in the ensemble are invoked according to the context values, i.e., a single model are invoked for a single context. This way, each data instance is evaluated by 8 models. As a comparison, an approach that uses only the activity as a context would construct as many models as there are activities (in our case 10), but would still evaluate the testing data instance with only one model – the model constructed for the corresponding activity (as shown in Figure 5-1).

The final EE estimation is provided by combining the outputs of the invoked models. For example, consider the following scenario with three contexts: a user is running with the heart rate of 140 min^{-1} and breath rate of 10 min^{-1} . Consider that the heart rate value falls in the second heart rate interval (*Low*), and the breath rate value into the first breath rate interval

(*Very low*). The data instance will thus be evaluated by the models $m_{A=Running}$, $m_{HR=Low}$ and $m_{BR=Very\ low}$, whose outputs will be combined (e.g., by averaging, choosing the median, etc.) to estimate the final EE. The empirical analysis of the data showed that choosing the median value is the most suitable for EE estimation. In this case the models that are not accurate for some situations are discarded and not taken in consideration, which is not the case if the average is chosen.

5.3 Experimental Setup

5.3.1 Experimental Activity Scenario

We used the same activity scenario as the one used for AR (Subsection 4.3.1). A total of ten healthy users (age 27.2 years (SD = 3.1); BMI 24.1 kg·m⁻² (SD = 2.3); weight 78.2 kg (SD = 10.9)) completed the two-week study. Before testing, height and weight (one layer of clothes, no shoes) were measured via InBody-720 body composition analyzer. Prior to participation, informed consent was obtained from the users.

Each user was observed by a medical supervisor during the execution of a pre-defined comprehensive activity scenario. The supervised measurements lasted approximately eight hours for each user and were recorded starting in the early morning. The activity scenario included 15 different atomic activities that were categorized into seven activity types according to the intensity and the type presented in Table 5-2: sedentary, light household activities and exercise (Light HH & exercise), moderate to vigorous household activities (Mod-Vig HH), walking, cycling light, cycling vigorous, running.

Table 5-2. Activity scenario.

Activity type	Atomic activities	METs
Sedentary	Lying, sitting, standing, on all fours, kneeling	1.0 – 1.5
Light HH & exercise	Washing dishes, working on a PC, lying and doing light exercise, walking doing light chores	1.5 – 2.5
Mod-Vig HH	Scrubbing the floor, shoveling snow - digging	2.5 – 3.5
Walking	Walking on a treadmill with 4km/h	4.0
Cycling light	Light stationary cycling: 1 W/kg of body mass, 65 RPM	5.0
Cycling vigorous	Vigorous stationary cycling: 2 W/kg of body mass, 65 RPM	7.5
Running	Running on a treadmill with 8km/h	8.0

5.3.2 Method Evaluation

The evaluation of CoReAmI was performed using the leave-one-person-out cross-validation technique [162]; that is, models were trained on the data of nine people and tested on the remaining person. This procedure was repeated ten times, once for each person. The same procedure was performed for the activity recognition classifier, whose output was used as a context in the EE estimation. This evaluation technique is the most commonly used in the ML community if the model is intended to be used on a user different from the ones used for training, which is the case in the EE estimation [154]. This method yields an estimate of how

well the model would do if it were applied to a population on which it was not trained. As for the evaluation metrics, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used, since they are the most commonly used metric in the EE estimation domain. They are defined as follows:

$$RMSE = \sqrt{\frac{1}{q} \sum_{1}^q (EE_{estimated} - EE_{true})^2} \quad (22)$$

$$MAE = \frac{1}{q} \sum_{1}^q |EE_{estimated} - EE_{true}| \quad (23)$$

where q is the number of data instances, $EE_{estimated}$ is the estimated EE and EE_{true} is the true EE measured by the Cosmed device.

Because CoReAmI is independent of the regression algorithm used for training, we compared five different algorithms (base learners): Multiple Linear Regression (MLR) [106], Support Vector Machine for Regression (SVR) [107], Gaussian Processes for Regression (GPR) [163], Model Trees (M5P) [99], and multilayer perceptron feedforward Artificial Neural Network (ANN) [164]. In addition, as a baseline for comparison, we evaluated the same regression methods without CoReAmI, i.e., single regression models were constructed over the whole context dataset.

Next, because in this particular application of CoReAmI an ensemble of regression models is constructed, we also compared it to two commonly used ensemble learning methods: Bagging [97] and Random Subspace [98]. Bagging is an approach that is based on bootstrapping, i.e., training multiple models on different subsets of the whole training dataset, constructed by sampling the whole dataset with replacement, and then aggregating the outputs from each model by averaging. Random Subspace method is an ensemble method, proposed by Ho [98], which also modifies the training data; however, this modification is performed in the feature space. That is, a pre-defined number of features is selected randomly from the whole feature set. This procedure is repeated multiple times, creating a different training set for each selection. Then, for each training set, a regression model is built. Similar to Bagging, the final output is provided by averaging. For both ensemble techniques, the same five base ML algorithms were compared as in the single-regression learning.

We also compared CoReAmI's estimated MET to the MET output of the BodyMedia commercial sensor (it should be noted that the BodyMedia sensor averaged the MET estimation over one-minute interval, while our methods over 10-second interval).

Finally, we re-implemented and compared the results of an approach first introduced by Staudenmayer et al. [154] and further improved by Trost et al. [155]. This approach uses an artificial neural network (ANN) trained with six features extracted from the chest accelerometer data only: 10th, 25th, 50th, 75th and 90th percentiles of the acceleration peak counts, and the lag one autocorrelation. This approach achieved better results compared to the conventional regression-based approaches in both of the studies.

For each comparison, tests to confirm the statistical significance of the MAE and RMSE results were performed using paired Student's T-test with a significance level of 5%.

5.4 Results

Table 5-3 shows the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) performance comparison between four approaches: single regression, Random Subspace, Bagging and our CoReAmI. The five base learners explained in Subsection 5.3.2 were tested for each of the approaches. The best performing base learner is marked with bold. Additionally, the best performing approach for each base learner is marked with gray background.

The results achieved by the single-regression methods show that in general the methods that use simple learning functions, e.g., linear or polynomial (SVR, GPR and MLR) are better compared to the more complex ones such as ANN and M5P. This is in a way expected since ANNs and M5P are more susceptible to overfitting, and this problem is even more likely to occur when the testing data are from a person that is not used in the training data. When the five methods are used as base learners in the two ensemble schemes, i.e., Random spaces and Bagging, the results are slightly worse than using single regression, except for the ANN and M5P, for which slight improvement is achieved. However, when our CoReAmI uses the same base learners, the achieved RMSE and MAE are significantly better (lower) compared to the other three approaches, i.e., single regression, Random Subspace and Bagging. The difference ranges from 0.08 METs to 0.24 METs for the RMSE, and from 0.05 to 0.21 METs for the MAE.

Table 5-3. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for CoReAmI's MET estimation compared to the: single regression, Random Subspace and Bagging using 5 base learners: artificial neural network (ANN), support vector regression (SVR), multiple linear regression (MLR), Gaussian processes for regression (GPR), and Model Tree (M5P). The best performance achieved by each of the aggregation techniques for each base learner is marked with bold style. The overall best performance for each aggregation technique is marked with gray background.

	Base learner	Single regression	Random Subspace	Bagging	CoReAmI
RMSE	ANN	1.094	1.059	1.054	0.850
	SVR	0.962	1.033	0.965	0.851
	MLR	0.967	1.033	0.969	0.854
	GPR	0.967	1.081	0.968	0.883
	M5P	1.113	0.991	0.966	0.887
MAE	ANN	0.820	0.770	0.740	0.613
	SVR	0.703	0.749	0.705	0.613
	MLR	0.713	0.766	0.715	0.622
	GPR	0.714	0.818	0.715	0.645
	M5P	0.787	0.734	0.688	0.637

The improvements of our CoReAmI compared to the single-regression approach confirms the general rule in ensemble learning, i.e., ensembles tend to train multiple weak learners, and by combining the learner's outputs they create a stronger and more robust model. The further comparison to the two standard ensemble approaches (Random Subspace and Bagging) shows the advantage of using semantics (context) to resample the training data instead of using bootstrapping (Bagging) or randomly selecting features (Random Subspace).

The CoReAmI approach is general and can use different techniques for aggregating the outputs of each model into a final one. In the tests shown in Table 5-3 we used the simplest aggregation technique, i.e., averaging. This technique was also used by the other two ensemble approaches: Bagging and Random Subspace, making the results more comparable. We additionally tested the performance achieved by the median technique. Table 5-4 shows the comparison of the RMSE and MAE achieved by averaging compared to the median technique. The results show that the RMSE and MAE achieved by the median are almost always better (lower), except for the RMSE achieved by the M5P. The rationale why the median should work better is that by choosing the median value the models that are not accurate for some situations are discarded and not taken in consideration, which is not the case if the average is chosen. Because SVR by using median achieved the best results overall (0.825 RMSE and 0.601 MAE), it is used in all further analysis.

Table 5-4. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) achieved by CoReAmI using two aggregation techniques: average and median; and five base learners: artificial neural network (ANN), support vector regression (SVR), multiple linear regression (MLR), Gaussian processes for regression (GPR), and Model Tree (M5P). The best performance achieved by each of the approaches for each base learner is marked with bold style. The overall best performance for each base learner is marked with gray background.

Base learner	RMSE		MAE	
	Average	Median	Average	Median
SVR	0.851	0.825	0.613	0.601
ANN	0.850	0.840	0.613	0.594
MLR	0.854	0.830	0.622	0.610
GPR	0.883	0.872	0.645	0.637
M5P	0.887	0.893	0.637	0.633

Since CoReAmI consists of eight contexts, we additionally show the results if only a single context is used. In Figure 5-4, one can see that the CoReAmI's EE estimation is better than the estimations provided by each of the base learners individually using RMSE metrics. This shows the advantage of using an aggregation function, i.e., by combining the individual models outputs using a median; the ensemble outperformed the individual models. This result is in accordance with the hypothesis presented by Dietterich [92], who studied the process of combining (aggregation) of the decisions provided by multiple models. He showed that it is better to find a good aggregation function instead of choosing the best single model and that a stronger generalization is achieved in this way.

Of particular interest is the comparison with the first regression model, which uses the activity as the only context: it only uses different regression models for different activities, like in several pieces of related work [76][133][78]. The results show that using the activity of the user as the only context is not sufficient, and that by combining multiple contexts one should expect better performance, i.e., a decrease of the RMSE by 23%.

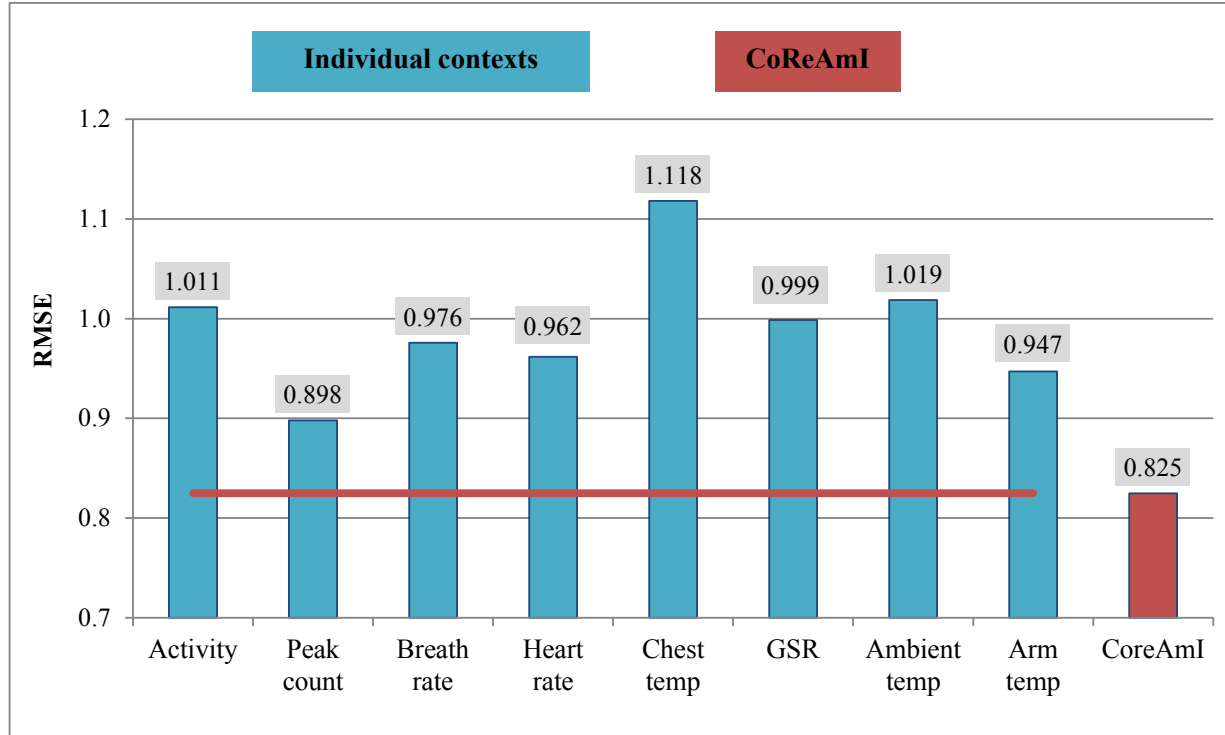


Figure 5-4. Mean Root Mean Square Error (RMSE) for the CoReAmI's MET estimation compared to each of the contexts used individually (only the context models learned for the particular context).

Figure 5-5 shows a scatter plot comparing the measured and estimated MET values for different activities. Three approaches are compared: our CoReAmI approach, the MET output of the BodyMedia sensor, and the ANN trained on chest-accelerometer data only (ANN-Acc).

The results show that in general, the estimations of CoReAmI better match the actual Cosmed MET values (the diagonal line in Figure 5-5) for almost all of the activities. The BodyMedia sensor has comparably good performance for the sedentary activities and for the more dynamic, fitness activities (walking, cycling, running), which is probably because the device is intended for physically active users. On the contrary, for everyday light and moderate household activities the performance is significantly worse than the CoReAmI's estimations. In addition, the results in Figure 5-5 show that the ANN-Acc approach largely underestimates the METs for the dynamic activities, especially for the cycling activity. This was expected, because this method uses only the torso acceleration while the cycling activity is an activity that does not include a lot of torso movement, but has a relatively high MET value. Moreover, this activity was omitted in the original study presented by the authors [155].

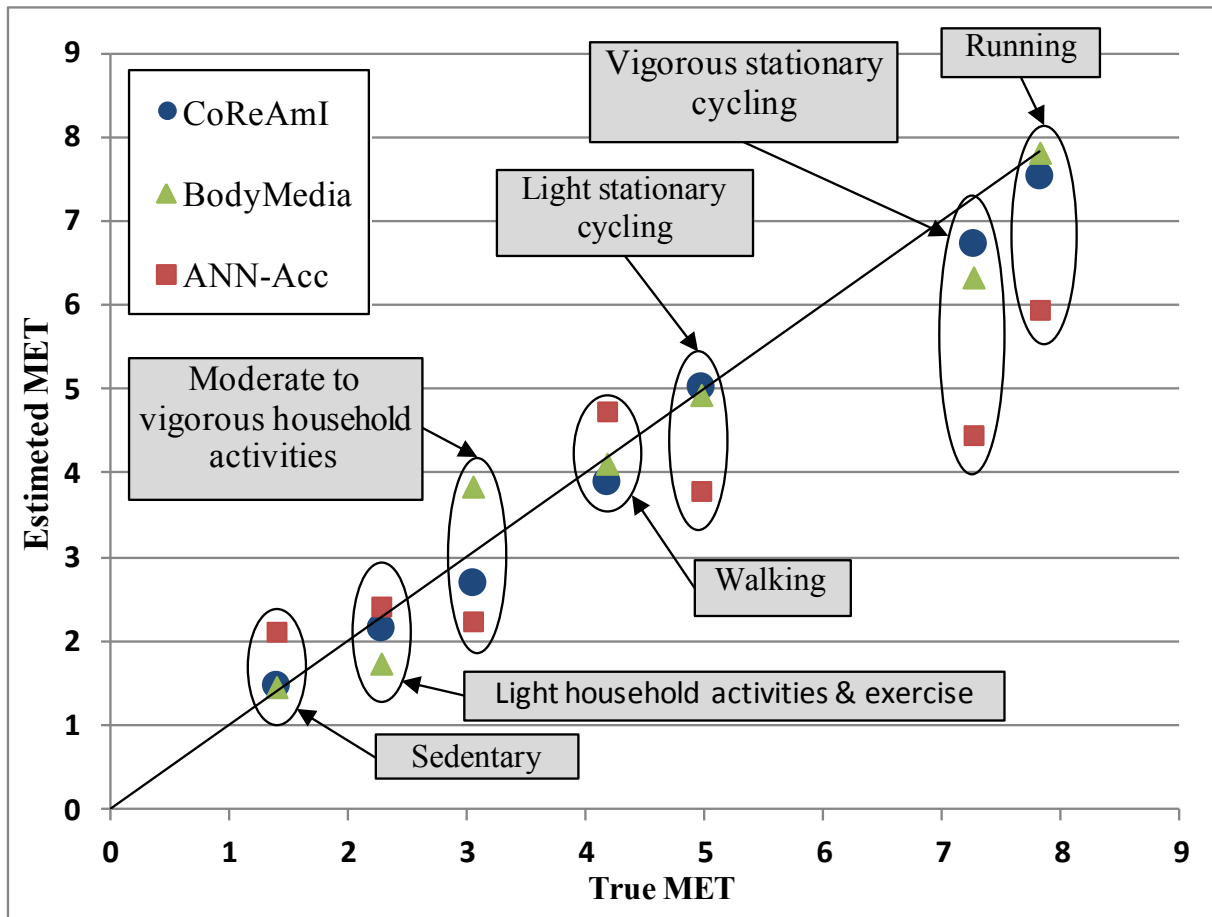


Figure 5-5. Measured and estimated METs for different types of activities using CoReAmI approach, output of the BodyMedia sensor and ANN trained on chest-accelerometer data only (ANN-Acc) approach [154][155].

The comparison in Figure 5-5 has a drawback because it averages the estimated EE over one type of activity, and this way it allows the errors of the methods (underestimations and overestimations) to cancel each other. For this reason we further analyzed the performance using the MAE and RMSE. Table 5-5 presents the results for CoReAmI, BodyMedia and ANN-Acc EE estimations for all the activities and for different activity types individually. When calculated for all the activities, CoReAmI has significantly lower MAE and RMSE compared to the BodyMedia and ANN-Acc. Per-activity analysis shows that CoReAmI also has significantly lower RMSE for all activity types compared to the BodyMedia and ANN-Acc approach: on average 0.45 and 0.94 lower than the BodyMedia and ANN-Acc approach, respectively. The MAE analysis shows that CoReAmI has significantly lower MAE for all the activity types if compared to the ANN-Acc and to the BodyMedia's estimations. The difference in performance of ANN-Acc compared to CoReAmI and BodyMedia additionally confirms that acceleration information is not sufficient for accurate EE estimation. These results are in accordance to the findings of Lester et al. [77], Liu et al. [158], and Vyas et al. [78], who showed that by using multiple sensors one should overcome this problem.

Table 5-5. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for the CoReAmI's EE estimation compared to the BodyMedia sensor and the ANN-Acc regression model. The best performance for each activity type is marked with bold style.

	Activities	CoReAmI	BodyMedia	ANN-Acc
RMSE	Overall activities	0.825	1.326	1.763
	Sedentary	0.410	0.490	0.950
	Light HH & games:	0.630	1.000	0.960
	Mod-Vig HH & sports	0.880	1.560	1.140
	Walking	0.770	0.830	0.960
	Cycling light	0.670	1.010	1.730
	Cycling vigorously	0.940	1.290	3.800
	Running	0.970	1.760	3.110
MAE	Overall activities	0.601	0.848	1.266
	Sedentary	0.410	0.490	0.950
	Light HH & games:	0.630	1.000	0.960
	Mod-Vig HH & sports	0.880	1.560	1.140
	Walking	0.770	0.830	0.960
	Cycling light	0.670	1.010	1.730
	Cycling vigorously	0.940	1.290	3.800
	Running	0.970	1.760	3.110

5.5 Summary and Discussion

This chapter presented the usage of CoReAmI for estimation of the EE. The CoReAmI approach extracted eight contexts and applied used context-based regression ensemble using the sensors data the following sensors: two accelerometers, Zephyr (heart rate, breath rate, and chest skin temperature), BodyMedia (GSR, arm skin temperature, ambient temperature). The results showed that on average, CoReAmI significantly outperforms the competing approaches.

Our findings are consistent with those of Crouter et al. [76], who showed that single regression models cannot accurately estimate the EE over a range of activities, and that using multiple models based on the context (in their case - activity) significantly improves the EE estimation. We went a step further and used not only the activity as a context, but multiple contexts (heart rate, breath rate, GSR, etc.), resulting in an ensemble of models invoked for the contexts in which the user is at the particular moment.

The general rule of preferring ensemble methods over single-regression approaches was justified with our results. That is, our context-based ensembles train multiple weak learners, and by combining the learner's outputs they create a stronger and more robust model.

Some may argue that the improvement is not worth the trouble of introducing such a complex approach compared to the single-regression approaches. However, once the context structure is defined and the models are trained, the usage in practice is relatively simple and requires relatively low computational power. In general, the computational cost of constructing an ensemble is not much larger than creating a single regression model [165]. The results show

that the difference in the errors – if they do not cancel each other out – can amount to several hundred calories per day. This is probably the most valuable for people who are particularly interested in precisely matching the caloric intake and output. This interest can have many reasons – engaging in certain sports or calorie restriction lifestyle, suffering from diabetes etc.

The CoReAmI approach has a number of strengths when used for EE estimation. First, the novel reasoning with the use of multiple contexts enables a more robust and more context-specific EE compared to the existing solutions. The resultant EE estimations were more accurate than conventional single-regression approaches (linear models, non-linear regression models, ANN-Acc), conventional ensemble approaches (Bagging and Random Subspace), and the more advanced multi-sensor BodyMedia device. Second, we showed that using the activity of the user as the only context is inferior to multiple contexts in terms of accuracy. Next, our methodology is independent of the ML algorithm used for training; therefore, an arbitrary algorithm can be used. This may be beneficial if the processing power of the device is limited and requires implementation of simple algorithms such as linear regression.

There are also few limitations that warrant consideration. First, since it is not easy to obtain valid, multi-sensor measures of EE, our method was developed using data from a limited number of people in controlled activity trials. Consequently, additional research is needed to evaluate the validity of the context-based approach under free-living conditions. This issue is also present when we compare CoReAmI to the MET output of the BodyMedia sensor, whose EE estimation model is trained on a scenario different from ours. However, this is not the case when we compare to the ANN-Acc approach, where the model was trained and tested the same way as for CoReAmI.

As a part of our current and future work, we are working on a multi-sensor prototype device [166][167] in collaboration with the Department of Communication systems at the Jožef Stefan Institute. This device should include several sensors inside a single enclosure, and should be able to extract the user's: accelerations, ECG [168][169], heart rate, breath rate[170], temperature, GSR and similar. We plan to analyze this data by applying the CoReAmI approach not only for energy expenditure estimation, but also for estimations and reasoning about the user's health in general: analysis of heart-related parameters, gait analysis, etc.

6 Fall Detection Domain

The third problem domain on which we applied the CoReAmI approach is the fall detection (FD). FD is a really important application in AmI because falls are among the most critical health problems for the elderly [171]. Approximately 30% of people over the age of 65 fall each year, and this proportion increases to 40% in those aged more than 70 [172]. About 20% of the elderly who fall require medical attention [173]. Furthermore, falls and the fear of falling are important reasons for nursing-home admission [174]. Falls are particularly critical when the elderly person is injured and cannot call for help. These reasons, combined with the increasing accessibility and miniaturization of sensors and microprocessors, are driving the development of fall-detection systems.

Even though FD has received significant attention in recent years, it still represents a challenging task for two reasons. First, most of the current approaches use accelerometers and define a fall as having greater accelerations than normal daily activities. However, since there are several everyday fall-like with high acceleration, such as sitting quickly or lying down quickly, focusing only on a high acceleration can result in many false alarms. Second, not all falls are characterized by a high acceleration. Rubenstein et al. [175] showed that 22% of the falls experienced by the elderly are slow and are caused by dizziness and vertigo (13%), and drop attacks (9%). Therefore, the detection of slow falls should be an intrinsic part of a fall-detection system.

In this chapter we present the application of the CoReAmI approach to the task of FD. The chapter is organized as follows. First, we present the related work in EE estimation. Then, we present the approach itself, including the sensor equipment, context extraction phase and the context-based EE estimation. In the next two sections the experimental setup and the results are presented. In the final section, a discussion and directions for future work are provided.

6.1 Related Work

Similar to AR, FD approaches can also be divided into those using non-wearable (i.e., ambient) and wearable sensors.

Non-wearable sensors

The most common non-wearable approach is camera-based [176][177]. Although this approach is physically less obtrusive to the user compared to the wearable sensors, it suffers from issues such as low image resolution, target occlusion and time-consuming processing. Probably the biggest issue is the user privacy: the user has to accept that a camera will record him/her.

In recent years, studies that use sound and vibration non-wearable sensors are gaining attraction. However, these sensors proved to be efficient only when combined with other sensors, especially the wearable inertial sensors [178]. We are considering these sensors as

future addition to our system, since additional sensing modalities would enhance the context-based reasoning.

Another approach using non-wearable sensors was proposed by Botía et al. [179] and Muñoz et al. [180]. Their system was able to detect most of the alarming situations using three types of sensors: infrared motion sensors, pressure sensors and main door open detector. In the first study, by Botía et al., the authors mainly focused on finding the best time intervals which should be considered in order to raise an alarm. In the second study, by Muñoz et al., the authors proposed an alert management tool for supporting the caregivers in their task of monitoring and validating alerts. The focus of this study is not the accuracy of detection of alarming situation, but the proposed support tool which enables caregivers to easily confirm or dismiss a potential alarming situation. In both cases, they showed that in the case of multiple persons, their system is prone to false alarms, which is an important disadvantage of the systems that use only non-wearable sensors.

Wearable sensors

Most of the studies that use wearable sensors are based on inertial sensors. Usually, they are focused only on fast falls [181][182], which are not difficult to detect using the acceleration signal. The non-fall events used to test for false positives are usually normal, everyday activities [183][177], not events chosen specifically because they are easily mistaken for falls. In contrast, we used complex falls and every-day events that appear like falls. An example where FD was evaluated on events difficult to recognize as falls or non-falls is the work by Li et al. [184]. By applying thresholds to two inertial sensors, they detected a fall with 90.1% accuracy. The recall value of their method on a fall event ending with sitting was 50% and for a non-fall event, quickly lying on a bed, was 40%. By combining one inertial and location sensor, we were able to achieve 99% and 100%, on similar events, respectively.

A combination of inertial and location sensors was described in Zinnen et al. [185]. However, their goal was AR for car-quality control and not FD. Their approach was based on high-level primitives that were derived from a reconstructed human-body model by using inertial sensor data. The location data was mainly used to estimate the person's location near the car. In our approach, beside the location of the user in the apartment, the location features were also used for the recognition of the user's activity.

A context-based approach to FD is presented in the study by Li et al. [79]. However, they used a different fall-detection method and different types of sensors to extract the context information, compared to our approach. In particular, they used five wearable accelerometers and two ambient sensors that monitored the vibration of the furniture. They combined the user's posture information extracted from the accelerometers, and the context information extracted from the environmental sensors, in order to detect the fall situations. Although they also analyzed slow falls and fall-like situations, their evaluation was performed on only three test subjects, while we tested our method on 11 subjects. The advantage of our location system, compared to the environmental sensors, is that it provides richer information about the user's situation, e.g., the user's location, the sensor's height, etc. The environmental sensors used in their research can only inform about the presence/absence of the user at a specific location where the sensor is installed. We tested all the combinations of 10 sensors and found a satisfactory performance with single sensor enclosure, while they analyzed only the fixed five accelerometer placements on the body.

In general, FD approaches that also exploit the activity of the user tend to be more successful than those relying on high acceleration only. Most of them try to recognize if the

user is lying after a potential fall trigger (e.g., high acceleration) [184][186]. Others recognize the fall as one of several elementary activities [178][183][181]. There are also some that use the activity information as input to the FD. For instance, to recognize a fall, Sixsmith et al. [187] and Naranjo et al. [188] used two and four levels of activeness, respectively. In this study, the user's activity is one of the four contexts which are extracted and used by CoReAmI.

6.2 Fall Detection with CoReAmI

To overcome the problems of the existing fall-detection methods discussed in the previous subsection, the CoReAmI approach was adapted to detect falls using wearable inertial and location sensors (shown in Figure 6-1). The approach uses context information from the both types of sensors to determine whether a fall has occurred. In the context extraction phase four contexts are extracted: the activity (extracted from both types of sensors), body accelerations (extracted from the inertial sensors), acceleration fall pattern (extracted from the inertial sensors) and location (extracted from the location sensors). Then, in the context modeling phase, for each context value a model is constructed using the other three sources of information. The modeling is performed using expert rules which describe a fall situation. In the final phase, a data instance is evaluated by multiple expert rules, which are invoked according to the context values. The decisions of each model are then aggregated by majority voting.

To explain the basic principle of the context-based reasoning, let us consider the following example in which a user is lying on a bed, i.e., a non-fall situation. If the activity of the user is considered as context, it is lying; therefore it is a possible fall situation. In this case, the rule uses the body movements, the acceleration fall pattern and the location of the user to model the situation. Because the location in the current case is the bed, the decision is – no fall. The same situation is considered by using the location and the body movements as a context. The final decision is presented by aggregating the decisions given by each of the models.

6.2.1 Sensors

The sensor equipment consists of inertial and location sensors (Figure 6-2). These types of sensors were chosen because inertial sensors are cheap and portable, and location sensors provide rich information about the user without significantly compromising the user's privacy.

Six inertial sensors were placed on the chest, waist, left thigh, right thigh, left ankle and right ankle (non-filled circles in Figure 6-2). Since only activities that are associated with the user's legs and torso were studied, the arm- and wrist-sensor placements were not considered. We used Xsens-MTx inertial sensors [189], but the methods developed in this research are general and can be applied to any type of inertial sensor. The Xsens-MTx is a complete MEMS inertial measurement unit (IMU) with an integrated 3-axis accelerometer and a 3-axis gyroscope. The accelerometer data is already explained in Subsection 4.2.1. The gyroscope is a device that measures the angular velocity, thus allows estimation of the device's orientation and rotation. Each of the Xsens-MTx sensors is connected to a Xbus Master unit, which synchronizes the data and sends it as a single data sample including measurements from all the sensors to a PC. Additionally, the Xbus Master supplies the Xsens-MTx's with power. This unit is the main communication centre that collects the data from the sensors and is connected to a PC through a USB cable or Bluetooth (BT) wireless communication. In our case we used a BT, because we needed the person to be able to move freely around the room.

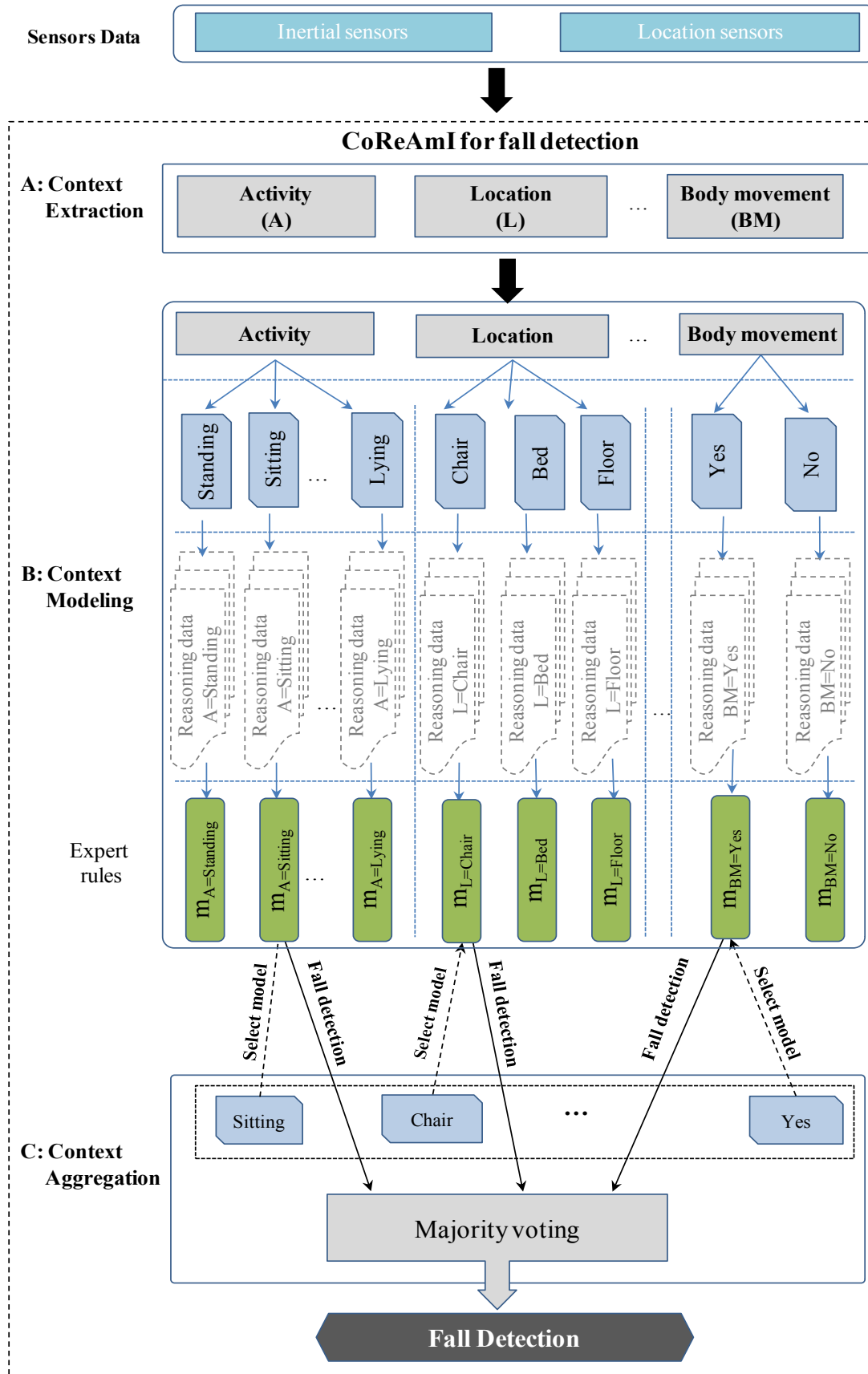


Figure 6-1. CoReAmI approach for fall detection.

Four location tags were placed on the chest, waist, left and right ankle (filled white circles in Figure 6-2). They emit UWB radio signals, which are detected by sensors fixed in the corners of a room, and their coordinates are computed. The location system used in this study is Ubisense [190]; it is a real-time location system (RTLS) used to track subjects indoors. The tags that were used in our research are *Series 7000 Compact Tags*. One tag is a small device that, when attached to something or worn by a person, allows them to be located to an accuracy of 15 cm (and sometimes to 1 m) in 3D in real-time. In addition, it includes additional features such as a LED for easy identification, a motion detector (to activate a stationary tag) and a click button to trigger events. In our case we used the button to activate the tags from sleep mode. Note that for simplicity the term sensor is also used for the wearable location tag. The low electricity consumption and power-management techniques result in long battery lifetime. In our application, the tag updates its position ten times a second and battery lifetime is over one month.

The data-sampling frequency of the location sensors was set to 10 Hz because of the Ubisense's hardware limitations. Although the inertial sensors do not have the same limitation, the data is sampled at the same frequency to simplify the synchronization.

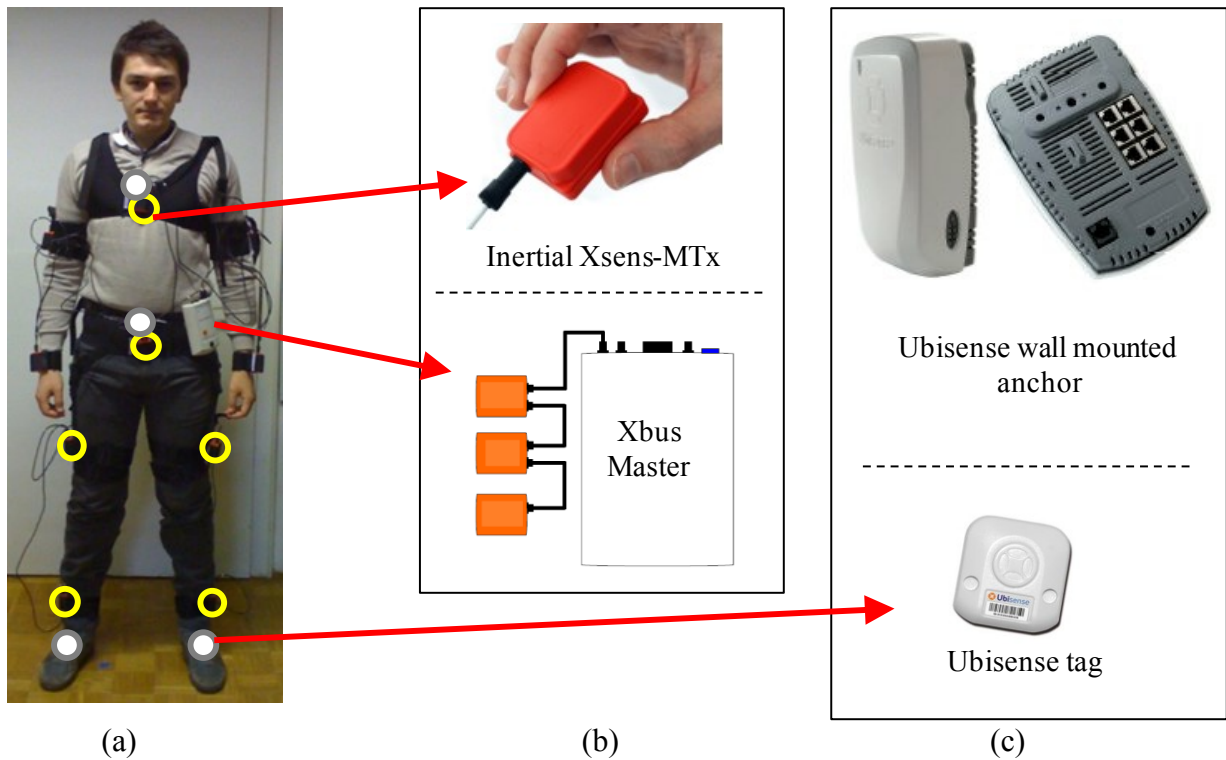


Figure 6-2. Sensor equipment.

- (a) User wearing the equipment: inertial sensors (non-filled circles) and location tags (filled circles);
- (b) The inertial sensors equipment: Xsens-MTx and the Xbus Master;
- (c) The location sensors equipment: Ubisense wall mounted anchors and Ubisense wearable tag.

6.2.2 Context Extraction

In order to extract each of the contexts, first the data is preprocessed. An **inertial sensor** provides raw data that consists of accelerations (from an accelerometer) and angular velocities (from a gyroscope) along three perpendicular axes. The raw data was filtered with low-pass and high-pass filters as described in Subsection 4.2.2.

The UbiSense output consists of the 3D coordinates of the sensors that are attached to the user's body. In a typical open environment, the localization accuracy is about 15 cm, but in practice it may occasionally drop to 1 m or more. Therefore, filtering was performed in order to tackle the problems with the UbiSense system [191]. First, a median filter computed each coordinate as the median of the measured values in a time window. This type of filtering removes large, short-term deviations of a measured coordinate from the true one. Second, the coordinates were corrected with a filter enforcing anatomic constraints based on the user's height and the body proportions. After that, a Kalman filter was used to smooth the data.

The following four contexts were extracted from the preprocessed inertial and location data: (i) acceleration fall pattern, (ii) the user's body movement, (iii) the user's location, and (iv) the user's activity.

The **acceleration fall pattern** (AFP) was used as one of the contexts, as well as a baseline for comparison. The rationale for this method was that the acceleration pattern during a typical fast uncontrolled fall (shown in Figure 6-3) is a decrease in the acceleration (free fall) followed by a rapid increase (impact with the ground). For our implementation of the AFP, the difference between the maximum and minimum accelerations within a one-second window was calculated. If the difference exceeded the threshold and the maximum appeared after the minimum, a fall was declared. The threshold was chosen empirically based on preliminary data [17].

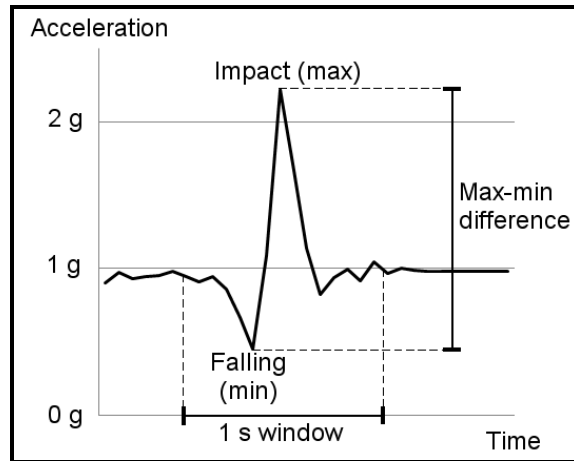


Figure 6-3. Acceleration pattern during a fall.

During motion the accelerometers produce a changing acceleration signal, and the fiercer the motion, the greater the change in the signal. Using these changes a feature was extracted: Acceleration Vector Changes (AVC) in order to determine the user's **body movements** [20]. This feature sums up the differences between consecutive values of the lengths of the acceleration vector, and divides the sum by the time interval (one second):

$$AVC = \frac{\sum_{i=1}^q |length_i - length_{i-1}|}{T_s - T_0} \quad (24)$$

q is the number of data samples, T_0 is the timestamp for the first data sample in the window, and T_s is the timestamp of the last data sample. By applying a threshold to the AVC value, the movement of a sensor is detected.

The **location** of the user was provided by the location system, which outputs the 3D coordinates of the location sensors that were attached to the user's body. This way it captured

the location of the user in the apartment and also the height of each sensor. Even though the location sensors provide relatively rich information about the user locations during the day, only the user's presence/absence in locations such as the bed, chair and floor was relevant for the FD. Further and more thorough analysis of the location data is not part of this research.

To recognize the **activities** of the user, ML was used. The idea of the ML approach was to learn a classification model that will be able to classify the target activities of the person wearing the sensors. The first step in the ML-based AR is the feature extraction procedure. Therefore, using a sliding-window technique the data from both types of sensors was first transformed into 30 features (25 from the inertial and five from the location sensors). The whole list of features can be found in Appendix B. Once the features are extracted, the feature vector was fed into the classification model, which recognized the activity of the user. The ML analysis was performed using the API of the software toolkit WEKA [109]. Among several methods tested, Random Forest (RF) yielded the best results in preliminary tests [20][21]. RF is an ensemble of Decision Trees in which the final decision is formed by a majority vote of the tree models [96].

Seven basic (atomic) activities that can also be interpreted as body postures were studied: standing, sitting, lying, sitting unusual (e.g., sitting on the ground), on all fours, going down and standing up. We decided only for these activities because they are common, atomic, everyday-life activities and are also the most relevant for the detection of falls and distinguishing them from non-falls.

6.2.3 Context-based Fall Detection

Table 6-1 shows the contexts and their values. For each discrete value of each context, expert rules are defined. That means for the "standing" activity 12 experts rules are defined, i.e., for each combination of the context values of the other three contexts.

Table 6-1. Extracted contexts and their possible values.

A – activity, L – location, BM – body movement, AFP – acceleration fall pattern.

A		L	BM	AFP
standing	sitting unusual	bed		
sitting	on all fours	floor	yes	yes
lying	going down	chair	no	no
	standing up			

An example rule that describes a fall situation when the context of the user is "lying" activity is the following:

$$\text{IF } (BM = \text{"no"} \wedge L = \text{"floor"} \wedge AFP = \text{"no"}) \text{ THEN "fall"} \quad (25)$$

The output of each rule is binary, either "fall" or "non-fall". The rule states that in order to detect a fall the user should not move and the locations should not be the bed. The same situation is considered by using the location, AFP and the body movements as a context. The final decision is presented by aggregating the decisions given by each of the rules, i.e., by choosing the majority decision. That means, in order to detect fall, at least three decisions should be fall. The example rules that are triggered when the user is lying on the bed and not moving is shown Table 6-2. In the third case, a false fall is detected because the expert rule

describes a lying situation where the user is not moving. However, this decision is overruled by the other three and non-fall is detected.

Table 6-2. Contexts and expert rules when the user is lying on the bed not moving.

Context:	Expert rule:
$A = \text{"lying"}$	IF ($BM = \text{"no"} \wedge L = \text{"bed"} \wedge AFP = \text{"no"}$) THEN "non-fall"
$BM = \text{"no"}$	IF ($A = \text{"lying"} \wedge L = \text{"bed"} \wedge AFP = \text{"yes"}$) THEN "non-fall"
$L = \text{"bed"}$	IF ($A = \text{"lying"} \wedge BM = \text{"no"} \wedge AFP = \text{"no"}$) THEN "fall"
$AFP = \text{"no"}$	IF ($A = \text{"lying"} \wedge BM = \text{"no"} \wedge L = \text{"bed"}$) THEN "non-fall"

The time interval for the reasoning was selected to be 10 seconds after empirical analysis of the data. This way, the interval is long enough for a reliable recognition, but still negligible compared to the time needed for help to arrive. In general, this is a parameter that can influence the performance in real-life situation and therefore should be appropriately adjusted. However, its adjustment analysis is not part of this research and is considered for future work.

Because the activity is recognized on a 1-second interval, during the 10-second reasoning interval it may contain different values, e.g., five lying, three standing up and two sitting activities. In order to represent the whole interval with one context value, we empirically selected 80% as the minimum percentage of same values that a context should have (e.g., the activity should be lying 8 seconds out of 10 to satisfy a rule that requires lying). Otherwise, if the activity values are represented with a smaller percentage than 80%, none of the activities is chosen for the particular interval. Additionally, the AFP is checked every second and is instantaneous, happens at a specific moment in time and does not last. Therefore, if AFP contained "yes" value in any second of the 10-second reasoning interval, a "yes" value was considered for the whole interval.

Given the values of the context shown in Table 6-1, there are 14 values that can be used as a context and for which expert rules are defined. This results in 256 possible rules. However, because of the specificity of our task (binary output, strictly defined decision space), it turned out that the following rules cover all the falls analyzed (even without analyzing each of the contexts individually, but simply creating a single rule over all the contexts):

$$\text{IF } (A = \text{"lying"} \wedge BM = \text{"no"} \wedge L = \text{"floor"} \wedge (AFP = \text{"no"} \vee AFP = \text{"yes"})) \text{ THEN "fall"} \quad (26)$$

$$\text{IF } (A = \text{"sit unusual"} \wedge BM = \text{"no"} \wedge L = \text{"floor"} \wedge (AFP = \text{"no"} \vee AFP = \text{"yes"})) \text{ THEN "fall"} \quad (27)$$

$$\text{IF } (A = \text{"on all fours"} \wedge BM = \text{"no"} \wedge L = \text{"floor"} \wedge (AFP = \text{"no"} \vee AFP = \text{"yes"})) \text{ THEN "fall"} \quad (28)$$

We used assumptions that the elderly do not usually lie or sit on the ground and are not on all fours for more than ten seconds while not moving. In principle, once the general reasoning scheme is established, adding more contexts or context values, thus adding more rules for different situations is a relatively easy task.

As one can note from the three rules, the output of the AFP context is not important. The reason for this is that the AFP is covered by the lying, sitting unusual and on all fours activities. That is, in all fall events where AFP was triggered (positive output) the user was lying, sitting unusual or was on all fours afterwards.

As a comparison and also to check which sensor type is better in which situation, we present the fall detection rules, which are based only on one type of sensors: inertial or location.

In the case of inertial sensors, the fall is detected using the activity, the body accelerations and AFP context. Our previous experiments and also some related work [181][182] showed that it was possible to detect a straightforward (fast) fall by using only AFP; however, lots of false positives appeared in other fall-like events: quickly lying down on a bed, quickly sitting on a chair, etc. Therefore, a potential fall detected by AFP was confirmed by the body movement and additional context information, i.e., the user's activity. As an example, a fall situation is defined by each of the following rules:

$$\text{IF } (AFP = \text{"yes"} \wedge A = \text{"lying"} \wedge BM = \text{"no"}) \text{ THEN "fall"} \quad (29)$$

$$\text{IF } (A = \text{"sit unusual"} \wedge BM = \text{"no"} \wedge (AFP = \text{"no"} \vee AFP = \text{"yes"})) \text{ THEN "fall"} \quad (30)$$

$$\text{IF } (A = \text{"on all fours"} \wedge BM = \text{"no"} \wedge (AFP = \text{"no"} \vee AFP = \text{"yes"})) \text{ THEN "fall"} \quad (31)$$

When the CoReAmI is using only location sensors, the FD is based on the activity and the location. The first advantage compared to the stand-alone inertial FD was the location information: the system was aware of some predefined "safe" locations, such as the bed. The second advantage was the z coordinate of the sensor location, which provides the height of the sensor and therefore distinguishes different activities, for example, sitting on the floor from sitting on a chair. The empirical analysis and preliminary results showed that the location-based AR is not sufficient and the three activities (lying, sitting unusual and on all fours), which are important for FD, are not always recognized as such. Therefore, we combined those three activities and applied the following rule:

$$\text{IF } (A = \text{"lying"} \vee \text{"sit unusual"} \vee \text{"on all fours"}) \wedge L = \text{"floor"} \text{ THEN "fall"} \quad (32)$$

6.3 Experimental Setup

6.3.1 The Experimental Scenario

A complex, 15-minute test scenario was specifically designed to investigate events that might be difficult to recognize as falls or non-falls. This scenario, shown in Table 6-3, was created in consultation with a medical expert. In Table 6-3 the numbers in parentheses represent the event numbers for easier referencing throughout the text. The events were recorded in a single recording including all the events.

Because typical fast falls are easy to detect due to high acceleration, only one such fall (1) was included. Three atypical falls not involving high acceleration, i.e., (2), (3) and (4), were included to test the use of the contextual activity information, i.e., that a person is not expected to sit/lay on the ground (as opposed to the chair/bed). Furthermore, the two events (5) and (6) involve high acceleration and could thus be misclassified as falls by acceleration-based methods, such as AFP. However, the methods that use the activity and location as contextual information should be able to detect that these are non-fall events. An event (7) was included that involves voluntarily lying on the ground, which could mislead the methods that use information other than acceleration. The events (8), (9) and (10) are normal and were included to verify that all the methods work correctly during normal events.

Additionally all the target activities are contained in the scenario (see Table 6-3): standing, sitting, lying, sitting unusual, on all fours, going down and standing up.

Table 6-3. The events in the scenario, the appropriate activities and event descriptions.

	#	Event	Activities	Description
Fall Events	(1)	Fast fall (tripping)	Standing/walking, going down, lying, standing up	Falling performed in different ways: forwards, backwards or to the sides.
	(2)	Slow fall (fainting)	Standing/walking, going down, lying, standing up	Losing consciousness and slowly falling to the ground (trying to hold onto furniture).
	(3)	Falling when trying to stand up	Sitting, standing up, going down, sitting on the ground, standing up	Trying to stand up from a chair, but having difficulties and slowly falling to the ground, ending up in a sitting posture on the ground.
	(4)	Sliding from a chair	Sitting, standing up, going down, sitting on the ground	Sliding from a chair and ending up in sitting unusual on the ground.
Fall-like Events	(5)	Quickly lying down on a bed	Standing/walking, going down, lying, standing up	Quickly lying down on a bed.
	(6)	Quickly sitting down on a chair	Standing/walking, going down, sitting, standing up	Quickly sitting down on a chair.
	(7)	Searching for something on the ground	Standing/walking, going down, on all fours, lying,	Going on all fours and afterwards going to lying posture in order to take an object from the ground.
Normal Events	(8)	Sitting down	Standing/walking, going down, sitting, standing up	Sitting down on a chair normally.
	(9)	Lying down	Standing/walking, going down, lying, standing up	Lying down on a bed normally.
	(10)	Walking	Standing/walking	Walking sequences between events.

The experimental scenario was recorded with all six inertial and four location sensors. Afterwards, the approach was tested with all 1023 combinations of sensors (single type, as well as both types).

The scenario was recorded by 11 young healthy volunteers (24–33 years, seven males and four females). It was repeated five times by each person, resulting in 55 recordings and a total of 550 events for the FD and total number of 105 438 segmented samples for the AR. Testing

elderly people was not feasible because the scenario was too strenuous and risky for them, but the volunteers were advised how to act by the medical expert in order to mimic elderly. Additionally, the data for three more people was recorded for tuning the basic parameters, e.g., thresholds, preliminary tests and choosing the best algorithms.

6.3.2 Method Evaluation

To evaluate the FD, one must decide how to weigh the undetected falls and the false alarms. Both are important: not detecting a fall may endanger a person's health, while false alarms make the system unlikely to be used in real life. Therefore, we used F-measure (F1), which weights undetected falls and false alarms equally. It is defined as a harmonic mean of recall (the percentage of the events recognized as falls/non-falls from all the fall/non-fall events) and precision (the percentage of the events truly being falls/non-falls of all the events recognized as such) [109]. The mathematical definitions are the following, where Q is the type of the event (fall or non-fall).

$$recall = \frac{\text{Number of correctly detected events of type } Q}{\text{Number of all the events labeled as } Q}, \quad (33)$$

$$precision = \frac{\text{Number of correctly detected events of type } Q}{\text{Number of all the events detected as } Q}, \quad (34)$$

$$F1 = \frac{2 * recall * precision}{recall + precision}, \quad (35)$$

Because inertial sensors for FD are quite popular in the literature, we also tested the CoReAmI by using only inertial sensors (rules no. 29–31). Therefore, a fall situation was defined using the contexts extracted from the inertial sensors only, i.e., activity (recognized by a classification model using only inertial features), body movements and AFP. Similarly, we tested the CoReAmI approach by using only location sensors (rule no. 32). Therefore a fall situation was defined using the contexts extracted from the location sensors only, i.e., activity (recognized by a classification model using only location features) and location.

We also compared CoReAmI performance to a ML approach (MLA). The basic principle of MLA is that a ML model is trained to detect a fall event. In our case, features extracted from the chest-inertial and chest-location sensor data were used. Therefore, the contextual location information was implicitly (through features) introduced in the MLA. More details about the MLA can be found in our work [25].

Tests to confirm the statistical significance of the results were also performed. Because of the small number of folds (11) and because the individual samples are paired (the same person's data for each combination), we used paired Student's T-test with a significance level of 5%.

6.4 Results

Figure 6-4 presents a matrix (5×7) of the best sensor combinations for CoReAmI FD. The inertial sensors are shown on the horizontal axis and the location on the vertical axis. Each rectangle in the matrix contains the sensor placements and the achieved F-measure marked with F as a percentage. For example, the (2, 3) rectangle represents the combination of two

location and three inertial sensors. It is the best of all combinations according to the F-measure = 99.7%. The dotted lines (diagonal) connect the rectangles that have the same number of sensors. Along each dotted line the best (according to the F-measure) rectangle is marked with a darker red color. These rectangles represent the best combination given the number of sensors.

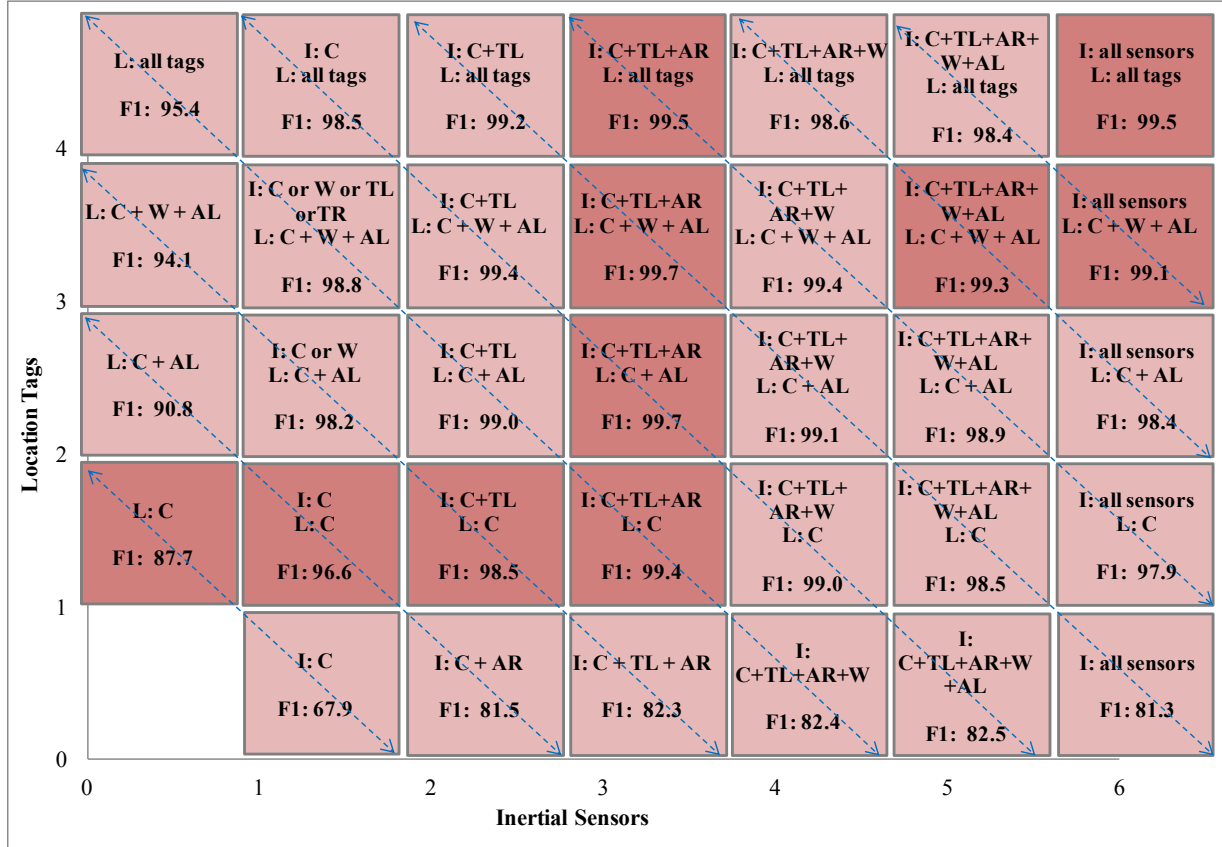


Figure 6-4. Matrix representation of the best sensor combinations using the Inertial (I) and Location (L) sensors. F - overall F-measure, C - Chest, W - Waist, AR - Ankle Right, AL - Ankle Left, TR - Thigh Right, TL - Thigh Left.

Another representation of the same results is shown in Figure 6-5. This is a 3D representation, where the third axis is the achieved F-measure.

Analyzing the results achieved with the inertial sensors alone (Figure 6-4 horizontal axis rectangles), one can see that the only important improvement is achieved when using two sensors instead of one. After this, adding up to five sensors did not significantly improve the F-measure; including a sixth sensor even decreased the performance.

For the location sensors, an increase in the number of sensors increases the performance all the way. The statistical tests showed that there is a significant difference in the performance of the system using one, two, three and four location sensors. Like with the inertial FD, the chest is the best-performing placement.

The statistical tests for the combined FD showed that the difference in performance is statistically significant only when the system is using two and three sensors. Four sensors or more do not significantly increase the performance of the system.

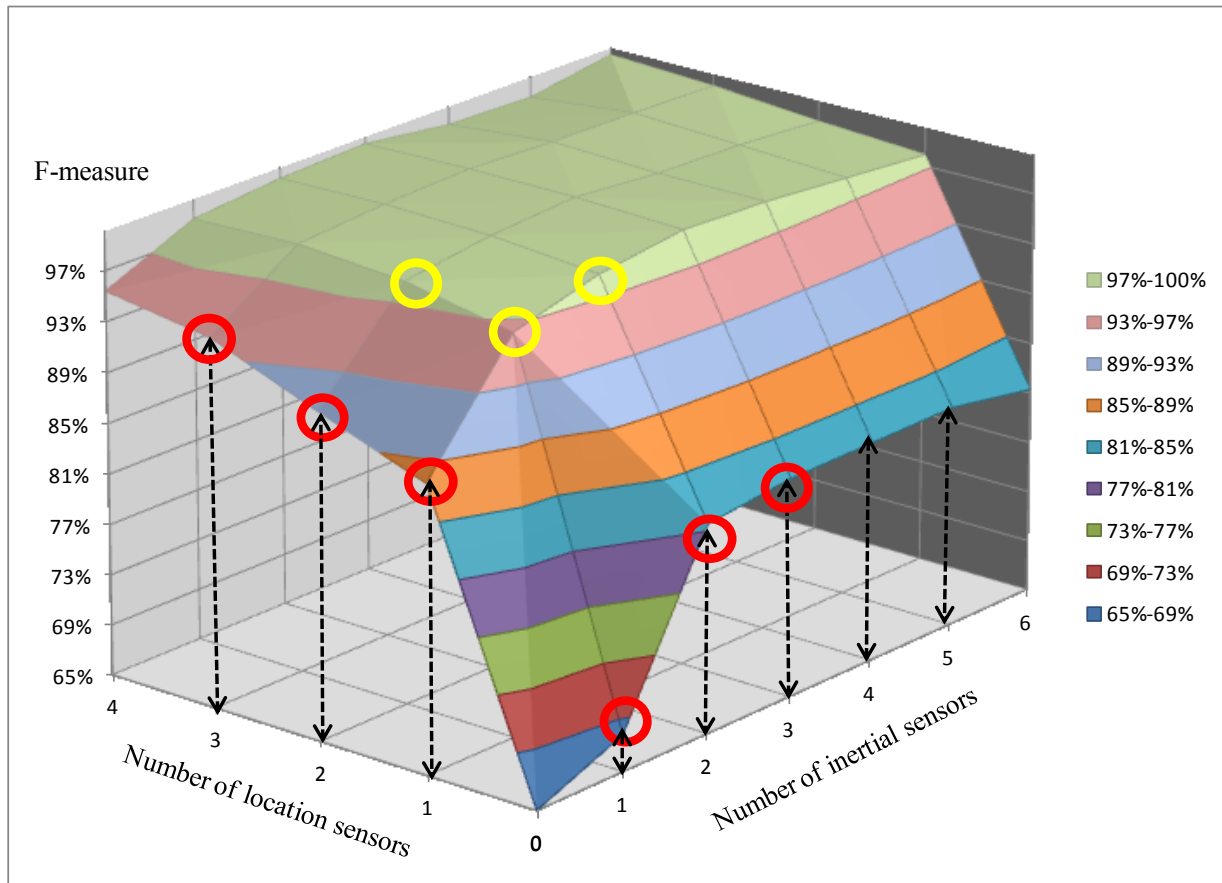


Figure 6-5. Best sensor combinations for CoReAmI fall detection.

The parts of the graph with a smaller number of sensors are of the greatest interest for practical usage (circles in Figure 6-5). The combination of sensors clearly outperforms the individual sensor types. For example, the performance values of the system using two sensors are 81.5% and 90.8%, for the inertial and location sensors, respectively. Their combination achieves 96.6%, an improvement of 15 p.p. and 6 p.p., respectively. This is the case for each number of sensors (dotted lines): the combination of two sensor types is better than each of the types used separately.

Since both sensor types can be put in the same enclosure, one can also examine the number of enclosures. Table 6-4 and Table 6-5 show the results when one and two sensor enclosures (equipped with the both sensor types) are analyzed, respectively.

Table 6-4. CoReAmI fall-detection analysis using only one sensor enclosure equipped with inertial and location sensor.

1-sensor enclosure (Inertial + Location)			
C	W	AL	AR
96.1%	95.6%	75.0%	72.1%

The performance of the system using only one sensor of one type is 68% and 88% for the inertial and location sensor, respectively. The results in Table 6-4 show that by combining them into one enclosure on the chest, the achieved F-measure is 96.6%, an improvement of 29 p.p. and 9 p.p., respectively. Furthermore, the combination of one inertial and one location

sensor placed on the chest outperforms each of the other sensor placement combinations: waist, left and right ankle.

The results in Table 6-5 show that the best performing 2-enclosures-placement (when both types of sensors are included in each enclosure) is the chest and left ankle achieving 98.3% performance. However, this is not the best performing 2-enclosures-placement, because the combination of one inertial and one location sensor on the chest and one inertial sensor in the left thigh (shown in Figure 6-4) achieves 98.5% performance. This shows that it is better to add inertial sensor alone on the thigh (98.5%) instead of adding both (inertial and location) to the left ankle (98.3%). The reason for this is the improvement in the AR module, which is greater when the thigh inertial sensor is introduced compared to the ankle inertial and location. The best performing combination of three sensor enclosures is chest (inertial and location), right ankle (inertial and location) and left thigh (inertial only).

Table 6-5. CoReAmI fall-detection analysis using only two sensor enclosures equipped with inertial and location sensor.

2-sensor enclosures (Inertial + Location)					
C+AL	C+AR	C+W	W+AR	W+AL	AR+AL
98.3%	98.0%	97.2%	97.6%	97.9%	83.7%

The rest of the discussion is a detailed analysis of the results achieved by the statistically significant simplest and the best combinations of the inertial-only, location-only and both types of sensors. The sensor types and placements are shown in Table 6-6 and the results are presented in Table 6-7. The events in Table 6-7 are divided into three groups: fall, non-fall (fall-like), and normal events. The numbers are the percentage of all fall/non-fall events being correctly recognized as fall/non-fall (true positive and negative rate). The last row represents the overall F-measure.

Table 6-6. CoReAmI for fall detection: the simplest and the best combinations of the inertial-only, location-only and both types of sensors.

	The simplest combination	The best combination
Inertial sensors	Chest	Chest + Right ankle
Location sensors	Chest	All four sensors
Combined sensors	Inertial: Chest Location: Chest	Inertial: Chest + Right ankle Location: Chest

The first two columns show the results achieved for the FD with inertial sensors. The first event in Table 6-7, tripping, is a typical fall that was recognized accurately because of the AFP rule. The second event, which is falling slowly, was difficult to recognize because of the low acceleration during this event. For this event, additional contextual information was necessary (e.g., the location of the user). The effect of the activity information of the user can be seen in the fall events that end with sitting unusual on the ground (events 3 and 4). In these cases the AR model correctly recognized sitting unusual on the ground. On the other hand, this has a negative impact on the performance when the sitting event is analyzed (events 5 and 8). In this case, the AR model was not accurate enough and recognized sitting unusual on the ground, resulting in a false positive. This issue was solved by including more sensors, which improved the AR method (e.g., the column Inertial-best).

The location sensors based FD was using the activity and the location information. Because of the location, it recognized all falls with high accuracy (events 1 to 4). However, some problems remained among the non-fall events, because of the relatively low accuracy of the AR model. Namely, sitting (events 5 and 8) and searching on the ground (event 6) were misclassified as sitting unusual on the ground or lying (on the ground), causing the system to detect a fall during the non-fall events. Improvements in the performance can be seen when the number of sensors is increased (the column Location-best), due to the improvements in the AR method.

The last two columns show the results achieved with the combination of both types of sensors and the contexts extracted from the both types of sensors. The improvements are clear in all of the events. The overall performance when two sensors (one inertial and one location) were used was 96.6%. Some problems only appeared among the non-fall events that ended with sitting (5 and 8) and the searching on the ground event (6). The reason lies in the AR method, which misrecognized the appropriate activities (sitting and on all fours). These problems were solved by including one more inertial sensor, which significantly improves the AR model and consequently the FD (the last column in Table 6-7).

Table 6-7. CoReAmI detailed fall detection results for each event.

		CoReAmI					
		Inertial (Activity + AFP + Movement)		Location (Activity + Location)		Combination (Activity + AFP + Movement + Location)	
		Simplest	Best	Simplest	Best	Simplest	Best
Fall Events	(1) Tripping – Quick falling	100%	100%	96%	100%	100%	100%
	(2) Fainting – Falling slowly	11%	11%	100%	100%	100%	100%
	(3) Falling from a chair slowly	68%	98%	95%	95%	99%	99%
	(4) Sliding from a chair	72%	99%	97%	97%	98%	99%
Non-Fall Fall-like Events	(5) Sit down quickly on a chair	55%	97%	75%	89%	91%	98%
	(6) Searching on the ground	85%	88%	25%	78%	80%	89%
	(7) Quickly lying down on a bed	34%	34%	100%	100%	100%	100%
Non-Fall Normal Events	(8) Sitting normally	68%	98%	80%	93%	93%	98%
	(9) Lying normally	100%	100%	100%	100%	100%	100%
	(10) Walking	97%	100%	92%	97%	100%	100%
Overall F-measure in %		67.9%	81.5%	87.7%	95.4%	96.6%	98.5%

Finally, two commonly used methods in the literature, the AFP approach and the machine-learning approach (MLA), were tested for comparison. The results are shown in Figure 6-6, by presenting the true-positive or true-negative rate for each fall or non-fall event, respectively. The AFP is described in Subsection 6.2.2. More details about the MLA can be found in our previous work (Luštrek et al. [25]). The basic principle of MLA is that a ML model is trained to detect a fall event. In our case, features extracted from the chest-inertial and chest-location

sensor data were used. Therefore, the contextual location information was implicitly (through features) introduced in the MLA.

The overall results showed that CoReAmI, in which the context is explicitly encoded with rules, outperformed the other two methods, which use: implicit context information (MLA) or only accelerations (AFP). The AFP outperformed CoReAmI only in two events (6, 8); however, this was due to the one-sided performance of the AFP (detects only high accelerations) at the expense of the overall performance.

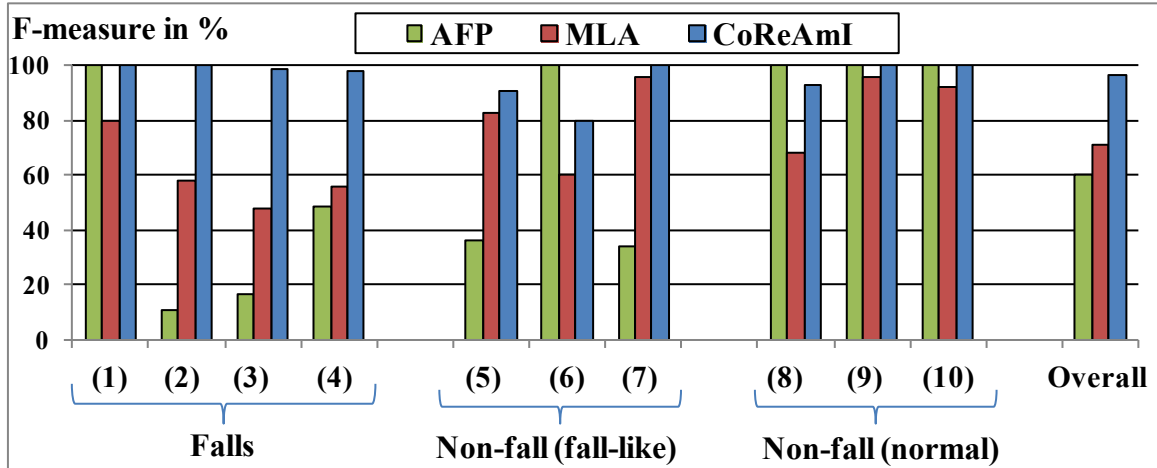


Figure 6-6. Comparison of the fall detection results achieved by our CoReAmI approach, the Machine-learning approach (MLA), and Acceleration Fall Pattern (AFP) approach. The event numbers correspond to the events given in Table 6-3.

6.5 Summary and Discussion

We presented a usage of the CoReAmI approach for fall detection which combines inertial and location sensors. The method exploits four contexts to detect a fall situation: the activity of the user, the body accelerations, the acceleration fall pattern and the location. Each context has multiple context values, which are used to construct expert rules.

Once we established the general reasoning scheme, adding more rules for different situations is an easy task. Also the addition of new sensors, such as sound and vibration, is relatively easy from the reasoning point of view; just adding a context and including it in the rules. Currently, the expert rules were designed manually. The automation of learning the best context relations is considered for future work.

The set of rules constructed when the two types of sensors are used (equation no. 26, 27 and 28) show that the output of the AFP context is not important. The reason for this is that the AFP is covered by the lying, sitting unusual and on all fours activities, which are detected after the instantaneous AFP. That is, in all fall events where AFP was triggered (positive output) the user was lying, sitting unusual or was on all fours afterwards. In these cases the fall was detected due to the detection of the activity and the other contexts (excluding the AFP). However, when only inertial sensors were used (equation no. 29) the AFP was an important trigger for a fall situation and was used in the rule.

We tested the performance with all possible combinations of the six inertial and four location sensors to find the best sensor placements, using the CoReAmI approach. The evaluation was performed on a complex test scenario, which included real-life, realistic events

that are difficult to recognize as falls or non-falls. The results showed that by combining the two types of sensors it is possible to detect complex fall situations by using the activity and the context information from both types of sensors. It is essential that both sensor types are employed, since they provide complementary information about the user's situation. Finally, the best practical solution proved to be the chest placement with a single sensor enclosure including one inertial and one location sensor achieving 96.6% for the fall detection employed in CoReAmI and 93.3% for the activity-recognition task only.

Additional analysis shows that, given the specificity of the problem (binary output, strictly defined decision space with a single context value in the reasoning interval), it was easier and sufficient to include the context inside the rules and construct a rule for the whole decision space (including all the contexts at once in a single rule), e.g., equation no. 26–32. This way, the CoReAmI approach was significantly simplified, i.e., the context modeling phase was represented by constructing several rules using all the available information (contexts), and the context aggregation phase was not needed (because only a single rule is triggered for each reasoning interval). This resulted in relatively faster and efficient reasoning, which allowed us to test thousands of sensor combinations in a reasonable time. To summarize, even though this modification significantly simplified the CoReAmI, the achieved performance showed us that carefully defining and extracting useful context is also an important aspect and sometimes sufficient for achieving high performance in the reasoning task.

For the future work, we plan to test CoReAmI for FD in real-life situations. This can be achieved by employing the system in elderly homes and monitoring their activities and events. Additionally, the interaction between the user and the system can be improved. This can be achieved by including the user's smartphone, tablet or PC as a medium for showing system's notifications (fall detected, system malfunction, etc.).

7 Conclusions and Future Directions

This thesis has addressed the problem of combining sensors data and reasoning in ambient intelligence domain by using multiple contexts. We have proposed a context-based approach called CoReAmI, which extracts multiple contexts from sensor data and reasons about the user by constructing multiple models for each context individually. This provides a multi-view perspective that makes more accurate reasoning. The approach is presented formally through definitions, algorithms, and flowcharts.

The central scientific hypothesis of the thesis was confirmed experimentally. The hypothesis states that extracting multiple sources of information and combining them with context-based approach (that is, using each source of information as a context) can lead to better reasoning performance than conventional approaches in the ambient intelligence domain.

To examine the validity of the hypothesis, we applied the CoReAmI approach on three problem domains in ambient intelligence: activity recognition, energy expenditure estimation, and fall detection. The results showed that CoReAmI significantly outperforms the competing approaches in each of the domains.

For the first two domains, machine learning methods were used to model each of the contexts, resulting in a context-based ensemble of classification (activity recognition) or regression (energy expenditure estimation) models. Our context-based ensemble not only exploited the complementarity of multiple models (as most other conventional ensemble approaches do), but also contained models that tend to be more accurate for a particular context than those trained on the entire training set. This is because each model was trained on a subset of the training set that is more homogeneous than the whole set, and used in the context of this subset; that is, to reason about samples similar to the ones in the subset. In other words, CoReAmI semantically splits the dataset into meaningful viewpoints (contexts) without using statistics about the data, as most of the conventional ensemble-based algorithms do (such as Bagging and Random Subspaces). Once a data instance has been evaluated by the context-based ensemble, the outputs of the context models are aggregated and the final decision is provided. This technique enabled us to take advantage of the general rule in ensemble-learning, which states that it is better to find a good aggregation function than to choose the best single model because it achieves stronger generalization [86][113][114].

For the third domain, expert rules were used to model the fall event by using contexts extracted from wearable inertial and location sensors. The empirical analysis of the data and the specificity of the problem (binary output, strictly defined decision space with a single context value in the reasoning interval) showed that it is sufficient to model the fall event by using the whole available information at once; that is, without the modeling and aggregation phases. Even though this modification significantly simplified the CoReAmI, the achieved performance showed that carefully defining and extracting useful context is also an important step and is sometimes sufficient for achieving high performance in the reasoning task.

The presented CoReAmI approach is fairly general and can be applied to many ambient intelligence problems for which the available information can be presented by multiple contexts. Probably the biggest limitation is that considerable human effort is

needed to present the context information appropriately (context extraction phase). However, if the contexts are already defined (which is often the case in machine learning tasks, for example by features extracted from the sensors data), one can continue with CoReAmI by adapting the context modeling and aggregation phases and significantly reducing the adaptation time.

7.1 Scientific Contributions

This thesis generated the following original contributions:

1. **A new, general, context-based reasoning approach in ambient intelligence, called CoReAmI.** The approach extracts multiple contexts from sensor data and performs reasoning about users using multiple models constructed for each of the contexts individually. The multiple models are constructed on subsets of the whole dataset created by partitioning the dataset by using each context value. That is, a particular subset contains data instances which correspond to a particular context value. When evaluating a data instance, the reasoning is performed by multiple models, invoked according to the current context of the user. Accordingly, multiple context-based views of the data are considered when making the final decision.
2. **Applying CoReAmI to three ambient intelligence problem domains, which resulted in:**
 - 2.1. **CoReAmI for activity recognition.** We studied the state-of-the-art approaches in activity recognition and observed that it is almost impossible to distinguish standing from sitting activity using a single accelerometer placed on the torso. We have managed to significantly improve the recognition of these two activities by adapting the CoReAmI approach for the activity-recognition problem domain.
 - 2.2. **CoReAmI for energy expenditure estimation.** We adapted and applied the CoReAmI approach to estimate the human energy expenditure. Our multi-context approach significantly improved the estimation performance compared to conventional approaches and approaches that are based on single context (such as the activity of the user). Additionally, the CoReAmI provided better energy expenditure estimations than the BodyMedia device, which is a state-of-the-art commercial device for energy expenditure estimation.
 - 2.3. **CoReAmI for fall detection.** We adapted and applied the CoReAmI approach to detect human falls. CoReAmI for fall detection significantly improved the detection performance compared to conventional approaches, such as: threshold-based approaches and approaches that are based only on machine learning.
3. **A novel method for context-based partitioning of a dataset** into multiple subsets and this way creating multiple views on the data by using each feature as a context.
4. **Contribution to the ambient intelligence community by preparing several ambient intelligence datasets** for human activity recognition, energy expenditure estimation and fall detection (some of which are already available at: <http://dis.ijs.si/ami-repository/>; others are to be made available in the near future).

7.2 Future Directions

The following list summarizes the future work of the CoReAmI approach:

- **Optimizing the hyper-parameters.** In the current version of CoReAmI, the tested machine learning algorithms (classification and regression) are used with the default algorithm parameters (hyper-parameters) as defined in WEKA. For future implementations, we plan to use the approach that finds the best combination of parameters, such as the Auto Weka software toolkit [192].
- **Context-based ensembles.** When machine learning algorithms are used to learn the context models, the CoReAmI modeling and aggregation phases are variants of the ensemble-learning approach. We plan to further define and release the CoReAmI context-based modeling and aggregation phase (without the context extraction phase) as an ensemble-learning algorithm for general purpose machine learning, such as Random Forest. We will also consider providing it as part of the WEKA toolkit.
- **Directions of how to use CoReAmI on a new problem domain.** We plan to simplify and define the entire process of applying and adapting the CoReAmI approach on a new domain. To achieve this, we plan to release the code, its documentation, and appropriate sample applications.
- **Dealing with missing context values.** The current version assumes that each context value is available for the reasoning approach. However, this is not the case in real life, where sensor malfunctions, and therefore missing sensor data, are common. We plan to adapt techniques that will deal with missing sensor data similar to the ones that have already been developed for the machine learning approaches, such as interpolation and expectation-maximization.
- **Dealing with redundant or similar context information.** The idea of this technique is to select only the most relevant contexts that contain unique information and to remove the ones that are redundant. Similar techniques have already been developed in machine learning. Their goal is to perform feature selection and remove features that are redundant or do not give any useful information. We believe that similar techniques can be applied to our approach.
- **Context grouping.** This idea suggests that multiple contexts can be grouped to form a new complex context. In the energy expenditure domain, for example, the heart rate and breath rate can be combined into a single context. However, when multiple contexts with multiple values are combined, the training (reasoning) data is significantly reduced (the reasoning data is the subset that contains the combined context values instead of one). Therefore, this technique requires a lot of data in order to exploit its potential.
- **Temporal reasoning.** In general, the time-order of the sensor information is important in ambient intelligence systems. In the current implementations of CoReAmI, we implicitly included the temporal component in the first and the second phase of CoReAmI: context extraction and context modeling phase. In the AR domain the temporal component was included by extracting contexts before and after the current activity. In the FD domain the temporal component was included in the expert rules, in particular for the rule that uses only inertial sensors (no. 29). In this rule it was important to know whether the user is lying down after or before an APF. Even though with these two examples we showed how the time can be included in

CoReAmI, we believe that this is not enough and more advanced approaches may be considered, such as event calculus (a state of the art temporal reasoning approach) [112].

- **Interval-based reasoning.** The current version of CoReAmI reasons about the user using a point-based approach; that is, the reasoning interval is analyzed as a point in time. Interval-based reasoning has attracted increased attention in recent years and provides a perspective that cannot be expressed by the point-based approaches [193]. We consider the interval-based analysis for future work.

8 Acknowledgements

First of all, I would like to thank my supervisor Prof. Dr. Matjaž Gams and co-supervisor Dr. Mitja Luštrek, who accepted me as a PhD student at the Jožef Stefan Institute (JSI). I am grateful for all the support and guidance that they have given me during my studies. This thesis would not have been possible without them and without the encouragement they have given me over the last four years I spent at JSI.

I would like to thank the members of my PhD committee for carefully reading the thesis and providing useful feedback: Marko Bohanec, Roman Trobec and Juan Carlos Augusto. I am grateful for their precise evaluation and for providing constructive comments that improved the quality of the thesis.

I wish to thank the Slovene Human Resources Development and Scholarship Fund who funded my research by enabling me a scholarship for PhD students in Slovenia from abroad.

I would also like to thank all my colleagues with whom I worked during my PhD studies. Especially Simon Kozina, with whom I have collaborated on several papers and created the RaReFall system which won the EvAAL activity recognition competition. Next, I would like to thank Boštjan Kaluža, who helped me with the initial idea of defining the context-based reasoning approach. Special thanks to my officemate Rok Piltaver, for making my time at the office more enjoyable. Also, I would like to thank all the people from the Department of Intelligent Systems at the JSI.

I am deeply grateful to my parents Slave and Roza, my brother Martin, and Maja Stojanoska, who supported me unconditionally during my years at JSI. Without your love, encouragement and support, none of this would have been possible.

Finally, I would like to express my gratitude to all my friends (in Slovenia, Macedonia and around the world) who have helped me and supported me during the years of my PhD studies at the Jožef Stefan International Postgraduate School.

9 References

- [1] Ducatel K, Bogdanowicz M, Scapolo F, Leijten J, Burgelman J (2001) Scenarios for ambient intelligence in 2010. Technical report. Retrieved September, 2014 from: <http://cordis.europa.eu/ist/istagreports.htm>
- [2] Aarts E. Ambient intelligence: a multimedia perspective. *IEEE Multimedia*. 2004; 11(1):12–19.
- [3] Shadbolt N. Ambient intelligence. *IEEE Intell Syst*. 2003; 18(4):2–3.
- [4] Ambient Assisted Living (AAL). Retrieved September, 2014. <http://www.aal-europe.eu/>.
- [5] United Nations 2009, World Population Ageing, Report. Retrieved September, 2014.
- [6] Bourouis A, Feham M, and Bouchachia A. A New Architecture of a Ubiquitous Health Monitoring System: A Prototype Of Cloud Mobile Health Monitoring System. *The Computing Research Repository (CoRR)*. 2012.
- [7] Dey A, Salber D, Abowd G, Futakawa M. The conference assistant: Combining context awareness with wearable computing. In *Proceedings of the 3rd International Symposium on Wearable Computing, ISWC*. 1999; 21–28.
- [8] The Confidence Project, European FP7, 2014. Retrieved September, 2014 from: <http://www.confidence-eu.org>
- [9] The CHIRON project JU ARTEMIS, 2014. Retrieved September, 2014 from: <http://www.chiron-project.eu/>
- [10] The Commodity12 project, European FP7, 2014. Retrieved September, 2014 from: <http://commodity12.eu/>
- [11] Gjoreski H, Kozina S, Gams M, and Luštrek M. RAReFall – Real-time Activity Recognition and Fall Detection System. *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM)*, live DEMO. 2014; 145–147.
- [12] Álvarez-García JA, Barsocchi P, Chessa S, Salvi D. Evaluation of localization and activity recognition systems for ambient assisted living: The experience of the 2012 EvAAL competition. *Journal of Ambient Intelligence and Smart Environments*. 2013; 119–132.
- [13] Gjoreski H, Kozina S, Gams M, Luštrek M, Álvarez-García JA, Hong JH, Ramos J, Dey AK, Bocca M, Patwari N. Competitive Live Evaluation of Activity-recognition Systems. *IEEE Pervasive Computing*, accepted, 2014.
- [14] Victory at the annual Activity Recognition competition EvAAL 2013. Retrieved December, 2014 from: <http://dis.ijs.si/?p=1320>
- [15] Young researchers from the Jozef Stefan Institute with remarkable results. Retrieved December, 2014 from: <http://ava.rtvsllo.si/predvajaj/mladi-raziskovalci-ijs-ja-s-svetovnim-uspehom/ava2.174241224/>
- [16] Gimón D, Gjoreski H, Kaluža B, Gams M. Using accelerometers to improve position-based activity recognition. In *Proceedings of the International Multiconference Information Society – IS*. 2010.
- [17] Gjoreski H, Gams M, Chorbev I. 3-Axial Accelerometers Activity Recognition. In *Web Proceedings of ICT Innovations*. 2010.

- [18] Gjoreski H, Gams M. Accelerometer Data Preparation for Activity Recognition. In Proceedings of the International Multiconference Information Society – IS. 2011.
- [19] Gjoreski H, Gams M. Activity/Posture Recognition Using Wearable Sensors Placed on Different Body Locations. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Soft Computing. 2011; 340–347.
- [20] Gjoreski H, Luštrek M, Gams M. Accelerometer Placement for Posture Recognition and Fall Detection. In Proceedings of the 7th International Conference on Intelligent Environments (IE). 2011; 47–54.
- [21] Gjoreski H. Adaptive Human Activity Recognition and Fall Detection using Wearable Sensors. Master Thesis.
- [22] Kozina S, Gjoreski H, Gams M, Luštrek M. Three-layer activity recognition combining domain knowledge and meta-classification. *Journal of medical and biological engineering*. 2013; 33(4).
- [23] Kozina S, Gjoreski H, Gams M, Luštrek M. Efficient Activity Recognition and Fall Detection Using Accelerometers. *Evaluating AAL Systems Through Competitive Benchmarking. Communications in Computer and Information Science*. 2013; 386: 13–23.
- [24] Gjoreski H, Kozina S, Luštrek M, Gams M. Using multiple contexts to distinguish standing from sitting with a single accelerometer. *European Conference on Artificial Intelligence (ECAI)*. 2014.
- [25] Luštrek M, Gjoreski H, Kozina S, Cvetkovic B, Gams M. Detecting Falls with Location Sensors and Accelerometers. In Proceedings of Twenty-Third IAAI Conference on Artificial Intelligence. 2011.
- [26] Luštrek M, Kaluža B, Cvetković B, et al. Confidence: Ubiquitous Care System to Support Independent Living, DEMO at European Conference on Artificial Intelligence (ECAI). 2012.
- [27] Kaluža B, Cvetković B, Dovgan E, Gjoreski H, Gams M, Luštrek M. Multiagent Care System to Support Independent Living. *International Journal on Artificial Intelligence Tools*. 2013.
- [28] Gjoreski H, Luštrek M, Gams M. Context-Based Fall Detection using Inertial and Location Sensors. In Proceedings of the International Joint Conference on Ambient Intelligence (AmI-12). 2012.
- [29] Gjoreski H, Gams M, Luštrek M. Context-based fall detection and activity recognition using inertial and location sensors. *Journal of Ambient Intelligence and Smart Environments (JAISE)*; 2014, 6(4): 419–433.
- [30] Gjoreski H, Kaluža B, Gams M, Milić R, Luštrek M. Ensembles of multiple sensors for human energy expenditure estimation. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp). 2013; 359–362.
- [31] Mazomenos E, Cvetković B, Kozina S, et al. Case studies. V: Maharanta, Koushnik, Bonfiglio, Silvio. *System design for remote healthcare*. New York, Springer. 2014; 277–332.
- [32] Gams M, Gjoreski H, Kaluža B, Luštrek M, Gams M. Method and system for context-based activity recognition. Ljubljana: Urad RS za intelektualno lastnino, 01. aug. 2014.
- [33] Gams M, Gjoreski H, Luštrek M. Method and System for Detecting a Person Driving a Vehicle while Using a Mobile Computing Device. United Kingdom, 01. aug. 2014.
- [34] Cook DJ, Augusto JC, Jakkula VR. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*. 2009; 5(4): 277–298.

- [35] Augusto JC, McCullagh P. Ambient Intelligence: Concepts and Applications. *Computer Science and Information Systems*. 2007; 4(1): 1–28.
- [36] Aztiria A, Izaguirre A, Augusto JC. Learning patterns in ambient intelligence environments: a survey. *Artificial Intelligence Review*. 2010; 34(1):35–51.
- [37] Wright D, Gutwirth S, Friedewald M, Vildjiounaite E, Punie Y. *Safeguards in a World of AmI*. Springer, Dordrecht, 2008.
- [38] IST Advisory Group. Retrieved September, 2014 from: http://cordis.europa.eu/fp7/ict/istag/about_en.html
- [39] Friedewald M, Da Costa O. Science and technology roadmapping: Ambient intelligence in everyday life (AmI@life). Technical report, 2003.
- [40] Hongwei H, Youzhi X, Hairong Y, Mubeen, S, Hongke Z. An Elderly Health Care System Using Wireless Sensor Networks at Home. *Sensor Technologies and Applications*. 2009; 158–163.
- [41] Hanson MA, Powell HC, Barth AT, et al. Body Area Sensor Networks: Challenges and Opportunities. *Computer*. 2009; 42:158–65.
- [42] Santofimia MJ, Fahlman SE, Del Toro X, Moya F, Lopez JC. A semantic model for actions and events in ambient intelligence. *Engineering Applications of Artificial Intelligence*. 2014; 24(8):1432–1445.
- [43] Strobbe M, Van Laere O, Dhoedt B, De Turck F, Demeester P. Hybrid reasoning technique for improving context-aware applications. *Knowledge And Information Systems*. 2012; 31(3):581–616.
- [44] Jess (2008) Jess, the rule engine for the java platform. Retrieved September, 2014 from: <http://www.jessrules.com/>
- [45] JBoss (2010) Drools—business logic integration platform. Retrieved September, 2014 from: <http://www.jboss.org/drools/>
- [46] Kaluza A. Probabilistic Risk Analysis for Multimodal Entry Control. *Expert systems with applications*. 2011; 38(6):6696–6704.
- [47] Bouamrane MM, Rector A, Hurrell M. Using owl ontologies for adaptive patient information modelling and preoperative clinical decision support. *Knowl Inf Syst*, 2010.
- [48] Wang KI, Abdulla WH, Salcic Z. Ambient intelligence platform using multi-agent system and mobile ubiquitous hardware. *Pervasive and Mobile Computing*; 2009 5(5):558–573.
- [49] Siirtola P, Laurinen P, Haapalainen E, Rönning J, Kinnunen H. Clustering-based activity classification with a wrist-worn accelerometer using basic features. *CIDM* 2009; 95–100.
- [50] Leake DB. *Case-based reasoning: experiences, lessons and future directions*. MIT Press, Cambridge, 1996.
- [51] Garcia Juan A. Recio. jCOLIBRI: a multi-level platform for building and generating CBR systems. PhD Thesis. 2008.
- [52] Glasgow J, Narayanan NH, Chandrasekaran B. *Diagrammatic reasoning: cognitive and computational perspectives*. AAAI Press, Menlo Park, 1995.
- [53] Novk V, Perfilieva I, Mockor J. *Mathematical principles of fuzzy logic*. Springer, Berlin. 1999.
- [54] Mylonas P, Vallet D, Castells P, Fernández M, Avrithis Y. Adapting applications in handheld devices using fuzzy context information. *Knowl Eng Rev*. 2008; 23(1):73–100.
- [55] Parsia C. Pellet: Owl 2 reasoner for java. 2010. Retrieved September, 2014.
- [56] Fact++ (2009) Fact++ owl-dl reasoner. Retrieved September, 2014 from: <http://owl.man.ac.uk/factplusplus/>

- [57] Racer (2010) Racerpro: Renamed abox and concept expression reasoner. Retrieved September, 2014 from: <http://www.racer-systems.com/>
- [58] Steinberg AN, Bowman CL. Revisions to the JDL Data Fusion Model. Handbook of Multisensor Data Fusion. CRC Press LLC, Florida, 2001.
- [59] Crowley JL, Demazeau Y. Principles and Techniques for Sensor Data Fusion. Signal Processing. 1991; 32(1–2): 5–27.
- [60] Salerno J, Hinman M, Boulware D. Building a Framework for Situation Awareness. In Proceedings of the Seventh Int. Conf. Information Fusion. 2004; 219–226.
- [61] Elmenreich W. A Review on System Architectures for Sensor Fusion Applications. SEUS, LNCS 4761, 2007; 547–559.
- [62] Akyldiz IF, Su W, Sankarasubramanian Y, Cayirci E. A survey on sensor networks. IEEE Communications Magazine. 2002; 40:102–114.
- [63] Sasiadek JZ, Hartana P. Sensor data fusion using Kalman filter. In Proceedings of the Third International Conference on Information Fusion. 2000; 2: 19–25.
- [64] Punska O. Bayesian Approaches to Multi-Sensor Data Fusion. PhD Thesis. 1999.
- [65] Sasiadek JZ, Hartana P. Sensor data fusion using Kalman filter. In Proceedings of the Third International Conference on Information Fusion. 2000; 2:19–25.
- [66] Giraud C, Jouvencel B. Sensor Selection: A Geometrical Approach. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. 1995; 2:555–560.
- [67] Marzullo K. Tolerating Failures of Continuous-Valued Sensors. ACM Transactions on Computer Systems. 1990; 8(4):284–304.
- [68] Russo F, Ramponi G. Fuzzy Methods for Multisensor Data Fusion. IEEE Transactions on Instrumentation and Measurement. 1994; 43(2):288–294.
- [69] Munoz MA, Rodriguez M, Favela J, Martinez-Garcia AI, Gonzalez VM. Context-Aware Mobile Communication in Hospitals. IEEE Computer. 2003; 36(9):38–46.
- [70] Schilit BN, Theimer MM. Disseminating Active Map Information to Mobile Hosts. IEEE Network. 1994; 8(5):22–32.
- [71] Schilit BN, Adams NL, Want R. ContextAware Computing Applications. In Proceedings of the Workshop on Mobile Computing Systems and Applications, IEEE Computer Society. 1994.
- [72] Dey AK, Abowd GD. Towards a Better Understanding of Context and Context-Awareness. In Proceedings of the ACM CHI, Workshop on "The What, Who, Where, When, and How of Context-Awareness", 2000.
- [73] Schmidt A, Beigl M, Gellersen H. There is more to Context than Location. Computers and Graphics. 1999; 23(6):893–901.
- [74] Ferreira D, Goncalves J, Kostakos V, Barkhuus L, Dey AK. Contextual experience sampling of mobile application micro-usage. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14). 91–100.
- [75] Korkea-aho M. Context-Aware Applications Survey. Retrieved September, 2014 from: <http://www.hut.fi/~mkorkeaa/doc/context-aware.html>
- [76] Crouter SE, Clowers KG, Bassett DR Jr. A novel method for using accelerometer data to predict energy expenditure. J Appl Physiol. 2006; 100(4):1324–31.
- [77] Lester J, Hartung C, Pina L, Libby R, Borriello G, Duncan G. Validated caloric expenditure estimation using a single body-worn sensor. In Proceedings of the 11th ACM international conference on Ubiquitous computing (Ubicomp). 2009; 225–234.

- [78] Vyas N, Farrington J, Andre D, Stivoric J. Machine learning and sensor fusion for estimating continuous energy expenditure. In Proceedings of the 23rd Conference on Innovative Applications of Artificial Intelligence (IAAI). 2011; 1613–1620.
- [79] Li Q, Stankovic J. Grammar-based, posture and context-cognitive detection for falls with different activity levels. In Proceedings of the 2nd Conference on Wireless Health. 2011.
- [80] Lan T, Wang Y, Mori G, Robinovitch S. Retrieving Actions in Group Contexts. International Workshop on Sign Gesture Activity (at ECCV), 2010.
- [81] Naguib H, Coulouris G. Location Information Management. In Proceedings of the Annual Conference on Ubiquitous Computing. 2001.
- [82] Jansson M. Context Shadow: An Infrastructure for Context Aware Computing. In Proceedings of Artificial Intelligence in Mobile System 2002.
- [83] Chen D, Schmidt A, Gellesen HW. An Architecture for Multi-Sensor Fusion in Mobile Environments. In Proceedings of the International Conference on Information Fusion. 1999; 2: 861–868.
- [84] Dey AK, Salber D, Futakawa M, Abowd GD. GVU Technical Report GIT-GVU-99-23. In Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology (UIST '99), 1999.
- [85] Shaofeng W, Lianying J, Aiguang L, Jiankang W. Body sensor networks for ubiquitous healthcare. Journal of Control Theory and Applications. 2011; 9(1): 3–9.
- [86] Gams M. Weak intelligence: through the principle and paradox of multiple knowledge. New York: Nova Science Publishers, 2001.
- [87] Gams M, Bojadžiev D, Knap Ž. Is intelligence multiple?. CC-AI, 1995; 12: 285–310.
- [88] Gams M. Strong vs. weak AI. Informatica, 1995; 19(4): 479–493.
- [89] Gams M, Karba N, Drobnič M. Integration of multiple reasoning systems for process control. Engineering applications of artificial intelligence, 1997; 10:41–46.
- [90] Gams M, Drobnič M, Karba N. Average-case improvements when integrating ML and KA. Applied intelligence, 1996; 6: 87–99.
- [91] Karba N, Gams M, Drobnič M. A general integrating schema applied at iron & steel works. Elektrotehniški vestnik, 1995; 62: 206–211.
- [92] Dietterich TG. Ensemble methods in Machine Learning. In Proceedings of the first International Workshop on Multiple Classifier Systems. 2000; 1–15.
- [93] KDD Cup. Retrieved September 2014, from: <http://www.sigkdd.org/kddcup/>
- [94] Random Forest visualization. Retrieved September, 2014, from: http://es.wikipedia.org/wiki/Random_forest
- [95] Mitchell T. Machine learning. McGraw-Hill Series in Computer Science, 1997.
- [96] Breiman L. Random forests. Machine Learning. 2001; 45:5–32.
- [97] L. Breiman, “Bagging Predictors,” Machine Learning. 1996; 24(2): 123–140.
- [98] T. K. Ho. The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998, 20(8): 832–844.
- [99] Wang Y, Witten I. Induction of model trees for predicting continuous classes. In Proceedings of the 9th European Conference on Machine Learning (ECML). 1997; 128–137.
- [100] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. COLT: Proceedings of the Workshop on Computational Learning Theory. 1998; 92–100.
- [101] Kozina S, Luštrek M., Gams M. Dynamic signal segmentation for activity recognition. In Proceedingd of the Space, Time and Ambient Intelligence (STAMI). 2011; 16–22.

- [102] Meko DM. Course: Applied Time Series Analysis. Laboratory of Tree-Ring Research, University of Arizona. Retrieved September, 2014, from: <http://www.ltrr.arizona.edu/~dmeko/geos585a.html>.
- [103] Tapia EM. Using Machine Learning for Real-time Activity Recognition and Estimation of Energy Expenditure. Ph.D. Thesis. 2008.
- [104] Rish I. An empirical study of the naive Bayes classifier. IJCAI Workshop on Empirical Methods in AI. 2001.
- [105] Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995; 20(3): 273.
- [106] David A. Freedman (2009). *Statistical Models: Theory and Practice*. Cambridge University Press. p. 26.
- [107] Drucker, Harris; Burges, Christopher J. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997); "Support Vector Regression Machines", in *Advances in Neural Information Processing Systems*, NIPS 1996; 155–161.
- [108] Okeyo G, Chen L, Wang H. Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*. 2014; 39:29–43.
- [109] Witten I, Frank E, Hall MA. *Data Mining: Practical machine learning tools and techniques*. 3rd ed, Morgan Kaufmann, 2011.
- [110] Zarandi MHF, Mohammadhasan N, Bastani S. A Fuzzy Rule-Based Expert System for Evaluating Intellectual Capital. *Advances in Fuzzy Systems*. 2012.
- [111] Shoham Y, Goyal N. Temporal reasoning in artificial intelligence. In *Exploring artificial intelligence*. H. E. Shrobe (Ed.). Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 419–438. (1988).
- [112] Artikis A, Sergot M, Paliouras G. A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on Events in multimedia (EiMM '10)*. 2010; 3–8.
- [113] Xu L, Krzyzak A, Suen CY. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*. 1992, 22(3): 418,435.
- [114] Bauer E, Kohavi R. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Mach. Learn*. 1999; 36, 105–139.
- [115] Wolpert DH. Stacked Generalization. *Neural Networks*. 1992; 5:241–259.
- [116] Breiman L. Stacked regressions. *Machine Learning*. 1996; 24(1):49–64.
- [117] Kearns MJ. *The Computational Complexity of Machine Learning*. Ph.D. Dissertation. Harvard Univ., Cambridge, MA, USA.
- [118] Gregory DA, Anind KD, Peter JB, Nigel D, Mark S, Pete S. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. 1999; 304–307.
- [119] Vyas N, Farrington J, Andre D, Stivoric JI. Machine Learning and Sensor Fusion for Estimating Continuous Energy Expenditure. In *Proceedings of the Innovative Applications of Artificial Intelligence Conference*. 2012.
- [120] PAMSys – The physical activity monitoring system. Retrieved September, 2014 from: <http://www.biosensics.com/pamsys.html>
- [121] Atallah L, Lo B, King R, Yang GZ. Sensor Placement for Activity Detection Using Wearable Accelerometers. In *Proceedings BSN (2010)*, 24–29.
- [122] Cleland I, Kikhia B, Nugent C, et al. Optimal Placement of Accelerometers for the Detection of Everyday Activities. *Sensors*. 2013; 13 (7).
- [123] Luštrek M, Cvetković B, Kozina S. Energy Expenditure Estimation with Wearable Accelerometers. In *Proceedings of the IEEE ISCAS*. 2012; 5–8.
- [124] Just stand organization. Retrieved September, 2014 from: <http://www.juststand.org>

- [125] Bonomi AG. Physical activity recognition using a wearable accelerometer. PhD Thesis. 2010.
- [126] Sukthanka Gr, Sycara K, A cost minimization approach to human behavior recognition. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). 2005; 1067–1074.
- [127] Kwapisz JR, Weiss GM, Moore SA. Activity Recognition using Cell Phone Accelerometers. *Human Factors*. 2010; 12:74–82.
- [128] Wu H, Lemaire ED, Baddour N. Activity Change-of-state Identification Using a Blackberry Smartphone. *Journal of Medical and Biological Engineering*, 2012; 32: 265–272.
- [129] Lai C, Huang YM, Chao HC, Park JH. Adaptive Body Posture Analysis Using Collaborative Multi-Sensors for Elderly Falling Detection. *IEEE Intelligent Systems*. 2010; 2–11.
- [130] Ravi N, Dandekar N, Mysore P, Littman ML. Activity Recognition from Accelerometer Data. In Proceedings of the 17th conference on Innovative applications of artificial intelligence. 2005; 1541–1546.
- [131] Banos O, Damas M, Pomares H, Rojas F, Delgado-Marquez B, Valenzuela O. Human activity recognition based on a sensor weighting hierarchical classifier. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*. 2012; 1–11.
- [132] B. Kaluža, "Reducing spurious activity transitions in a sequence of movement," *Proc. 18th International Electrotechnical and Computer Science Conference*, B: 163–166, 2009.
- [133] J. Lester, T. Choudhury, N. Kern, G. Borriello and B. Hannaford, "A Hybrid Discriminative/Generative Approach for Modeling Human Activities," *Proc. International joint conference on Artificial intelligence, IJCAI*, 2005; 766–772.
- [134] Lara OD, Péreza AJ, Labrador MA, Posada JD. Centinela: A Human Activity Recognition System Based on Acceleration and Vital Sign Data. *Pervasive and Mobile Comp.* 2012; 8(5):717–729.
- [135] Lee MW, Khan AM, Kim TS. A Single Tri-Axial Accelerometer-Based Real-Time Personal Life Log System Capable of Human Activity Recognition and Exercise Information Generation. *Personal and Ubiquitous Computing*. 2011; 15(8):887–898.
- [136] Khan AM, Lee YK, Kim TS. Accelerometer Signal-based Human Activity Recognition Using Augmented Autoregressive Model Coefficients and Artificial Neural Nets. In Proceedings of the IEEE EMBS. 2008; 20–24.
- [137] Curone D, Bertolotti GM, Cristiani A, Secco EL, Magenes G. A Real-Time and Self-Calibrating Algorithm Based on Triaxial Accelerometer Signals for the Detection of Human Posture and Activity. *IEEE TITB*. 2010; 14(4):1098–1105.
- [138] Mathie M, Celler B, Lovell N, Coster A. Classification of basic daily movements using a triaxial accelerometer. *Med. Biol. Eng. Comput.* (2004), 42, 679–687.
- [139] Ermes M, Parkka J, Mantyjarvi, J, Korhonen I. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Trans. Inf. Technol. Biomed.* 2008; 12(1):20–26.
- [140] Lester J, Choudhury T, Borriello G. A Practical Approach to Recognizing Physical Activities. *Pervasive Computing, Lec. Notes*. 2006; 3968:1–16.
- [141] Song SK, Jang J, Park SJ. Dynamic Activity Classification Based on Automatic Adaptation of Postural Orientation. In Proceedings of the EMBC. 2009, 6175–6178.
- [142] Olguín D, Pentland A. Human Activity Recognition: Accuracy across Common Locations for Wearable Sensors. In Proceedings of the ISWC. 2006.

- [143] Khan AM. Human Activity Recognition Using A Single Tri-axial Accelerometer. PhD Thesis. 2011.
- [144] Shimmer Sensor. Retrieved September. 2014 from: www.shimmer-research.com
- [145] Freedson PS, Pober D, Janz KF. Calibration of accelerometer output for children. *Med Sci Sports Exerc.* 2005; 37(11 Suppl):S523–30.
- [146] Hendelman D, Miller K, Bagget C, Debold E, Freedson PS. Validity of accelerometry for the assessment of moderate intensity physical activity in the field. *Med Sci Sports Exerc.* 2000 Sep;32(9 Suppl):S442–9.
- [147] Montoye HJ, Washburn R, Servais S, Ertyl A, Webster JG, Nagle FJ. Estimation of energy expenditure by a portable accelerometer. *Med Sci Sports Exerc.* 1983; 15(5):403–7.
- [148] Swartz A, Strath SJ, Bassett DJ, O'Brien W, King GA, Ainsworth BE. Estimation of energy expenditure using CSA accelerometers at hip and wrist sites. *Med Sci Sports Exerc.* 2000 Sep;32(9 Suppl):S450–6.
- [149] Fruin ML, Rankin JW. Validity of a multi-sensor armband in estimating rest and exercise energy expenditure. *Med Sci Sports Exerc.* 2004; 36(6):1063–9.
- [150] Bassett DR, Ainsworth BE, Swartz AM, Strath SJ, O'Brien WL, King GA. Validity of four motion sensors in measuring moderate intensity physical activity. *Med Sci Sports Exerc.* 2000; 32(9):S471–S80.
- [151] Rumo M, Amft O, Tröster G, Mäder U. A stepwise validation of a wearable system for estimating energy expenditure in field-based research. *Physiol Meas.* 2011; 32(12):1983–2001.
- [152] Sazonova N, Browning RC, Sazonov E. Accurate Prediction of Energy Expenditure Using a Shoe-Based Activity Monitor. *Med Sci Sports Exerc.* 2011; 43(7):1312–21.
- [153] Rothney M.P, Neumann M, Béziat A, Chen KY. An artificial neural network model of energy expenditure using nonintegrated acceleration signals. *J Appl Physiol.* 2007;103(4):1419–27.
- [154] Staudenmayer J, Pober D, Crouter S, Bassett D, Freedson PS. An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer. *J Appl Physiol.* 2009; 107:1300–7.
- [155] Trost SG, Wong WK, Pfeiffer KA, Zheng Y. Artificial neural networks to predict activity type and energy expenditure in youth. *Med Sci Sports Exerc.* 2012; 44(9):1801–1809.
- [156] Liu S, Gao RX, John D, Staudenmayer J, Freedson P. Multi-sensor data fusion for physical activity assessment. *IEEE Trans Biomed Eng.* 2012; 59(3):687–96.
- [157] John D, Liu S, Sasaki JE, et al. Calibrating a novel multi-sensor physical activity measurement system. *Physiol Meas.* 2011; 32(9):1473–89.
- [158] Liu S, Gao RX, Freedson PS. Computational Methods for Estimating Energy Expenditure in Human Physical Activities. *Med Sci Sports Exerc.* 2012; 44(11):2138–46.
- [159] Albinali F, Intille SS, Haskell, W, Rosenberger, M. Using wearable activity type detection to improve physical activity energy expenditure estimation. In *Proceedings of the 12th ACM international conference on Ubiquitous computing (Ubicomp).* 2010; 211–320.
- [160] Lee JM, Kim Y, Welk GJ. Validity of Consumer-Based Physical Activity Monitors. *Med Sci Sports Exerc.* 2014.
- [161] Mannini A, Sabatini AM. Machine Learning Methods for Classifying Human Physical Activities from on-body sensors. *Sensors.* 2010; 10:1154–1175.
- [162] Venables WN, Ripley BD. *Modern Applied Statistics with S.* New York (NY): Springer-Verlag; 2002.

- [163] David J.C. Mackay (1998). Introduction to Gaussian Processes. Dept. of Physics, Cambridge University, UK.
- [164] Russell, S.J.; Norvig, P. Artificial Intelligence: A Modern Approach. 3. Upper Saddle River (NJ): Prentice Hall; 2010. p. 729
- [165] Zhou ZH. Ensemble Methods: Foundations and Algorithms. Chapman and Hall/CRC. 2012.
- [166] Trobec R, Depolli M, Avbelj V. Wireless network of bipolar body electrodes. In Proceedings of the 7th International Conference on Wireless On-demand Network Systems and Services (WONS). 2010; 145–149.
- [167] Trobec R, Avbelj V, Rashkovska A. Multi-functional Wireless Body Sensor. In Proceedings of the First International Conference on Global Health Challenges. 2012; 71–74.
- [168] Trobec R, Tomašić I. Synthesis of the 12-lead electrocardiogram from differential leads. IEEE Trans. Inf. Technol. Biomed. 2011; 15: 615–621.
- [169] Tomašić I, Frljak S, Trobec R. Estimating the Universal Positions of Wireless Body Electrodes for Measuring Cardiac Electrical Activity. IEEE Transactions on Biomedical Engineering. 2013; 60:3368–3374.
- [170] Trobec R, Rashkovska A, Avbelj V. Two proximal skin electrodes – a respiration rate body sensor. Sensors. 2012; 12:13813–13828.
- [171] Friedman SM, Munoz B, West SK, Rubin GS, Fried LP. Falls and Fear of Falling: Which Comes First? A Longitudinal Prediction Model Suggests Strategies for Primary and Secondary Prevention. Journal of the American Geriatrics Society. 2002; 1329–1335.
- [172] World Health Organization (WHO). Global brief for World Health Day 2012, Good health adds life to years. Retrieved, September 2014, from: http://whqlibdoc.who.int/hq/2012/WHO_DCO_WHD_2012.2_eng.pdf
- [173] Gillespie LD, Robertson MC, Gillespie WJ, et al. Interventions for preventing falls in older people living in the community (Review). The Cochrane Library, 4. 2009.
- [174] Tinetti ME, Williams CS. Falls, Injuries Due to Falls, and the Risk of Admission to a Nursing Home. The New England Journal of Medicine. 1997; 337:1279–1284.
- [175] Rubenstein LZ, Josephson KR. The epidemiology of falls and syncope. Clinics in Geriatric Medicine. 2002; 18(2):141–158.
- [176] Khawandi S, Daya B, Chauvet P. Automated monitoring system for fall detection in the elderly. In Proceedings of the International Journal of Image Processing (IJIP). 2010; 4(5).
- [177] Machajdik J, Zambanini S, Kampel M. Fusion of Data from Multiple Cameras for Fall Detection. In Proceedings of Workshop on Behavior Monitoring and Interpretation (BMI'10). 2010; 1–7.
- [178] Doukas N, Maglogiannis I. Emergency Fall Incidents Detection in Assisted Living Environments Utilizing Motion, Sound, and Visual Perceptual Components. Information Technology in Biomedicine, IEEE Transactions on. 2011; 15(2): 277–289.
- [179] Botía JA, Villa A, Palma JT, Pérez D, Iborra E. Detecting Domestic Problems of Elderly People: Simple and Unobstrusive Sensors to Generate the Context of the Attended. In Proceedings of the 1st International Workshop on Ambient Assisted Living. 2009; 819–826.
- [180] Muñoz A, Serrano E, Villa A, Valdés M, Botía JA. An Approach for Representing Sensor Data to Validate Alerts in Ambient Assisted Living. Sensors. 2012; 6282–6306.

- [181] Lin HT, Hsieh YJ, Chen MC, Chang W. ActionView: a movement-analysis ambulatory monitor in elderly homecare systems. *IEEE International Symposium on Circuits and Systems*. 2009; 3098–3101.
- [182] Zhang T, Wang J, Xu L, Liu P. Fall Detection by Wearable Sensor and One-Class SVM. *Intelligent Computing in Signal Processing and Pattern Recognition*. 2006; 345:858–863.
- [183] Jantaraprim P, Phukpattaranont P, Limsakul C, Wongkittisuksa B. Evaluation of fall detection for the elderly on a variety of subject groups. In *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology*. 2009.
- [184] Li Q, Stankovic J, Hanson M, Barth AT, Lach J, Zhou G. Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information. In *Proceedings of the Sixth International Workshop on Wearable and Implantable Body Sensor Networks*. 2009; 138–143.
- [185] Zinnen C, Wojek, Schiele B. Multi Activity Recognition Based on Bodymodel-Derived Primitives. In *Proceedings of the 4th Int. Symposium on Location and Context Awareness*. 2009.
- [186] Nguyen TT, Cho MC, Lee TS. Automatic fall detection using wearable biomedical signal measurement terminal. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2009; 5203–6.
- [187] Sixsmith A, Johnson N. Smart sensor to detect the falls of the elderly. *Pervasive Computing, IEEE*. 2004; 3(2):42–47.
- [188] Naranjo HD, Roa LM, Reina TJ, Estudillo VMA. Personalization and adaptation to the medium and context in a fall detection system. *IEEE Trans Inf Technol Biomed*. 2012; 16(2):264–71.
- [189] XSens sensors website. Retrieved September, 2014 from: <http://www.xsens.com>
- [190] Ubisense RTLS website. Retrieved September, 2014 from: <http://www.ubisense.net>
- [191] Kaluža B, Dovgan E. Glajenje trajektorij gibanja cloveskega telesa zajetih z radijsko tehnologijo. In *Proceedings of the International Multi-Conference Information Society*, 2009.
- [192] Thornton C, Hutter F, Hoos H, Leyton-Brown K. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. In *Proceedings of the KDD*. 2013; 847–855.
- [193] Monica DD, Goranko V, Montanari A, Sciavicco G. Interval Temporal Logics: a Journey. *Bulletin of the EATCS*, 2011; 105: 73–99.

Index of Figures

Figure 2-1. Relation between AmI and other areas in computer science [35].	8
Figure 2-2. Average power consumption of continuous ambulatory monitoring applications [41].	10
Figure 3-1. Visualization of the decision space divided by a single regression model and multiple regression models (ensemble-learning approach) [94].	18
Figure 3-2. An example of a context-based view of the data. Context-based data partitioning is performed using the feature B as a context.	20
Figure 3-3. CoReAmI reasoning flowchart.	21
Figure 3-4. Context extraction phase in the CoReAmI approach.	23
Figure 3-5. Synchronized multiple sensor data (approximately five minutes of data).	24
Figure 3-6. Activity extraction procedure.	26
Figure 3-7. Context-based modeling phase in CoReAmI.	29
Figure 3-8. Graphical illustration of an example dataset and dividing the reasoning data according to the context values, i.e., context-based data partitioning. The breath-rate feature is chosen as a context.	30
Figure 3-9. Context-based classification for activity recognition.	32
Figure 3-10. Context-based regression for energy expenditure.	34
Figure 3-11. Context-based reasoning for fall detection.	35
Figure 4-1. The RAReFall system.	48
Figure 4-2. Overview of the context-based activity-recognition process. The baseline approach combined with the CoReAmI approach for the sitting and the standing activity.	50
Figure 4-3. Shimmer sensor platform [144].	51
Figure 4-4. Baseline activity recognition data flow.	51
Figure 4-5. 3D Acc data for a sitting activity, including the activity segments before and after.	54
Figure 4-6. Graphical representation of the sensor tile (inclination) angles.	56
Figure 4-7. CoReAmI for activity recognition.	57
Figure 4-8. Recall and precision for the sitting (left) and the standing (right) activities achieved by the baseline approach and CoReAmI.	60
Figure 4-9. Comparison of the accuracy achieved for the standing and sitting activities using: each of the context models individually and CoReAmI.	61
Figure 4-10. Comparison of the accuracy and the F-measure for the standing and sitting activities achieved by the baseline, J48 trained with the context features (J48-ML), Random Forest trained with the context features (RF-ML), and CoReAmI.	61
Figure 5-1. EE estimation using activity as a context.	67
Figure 5-2. CoReAmI for EE estimation.	68

Figure 5-3. A person wearing the sensor equipment for EE estimation.	69
Figure 5-4. Mean Root Mean Square Error (RMSE) for the CoReAmI's MET estimation compared to each of the contexts used individually (only the context models learned for the particular context).	76
Figure 5-5. Measured and estimated METs for different types of activities using CoReAmI approach, output of the BodyMedia sensor and ANN trained on chest-accelerometer data only (ANN-Acc) approach [154][155].....	77
Figure 6-1. CoReAmI approach for fall detection.	84
Figure 6-2. Sensor equipment. (a) User wearing the equipment: inertial sensors (non-filled circles) and location tags (filled circles); (b) The inertial sensors equipment: Xsens-MTx and the Xbus Master; (c) The location sensors equipment: Ubisense wall mounted anchors and Ubisense wearable tag.....	85
Figure 6-3. Acceleration pattern during a fall.	86
Figure 6-4. Matrix representation of the best sensor combinations using the Inertial (I) and Location (L) sensors. F - overall F-measure, C - Chest, W - Waist, AR - Ankle Right, AL - Ankle Left, TR - Thigh Right, TL - Thigh Left.	92
Figure 6-5. Best sensor combinations for CoReAmI fall detection.	93
Figure 6-6. Comparison of the fall detection results achieved by our CoReAmI approach, the Machine-learning approach (MLA), and Acceleration Fall Pattern (AFP) approach. The event numbers correspond to the events given in Table 6-3.	96

Index of Tables

Table 3-1. Context extraction time complexity, where x_i is a sensor data sample value, and p is the number of data samples in one window.	40
Table 4-1. EvAAL-AR '12 and '13 teams and results (score: from 0 to 10)	49
Table 4-2. The contexts extracted for the context-based AR.	55
Table 4-3. Confusion matrix for the baseline AR.	60
Table 4-4. The confusion matrix of CoReAmI AR.	60
Table 5-1. Contexts extracted from each sensor.	70
Table 5-2. Activity scenario.	72
Table 5-3. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for CoReAmI's MET estimation compared to the: single regression, Random Subspace and Bagging using 5 base learners: artificial neural network (ANN), support vector regression (SVR), multiple linear regression (MLR), Gaussian processes for regression (GPR), and Model Tree (M5P). The best performance achieved by each of the aggregation techniques for each base learner is marked with bold style. The overall best performance for each aggregation technique is marked with gray background.	74
Table 5-4. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) achieved by CoReAmI using two aggregation techniques: average and median; and five base learners: artificial neural network (ANN), support vector regression (SVR), multiple linear regression (MLR), Gaussian processes for regression (GPR), and Model Tree (M5P). The best performance achieved by each of the approaches for each base learner is marked with bold style. The overall best performance for each base learner is marked with gray background.	75
Table 5-5. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for the CoReAmI's EE estimation compared to the BodyMedia sensor and the ANN-Acc regression model. The best performance for each activity type is marked with bold style.	78
Table 6-1. Extracted contexts and their possible values. A – activity, L – location, BM – body movement, AFP – acceleration fall pattern.	87
Table 6-2. Contexts and expert rules when the user is lying on the bed not moving.	88
Table 6-3. The events in the scenario, the appropriate activities and event descriptions.	90
Table 6-4. CoReAmI fall-detection analysis using only one sensor enclosure equipped with inertial and location sensor.	93
Table 6-5. CoReAmI fall-detection analysis using only two sensor enclosures equipped with inertial and location sensor.	94
Table 6-6. CoReAmI for fall detection: the simplest and the best combinations of the inertial-only, location-only and both types of sensors.	94
Table 6-7. CoReAmI detailed fall detection results for each event.	95

Index of Algorithms

Algorithm 1. Context extraction phase in CoReAmI.	25
Algorithm 2. Context modeling phase in CoReAmI. The constructModel() function constructs a model given a dataset. The context_based_partitioning() function creates a subset (reasoning) dataset, given a dataset and context value.....	28
Algorithm 3. Context aggregation phase in CoReAmI.	37
Algorithm 4. Execution of CoReAmI – evaluating an observation sequence. The function findCorrespondingModel () returns a model from the set of models, which corresponds to the particular context value – given as input.	43
Algorithm 5. Low-pass filter.	52
Algorithm 6. High-pass filter.....	53

Appendix A: Extracted Features for the Baseline Activity Recognition Approach

The features described in this appendix are extracted for every window for which the activity is recognized. The index i in the following equations represents the index of the acceleration data sample in the window.

- The average acceleration along the x, y and z axes. Only the equation for the x axis is shown, the other two are the same except that y and z are substituted for x, and the average length of the acceleration vector within the window.

$$\bar{x} = \frac{1}{N} \cdot \sum_{i=1}^N x_i$$

$$a_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

$$\bar{a} = \frac{1}{N} \cdot \sum_{i=1}^N a_i$$

N is the number of acceleration measurements within the window, x_i is the i -th acceleration measurement along one axis, and a_i is the length of the i -th acceleration vector.

- The variance of the acceleration along x and z axes and the variance of the length of the acceleration vector. Only the equation for the x axis is shown:

$$\delta_x^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}$$

- The maximum and the minimum acceleration along the x, y and z axes and the maximum and the minimum length of the acceleration vector:

$$M_x = \max\{x_i | i = 1 \dots N\}$$

$$m_x = \min\{x_i | i = 1 \dots N\}$$

- The difference between the maximum and the minimum acceleration along the x, y and z axes and the difference between the maximum and the minimum length of the acceleration vector:

$$diff_x = M_x - m_x$$

- The angle of change in the acceleration between the maximum and the minimum acceleration along the x and y axes:

$$spd_x = \tan^{-1} \frac{M_x - m_x}{t(M_x) - t(m_x)}$$

$t(M_x)$ and $t(m_x)$ are the timestamps when the maximum and the minimum acceleration were measured.

- The orientation (inclination angles) of the accelerometer along the x, y and z axes:

$$\varphi_x = \cos^{-1} \left(\frac{\bar{x}}{\sqrt{\bar{x}^2 + \bar{y}^2 + \bar{z}^2}} \right)$$

- The index of dispersion along the x and z axes and the index of dispersion of the length of the acceleration vector:

$$d_x = \frac{\delta_x^2}{\bar{x}}$$

- The sum of absolute differences between the consecutive lengths of the acceleration vector (s):

$$s = \frac{\sum_{i=1}^{N-1} |a_{i+1} - a_i|}{t_N - t_1}$$

t_1 and t_n are the starting and the ending time of the window.

Appendix B: Extracted Features for the Activity Recognition Performed for the Fall Detection Approach

Inertial Features

The features extracted from the inertial sensor data and used in the activity recognition. The total number of features per sensor is 25: 8 for the gyroscope data and 17 for the accelerometer data, divided into four groups:

- Statistical features (total 20). The Mean Value and the Standard Deviation were extracted for both the acceleration and gyroscope data; additionally, the Root Mean Square (RMS) was calculated only for the accelerometer data. A feature-selection analysis showed that the RMS was a redundant feature for the gyroscope data.
- Movement intensity feature. Using the changes of the acceleration vector the following feature was extracted: This feature sums up the differences between consecutive values of the lengths of the acceleration vector, and divides the sum by the time interval (one second):

$$AVC = \frac{\sum_{i=1}^q |length_i - length_{i-1}|}{T_s - T_0}$$

q is the number of data samples, T_0 is the timestamp for the first data sample in the window, and T_s is the timestamp of the last data sample. By applying a threshold to the AVC value, the movement of a sensor is detected.

- Sensor inclination angles (total 3). Since most of the time the main component of the acceleration vector was the gravity, they were calculated as the angles between the acceleration vector and each of the axes. For instance, the angle φ_x between the acceleration vector and the x axis is computed as follows (where the values a_x , a_y and a_z represent the actual acceleration vector):

$$\varphi_x = \arccos\left(\frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right),$$

- Difference between the maximum and minimum value of the acceleration vector in the current data window.

Location Features

The following features were extracted from the location sensor data:

- The z (height) coordinate of the sensor,
- The Euclidian distances between each pair of sensors,
- The z -distances between each pair of sensors (difference in heights),
- The Euclidian distances between each pair of sensors in the xy plane,
- Two velocity-based features: the first one is the absolute velocity of the sensor, and the second one is computed as the velocity of the sensor in the z direction.

Appendix C: Bibliography

C1 Publications Related to This Thesis

C1.1 Journal Articles

1. Gjoreski H, Kozina S, Gams M, Luštrek M, Álvarez-García JA, Hong JH, Ramos J, Dey AK, Bocca M, Patwari N. Competitive Live Evaluation of Activity-recognition Systems. *IEEE Pervasive Computing*, accepted, 2014. **JCR: 2.103**
2. Gjoreski H, Gams M, Luštrek M. Context-based fall detection and activity recognition using inertial and location sensors. *Journal of Ambient Intelligence and Smart Environments*, 2014, vol. 6, no. 4, p.p. 419-433. **JCR: 1.082**
3. Kozina S, Gjoreski H, Gams M, Luštrek M. Three-layer activity recognition combining domain knowledge and meta-classification. *Journal of medical and biological engineering*, 2013, vol. 33, no. 4, p.p. 406-414. **JCR: 1.076**
4. Kaluža B, Cvetković B, Dovgan E, Gjoreski H, Mirchevska V, Gams M, Luštrek. A Multiagent care system to support independent living. *International Journal on Artificial Intelligence Tools*, 2014, vol. 23, no. 1. **JCR: 0.321**

C1.2 Conference Proceedings

1. Gjoreski H, Kozina S, Luštrek M and Gams M. Using multiple contexts to distinguish standing from sitting with a single accelerometer. *European Conference on Artificial Intelligence (ECAI 2014)*, 2014, 1015-1016.
2. Gjoreski H, Luštrek M, Gams M. Developing a sensor firmware application for real-life usage. In *Proceedings of the International Multiconference Information Society – IS*, 2014; 2014, 34-37.
3. Gjoreski H, Rashkovska, A, Kozina S, Luštrek M, Gams M. Telehealth using ECG sensor and accelerometer. In *Proceedings, MIPRO 2014, 37th International Convention*, 2014; 283-287.
4. Gjoreski H, Kozina S, Gams M, Luštrek M. RAReFall - real-time activity recognition and fall detection system. *Percom workshops, 2014 IEEE International Conference on Pervasive Computing and Communication*, 2014; 145-147
5. Gjoreski H, Kozina S, Luštrek M, Gams M. Recognizing human activities and detecting falls in real-time. *6th Jožef Stefan International Postgraduate School Students' Conference*, 2014; 121-130.
6. Kozina S, Gjoreski H, Gams M, Luštrek M. Efficient Activity Recognition and Fall Detection Using Accelerometers. *Evaluating AAL Systems Through Competitive Benchmarking. Communications in Computer and Information Science*. 2013; 386: 13-23.

7. Gjoreski H, Kaluža B, Gams M, Milić R, Luštrek M. Ensembles of multiple sensors for human energy expenditure estimation. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp). 2013; 359–362.
8. Gjoreski H, Kozina S, Gams M, Luštrek M. Potential usage of smartphone inertial sensors in healthcare applications. Jožef Stefan International Postgraduate School Students Conference, 2013; 126-135.
9. Gjoreski H, Luštrek M, Gams M. Context-based fall detection using inertial and location sensors. Lecture notes in computer science (LNCS 7683), Ambient Intelligence: Third International Joint Conference, AmI 2012.
10. Luštrek M, Kaluža B, Cvetković B, Dovgan E, Gjoreski H, Mirchevska V, Gams M. Confidence: Ubiquitous Care System to Support Independent Living, DEMO at European Conference on Artificial Intelligence (ECAI). 2012.
11. Žbogar M, Gjoreski H, Kozina S, Luštrek M. Improving accelerometer based activity recognition. 15. International Multi-Conference Information Society - IS, 2012; 167-170.
12. Gjoreski H, Luštrek M, Gams M. Accelerometer Placement for Posture Recognition and Fall Detection. In Proceedings of the 7th International Conference on Intelligent Environments (IE). 2011; 47-54.
13. Gjoreski H, Gams M. Activity/Posture Recognition Using Wearable Sensors Placed on Different Body Locations. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Soft Computing. 2011; 340-347.
14. Luštrek M, Gjoreski H, Kozina S, Cvetkovic B, Gams M. Detecting Falls with Location Sensors and Accelerometers. In Proceedings of Twenty-Third IAAI Conference on Artificial Intelligence. 2011; 1662-1667.
15. Gjoreski H, Gams M. Accelerometer Data Preparation for Activity Recognition. In Proceedings of the International Multiconference Information Society – IS. 2011; 49-52.
16. Gjoreski H, Gams M, Chorbev I. 3-axial accelerometers activity recognition. ICT Innovations 2010; 51-58,
17. Gimon D, Gjoreski H, Kaluža B, Gams M. Using accelerometers to improve position-based activity recognition. In Proceedings of the International Multiconference Information Society – IS. 2010; 15-18.

C1.3 Book Chapter

1. Mazomenos E, Rodríguez JM, Barca CC, Tartarisco G, Pioggia G, Cvetković B, Kozina S, Gjoreski H, Luštrek M, Solar H, Marinčič D, Lampe J, Bonfiglio S, Maharatna K. Case studies. System design for remote healthcare. New York, Springer. 2014; 277-332.

C1.4 Patent Applications

1. Gams M, Gjoreski H, Kaluža B, Luštrek M, Gams M. Method and system for context-based activity recognition. Ljubljana: Urad RS za intelektualno lastnino, Ljubljana, 01. August 2014.
2. Gams M, Gjoreski H, Luštrek M. Method and System for Detecting a Person Driving a Vehicle while Using a Mobile Computing Device. United Kingdom, 01. August 2014.

C2 Other Publications

C2.1 Conference Proceedings

1. Gjoreski M, Gjoreski H, Kulakov A. Automatic recognition of emotions from speech.. In Proceedings of the International Multiconference Information Society – IS, 2014; 38-41.
2. Gjoreski H. Risk assessment model for congestive heart failure. 4th Jožef Stefan International Postgraduate School Students Conference, 2012; 125-130.
3. Cvetković B, Kaluža B, Gjoreski H, Luštrek M. Hybrid recommender system for personalized POI selection. In Proceedings of the International Multiconference Information Society – IS. 2013; 19-22.
4. Gjoreski H, Cvetković B, Kaluža B, Luštrek M. Sightseeing route planning. In Proceedings of the International Multiconference Information Society – IS. 2013, 39-42.
5. Gjoreski M, Gjoreski H, Piltaver R, Gams M. Predicting the arrival and the departure time of an employee. In Proceedings of the International Multiconference Information Society – IS, 2013; 43-46.
6. Luštrek M, Kaluža B, Cvetković B, Gjoreski H. E-turist : inteligentni elektronski turistični vodnik", In Proceedings of the International Multiconference Information Society – IS, 2013, 75-78.

C2.2 Patent Application

1. Gams M, Piltaver R, Gjoreski H. Postopek identifikacije osebe, ki vstopa v prostor. Ljubljana: Urad RS za intelektualno lastnino, Ljubljana, 19. September, 2013.

Appendix D: Biography

Hristijan Gjoreski was born on August 12, 1987 in Prilep, Republic of Macedonia. In 2006 he started his undergraduate studies at the Faculty of Electrical Engineering and Information Technology in Skopje, Macedonia. He finished his undergraduate study in 2010 with great honor. During his undergraduate studies he was awarded a scholarship by the Ministry of Education of Macedonia, which was given to the best 125 students with remarkable success in the studies.

In 2010, he enrolled in the master programme "Information and Communication Technologies" at the Jožef Stefan International Postgraduate School in Ljubljana, Slovenia. In 2011, he successfully defended the master thesis titled "Adaptive human activity recognition and fall detection using wearable sensors", under the supervision of Prof. Dr. Matjaž Gams. During the master studies, he was awarded a scholarship by the Department of Intelligent Systems at the Jožef Stefan Institute in Slovenia.

In the fall 2011 he applied for a PhD degree in "Information and Communication Technologies" at the Jožef Stefan International Postgraduate School. For his PhD studies he was awarded a scholarship for PhD students in Slovenia from abroad, enabled by the Slovene Human Resources Development and Scholarship Fund.

Since 2010, he is a research assistant at the Department of Intelligent Systems at the Jožef Stefan Institute, under advisement of Prof. Dr. Matjaž Gams and Dr. Mitja Luštrek. His main research interest is the analysis of human behavior and health using primarily machine-learning techniques applied on wearable sensors' data. He has experience with various sensing technologies and analysis of their data: inertial sensors, ultra-wideband location system, heart rate, body temperature, GSR, and other physiological sensors. He has used these sensors' data in order to extract useful information about the user wearing the sensors: recognize activity, detect a fall and estimation of human energy expenditure.

His research work has been presented at dozens of international conferences and published in several international journals. He was also part of the team that won the annual international competition in activity recognition – EvAAL 2013. For his work in the ambient intelligence domain he has also received the best student paper award at the Jožef Stefan International Postgraduate School Students' Conference (IPSSC 2014). The majority of his research experience in the ambient intelligence and artificial intelligence is mainly from three European research projects: Confidence, Chiron and Commodity12. He also collaborated in two projects funded by the Slovenian Ministry for Education, Science, Culture and Sport: eTurist and Asistent.