

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Remote M2M Healthcare: Applications and Algorithms

João Carlos Figueiredo Rodrigues Prudêncio

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Ana Cristina Costa Aguiar (Assistant Professor)

Second Supervisor: Daniel Enrique Lucani Rotter (Assistant Professor)

18th June, 2012

Remote M2M Healthcare: Applications and Algorithms

João Carlos Figueiredo Rodrigues Prudêncio

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Pedro Souto

External Examiner: Pedro Abreu

Supervisor: Ana Aguiar

18th June, 2012

Abstract

The ageing population has been increasing in the last few years. The life expectancy is today much higher, contributing to the appearance of chronic diseases in a great part of the population. The elder ones, the handicapped and those suffering from chronic illness need a constant monitoring of their vital signs. Mobile Healthcare is an emerging market that comes as an alternative to the traditional healthcare methodology and aims to reduce the number of caretakers needed and to improve the quality of life of the population.

However, most of the applications existent in the market are closed and proprietary vertical solutions, not easily extended or integrated with other services. Machine-to-Machine Communications is a type of communications characterized by a large number of devices interconnected and communicating without the need of human intervention. It aims at being interoperable and its potential is clearly evidence by the standardization efforts from ETSI, 3GPP and IETF. The main purpose of this thesis is to develop an application that demonstrates the features of M2M leveraging an existing infrastructure of a major Telecommunication Company and enables remote monitoring of vital signs by multiple parties using the same infrastructure. The smartphone will be used, taking advantage of the available sensors, and a sensor for the heart rate will be added, to monitor the position, physical activity and heart rate. All this data will be sent to the parties interested, a relative or a caretaker, using the paradigm publish/subscribe available with the Extensible Messaging and Presence Protocol (XMPP).

A remote service able to recognize the physical activity being performed by a user was developed using the acceleration and rotation returned by embedded sensors of the smartphone. Based on previous studies in this area, a set of features are extracted from the acceleration. Initially a rotation of the acceleration data is performed using the rotation information. The rotated data is used as an input to an Autoregressive model in order to obtain coefficients that represents the signal of acceleration. Moreover, an integration is performed to obtain the velocity and total displacement traveled. After extracting these features the data is sent to a classifier - an Artificial Neural Network. The classifier after an initial phase of training will be able to distinguish between 5 different activities: standing, walking, running, walking upstairs and walking downstairs.

Resumo

O envelhecimento da população tem aumentando significativamente nos últimos anos. A esperança média de vida é hoje muito maior, contribuindo para o aparecimento de doenças crónicas em grande parte da população. Os idosos, os deficientes, ou aqueles que sofrem de alguma doença crónica precisam de uma monitorização contínua dos seus sinais vitais. "Mobile Healthcare" é um mercado emergente que é uma alternativa para a metodologia tradiocional aplicada na área da saúde e tem como objectivo reduzir o número de prestadores de cuidados necessários e para melhorar a qualidade de vida da população.

No entanto, a maioria das aplicações existentes no mercado são fechadas e soluções verticais proprietárias, que não são facilmente estendidas ou integradas com outros serviços. "Machine-to-Machine-Communications" é um tipo de comunicação caracterizada por um grande número de dispositivos interligados e comunicando entre si sem a necessidade de existir intervenção humana. É amplamente conhecido por ser interoperável e tem um potencial de crescimento claramente evidenciado pelos esforços de padronização da ETSI, 3GPP e IETF.

O objetivo principal desta tese é desenvolver uma aplicação que permite a monitorização remota de sinais vitais por várias partes, utilizando a mesma infraestrutura. O smart phone será utilizado, aproveitando-se os sensores disponíveis e um sensor para a frequência cardíaca será adicionado, para monitorar a posição, a atividade física e a frequência cardíaca. Todos esses dados serão enviados para as partes interessadas, um parente ou um prestador de cuidados, usando o paradigma "publish/subscribe" disponível com o protocolo "Extensible Messaging e Presence Protocol" (XMPP).

Foi desenvolvido um serviço remoto capaz de reconhecer a actividade física que um utilizador está a praticar utilizando a aceleração e rotação, valores obtidos através de sensores incorporados no smartphone. Baseado em estudos anteriores nesta área, um conjunto de características são extraídas a partir da aceleração. Inicialmente uma os dados da aceleração sofrem uma rotação através da informação obtida do sensor de orientação. Os dados rodados são utilizados como entrada de um modelo Autoregressivo, com o intuito de obter os coeficientes que representam o sinal da aceleração. Além disso, uma integração é realizada para obter a velocidade e o deslocamento total percorrido. Após extrair estas características, os dados são enviados para um classificador - uma Rede Neuronal. O classificador após uma fase inicial de treino será capaz de distinguir entre 5 actividades diferentes: em pé, andar, correr, subir escadas e descer escadas.

Acknowledgements

Along the development of this thesis I had the valuable collaboration of teachers, family, friends and colleagues who, directly or indirectly contributed to my work and made me believe in this project.

Firstly, a special thanks to my supervisor, Ana Aguiar who, for her commitment, help, and expertise in the area of M2M and Machine Learning made this project possible. The countless hours spent for my guidance and wise counsel, made this idea a good contribution in area of research of mobile healthcare. I thank also the important help in data collection for the training algorithm to recognize physical activities.

A word of appreciation also to Daniel Lucani, my co-supervisor, for the availability and the know-how in the area of Signal Processing and Machine Learning.

This project also would not be possible without the collaboration of the Instituto de Telecomunicações (IT), the cession of the space and the material that was essential to complete the dissertation. A big thanks also to PT Inovação, in particular to Telma Mota and Filipe Pinto for useful advice and ideas and to Edgar Domingues and João Gonçalves that in addition to the availability, have been tireless in helping to solve problems related to the MyContext Framework.

In addition, I am deeply grateful to my friends willingness, namely Sofia Maciel, André Figueroa and João Campos, that spent an afternoon running and walking upstairs and downstairs, to collect data. A very special thank also for all my friends in general, each in its own way, make my daily life unique and unforgettable.

Continuing, I can not forget the contribution of my parents, encouragement and motivation in this final phase of the course and throughout my academic life in general. I thank them all the dedication, companionship, advice and guidance that help me along the development of this thesis.

Last but not the least, a special thanks to Marta for her willingness and readiness to cooperate and help in this thesis from the beginning. One of the most obvious contribution was the collection of data, but more importantly was the privilege to share the frustrations, difficulties and successes everyday. Her support and optimism were certainly a great inspiration during this period.

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Goals	2
1.4	Outline	3
2	State of the Art	5
2.1	Mobile Healthcare	5
2.2	Machine-to-Machine Communications	7
2.2.1	M2M Standardization	8
2.2.2	XMPP	10
2.3	Human Activity Recognition	13
2.3.1	Features extraction	15
2.4	Classification	19
2.4.1	Artificial Neural Networks	19
2.4.2	Support Vector Machines	24
2.5	Geographic Fencing	24
2.6	Summary	26
3	Problem Formulation	29
3.1	How to use an M2M Platform in an healthcare scenario	29
3.2	Detection of Physical Activity using a smartphone	31
4	An M2M Healthcare application	33
4.1	Existing Platform	33
4.2	Application Design	36
4.2.1	General Architecture	36
4.3	Implementation	37
4.3.1	Context information	37
4.3.2	Source Providers	40
4.3.3	Composite Providers	40
4.3.4	VitalTracker application for mobile phones	41
4.3.5	Vitaltracker application for tablets	45
5	Physical Activity Detection using Smartphones	49
5.1	Methodology	49
5.2	Sensors available	50
5.3	Data collection	51

CONTENTS

5.4	Features extraction	51
5.4.1	Rotation of the accelerometer data	53
5.4.2	Autoregressive modeling	54
5.4.3	Extraction of velocity and displacement	57
5.5	Classification	59
5.5.1	Artificial Neural Network	59
5.5.2	Standardization	60
5.5.3	Metrics	62
5.5.4	Training and testing	63
6	Results	67
6.1	Remote M2M Application	67
6.2	Physical Activity Detection using Smartphones	68
7	Conclusions	77
	References	79

List of Figures

2.1	An architecture of mobile healthcare systems [Shi11a]	6
2.2	Different stages in the healthcare model [Shi11b]	7
2.3		8
2.4		9
2.5	Message exchanges in a standard instant-messaging session [SA05].	11
2.6	Comparison between polling and a pubsub system [SAP09].	12
2.7	Acceleration signal recorded by a capacitive accelerometer for each sensing axis during cycling, rowing, and exercising on an elliptical trainer [BGYW09].	14
2.8	Comparison among the magnitude spectra of the prediction residuals obtained using Burg’s method, the warped Yule-Walker method, and the warped Burg’s method.	17
2.9	Uniaxial accelerometers measure the component a_u of an equivalent acceleration a in the direction U of the sensitive axis of the accelerometer [JKC07].	17
2.10	Example of the usage of the tilt angle to distinguish between some static postures, with the accelerometer position in the chest of the user [KNM ⁺ 06].	18
2.11	3-D feature plot for the four transitions after the LDA [KLLK10].	19
2.12	Biological neuron [JMM96]	20
2.13	Analog Schema of a Biological neuronal network [Gra07]	20
2.14	An example of neural network with three intermediate layers. [PM92]	23
2.15	Sigmoid function [Fau93].	23
2.16	Example of a separable problem in a 2 dimensional space [CV95].	25
2.17	Simple polygon with a point inside and another outside of its area [JZ11].	26
2.18	Example of geo fencing application [WWL ⁺ 09].	26
3.1	Remote monitoring systems of today.	30
3.2	Remote monitoring systems using M2M.	30
4.1	Architecture of the context framework [JG11]	35
4.2	General architecture proposed for the monitoring system.	37
4.3	Block diagram of the vitaltracker application.	42
4.4	Zephyr heart rate sensor communicating with a mobile phone.	43
4.5	Screenshots of the VitalTracker application.	46
4.6	Screenshots of the VitalTracker application for tablets.	47
5.1	Block diagram of the physical detection system developed.	50
5.2	Plot of the acceleration in each axis, extracted by the Android application for the physical activity standing.	52
5.3	Axis relative to the smartphone.	53
5.4	Android application screenshots to collect data.	54

LIST OF FIGURES

5.5	Comparison of the acceleration signal before and after a rotation.	55
5.6	Transform matrix.	56
5.7	Evolution of the error of the autoregressive modeling when increasing the order.	57
5.8	Plot of the original signal, velocity signal after integration and the displacement after double integration.	58
5.9	Iterative process to decide the number of hidden layers and neurons.	61
5.10	Example of ANN with 2 layers each one with 2 neurons, 15 input neurons and 5 output neurons.	62
5.11	Example a confusion matrix applied to the classification of physical activities.	63
5.12	Example an ANN with one layer and one hidden layer generated by Neuroph Studio.	64
5.13	Plot of the evolution of the error in the training phase generated by Neuroph Studio.	64
5.14	Confusion matrix of the classification process an Artificial Neural Network developed on Neuroph Framework.	65
5.15	Confusion matrix of the classification process using an Artificial Neural Network on R	65
5.16	Confusion matrix of the classification process using Support Vector machines on R	66
6.1	Test of the system developed, using two mobile phones as context sources, and another mobile phone and a tablet as context consumers.	69
6.2	Vital Tracker application in a tablet, monitoring two users and showing a geo fence alarm.	70
6.3	Route at FEUP.	71
6.4	Confusion matrix of the test using all the data collected by the 5 individuals.	71
6.5	Confusion matrix of the test using the data from the subject that trained the network, using the same smartphone that was used in training too.	72
6.6	Evolution of the physical activities along the route for the subject and smartphone used in training.	72
6.7	Confusion matrix of the test using the data from the subject that trained the network, using a different smartphone.	73
6.8	Evolution of the physical activities along the route for the subject used in training, with a different smartphone.	73
6.9	Evolution of the physical activities along the route for a subject that was not used in the training phase.	74
6.10	Confusion matrix obtained after training the network with the data from the 5 individuals traveling along the defined route using LG smartphone.	74
6.11	Confusion matrix obtained after training the network with the data from the 5 individuals traveling along the defined route using the Samsung smartphone.	75

Abbreviations

XMPP	Extensible Messaging and Presence Protocol
M2M	Machine-to-Machine
GPRS	General Packet Radio Service
IETF	Internet Engineering Task Force
TCP	Transmission Control Protocol
SSL	Secure Sockets Layer
DNS	Domain Name System
IP	Internet Protocol
DMBS	Database Management System
ADMBS	Active Database Management System
HAR	Human Activity Recognition
LDA	Linear Discriminant Analysis
ANN	Artificial Neural Network
ADALINE	Adaptive Linear Neuron
LMS	Least Mean Square
GPS	Global Positioning System
PIP	Point in Polygon

Chapter 1

Introduction

There have been several efforts in the area of Mobile Healthcare for remote monitoring, which consists of a system where the patients wear sensors that transmit the sensor data to a mobile phone or other device, which forward it to a backoffice. Ageing population is affecting the society, and there is a lack of personell to take care of the ageing. Furthermore, the number of chronic diseases are increasing, leading to a great number of people that requires continuous monitoring. Mobile Healthcare is an alternative to the traditional healthcare methodology, helping to reduce the number of caretakers needed and improving the quality of life of the population.

Parallel to this situation, is the emerging market of Machine-to-Machine communications (M2M). By definition, this type of communications is characterized by a set of devices interconnected, communicating autonomously without human intervention. A typical use case of this technology is in the scenario of Healthcare [LLL⁺11], where vital signs are monitored, enabling remote monitoring and diagnosis. However, most of the solutions that are in the market today are closed and proprietary. Each solution is very specific, not easily adaptable to the needs of all people, and is not reusable. Moreover, in this kind of applications the security, reliability and efficiency of the communications are crucial, since human lives are at stake.

The general purpose of this thesis is to develop an application for Remote Healthcare, using Machine-to-Machine communications (M2M). The technologies to be used must be open, interoperable and easily extensible. In order to demonstrate the advantages of this architecture services related to health and wellbeing monitoring will be developed like: geo fencing, human activity recognition and heart rate monitoring.

1.1 Context

“The increasing feasibility and convenience of mobile healthcare has already introduced several significant challenges for healthcare providers, policy makers, hospitals, and patients.” [RPB10]

The ageing population has been increasing in the last few years. According to the World Health Organization, the proportion of people over 60 is growing much faster than any other group of people [Org04]. This fact is a result of the longer life expectancy and better healthcare resources and techniques. Related to the life expectancy is the appearance of much more chronic diseases. It is estimated that 48% of the population have at least one chronic ailment [Shi11a].

The elder ones, the handicapped and those suffering from chronic diseases need constant monitoring. In the actual healthcare in public and private institutions, that means that domicile visits to the house of each patient must be done frequently or the person must frequently visit a doctor. The health expenditures are increasing, leading to a health crises in the actual days. In the United States, it is estimated that 15% of the spendings are related to healthcare in 2010 [fMS11]. Considering this, there is an huge impact, both economical and social, of the techniques and methods employed in this crucial area of the society. To face this difficulties, the new market of Mobile Healthcare is arising.

1.2 Motivation

Although there are already some applications in the area of Mobile Healthcare, there are issues related to interoperability. The solutions existent in the market are not easily extended, re-used or changed by a third party. Moreover, there is a lack of standards and compatibility between existent systems [Shi11b]. Most of them are closed and proprietary vertical solutions. The solutions existent in the market use expensive hardware and software, applied to a specific health condition of a patient. However, the proliferation of smart phones offer an opportunity to change this situation. Most smart phones come with a wide range of integrated sensors, like accelerometer or GPS. Likewise, the price of devices is decreasing and one can buy a smart phone with all the necessary features for a low price. Therefore, a growing share of the population has a smart phone and the healthcare software should take advantage of these devices.

In addition, the developers must be aware that the mobile phone will be used in a daily life routine, so the monitoring algorithms must be adaptable to this situation. The energy efficiency, security and reliability are important requirements that cannot be forgotten when developing software to be used in a healthcare scenario. The smallest ambiguity in transmitted data can lead to a fatal consequence. Furthermore, the latency of the transmission must be minimal since monitoring a patient is an activity that must be done in near real time.

1.3 Goals

The main purpose of this thesis is to develop an application that enables remote monitoring of vital signs by multiple parties using the same infrastructure. The Extensible Messaging and Presence Protocol (XMPP) will be used as the communication protocol, because it is an open technology for real-time communication, which powers a wide range of applications and that can be used to implement M2M communications. A framework to enable the communication between multiple

Introduction

devices following the XMPP protocol already exists, and was developed by PT Inovação. It is a distributed application that consists of multiple modules, one of them being a smartphone application. The sensors of each mobile phone will be used, mainly the accelerometer and GPS. Additionally another sensor will be used to monitor the heart rate of the patient.

On the other hand, there are a set of services that will consume this data, and generate important information and alarms. The main services are:

- Heart abnormalities detector: The heart rate should be used to detect a tachycardia or a bradycardia. Moreover, the information from the other sensors can also be used, in order to detect if a patient is in rest or doing some activity. For instance, if the patient is resting and the heart rate is very high something must be wrong.
- Activity Monitoring: Using the accelerometer data the type, duration and intensity of different physical activities must be monitored. If the patient is doing an activity that is not adequate for his health condition, or if the patient is in rest for a long time, warnings are generated. Moreover, the information relative to the heart rate, should also be used in order to achieve a better accuracy. The purpose is to use techniques of machine learning in order to recognise a set of activities.
- Geo-Fencing: A geographic barrier will be specified by the caretakers, if the patient breaks these barrier an alarm will be sent. This kind of service is useful, for instance, for elderly users with dementia or chronic disorientation.

Note that, these services will be developed as a proof of concept for an M2M architecture. Since the purpose is not to obtain a Medical certification, but to propose an example of application and enable extensibility to new services.

In order to use these services, two different agents will be developed. The first will be an Android agent for a smartphone, that will enable the patient to be monitored. Another functionality is to track the vital signs and to receive alarms of a relative, using the same application. For the caretakers, a tablet application will consume the sensor data, and it will be possible to monitor multiple patients, simultaneously sharing the same infrastructure independently of the hardware.

Concluding, the purpose is to use open technologies and standards, enabling the interoperability of the agents and services. In the area of human activity recognition, the main goal is to demonstrate the application in a real daily life routine, the patient will use the smart phone in his pocket as usual, and 5 different physical activities will be recognized: standing, walking, running, walking upstairs and downstairs.

1.4 Outline

The rest of this thesis is structured as follows:

Chapter 2 introduces the fundamental concepts in the area of Machine-to-Machine Communications and the ongoing process of standardization. The state of the art in the algorithms of detecting

Introduction

of automatic activity recognition and fencing are presented next.

Chapter 3 presents the problem formulation in the area of M2M applications and the problem of physical activity recognition.

Chapter 4 explains the M2M healthcare application that was developed and how the integration with an existing platform was done.

Chapter 5 describes the work done in the area of physical activity detection, presenting the techniques and algorithms used to accomplish this functionality.

Chapter 6 presents the results obtained in recognizing the physical activities, and shows the final application that integrates a service to detect the activities, geographic position and heart rate.

Finally, chapter 7 presents the conclusions.

Chapter 2

State of the Art

In order to understand the relevance of this thesis, it is essential to explain first some concepts. Firstly, the emerging area of Mobile Healthcare, and the different example of scenarios, are explained. One of the main purposes of this thesis is to use a widely known communications paradigm known by Machine-to-Machine. It will be explained how it works, example of applications will be listed, as well as, the XMPP protocol will be explored and standardization activities.

Finally, the existing algorithms of automatic detection services will be explored. For instance, the state of the art in the area of human activity recognition and Geo-fencing will be covered. Moreover, algorithms for machine learning, more specifically artificial neural networks, are described, because they will be the first approach.

2.1 Mobile Healthcare

Healthcare consists in a set of procedures in order to diagnosis, treat and prevent diseases. Nowadays, the Healthcare system is extremely expensive, as explained before, since there are much more chronic diseases and the population is getting older. In order to achieve a better quality of service in healthcare and at the same time reduce the costs, an emerging market is rising: the Mobile Healthcare market. M-Healthcare consists of the activities of healthcare empowered by the computer technologies and communications. In figure 2.1 it is possible to identify the different components usually present in a mobile healthcare system. It is crucial that the patient is connected to a device that communicates with a central server that treats the incoming messages with information related to the health status of the patients, and that relevant information is forwarded to medical staff ready to respond to a specific situation [Shi11a].

The practice of healthcare can be divided in different stages: data acquisition, interpretation and decision-making on treatment/prescription. In figure 2.2 it is possible to analyze the different stages in the medical practice. The evolution of technology and science enables the acquisition of

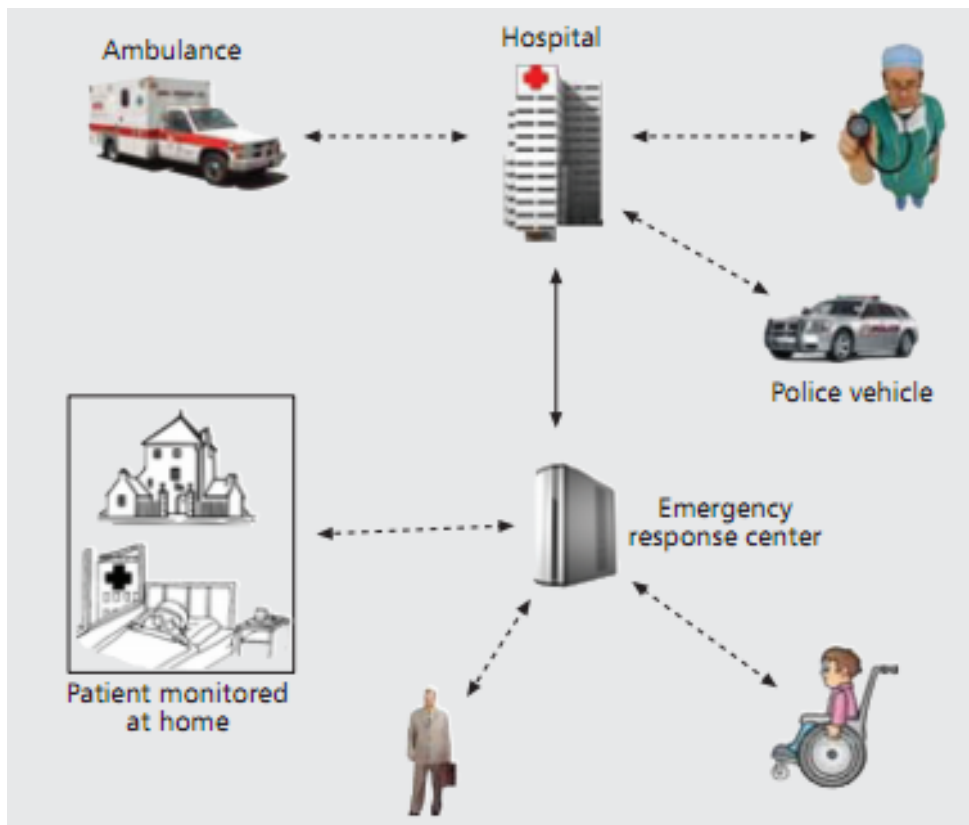


Figure 2.1: An architecture of mobile healthcare systems [Shi11a]

multiple vital signs, that contains important information for the patient health. Moreover, different services or machines process the data, like the heart beat, sweat, pulsation, X-ray images to produce richer information on the user's health. All this data must be decoded and converted to human-readable information. Based on this information, and the patient's history the doctors try to give a diagnose, sometimes using as an auxiliary method decision making software [Shi11b].

Recent enhancements of communication and computing enable the mass deployment of m-health systems. The European Commission efforts to create policies and strategies to globally implement telemedicine systems in crucial areas is a signal of change [Shi11a]. Moreover, the devices are increasing their miniaturization, capacity, computational power and the data rates of wireless communications are accelerating. Mobile systems can be divided in two distinct communication paradigms: current 2.5G and 3G systems; beyond 3G mobile systems [MTIPL06]. In this decade a boom of wireless technologies happened. For instance, the General Packet Radio Service (GPRS), which is a packet oriented service that transmits data over 2G communications system. The increasing of mobile devices used in healthcare scenarios can be explained by the quality of service gained in the usage of these systems, because of the huge amount of medical information that are obtained, in near real time. Moreover, less personnel is needed and as a consequence the costs are reduced [Shi11b]. On the other hand, there are still a lot of challenges in this market. The lack of standard and global policies in the usage of this system has a great

State of the Art

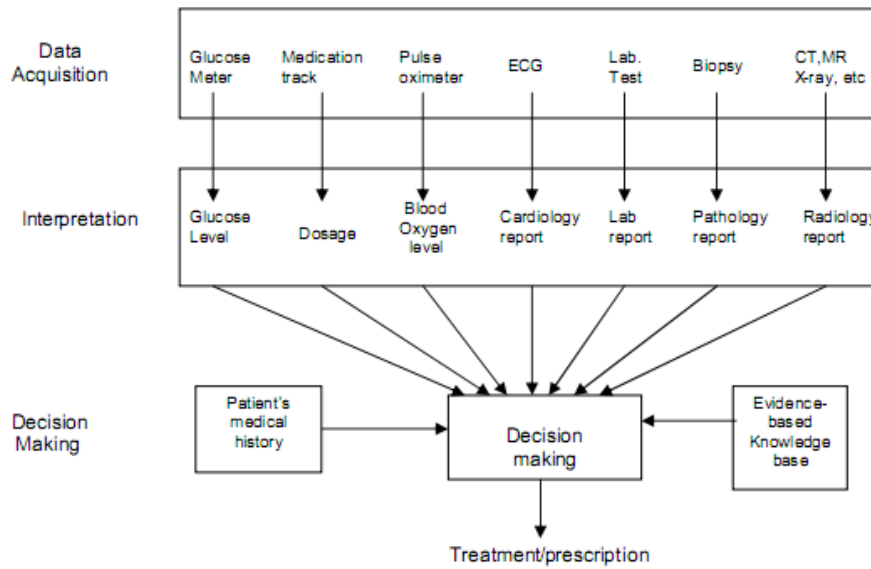


Figure 2.2: Different stages in the healthcare model [Shi11b]

impact. Moreover, the compatibility of mobile systems with the already deployed systems is a an issue not easily solved. Although, theoretically costs are reduced because of the automation of the processes, the 2G and 3G communications have a high cost. Besides the technological issues, the staff have few knowledge of using mobile technology, and the healthcare industries don not feel comfortable do adopt them [MTIPL06]. It is expected however that in a near future the 4G technologies will dominate the communications market, integrating the existent technologies like GPRS, 3G, Bluetooth, and many others. Moreover, it will support multimedia services at an high rate of transmission, maintaining a low cost, enabling the mass deployment of telemedicine services. The focus of this advanced technology will be to integrate multiple services, in order to use the one that are available at the time. So, it will enhance the personalization and high usability of the systems [MTIPL06].

2.2 Machine-to-Machine Communications

Machine-to-Machine Communications (M2M) is characterized by involving a large number of intelligent devices interconnected, communicating with each other, taking decisions autonomously and without human intervention. The main idea is to enable that the components of this architecture maintain interconnected, highly scalable and guarantee interoperability, with a low-cost associated. M2M is an emergent market that has a wide range of applications in pervasive and ubiquitous scenarios [LLL⁺11]. The proliferation of the Internet services and higher broadband available at a lower cost are factors that motivates the investment in areas like M2M and cloud computing, that can enhance the revenues of the companies, offering new services to the market.

Ubiquitous connectivity of billions of devices connected, is also known as "Internet of Things" and is explained by three reasons [WTJ⁺11]:

- Large number of devices: components of M2M are available in the market, at any range of prices;
- Scalable connectivity: all the devices can be connected;
- Cloud-based services: distributed processing by all the devices working together.

The general architecture of a M2M network is present in figure 2.3 and is described by the following components:

- M2M node: replies and transmits the data;
- M2M area network: area of connectivity between the nodes. Example: Personal Area Network;
- M2M gateway: ensures that the devices are working together and that exists connectivity to the network;
- M2M wireless network: provides connection between the gateway and the application domain;
- Application domain: processes the information depending of the scenario of the business process.

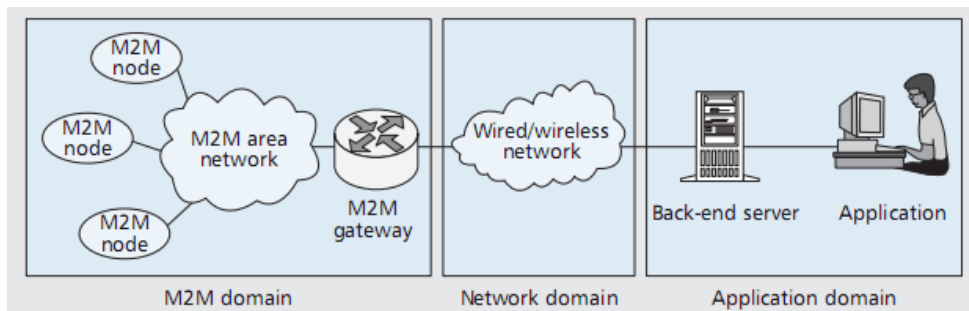


Figure 2.3

2.2.1 M2M Standardization

The growing potential of M2M communications is clearly evidenced by the standardization efforts from several bodies, including ETSI [TS10b] [TS10a], 3GPP [TR11a] [TR11b] and IETF [IET08] [IET10b] [IET10a].

An overview of the architecture proposed by ETSI is present in figure 2.4. Basically, is proposed an architecture based on existing standards by Institutions like IETF and 3GPP respecting

State of the Art

the communication protocol. On the other hand, another layer respecting the device domain is established, where standards like Zibbee, M-Bus, and others are integrated in the field of the communications between the devices.

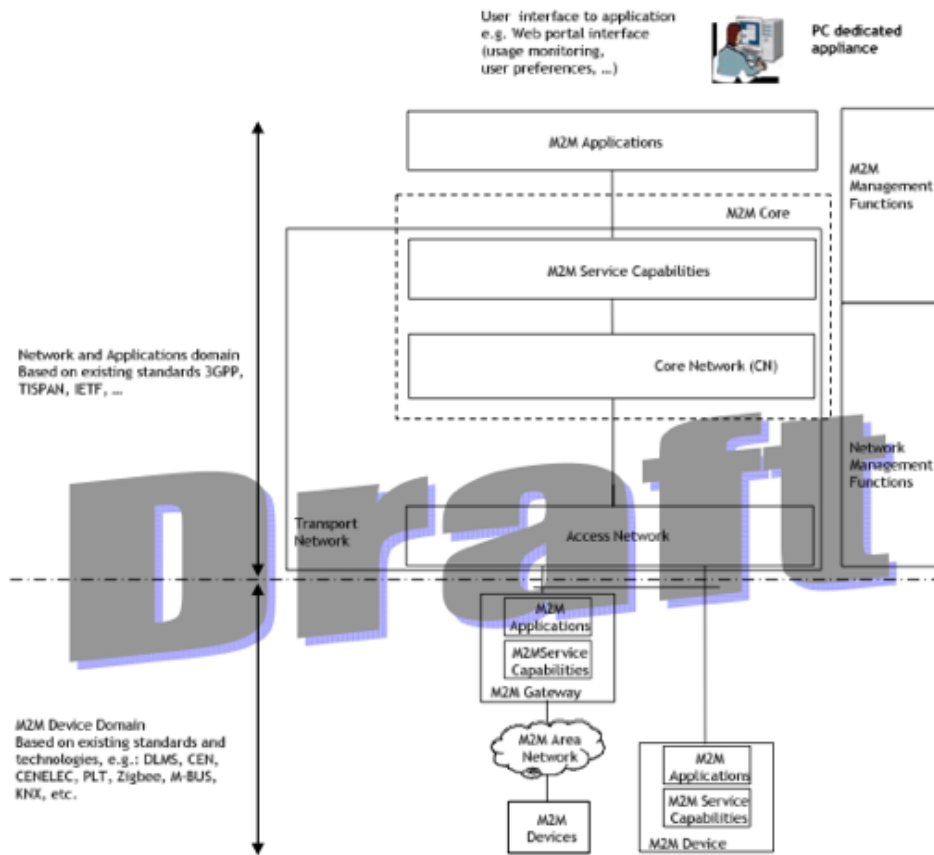


Figure 2.4

In the field of applications of remote healthcare using M2M, there are also efforts in order to define standards for the development and deployment of these applications [ETS11]. Usually there are multiple stakeholder in the system:

- Patient: is the individual of interest, being monitored and transmitting the data related to the vital signs;
- Remote Monitoring Device: device with one or more sensors, collecting the patients information and sending the data through via a gateway;
- M2M service capability provider: provides communication services to the applications, supporting specific functionalities;
- Application entity: application that uses all the information in order to help the caretaker to monitor specific health conditions;

2.2.2 XMPP

The Extensible Messaging and Presence Protocol (XMPP) is an open technology aiming to provide functionalities of presence-awareness and real-time communication using the Internet. It was originally developed by the open-source community Jabber with the purpose of providing an open protocol for Instant Messaging [SA09]. There have been efforts in standardization by the Internet Engineering Task Force (IETF) producing RFC documents [And04a] [And04b] and the XMPP Standards Foundation also published multiple extension specifications [And10]. Furthermore, this technology enables the exchange of complex messages, not only text, like geographic location, activity or streaming of video. Since this protocol is based on the markup language XML, it widely enables extensibility, being easy to create new payloads to send over the network. XMPP is based on the Transmission Control Protocol (TCP), and the payload of the messages are in XML. Regarding security issues, a layer of secure sockets layer is used (SSL). [SA05]. XMPP applications follows a client-server paradigm, where multiple messages are exchanged in order to established a connection, and to authenticate the clients. When streaming XML documents, the file respect some specific language. The basic element of this language is the stanza, which are words in XML that have a specific meaning. One can use the following stanzas in a XML document in the context of XMPP:

- <message/> when an entity sends information to another;
- <presence/>used to inform the status of a user, associating optional information if desired;
- <iq/> request-response mechanism, enabling entities to request some kind of information.

Figure 2.5 illustrates the different interactions and message exchanges in a typical instant-messaging session. The user sends a stream to the server and then exchange messages and presence status among other users.

A core concept of XMPP tecnology is the JabberID(JID), which relies on the Domain Name System(DNS) in order to identify and address the messages to a specif location through the Internet Protocol(IP). Put in other words, is just a synonym to an IP, aiming to be more user-friendly and easily remembered. Each user must be associated to a JID and can also provide a resource or an identifier to a particular connection. For instance, one can use the JID "user@domain.tld" and the resource "pc-home" [SAP09].

2.2.2.1 Publish Subscribe

Most software solutions that need to be aware of a change in another system usually adopt the strategy of polling loop. This strategy consists in making an HTTP Request at a predefined interval until the modification in the system happens. This approach is time consuming and it is heavy for the server that has to handle all these requests. Instead of polling, exist an extension in XMPP that is based on the design pattern "publish/subscribe" or "observer". Basically, it is possible to create a node at a pubsub service, and another client can subscribe to that node. Whenever someone publish

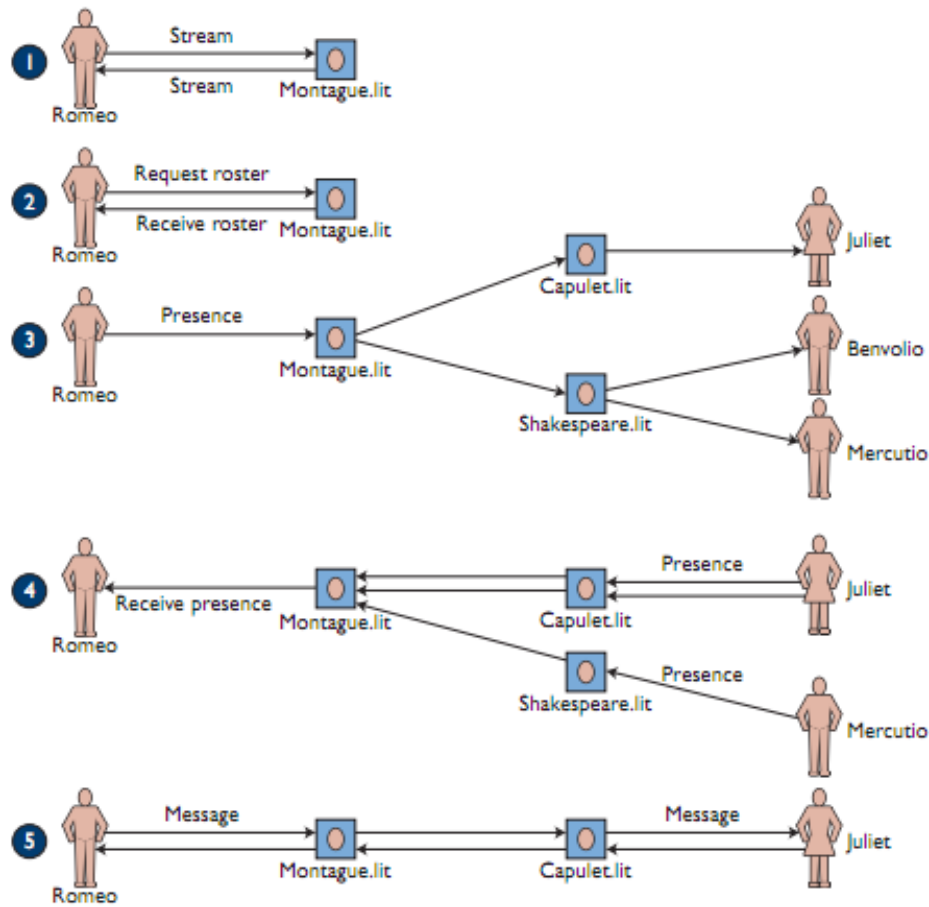


Figure 2.5: Message exchanges in a standard instant-messaging session [SA05].

something to the node, all the clients that had subscribed before, receive a notification [Mil10]. Figure 2.6 demonstrates a sequence of messages in each paradigm: polling and publish/subscribe. A simpler variation of the publish/subscribe paradigm exists in the presence extension, where the users subscribe to the presence status of other users [SA06]. An alternative solution to solve the issue of polling is to use Active Database Management System(ADMBS), creating a higher layer in simple Database Management System(DMBS), to enable handle of events and the configuration of action when some specific event occur [Hoa12]. However, this approach do not use the XMPP protocol and is not easily extensible. In order to create a new event, a trigger must be created in the database, not allowing the distributed services consuming the information as in M2M paradigm.

To better understand the messages that must be sent in order to implement a pubsub client-server application, a sequence of messages are shown bellow. First, a node must be created, giving a name to it and specifying the server that will host this node. After creating the node, any user depending on the configurations of the server and the node [Mil10], can subscribe to it.

```

1 <iq from="john@porto.it"
2   id="asfg45"

```

State of the Art

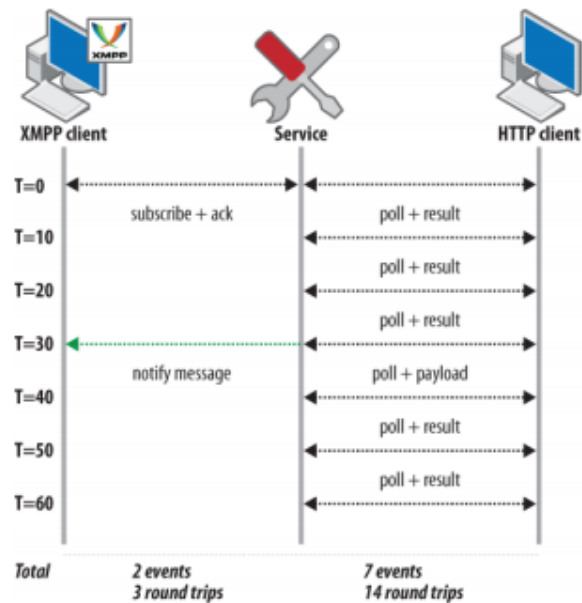


Figure 2.6: Comparison between polling and a pubsub system [SAP09].

```

3   to="pubsub.porto.it"
4   type="set">
5   <pubsub xmlns="http://jabber.org/protocol/pubsub">
6     <subscribe node="example-node" jid="john@porto.it"/>
7   </pubsub>
8 </iq>

```

Any user can now publish information, depending also on the configurations of the node. The message sent can have a payload or not. In the example below, a simple payload including a simple status field is sent.

```

1 <iq from="mary@porto.t"
2   id="ovo45"
3   to="pubsub.porto.it"
4   type="set">
5   <pubsub xmlns="http://jabber.org/protocol/pubsub">
6     <publish node="example-node">
7       <item>
8         <there xmlns="http://porto.it/simple-node" status="true"/>
9       </item>
10    </publish>
11  </pubsub>
12 </iq>

```

Finally, all the users that had previously subscribed to the node will receive the information.

In this example, the user john had subscribed so he will receive the information indicating that the status has changed to true.

```

1 <message from="pubsub.porto.it" to="john@porto.it">
2   <event xmlns="http://jabber.org/protocol/pubsub#event">
3     <items node="example-node">
4       <item id="ov263">
5         <there xmlns="http://porto.it/simple-node" status="true"/>
6       </item>
7     </items>
8   </event>
9 </message>

```

2.3 Human Activity Recognition

Physical activity can be described as any movement that needs muscle contraction. When practising any physical activity an amount of energy is expended, depending on the duration, type and intensity of the exercise. Nowadays, it is well known that doing some kind of activity is essential for health promotion and preventing diseases. Thus, one of the goals of the caretakers is to promote healthy habits, adequate for the health status of each individual.

In order to monitor the patients physical exercise, the field of automatic recognition of this activity by means of technology is an increasing scientific area of study. One method to achieve a recognised activity is to use a camera and with algorithms in the area of Computer Vision retrieve the patterns associated with each activity. However, this techniques only achieves a good accuracy when tested in a laboratory with controlled objects. Although many investigations have been done in this arena, techniques that achieve a good accuracy are not yet available [ATKI08]. Moreover, since the purpose of this thesis is to develop a system to monitor the patient in his daily routine, maintaining the quality of life of the individual using unobtrusive techniques, cameras are not an option.

Accelerometry comes as an alternative to the Computer Vision area, because an accelerometer is a portable sensor readily embedded in state of the art mobile phones and can achieve a good accuracy. Accute measures largely depends on the number of accelerometers used, position of the body and other factors. There are two kinds of accelerometers: piezoresistive and capacitive [BGYW09]. Piezoresistive accelerometers were the first kind of this type of devices, and can detect dynamic forces caused by the movement. However, it is not possible to detect the orientation of the body when doing static activities like sitting or laying down. Further research concludes that capacitive accelerometers can detect both forces acting on the system. They use the principle of change in resistance applied by the forces. Although these devices are less obtrusive and can achieve better results, they have a shorter battery autonomy. In figure 2.7 the raw data that the is retrieved from the capacitive accelerometer, while doing different activities, is presented.

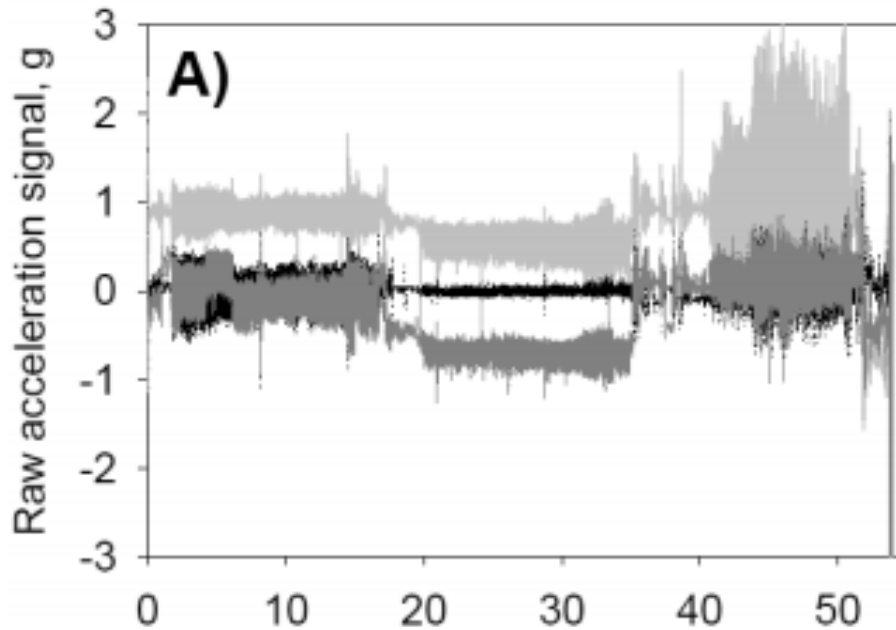


Figure 2.7: Acceleration signal recorded by a capacitive accelerometer for each sensing axis during cycling, rowing, and exercising on an elliptical trainer [BGYW09].

There are many methodologies in the field of human activity recognition (HAR). The purpose of this chapter is to present some of the more advanced techniques that achieved good results. The automatic detection of physical activity usually starts with the training phase which includes three stages:

- Data collection: Retrieving the raw data from the accelerometer;
- Processing the data, extracting useful features;
- Train the classifier using the features and indicating the output in each situation (the physical activity).

The data collection usually is done in a controlled situation, where the individual records the data from the accelerometer while doing pre-defined situations. For instance, the individual walks for 30 seconds, then runs for another 30 seconds and so on. The data collected in this phase should be used for the following ones. The raw data is a signal for each axis, as demonstrated in the figure 2.7, with the duration of the physical activity.

After the training phase, the trained classifier can be used to identify the activity that is being practiced by the user. The stages in this phase are similar, but the classifier is now on executing mode:

- Data collection: Retrieving the raw data from the accelerometer;
- Processing the data, extracting useful features;

- Send the features to the trained classifier;
- Recognise the physical activity.

2.3.1 Features extraction

In the phase of extracting features, some authors propose to use three different features: autoregressive modelling; Signal magnitude area; Tilt angle [KLK08].

2.3.1.1 Autoregressive model

An autoregressive model is used to predict multiple situations. It belongs to the group of linear prediction calculations, aiming to predict the output of a system with the knowledge from previous outputs. The forecasting problems uses times series data, which is a set of relations between an input and output. The data must be obtained at a constant rate in order to achieve a consistent model. An example of a times series is in 2.7. One of the important phases is the selection and fitting of the model, consisting in choosing one method, and estimating the unknown parameters.

“Regression analysis is a statistical technique for modeling and investigating the relationships between an outcome or response variable and one or more predictor or regressor variables.” [Mon08]

An auto-regressive model calculates a number of coefficients, depending of the order that is used. The expression of this model is the following:

$$y_n = - \sum_{m=1}^p a_m y_{n-p} + e_n \quad (2.1)$$

Where y_n are the signal samples, a_m are the model coefficients, and e_n is the residual. There are various techniques to calculate the parameters of this series: The least squares method; Yule-Walker method; Burg’s method.

Every regression model has one or more prediction variables and regression coefficients. The coefficients are unknown and must be calculated through sample of data. The error means the deviation between the original data and the model. The following formula generalizes a standard equation for a multiple linear regression model, it can be generalized to a simple model only with one predictor variable [Mon08].

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (2.2)$$

One method well known to solve this equation is the least squares method. This technique chooses the values for the coefficients in order to minimize the sum of the squares of the errors. The least-squares method, does not guarantee the correctness of the model, because of small changes in estimations may lead to wrong values [MDHD96].

On the other hand, the Yule-Walker method consider the first and last data points in the summation, and uses an autocovariance function to replace these theoretical values with estimated ones [MDHD96].

$$y_m = \sum_{k=1}^p \varphi_k y_{m-k} + \sigma_\varepsilon^2 \delta_{m,0} \quad (2.3)$$

This method should not be used if the autocovariance matrix is not well conditioned, which can result in a model that has wrong values due to a poor estimation of the parameters.

Finally, the Burg's method starts estimating the reflection coefficients, instead of estimating the autoregressive parameters directly. These parameters are estimated using the Levinson-Durbin algorithm. Burg's method is the one that achieves a high reliability, leading to a stable model [RKEV03] [MDHD96]. In the figure 2.8 it is possible to verify the differences between the three methods, mainly respecting to the prediction residuals obtained in each one.

2.3.1.2 Tilt Angle

Another feature that can be useful to distinguish between a set of different activities is the Tilt Angle. These values correspond to the angle between the vector of the gravity (g) and the z axis. It is an important feature to distinguish between static activities, for instance lying and sitting [KNM⁺06] [JKC07]. Note that the human body is constituted by a set of rigid bodies that are connected through the joints. When in a static position the orientations of these body parts do not change significantly, and this fact can be used to detect static positions. Another important notion is that this feature only works well when there is not acceleration in the environment, like when you are sitting in a car [VBdV⁺96]. This angle can be calculated through the following expression:

$$\Theta = \arccos(z) \quad (2.4)$$

Figure 2.9 illustrates how this angle is extracted from the system. Basically, the component of the acceleration in the direction of the z axis is measured. As an example, in figure 2.10 a set of static activities are listed, and it is explained how to use the tilt angle to recognize these activities.

2.3.1.3 Signal Magnitude Area

The activities in rest and in movement, have a completely different signal. It is intuitive that the degree of accelerations will be much smaller when resting. So, the measure of the signal

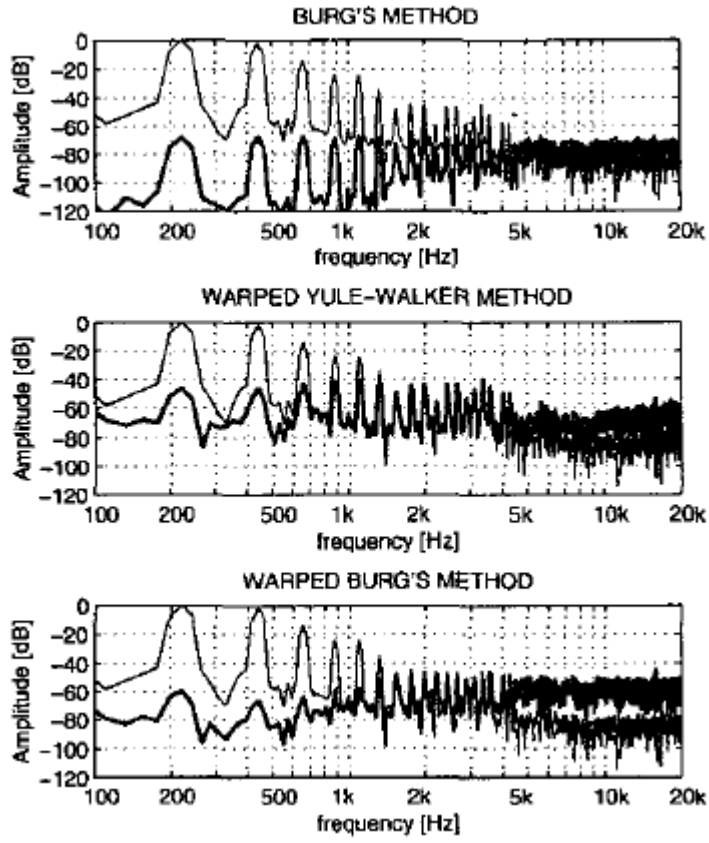


Figure 2.8: Comparison among the magnitude spectra of the prediction residuals obtained using Burg's method, the warped Yule-Walker method, and the warped Burg's method.

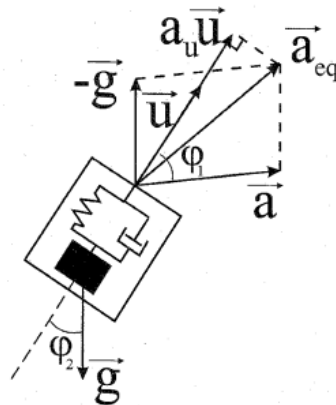


Figure 2.9: Uniaxial accelerometers measure the component a_u of an equivalent acceleration a_{eq} in the direction U' of the sensitive axis of the accelerometer [JKC07].

magnitude area (SMA), can verify the variations of the signal in the three axes [KNM⁺06]. To calculate this value you should use the following expressions:

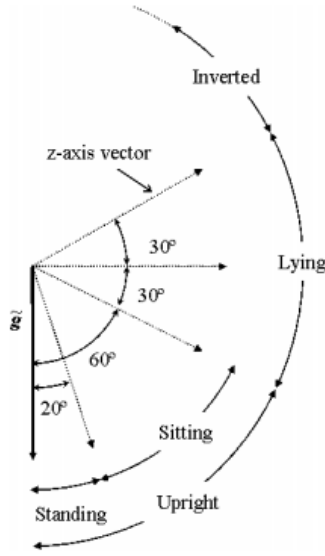


Figure 2.10: Example of the usage of the tilt angle to distinguish between some static postures, with the accelerometer position in the chest of the user [KNM⁺06].

$$SMA = \frac{1}{t} \left(\int_0^t |x(t)| dt + \int_0^t |y(t)| dt + \int_0^t |z(t)| dt \right) \quad (2.5)$$

The value $x(t)$, $y(t)$ and $z(t)$ are the acceleration in each axis, respectively. SMA is calculation using a time interval specified, for instance 1 second.

2.3.1.4 Linear Discriminant Analysis

The SMA, tilt angle and the autoregressive coefficients are useful features that can constitute a vector to be used as an input to a machine learning algorithm. However, in order to classify, and obtain the patterns in the features above, it is useful to apply a Linear discriminant analysis (LDA). This technique extracts linear combinations of variables that better represents the original data. Moreover, it aims to create a list of classes, based on the differences between the independent variables [KLLK10]. It is possible to compare the features before and after applying this technique in figure 2.11 and ???. To proceed to the calculation, one can use the following expression:

$$Disc = \operatorname{argmax} \frac{|D^T S_B D|}{|D^T S_W D|} = [d1, d2, \dots, dt]^T \quad (2.6)$$

The values SW and SB are the matrices obtained, in respect to the different classes.

After extracting the features, it can be used all of them or some depending on the complexity, number of activities that one pretends to identify, an algorithm of machine learning must be applied.

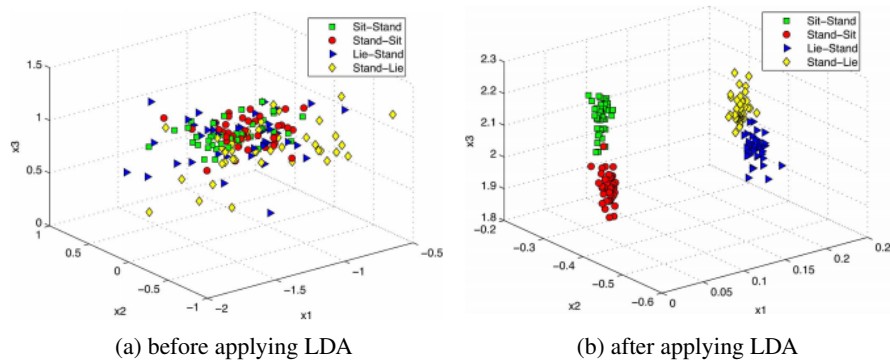


Figure 2.11: 3-D feature plot for the four transitions after the LDA [KLLK10].

2.4 Classification

Classification consists in extracting information from sensed data and group that information in different classes. The grouping is based on a previous training, where a dataset is provided, as well as the relation between the input and the classes desired. In machine learning there are two great paradigms: supervised and unsupervised learning. Classification problems use a supervised learning algorithm. In the field of classification exist a wide range of algorithms. This chapter will cover support vector machines and artificial neural networks since they are the most used kn:BIS06.

The focus will be artificial neural networks, since there are evidence that is possible to achieve great accuracy using it in the field of automatic activity recognition citepkn:KYT10 [KLK08]. Since a neural network is a learning system that is adaptive and robust, even when fuzzy data are presented the classifier will understand the features, even if they are different from the ones used in training .

2.4.1 Artificial Neural Networks

The study of Artificial Neural Networks (ANN) began in the 40's, and continues to be one crucial area of scientific interest, being an useful tool for automatic computational learning. By definition, an ANN is a computational network that aims to simulate the process of a real network of the human brain, and is used to solve complex or stochastic problems that traditional rule-based algorithms cannot solve. For better understanding the concepts behind ANN's, the neurophysiological and biological issues of the human brain will be briefly explained first.

2.4.1.1 Biological neural network

The biological neural system consists of a set of neurons that are connected through an axon. The neuron contains three different components: dendrites, soma and axon. The information received through electric impulses from other neurons are processed by the dendrites. Under certain situations, for instance in case of sufficient amount of signals, the signal is sent through

the axon, in order to reach other cells. Between the dendrites and the nucleus of the nerve cells, exist a terminal that is called synapse. A synapse is the barrier between two neurons, and it is where chemical reactions happen in order to transmit the signal. Neurotransmitters are used to enhance or inhibit the receptor of the neuron to transmit the impulses. Furthermore, depending on the results obtained, this can be adapted, existing a dependency on past events (memory), similar to what happens in an Artificial Neural Network [JMM96].

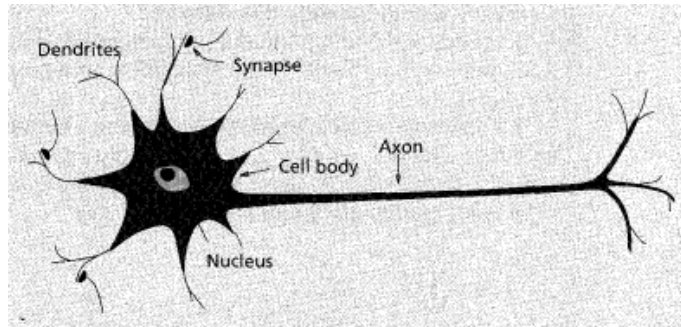


Figure 2.12: Biological neuron [JMM96]

A neuron can have many synapses and can transmit the signal to many neurons. Moreover, a neuron can receive impulses from many neurons, as figure 2.12 demonstrates. These characteristics inspired the scientific community to create an artificial analogy, to be used in computation. In a similar way, it is possible to assign a weight to each dendrite, giving more importance to information coming from a specific neuron. In figure 2.13 is a schema of a neural network that can be implemented digitally using a programming language. [Gra07].

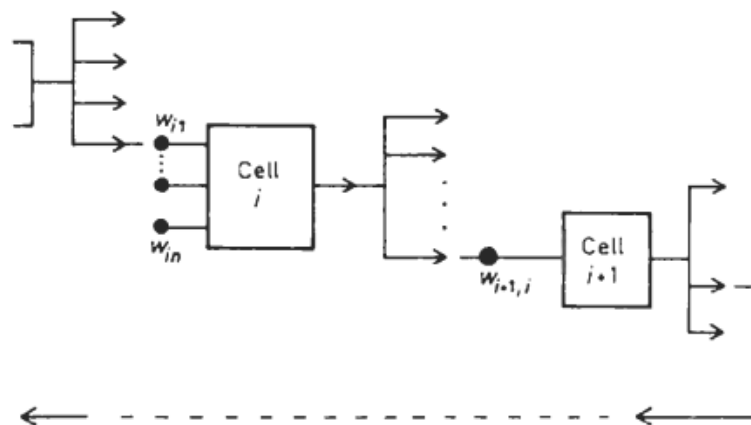


Figure 2.13: Analog Schema of a Biological neuronal network [Gra07]

2.4.1.2 History of Artificial Neural Networks

The study of the combination of neurons in a simple system, to enhance the computational capacity, started by Warren McCulloch in the 40's. Any McCulloch-Pitts neuron have a specific function, assigning a weight to each one. With the combination of the neurons in a net, the information can pass through. Moreover, the output of the net can be defined with a logic function, being an important discover in the area of ANN's . Donald Hebb create a general law in 1949, that says that if two neurons contribute at the same time to the net, they should be strongly interconnected. This premise was enhanced and generalized, and was used in the correlation matrix learning of Kohonen and Anderson [Gra07].

In the 50's and 60's, there was an increasing interest in the field of ANN's, and a new concept born: the perceptron. A perceptron is similar to a neuron, receive input by other perceptrons with a weight associated to each input, and transmits the information to more perceptrons. The weights are adjusted in order to minimize the error of the output. However, there were not proofs about the convergence of this learning algorithm to a good weight distribution in all cases. The delta rule, proposed by Bernard Widrow and Marcian Hoff, adjusts the weights in order to minimize the difference between the input and the output of the net, also called the mean squared error. This discover lead to the development of ADALINE (Adaptive Linear Neuron) and was an inspiration to the development of the backpropagation rule [JMM96].

In the 80's John Hopfield proposed a neural network with fixed weights and adaptive activation function. This kind of network is useful to solve problems that needs an associative memory like constraint satisfaction problems. Another huge discover was the backpropagation algorithm, that transmits the information about the errors in each unit, and changes the weights based on that values [Fau93]. This algorithm will be explained in detail in following section.

Artificial Neural Networks have a wide range of applications, from pattern classification, to clustering or even optimization issues. Note that in this thesis the focus of the problem of pattern classification, where the aim is to identify the type of activity based on the input from the accelerometer signal. Considering this, this chapter will emphasize this topic, exploring the algorithms that can be used to achieve this recognition.

2.4.1.3 Overview of ANN's algorithms

In the area of machine learning, a classification problem can be defined as the field of assigning a predefined class to a given input. The area of pattern classification includes these problems, and has the generic purpose of providing an output, based on all the inputs that have some kind of connection. An example of pattern recognition is to recognize an handwritten signatures of a set of persons. A multilayer neural network, using the backpropagation algorithm can solve with an high accuracy the problems of pattern recognition.

In order to calculate the output of a giver neuron, a weighted sum of its input signals is done. McCulloch and Pitts proposed a threshold unit, where the output is 1 if the sum reached a given value. In the following formula you can see how the output is calculated:

$$y = \sum_{j=1}^n w_j x_j - u \quad (2.7)$$

Where y is the output, w_j is the weight of the neuron j , x_j is the signal and u is the given threshold. This model can be used in multiple ways, after choosing the correct weights.

The way that the net is organized, how the neurons are connected is called the network architecture. If the network is a graph which the nodes are connected with no loops, it is a feed-forward network. On the other hand, if there are loops the architecture is called recurrent. Depending on the network architecture, different algorithms can be used. The most used are the feed-forward, where a set of output values are created and they do not have memory, recurrent functions are based in the reverse concept.

Usually, the action of selecting the correct weights is a process called learning algorithms. Note that there are two major different paradigms in the learning process:

- supervised: the output for a set of inputs is given;
- unsupervised: no output is given, just data as input and the network must find the patterns.

In the field of pattern classification, the most used learning algorithms are the supervised. In this case, the desired output for a given input is defined for a dataset aiming to train the network. Moreover, exist a lot of different approaches, or learning rules, in the supervised algorithms: error-correction; Boltzmann; Hebbian; competitive.

Error-correction focus on the fact that the output perceived in the training process do not always correspond to the desired output. So, the method calculates the error, modifying the weights to progressively reduce the error. Note that initially the weights are set to random small numbers. Based on this concept is the backpropagation algorithm, that was proposed in 1986 and aims to assign weights when training the network, in an architecture of the type multi-layer perceptrons. Note that there is an input layer, one or more intermediate or hidden layers, followed by the output layer as exemplified in figure 2.14 . Moreover, there are no connections between elements in the same layer, but the nodes in adjacent layers are all interconnected. The output layer in the stage of training is set to be 1 or 0, related to one specific input [PM92] .

In order to calculate the output of a neuron, based on the input, one may use the following expression:

$$y_j^h = \frac{1}{1 + e^{-x_j^h}} \quad (2.8)$$

Where x is the input, h is the layer and j is the index of the input vector. However, this is just an example of activation function, and the most widely used sigmoid function, illustrated in figure . This function has the characteristic of easily identify both possibilities without ambiguity [Hop88]. This fact is a very important issue in neural computation, because of the simple

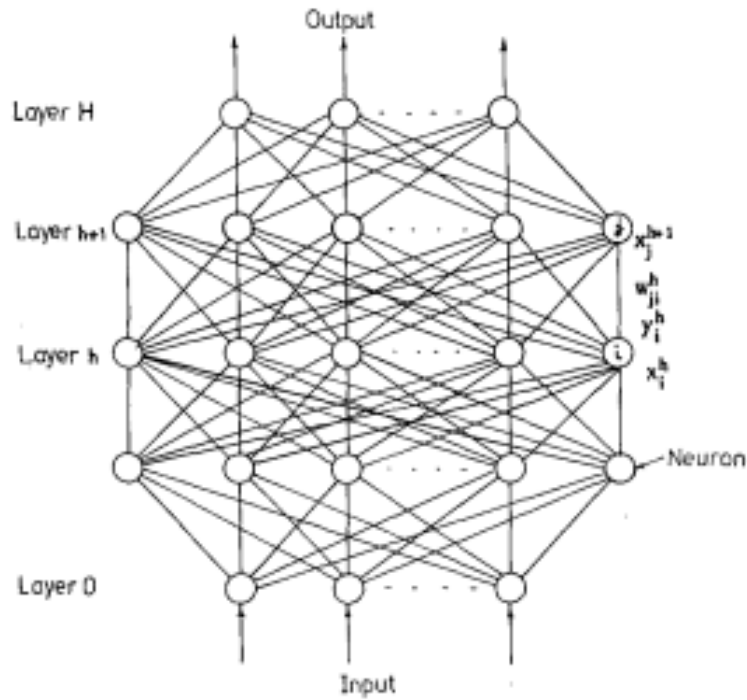


Figure 2.14: An example of neural network with three intermediate layers. [PM92]

relationship between the point and the derivative of that value. The parameters of the function can be defined to scale the function to other range, rather than from 0 to 1 [Fau93].

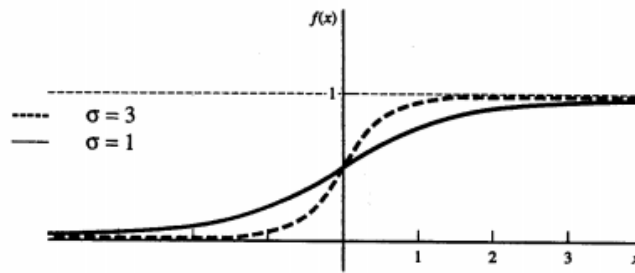


Figure 2.15: Sigmoid function [Fau93].

After one iteration in the training process, the error in the output must be calculated. One widely used method is the least mean square (LMS), one can use the following expression to calculate this value:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^N - d_{j,p})^2 \quad (2.9)$$

Where y is the output in the neuron j of the layer N , based on the input p ; d is the desired

output for this input. The purpose is to minimize the error $E(w)$, updating the weights iteratively. To proceed to this approach, the literature suggests that the method of gradient descent can be used [PM92] [Ser09]. In order to update the weights one can use the expression:

$$\Delta w_{ji}^h(t) = -\varepsilon \frac{\partial E}{\partial w_{ji}} + \alpha \Delta w_{ji}^h(t-1) - hdec \cdot w_{ji}^h(t-1) \quad (2.10)$$

Where E controls the rate of the descent, α is the coefficient of damping, $hdec$ is the percentage of decay and t is the number of iterations. The main idea is to initially forward to compute the error, and then backward the error updating the weights from the output layer to the input. Moreover, the error is reduced by a factor α . When a new input is present a new cycle starts, considering that α controls the stability and convergence of the model. This control value should be between 0 and 2, independently of the magnitude of the input [WL90].

2.4.2 Support Vector Machines

Support Vector Machines is a learning algorithm widely known in the field of pattern recognition, regression and novelty detection. It consists in mapping the input vectors in a feature space, where a linear decision is developed. This linear decision has special properties guaranteeing high generalization capacities. In the training phase, for each given input a point is assigned to an area of the space, existing two possible classes. When new datasets are given, the algorithm assign one of this category depending on the proximity of each area [Bis06].

Figure 2.16 shows an example of a problem in a two dimensional space. The hyperplane that maximizes the distance between it and all the data points, is known as the optimal hyperplane. To find an hyperplane that has this characteristic is not required to provide many training data. It is an efficient algorithm that have a huge power of generalization [CV95]. The probability of obtaining an error in classification is related to the number of support vectors and the number of training vectors:

$$E \leq \frac{E[\text{number of support vectors}]}{\text{number of training vectors}} \quad (2.11)$$

2.5 Geographic Fencing

GPS (Global Positioning System) is a technology that is widely used to retrieve the geographical location of a device. Nowadays, it is a well known technology, that is used in the daily routine of many people in their professional or personal life. The GPS measures the distance between the receiver and other satellites, the position is forecasted and broadcasted to the personal device of the user. Based on the position of well known satellites, it is possible to estimate the position of the user [Xu07].

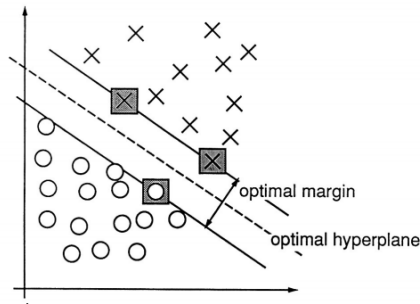


Figure 2.16: Example of a separable problem in a 2 dimensional space [CV95].

Geographic Fencing consists of defining a geographical barrier where the user can navigate, whenever the user enters or exits the fence an alarm notification is generated. Usually the user has a GPS device, and the geographical data are continuously being sent to the network, in order to be monitored by another device in a distinct place [BA11].

In order to retrieve the position of the user relative to a certain frontier, one can use an algorithm to solve a generic problem of point in polygon (PIP). This problem is common in applications in the field of computer graphics, computer vision or geographical information systems. A possible approach to the problem is to use the ray casting algorithm, widely used in computer graphics to detect spatial selections of objects [JZ11].

Consider a simple polygon, like in figure 2.17, that is not self-interconnected, meaning that the edges of the polygon don't intersect. Assuming that the polygon is defined counter-clockwise, the algorithm consists in tracing a ray horizontally, from right to left, to the polygon, crossing the point that needs to be detected as an insider or outsider [JZ11]. Whenever that ray intersects one edge of the polygon a value is added to a list, according to the following constraints:

- -1: if the edge crossed is defined bottom-up;
- 0: if the edge is not crossed;
- 1: if the edge crossed is defined up-bottom.

When the ray reaches the point of interest, all the values are summed. Finally, based on this result one can conclude the following:

- 0: the polygon is external to the polygon;
- 1: the point is internal to the polygon.

Figure 2.17 illustrates a polygon, where the point Q is inside the polygon and the point P is outside. Using the algorithm, considering point Q, the ray intersects P3P4, P4P5 and P5P6. Summing all the values obtained in each intersection the value 1 is obtained, confirming that the point is inside the polygon.

A geo fencing application example can be analyzed in figure 2.18. It can be useful to track the position of the elderly [WWL⁺09] or people with dementia or chronic disorientation [ANMF10].

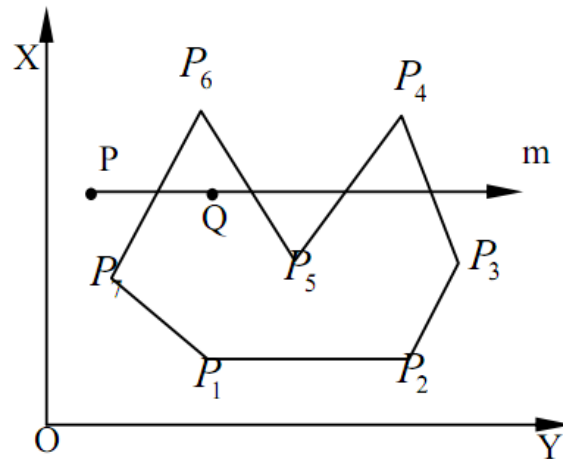


Figure 2.17: Simple polygon with a point inside and another outside of its area [JZ11].

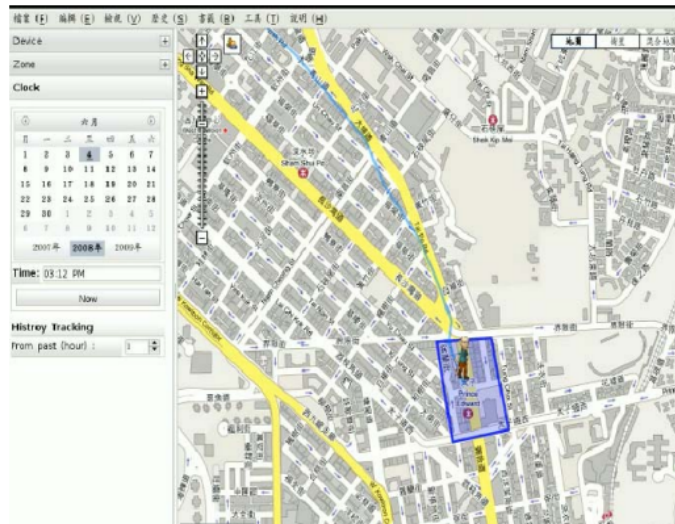


Figure 2.18: Example of geo fencing application [WWL+09].

2.6 Summary

This chapter presents the state of the art in distinct areas. Firstly, the concept of mobile healthcare is presented, including the challenges and motivations of developing this systems. Machine-to-machine communications is a paradigm that can be used in this scenario. So, the general architecture and the standards already existent are explained. Respecting automatic activity recognition is possible to conclude that existent investigations demonstrated the high accuracy in detecting several activities only using accelerometers. Classification enables to identify these activities after using a supervised learning algorithm. Artificial neural networks is widely used in this field because of its capacity of generalization and adaptive characteristics. The concept of geographic fencing was also present. Ray casting is a great solution to solve general point-in-polygon problems and can

State of the Art

also be applied in this area.

State of the Art

Chapter 3

Problem Formulation

3.1 How to use an M2M Platform in an healthcare scenario

One of the main purposes of this thesis is to develop a system of remote healthcare monitoring that demonstrates the potential of M2M communications. Although systems to monitor the health of a patient already exist, there is a lack of open systems that can easily be extended with other services or devices. Moreover, most of the solutions are proprietary and closed solutions and can not be adapted to specific scenarios without acquiring the whole system.

Figure 3.1 illustrates the problems of the current monitoring systems. A mobile phone is publishing data to a service that will then be forwarded to another device. However, if the user wants a new sensor, since the existing solution is closed and proprietary, an additional service must be used, as well as a different stream of data. Another problem of the current architecture is that the data existent in one service can not be reused for other purposes and there are unnecessary streams of data.

On the other hand, Figure 3.2 shows the scenario using the M2M paradigm, where the data can be available to multiple services and only one stream of data from the user is sufficient, adopting the standards of M2M. This enables interoperability and eases reusing existent services.

The aim is to develop an application of remote monitoring in a scenario of healthcare, using the paradigm of M2M. PT Inovação is working on a platform for Context Aware applications called MyContext, that will be used to develop the application.

The application must enable monitoring the heart rate, position and physical activity of a user. The user that is using the smartphone must visualize the information, but other users must also subscribe and monitor the data at real time. Moreover, it must be possible for a user to follow multiple users and receive alarms when a user has an heart rate above the normal values, or have left a geographic barrier that was defined.

Problem Formulation

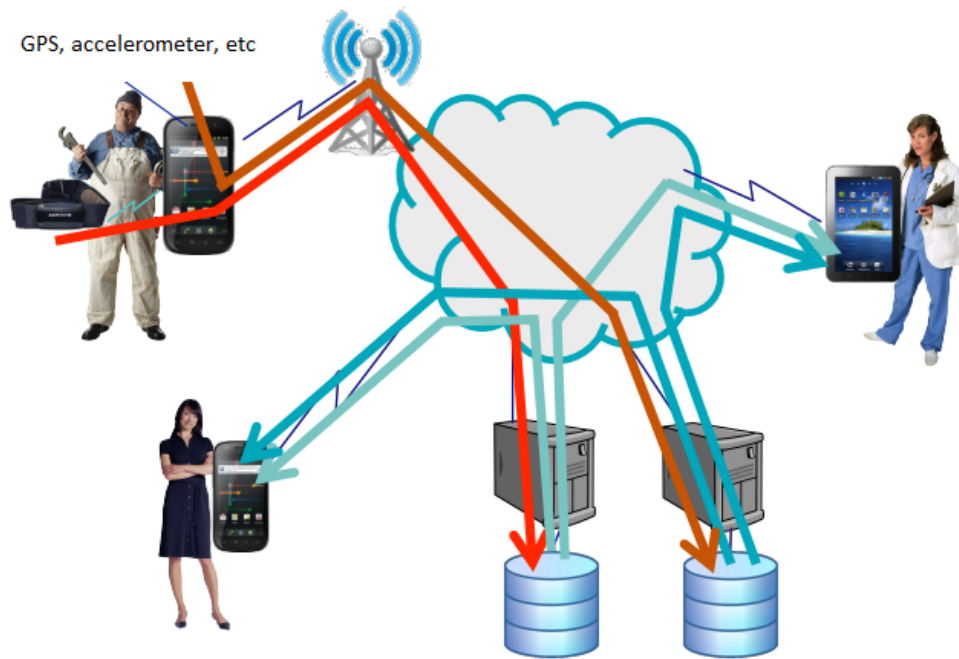


Figure 3.1: Remote monitoring systems of today.

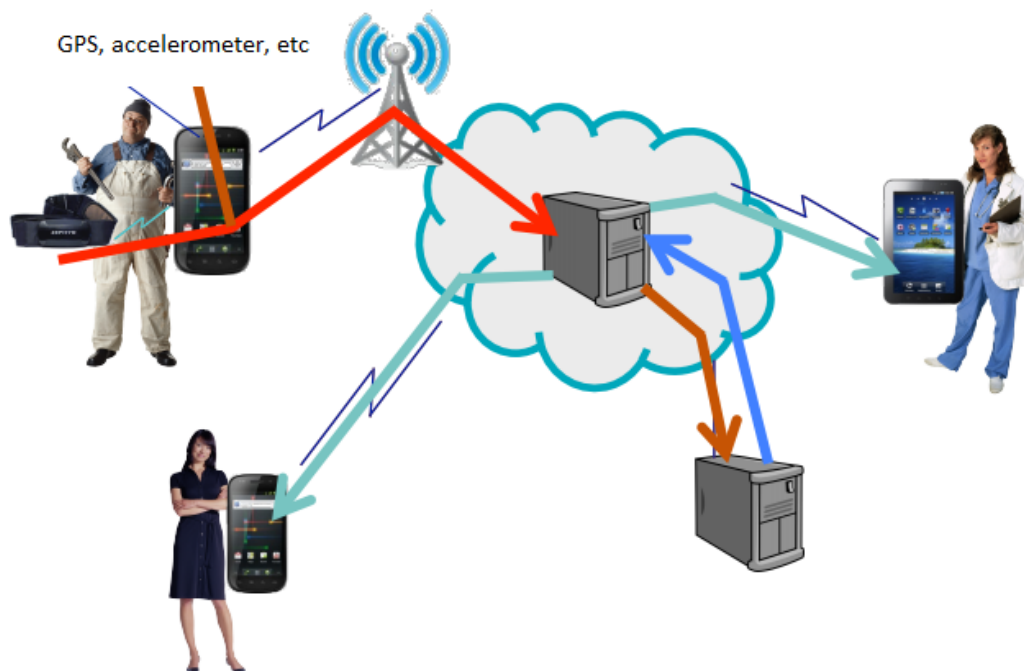


Figure 3.2: Remote monitoring systems using M2M.

In order to enable the monitoring of the physical activity of a user using the application, it is necessary to use techniques of Machine Learning like the following section describes. An Artificial

Neural Network will be trained using acceleration data, and then the network will be exported to the service of physical recognition, that will use this network to recognize the activities in real time.

3.2 Detection of Physical Activity using a smartphone

The detection of the physical activity of an individual is not a novelty. As demonstrated in Section 2.3 several efforts on this area have been done previously, mainly using dedicated accelerometers which enable the recognition of the physical activity with high accuracy. The studies that achieve better results use one or more accelerometers in specific and fixed positions of the body of the individual. Moreover, the hardware used on most studies are expensive dedicated accelerometers that return the acceleration on each axis with reduced noise. Although these systems of automatic detection of physical activity are useful as proof of concept, they are not affordable, comfortable neither easily used on a daily basis by the general population.

A system to recognize physical activities can be useful in the area of healthcare. A caretaker can be monitoring the activity of a user, and beware of dangerous situations for a particular individual. For instance, walking upstairs or downstairs can be dangerous for the elderly, since the person has an high probability of falling.

One of the main purpose of this thesis is to develop and adapt algorithms to automatically recognize the physical activity using only a standard smartphone and its embedded sensors. The user should be able to put the smartphone on his pocket or in any other part of the clothes, and the recognition of the physical activity is performed in a transparent and unobtrusive way. Smartphones with an embedded accelerometer are a recent trend and algorithms to do an accurate classification of the physical activity are not yet ready to achieve good results on these devices. There are a lot of difficulties and challenges to achieve a good accuracy when using a smartphone with the purpose described above:

1. Noisy data: The data from the accelerometer of a state of the art smartphone has a lot of noise. Even when the device is in a static position, constant changes on the acceleration data occur, that may affect the activity recognition;
2. Sampling rate: The rate at which the data is received from the accelerometer is not constant, and is highly dependent on the version of operative system and the type of accelerometer embedded on the smartphone.
3. Orientation: Since the hardware used is installed in the smartphone, there is no pre-defined knowledge about the position of the device. When changing the orientation of the mobile phone, the values of the acceleration along each axis will change, since the axis will suffer a rotation. For instance, the values of the acceleration when walking with the mobile phone in the hand will be different than walking with the mobile phone on the pocket;

Problem Formulation

4. Device dependent: Each device may have a different accelerometer, with a distinct sampling rate and precision;

The physical activities that are going to be recognized are: standing, walking, running, walking upstairs and downstairs. Standing and walking were chosen because they are the activities more common on the daily life of a person. Moreover, knowing if a person is standing is also useful in a scenario of Healthcare, where the caretaker can analyze if the user is being sedentary. Running is an activity that involves a great physical effort so the heart rate of the person performing this activity reaches high levels. So, in a scenario of Healthcare it is also important to beware of a person that is running, because the user may have heart problems. At last, walking downstairs and upstairs are also dangerous activities because the person performing the activity, specially the elderly, have an high risk of falling when practicing these activities.

Chapter 4

An M2M Healthcare application

One of the main purposes of this thesis is to develop a system for continuous healthcare monitoring as a proof of concept for an existing M2M framework. Machine-to-Machine communications is the paradigm that was adopted, where multiple devices that are connected to sensors are publishing data and communicating with each other.

The application was designed in order to demonstrate how M2M paradigm can enable that different sensor streams can be efficiently shared by multiple services. The sensors used were an heart rate sensor, accelerometer, rotation and GPS. Services that forwards these information to devices that had previously subscribed to it, and other services that will do calculates with the data and detect an abnormal situation were developed. A framework developed by PT Inovação called MyContext, based on the XMPP protocol, was used in order to establish the communication between the devices. The framework will be explained and also how this framework was used to solve this specific problem.

4.1 Existing Platform

PT Inovação has been working on a framework that provides communications facilities and libraries for the development of context-aware applications called MyContext. It is based on the XMPP protocol and re-use transport and security approaches of the protocol [JG11]. The paradigm used to send and receive information between the application components that use this framework is the publish-subscribe pattern [Mil10]. This thesis aims to develop a distributed system that is constituted of multiple applications, running on different devices, each one with a different purpose, exploiting the full functionalities of the framework. The system developed is the first deployed applications that used this framework, which is still on its infancy. This was an important work to PT Inovação, since the iterative development process followed provided feedback of the bugs and drawback that were encountered, and enabled the evolution of the existing platform.

An M2M Healthcare application

The advantages of using this framework instead of another open source XMPP library, like Smack, are the following:

- Extensibility: it is easy to create new types of context and new applications that use those contexts. A context can be raw data returned by a sensor or information extracted from the raw data.
- Local communication: it is possible to consume data from sensors in the same device.

Since the framework enables the agile development of context awareness applications, one of the main concepts of this framework is the context. Context information is the data that is exchanged between the multiple parties involved, and is defined using XML defined by a schema. The following XML code represents a schema of the GPS context, where the values that will be exchanged are listed, latitude, longitude and altitude, as well as the minimum and maximum occurrences of each field:

```
1
2 <?xml version="1.0" encoding="UTF-8"?>
3 <schema targetNamespace="http://iex.ptin.pt/ctx/gps"
4 xmlns="http://www.w3.org/2001/XMLSchema"
5 xmlns:tns="http://iex.ptin.pt/ctx/gps">
6   <complexType name="GpsType">
7     <sequence>
8       <element ref="tns:altitude" maxOccurs="1" minOccurs="0">
9     </element>
10      <element ref="tns:latitude" maxOccurs="1" minOccurs="1">
11    </element>
12      <element ref="tns:longitude" maxOccurs="1" minOccurs="1">
13    </element>
14    </sequence>
15  </complexType>
16  <element name="gps" type="tns:GpsType"></element>
17  <element name="latitude" type="double"></element>
18  <element name="longitude" type="double"></element>
19  <element name="altitude" type="double"></element>
20 </schema>
```

The architecture of this framework is illustrated in Figure 4.1 where different devices can consume and be the source of context. It is available an Android API to develop applications that are a source of context, which means the information is sent following a pre-defined context schema, to the context broker that will then forward this information to the interested parties, services and applications, that had previously subscribed to this type of context.

Context consumers are application, M2M nodes, that subscribe to certain type of context and receive the data that is sent by sources. A consumer can be a mobile application, website or desktop application.

An M2M Healthcare application

The framework uses the concept of provider, which can be seen as a service in the M2M paradigm, being a server-side component that provides context information. There are three different types of providers:

- Source provider: Responsible of consuming the context of a user device or application that is publishing information. In order for the data to reach other devices, this provider publishes the context to the context broker that will forward this information to the devices that subscribed to this context. For each context, a context provider must be running when the devices are publishing this type of context, to assure that the data reaches the consumers interested. If a provider like this does not exist, or is not running, the information will remain local on the device, and can only be consumed on the device.
- Composite provider: It is similar to a source provider, however it provides new context based on the context data sent by a source provider. It consumes one or more types of context and do specific calculations inferring another type of context. For instance, a composite provider consumes the acceleration and returns the physical activity being performed.
- External information provider: Forwards context data received from the source providers. This type of provider will not be used in the scope of this thesis.

Finally, the context broker is the responsible for the management of all requests and the distribution of the information provided through all the consumers that previously subscribed to it.

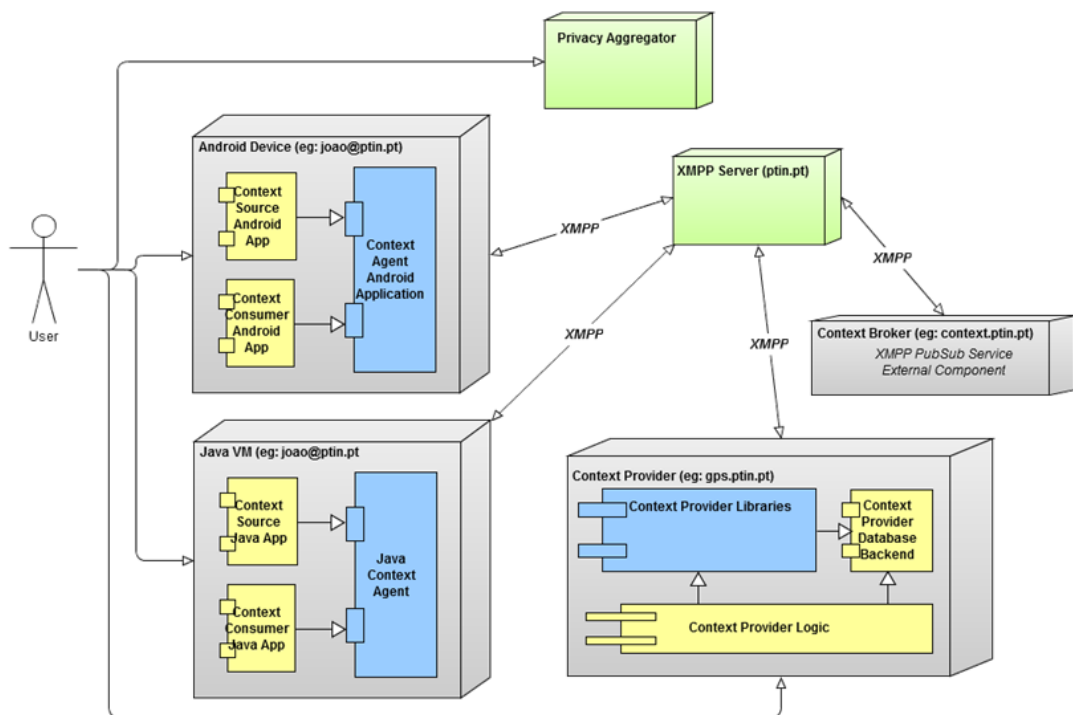


Figure 4.1: Architecture of the context framework [JG11]

An M2M Healthcare application

An application to be installed on the devices was also developed by PT Inovação and is called "Context Agent", since its purpose is to receive context being sent by one or more applications in the device, and forward that data to the context broker placed in a remote server. Moreover, it is responsible of receiving data from the context broker and send it to the applications that had subscribed to that type of context. PT inovação had supplied this application as a black box, providing only the API to use it.

The framework does not enable access control functionalities, it is not possible to retrieve a list of registered users for instance. Implementing these functionalities is out of scope of this thesis. There are two libraries available at the moment that enable the development of context awareness applications. An Android API for consumer and source applications and a context provider API for the Java programming language.

4.2 Application Design

4.2.1 General Architecture

The framework MyContext will be used to support all communications between the components of the distributed monitoring application. This framework enables the use of the extension Pub-Sub, being possible to publish information (by the android agents) and this information can be subscribed or consumed by other applications.

A smartphone application will be developed for the Android Platform. Using M2M terminology, this agent will be an M2M node responsible for collecting the data from multiple sensors: GPS, accelerometer, orientation. Moreover, the mobile device will establish a Bluetooth communication with an external sensor for heart rate monitoring (Figure 4.4). An application containing this agent will be available for both the users and the relatives of the patient, allowing the remote monitoring of a relative, but the user can also monitor his own vital signs (Figure 4.2). In terms of MyContext framework, this application will be both a context consumer and a context source.

Besides the mobile agent for the relatives, will be developed an Android application for a tablet, where the caretakers can monitor multiple patients. This application will enable that the caretakers subscribe more than one user and monitor them at the same time. Besides that, it will be possible to receive alarms when an anomaly situation happens. This application is a context consumer.

To enable the generation of alarms three services will be developed: geo fencing; activity recognition and heart abnormally detector. These applications are composite providers since they will consume the data from the Android applications and process it with algorithms to detect specific events. When an alarm situation is detected an alarm will be published to the context broker and all the context consumers (smartphone or tablet application) that had previously subscribed to these situations for a certain user will receive that information.

An M2M Healthcare application

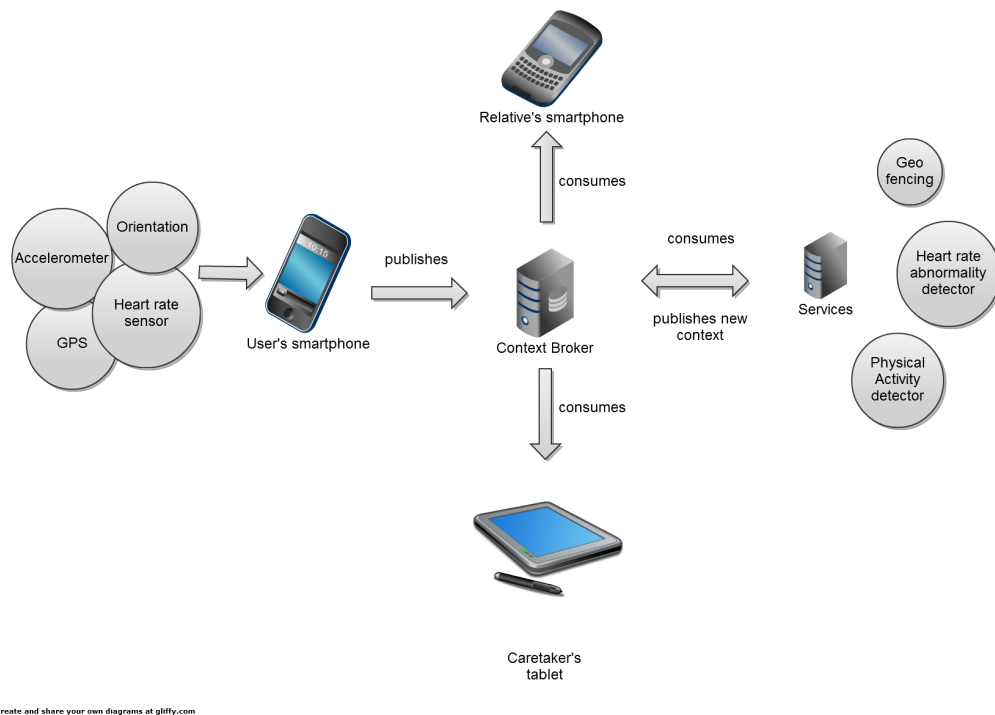


Figure 4.2: General architecture proposed for the monitoring system.

4.3 Implementation

In this section it will be explained how the MyContext framework was used in this scenario. An important step was to create the type of contexts that are going to be used, like the heart rate, physical activity and geographic position. Moreover, the providers that were created, source or composite providers, will also be detailed since they are crucial to forward the information data that is published and to create new type of contexts.

In order to publish and consume the context two applications were developed: a tablet application for the caretakers to monitor multiple users and a smartphone application for the users to monitor themselves or another person.

4.3.1 Context information

As explained before, for each type of information that is exchanged a context must be defined, creating a XML schema. In this section the multiple context that are going to be used in the scope of the application developed will be explained. Since the general format of the context was already described in the previous section, the multiple contexts that are going to be presented will be shown in a compact format, illustrating the variables that are going to be sent. Moreover, one context exists for each type of information, one for the raw sensor data and another for the information obtained in the composite providers using the raw data.

4.3.1.1 Location

One of the functionalities is sharing the geographic position of the user. So, the data that is going to be retrieved from the GPS sensor and consequently is going to be published is the latitude and longitude.

```
1 <element name="gps" type="tns:GpsType"></element>
2 <element name="latitude" type="double"></element>
3 <element name="longitude" type="double"></element>
```

A service, or composite provider, of geo fencing will consume the GPS data, and detect if the user is outside a specific boundary. So, the composite provider will publish to the context broker, context containing 4 points representing the polygon of the geographic barrier and a field indicating if the user is outside or not of the frontier.

```
1 <element name="geofencing" type="tns:GeofencingType">
2 </element>
3 <element name="latitude" type="double"></element>
4 <element name="longitude" type="double"></element>
5 <element name="latitude1" type="double"></element>
6 <element name="longitude1" type="double"></element>
7 <element name="latitude2" type="double"></element>
8 <element name="longitude2" type="double"></element>
9 <element name="latitude3" type="double"></element>
10 <element name="longitude3" type="double"></element>
11 <element name="latitude4" type="double"></element>
12 <element name="longitude4" type="double"></element>
13 <element name="status" type="string"></element>
```

4.3.1.2 Physical Activity

Another service available is the one responsible for recognizing the physical activity of the user. To enable this, the mobile application will publish information representing the physical activity. Using the data from the accelerometer and rotation sensor, useful features are going to be extracted representing the physical activity that the user is performing as described in Section 5.4. The list of fields that represent the physical activity being performed, and that is going to be published from the mobile phone are the following:

```
1 <element name="featuresactivity"
2 type="tns:FeaturesActivityType"></element>
3 <element name="bmp" type="double"></element>
4 <element name="coef1" type="double"></element>
5 <element name="coef2" type="double"></element>
```

An M2M Healthcare application

```
6   <element name="coef3" type="double"></element>
7   <element name="coef4" type="double"></element>
8   <element name="coef5" type="double"></element>
9   <element name="coef6" type="double"></element>
10  <element name="coef7" type="double"></element>
11  <element name="coef8" type="double"></element>
12  <element name="coef9" type="double"></element>
13  <element name="f10" type="double"></element>
14  <element name="f11" type="double"></element>
15  <element name="f12" type="double"></element>
16  <element name="f13" type="double"></element>
17  <element name="f14" type="double"></element>
18  <element name="f15" type="double"></element>
```

There are 9 values representing the coefficients resulting from the Autoregressive model applied to the acceleration data. The other values contain the velocity and displacement in each axis (see Chapter 7).

A composite provider will consumer the data representing the physical activity and will use a classifier to recognize the activity being performed. After the recognition it will publish that information to the context broker.

```
1   <element name="physicalactivity" type="tns:PhysicalActivityType">
2   </element>
3   <element name="type" type="string"></element>
```

4.3.1.3 Heart rate

The information about the vital signals that is going to be published through the smartphones is the heart rate of the user. This data will be extracted from the heart rate sensor from Zephyr, and will continuously be sent at a rate of 1 second.

```
1   <element name="heartrate" type="tns:HeartrateType">
2   </element>
3   <element name="bmp" type="int"></element>
```

A composite provider will consume the context of the heart rate, and will detect if it is a normal value or not. In case of an abnormal value is detected an alarm will be published, referring if it is a bradycardia or a tachycardia.

```
1   <element name="heartabnormality"
2   type="tns:HeartabnormalityType"></element>
3   <element name="bmp" type="int"></element>
4   <element name="abnormality" type="string"></element>
```

4.3.2 Source Providers

The information being published by the smartphone can be subscribed and be consumed in the same device. If the purpose is to enable that external devices can consume that information, a source provider must be defined for each type of context.

So, a source provider that consume the context of GPS and publishes that context to the context broker will be developed. The context broker then forwards the information based on the subscription list of the users interested in that information. In a similar fashion, a source provider that consumes and publishes the context representing the physical activity and another that consumes and publishes the heart rate, was developed.

In order to ensure that these three different contexts are being forwarded between different devices, the respective source providers, which are Java applications must be running on a remote server.

4.3.3 Composite Providers

4.3.3.1 Heart Abnormality

The service of heart abnormality is responsible of detecting if the heart rate of the user is normal. To accomplish this, this service will consume the heart rate data that is being published by all the users. After that, a threshold is defined and a simple verification is done on whether the heart is within the interval [60,100], like the following code illustrates:

```
1 if(bmp != 0 && bmp < 60 )
2   heartabnormalityCtx.setAbnormality("bradycardia");
3 else if(bmp > 100 && speed == 0)
4   heartabnormalityCtx.setAbnormality("tachycardia");
5 else
6   heartabnormalityCtx.setAbnormality("normal");
```

If an abnormality is detected, a bradycardia or tachycardia, a new context is published, labeled heart abnormality. After creating this type of context it is published to the context broker. All the users that had previously subscribed to heart abnormalities of this user will receive this information.

4.3.3.2 Physical Activity

A service for detecting the physical activity being performed was also developed. This service consumes the context of the features representing the physical activity. These features are sent to an Artificial Neural Network (ANN) previously trained, further explained on Section 5.5. The ANN returns the type of physical activity that was performed, and creates a new context containing

this information, publishing it to the context broker.

The ANN trained is stored in a file that contains the structure of the network and the weights on each neuron. When the service starts to run it loads the network and is ready to recognize the physical activities. At the beginning it also loads the median and standard deviation of each input of the network. These values are used to proceed to a standardization of the context data received from a user relative to a physical activity being performed, before sending it to the network (see Chapter 7).

4.3.3.3 Geo Fencing

Another composite provider was developed that aims to detect if a user is outside a specific geographic boundary. It consumes the GPS data of each user.

There is a file containing the geographic boundaries for each user, as the following list illustrates:

```
1 jprud:41.18 -8.63,41.18 -8.65,...
2 aaguiar:40.6 -8.65,40.63 -8.65...
3 dlucani:41.18 -8.59,41.18 -8.59,..
```

Each line represents the boundary of a user, specifying four geographic points, latitude and longitude, constituting a polygon. When the service starts to run it loads the file and stores this information in memory.

At the reception of a new context representing a geographic position of a user, the service checks if a user has a barrier defined, and in case of the frontier being defined it checks if the user is inside or outside the boundaries.

In order to verify if a point is inside a specific boundary, an algorithm based on Ray Casting is used by W. Randolph Franklin in 1970, being implemented using the Fortran language at that time [Fra06]. The algorithm starts to trace a ray horizontally starting at the test point. Every time the ray intercepts an edge a variable is incremented. At the end, the point is said to be outside or inside the polygon depending on the number of interceptions as further explained on Section 2.5.

A new context is published after checking if the point is inside the boundary defined, and its published to the context broker, containing the four points of the polygon and the information about the position of the point in relation to the polygon. The only situation where no context about geo fencing is published is when no barrier is defined for the user sending the GPS data.

4.3.4 VitalTracker application for mobile phones

An application to be used by a person that wants to monitor his heart rate, geographic position and physical activity was developed. Besides being possible to monitor his own information, it is also possible to monitor the data of another person. This application was developed using the Android SDK and the main modules of the application are described in Figure 4.3.

Besides the application developed, another application developed by PT Inovação must be installed

on the smartphone and running at the same time of the VitalTracker application. As explained on Section 4.1, from the application Vitaltracker it is possible to send and receive data from the Context Agent using MyContext API.

The Vitaltracker application developed can be divided in three distinct modules: UI, context source services and context consumers services (see Figure 4.3). In Android SDK an activity is responsible for creating a window where the user can interact with the UI. So, the UI represented in a diagram in Figure 4.3 is an activity, where the user can select the context data that wants to send and receive. After the user selects the context data that he is interested in, a service for each kind of data is created and starts to run. The concept of service in Android SDK is a component that belongs to the main application but does not have a user interface, and it is usually used to perform long tasks. Moreover, if the user minimizes the applications, the services continue to run. So, it is useful to use services in this scenario since the purpose is to be continuously sending the context data. So, a service for each type of context was defined, responsible of sending the context data to the Context Agent, that will forward the information to the context broker.

Moreover, three other services were defined with the purpose of consuming context data. After these consumer services receive the data, they will communicate with the UI in order to update it. Note that in the case of the consumer services, they will not be started by the VitalTracker application. Instead, a reference of each service is passed to the Context Agent application. When the last receives information from the context broker with content was subscribed by the VitalTracker application, it will then start these services.

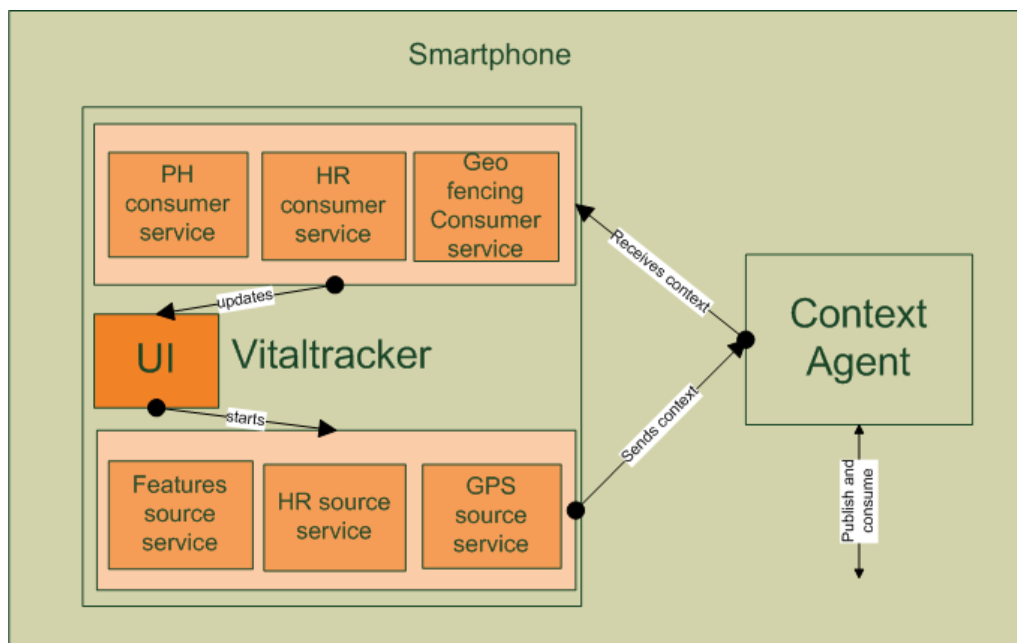


Figure 4.3: Block diagram of the vitaltracker application.

4.3.4.1 Source Services

There are three different Android services that publish context. Each Service runs on the background, continuously collecting data from the sensors and forwarding that data to the Context Agent.

A service for collecting the heart rate communicates via bluetooth to establish a connection and receive the data. After connecting with the device, it receives the heart rate of the individual, who is carrying the sensor at his chest, at a rate of 1 time per second. Figure 4.4 shows how the smart-phone communicates with the heart rate sensor.

Another service was developed to get the actual position of the user. Since the application needs to have Internet connection in order to use it, the position of the user is returned also using the network provider, if the GPS signal is not available. Every time the user change its location the service sends the longitude and latitude to the Context Agent application, which will then forward this information to the context broker.

Finally, a service will receive data from the accelerometer and orientation sensor. The first time, it receives an event from these sensors, it stores the current time. In the following events, it verifies if 5 seconds have passed since the first event, and when this happens, a process of features extraction is used in order to get useful information about the 5 seconds of physical activity, as explained in Section 5.4. After extracting these features, they are published to the context broker, and the counter starts again to get another 5 seconds of data from acceleration and orientation sensors.

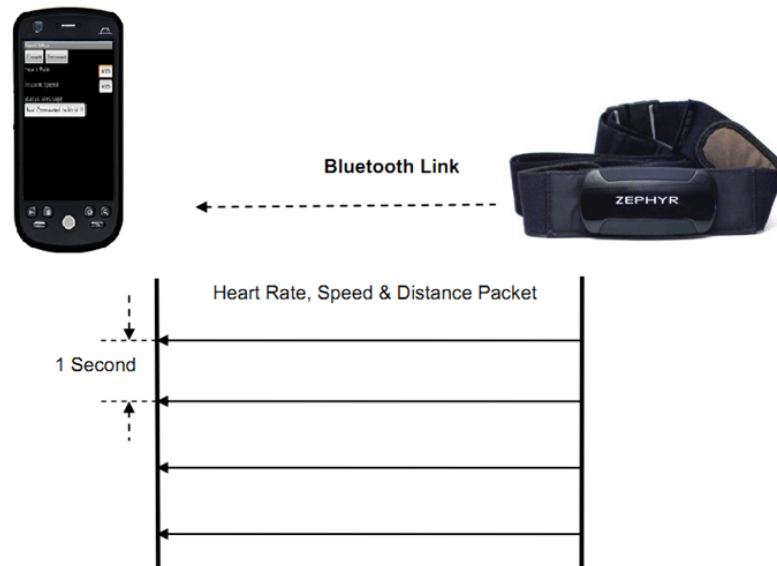


Figure 4.4: Zephyr heart rate sensor communicating with a mobile phone.

4.3.4.2 Consumer Services

Another type of Android services that will be running are Services that are waiting to receive context information. This context is published by the source service of a user, if the user is monitoring himself, or from the source service of another user. However, the source service of each context will not be started by the VitalTracker application. Instead, the reference of the service is passed to the Context Agent application at the moment of subscription of the context. The following code illustrates how to subscribe to a context using the MyContext API:

```
1 ContextSubscriberInterface aca = new AndroidContextAgent(this);
2 aca.subscribe("http://iex.ptin.pt/ctx/physicalactivity",
3     "user@mycontext.ptinovacao.pt",
4     PhysicalActivityConsumerService.getComponentName());
```

The username of the user to be subscribed is passed, as well as the name of the context, in this example the subscribed context is the physical activity. Moreover, the name of the service that is ready to receive this kind of information is given. When the Context Agent receives a context of this type it will start the consumer service.

After a consumer service receives the context information, it must send this data to the main activity in order to update the User Interface. The VitalTracker application is highly modular, built on all these independent services ready to send and receive different contexts. It is possible to put these services in different applications in order to achieve a higher level of decoupling. So, to communicate between these services and the user interface the concept of broadcast messages was used to send data. A broadcast message in Android enables the communication between different components in the same application, or between different applications on the same device. In order to send a message of this type, an Intent is created that provides the facility to call an operation in an Activity, encapsulating all the information needed, depending on the context information. The following code illustrates how this can be coded:

```
1 Intent intent = new Intent();
2 intent.putExtra("username", "joaoprudencio");
3 ...
4 intent.setAction("pt.it.porto.m2m.vitaltracker.geofencing");
5 this.sendBroadcast(intent);
```

Note that in this case an identifier is associated to the message: "pt.it.porto.m2m.vitaltracker.geofencing". So, in order to an application receive geo fencing context, it must create a broadcast receiver to handle the message and update the user interface. The following code shows how to create a broadcast receiver that is ready to receive the messages sent by the consumer service:

```
1 IntentFilter filterGeofencing = new IntentFilter(
```

An M2M Healthcare application

```
2 "pt.it.porto.m2m.vitaltracker.geofencing");
3 this.registerReceiver(geofencingReceiver, filterGeofencing);
4 private final BroadcastReceiver geofencingReceiver
5     = new BroadcastReceiver() {
6     public void onReceive(Context context, Intent intent) {...}
7 }
```

All the consumer services follows the same logic, the only thing that changes is the handle of the messages and the way the user interface is updated.

4.3.4.3 User Interface

In this section the user interface of the VitalTracker application will be presented. There are two main use cases for this application: the user can monitor his own context information or he can monitor information of other users. So, the first menu of the application enables the user to choose between these two options as Figure 4.5b shows.

If the user selects the option for monitoring another user another menu appears for the selection of that user. Figure 4.5a shows that menu where the user can type the username of the person he wants to monitor.

If the user selects the option to monitor himself he will be forwarded directly to the menu present in Figure 4.5c. Moreover, the user that pretends to monitor another person will also reach this menu after selecting the user that he pretends to follow. This menu enables the user to select the type of data that he pretends to visualize from another user, or from himself.

Finally, after selecting the context data of interest, the user can pull the bar on the bottom and the screen shown in Figure 4.5d will appear. This menu will enable the user to monitor new data received, like the geographic position, heart rate and physical activity.

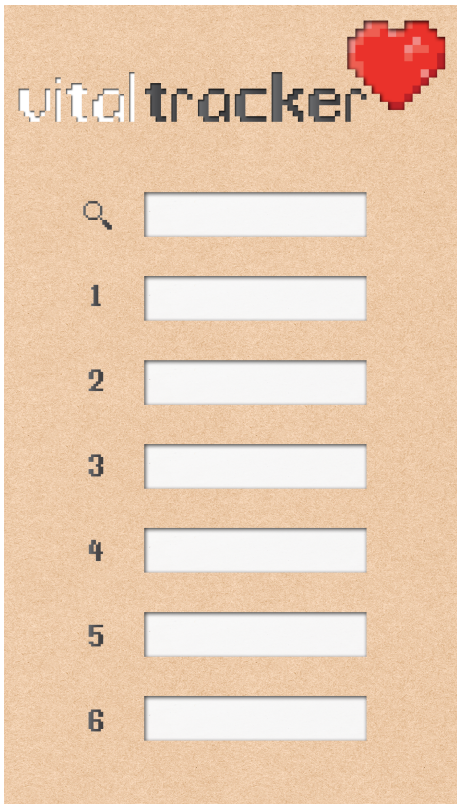
4.3.5 Vitaltracker application for tablets

Besides the VitalTracker application developed to be used on a smartphone, another one was created targeting a tablet, also with the operative system Android installed. This application is similar to the VitalTracker but it only enables the monitoring of other users. Moreover, it is possible to monitor more than one user at the same time.

Since the functionalities are similar, the same consumer services were used, and the source services will not be used because no context is created with this application, only consumed.

Figure 4.6a shows the menu that enable the subscription of multiple users. After the subscription of the users it is possible to select the context information that the user pretends to receive (Figure 4.6b). Finally, it is possible to visualize the context data being received from multiple users.

An M2M Healthcare application



(a) Subscription of user.



(b) Option between monitor himself or other person.



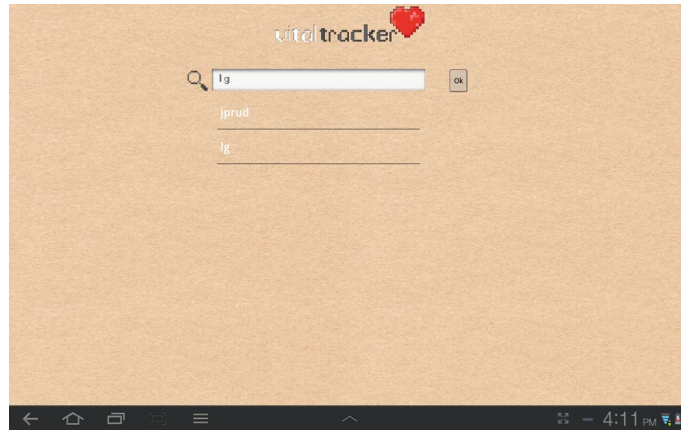
(c) Subscription of context.



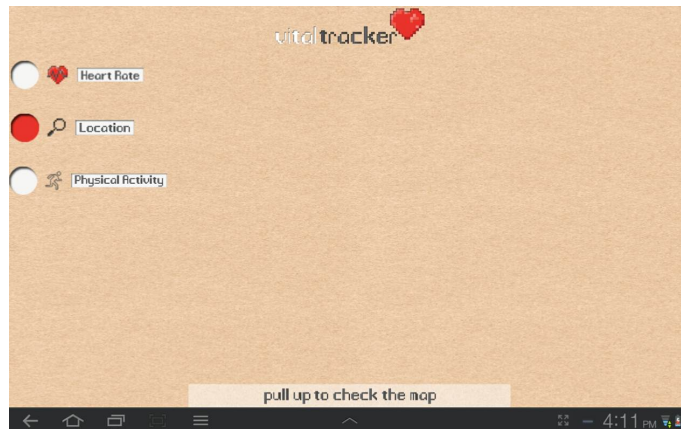
(d) Visualization of incoming information.

Figure 4.5: Screenshots of the VitalTracker application.

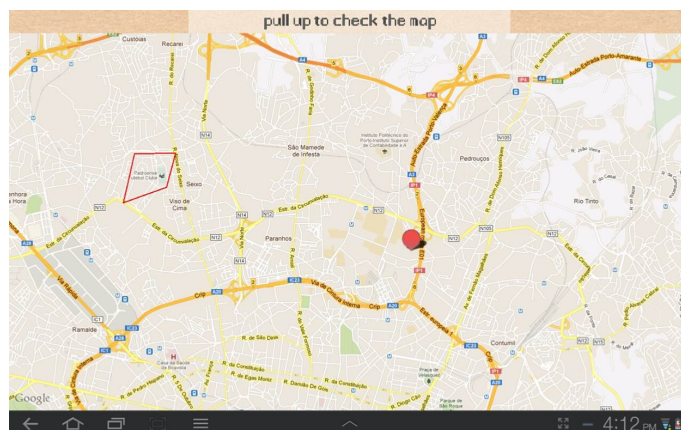
An M2M Healthcare application



(a) Subscription of users



(b) Subscription of context.



(c) Visualization of incoming information.

Figure 4.6: Screenshots of the VitalTracker application for tablets.

An M2M Healthcare application

Chapter 5

Physical Activity Detection using Smartphones

This Chapter describes the design of the physical activity detection service, one of the components of the M2M application described in the previous Chapter.

5.1 Methodology

Since the purpose is to develop a system affordable to the general public the target of the software developed is a smartphone using the Android operative system. It is one of the most widely used operative systems, and a smartphone with this software installed can be bought at a very low price. However, the algorithms that will be presented are general to all kind of devices.

In order to classify the activity being performed by an individual a set of steps will be taken. The system developed have two different stages: the training phase and the test phase. Firstly, the system must learn the pattern of the acceleration correspondent to each activity. After the training, the system is ready to recognize the physical activities performed, but only the ones trained in the previous phase. In both phases the process is similar:

- Data collection: Retrieving the raw data from the accelerometer;
- Processing the data, extracting useful features;
- Send the data to a classifier.

Each phase will be explained further in this chapter. Figure 5.1 describes the general process, starting with the data collection, then features are extracted and finally the data is sent to an Artificial Neural Network - the classifier that was chosen. ANN will be used, since there are evidence that is possible to achieve great results using it in this type of problems [KLLK10] [KLLK08].

An ANN is a learning system that is adaptive and robust, even when fuzzy data are presented the classifier will understand the features, even if they are different from the ones used in training . Support Vector Machines will also be used to compare the results with a different classifier.

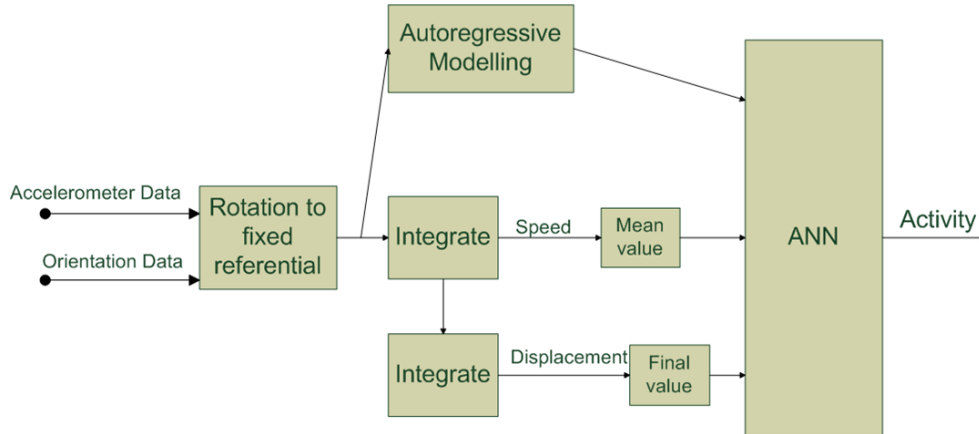


Figure 5.1: Block diagram of the physical detection system developed.

As explained in Section 3, it was decided that the system should recognize five different activities: standing, walking running, walking upstairs and walking downstairs.

5.2 Sensors available

Smartphones are embedded with many sensors and Android SDK enables the programmer to retrieve the data from the sensors [And12]. The following list shows a set of possible sensors that a smartphone with the Android operative system installed, but not all the smartphones comes with the full list embedded:

- Accelerometers: measure the acceleration applied to the smartphone along three axis. The force of gravity is included in the acceleration;
- Light: returns the light intensity detected;
- Magnetic field: measures the magnetic field around three axis;
- Gyroscope: returns the rate of rotation around each axis;
- Gravity: measures the force of gravity applied to the smartphone;
- Temperature: returns the temperature of the ambient;
- Proximity: calculates the distance between the smartphone and an object;
- Pressure: measures the ambient air pressure;
- Orientation: use data obtained by the magnetic field to detect the orientation of the smartphone.

5.3 Data collection

An Android application was developed to obtain the acceleration data, enabling an individual to select the physical activity that he is going to perform and saving the accelerometer data while he is performing it.

The data returned by the accelerometer is the acceleration relative to each axis (figure 5.3) of the mobile phone. Figure 5.2 demonstrates the data in a time frame of six seconds. Note that it is a discrete signal and the accelerometer returns the data at a specified rate. Android SDK enables the programmer to specify the rate pretended and there are three options available: fastest, normal, UI and game. However, it is not possible to define a specific rate in milliseconds and the data is not retrieved always at the same rate. Moreover, the signal has noise and the acceleration relative to gravity is also present.

Besides the accelerometer data, it is also important to record the orientation of the device, so the data from the orientation sensor will also be collected. In a similar way to the acceleration, it is possible to specify an approximate rate and the data is not retrieved at a constant rate. The data retrieved from the orientation sensor is a list of three values:

- azimuth: rotation around the Z axis (0 to 359).
- pitch: , rotation around the X axis (-180 to 180).
- roll: rotation around the Y axis (-90 to 90).

The application developed stores each activity performed in a file, associating the values of the orientation sensor and the accelerometer to a timestamp. When the user performs all activities he should upload the file to a server that is prepared to receive this data, using the option in the menu as figure 5.4 shows.

5.4 Features extraction

After extracting the raw data from the accelerometer it is necessary to process the data to retrieve useful information, and then send that information to a classifier. One of main issues is that the acceleration data depends on the orientation of the device, so it is essential to proceed to a transformation of the data so that the data is not dependent on the orientation.

After obtaining a signal of the acceleration transformed, two distinct techniques will be applied using this data: autoregressive modeling and integration. Applying these two techniques will result in a list of values that represent the original signal, and this list is the one that will be sent to the classifier.

The autoregressive model is useful to obtain a small number of coefficients that represents the original signal. Instead of sending to the classifier approximately 600 values for each 5 seconds of an activity, 9 coefficients are sent, 3 for each axis.

Integrating the signal of acceleration, the velocity is obtained. This value is useful to distinguish

Physical Activity Detection using Smartphones

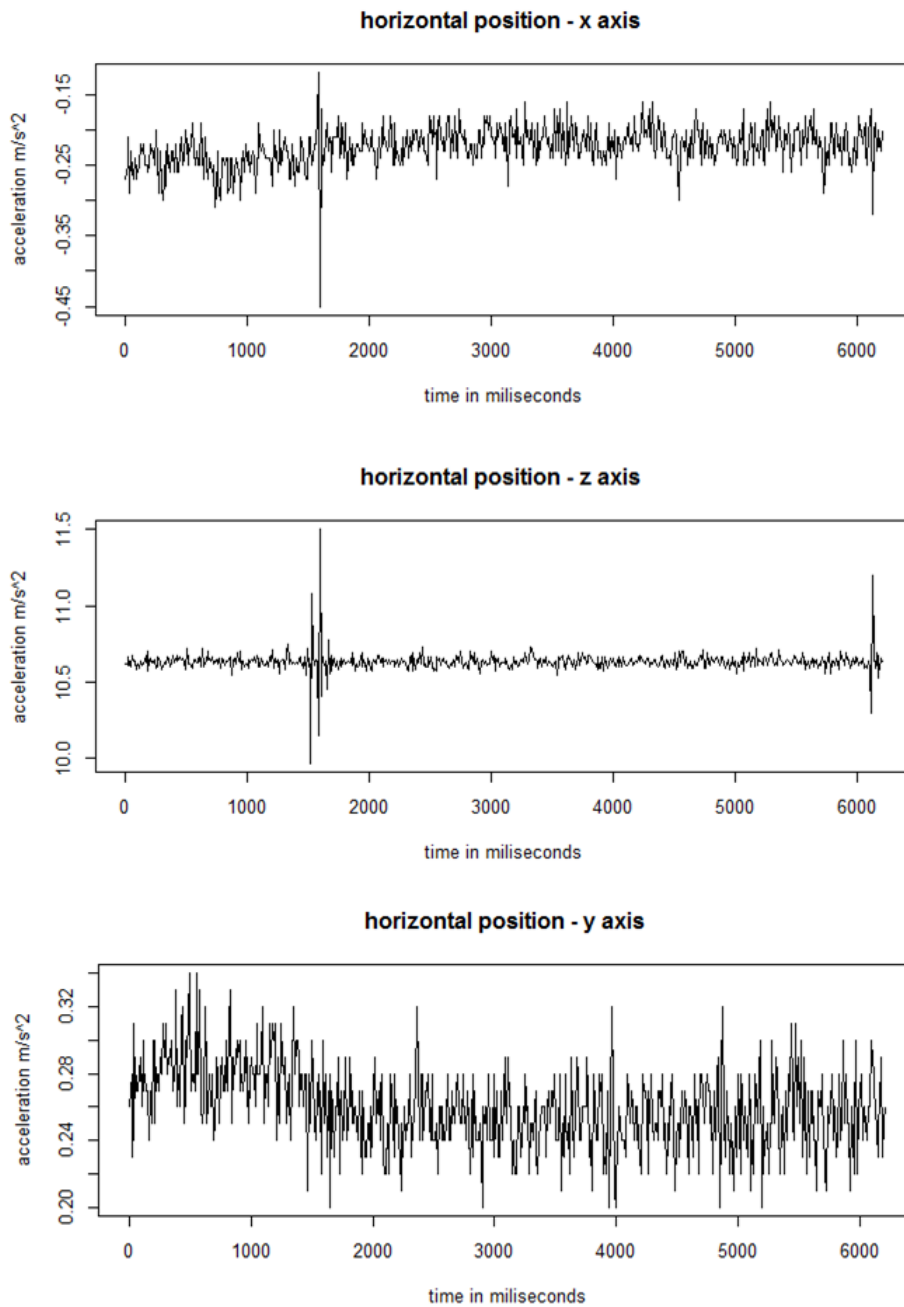


Figure 5.2: Plot of the acceleration in each axis, extracted by the Android application for the physical activity standing.

between walking and running for example. Moreover, the signal of velocity is also integrated obtaining the displacement travelled. The total displacement is also useful to distinguish between walking upstairs or downstairs with other activities.

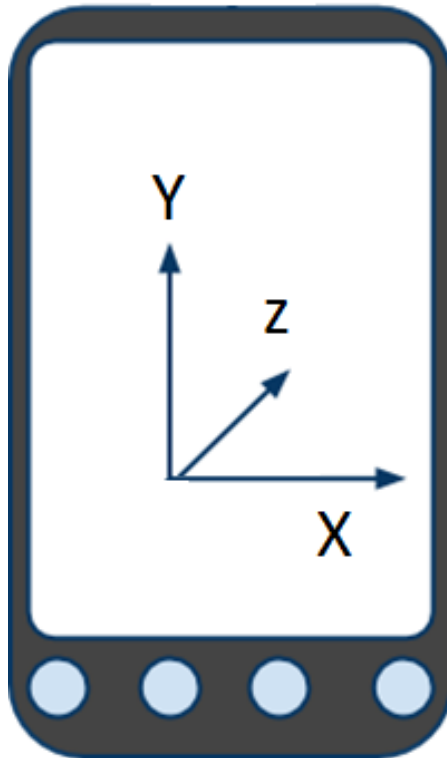


Figure 5.3: Axis relative to the smartphone.

5.4.1 Rotation of the accelerometer data

The acceleration on each axis returned by the accelerometer is dependent on the orientation of the device. So, the acceleration data of a person walking with his smartphone on the pocket will be different from the same person walking with the smartphone on his hand. The purpose of this thesis is to enable that a person can use the smartphone at any position without interfering with the physical activity recognition. To enable this functionality, the rotation of the device in relation to each axis of the smartphone must be used to rotate the accelerometer data.

Figure 5.5a shows a plot of the acceleration in each axis for a timeframe of 5 seconds, while the person was carrying the smartphone on his hand. In this case, there is a rotation of approximately 45 degrees around X axis. The strategy is to rotate the accelerometer data -45 degrees around X axis to obtain the signal with a standard orientation. Figure 5.5b shows the signal of the accelerometer after the rotation.

The strategy used is to apply a rotation on each point. Consider the rotation matrices on figure 5.6, that are used to apply a rotation in Euclidean space. If the purpose is to rotate a point in relation to an axis, one should multiply the corresponding matrix, replacing the desired angle, by the column vector containing the coordinates of the point. For instance, to make a Z-axis rotation after applying the multiplication the coordinates of the new point will be the following:

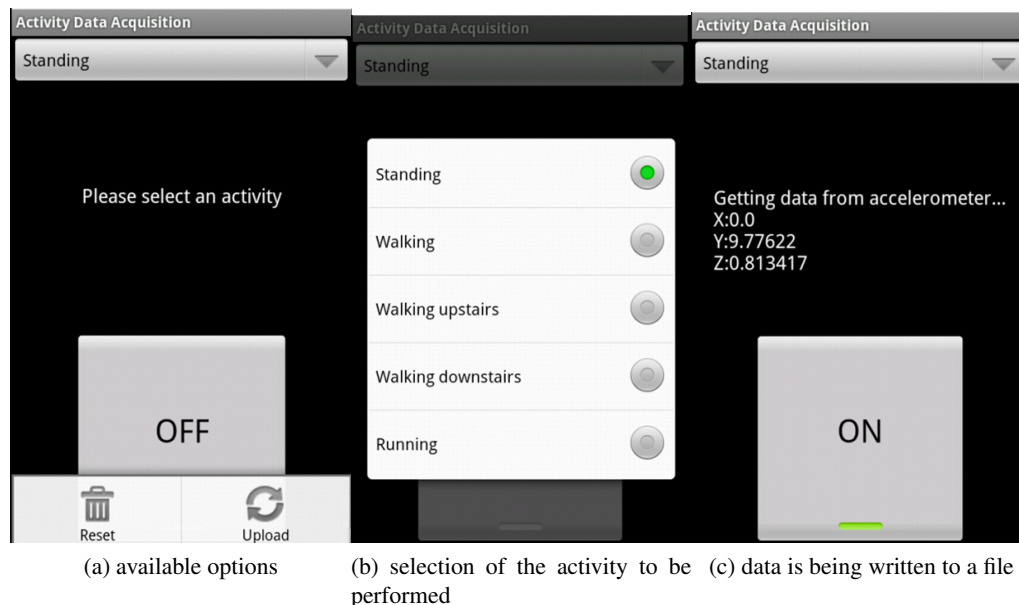


Figure 5.4: Android application screenshots to collect data.

$$x' = x * \cos q - y * \sin q \quad (5.1)$$

$$y' = x * \sin q + y * \cos q \quad (5.2)$$

$$z' = z \quad (5.3)$$

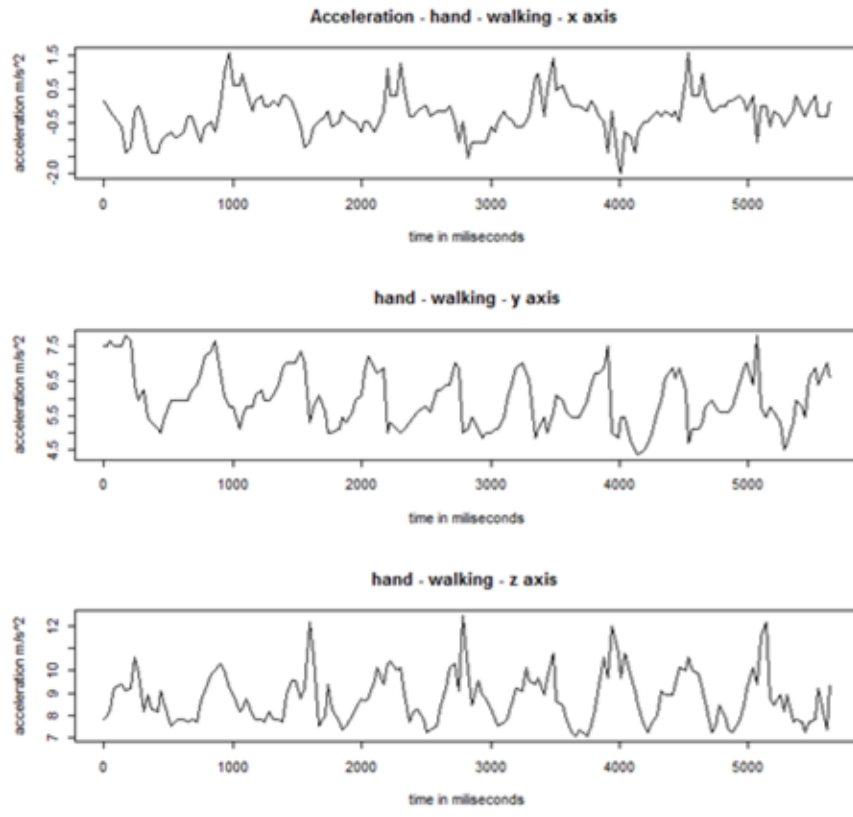
An implementation of an algorithm that use the matrices in figure 5.6 to rotate a point about an arbitrary axis developed by Glenn Murray was used [Mur11]. The algorithm was used to rotate each point of the acceleration about the three axis using the values retrieved by the orientation sensor. After obtaining all the rotate points, a new dataset is constructed will these orientation independent values.

5.4.2 Autoregressive modeling

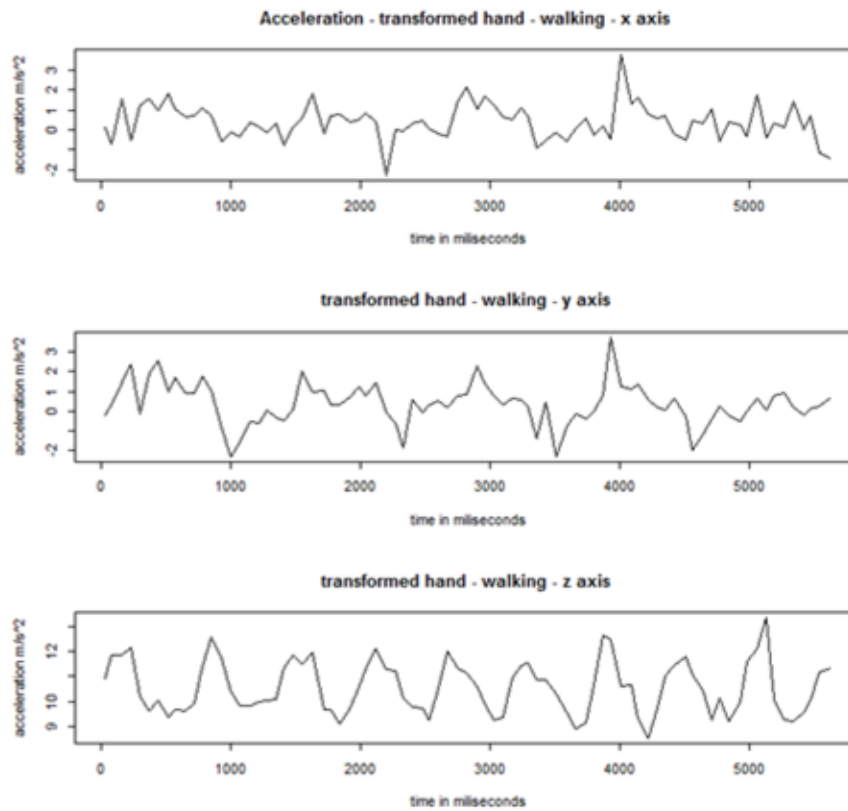
The accelerometer returns three distinct signals representing the acceleration on each axis. This raw data has the problem of the orientation dependency, but after solving that issue applying a rotation, important information about the physical activity performed is available. However, depending on the rate of the accelerometer and the hardware chosen, the amount of data is always huge. For instance, on the LG Maximo Black smartphone, 250 events of the accelerometer were obtained in a timeframe of 5 seconds, so the result are three signals each one with 250 points. As the purpose is to send this data to a classifier in order to learn the multiple activities, 750 points have to be given to the classifier for each timeframe of 5 seconds.

As described in section 2.3.1.1 it is useful to apply a technique of linear prediction that aims to predict an output of a system using its previous outputs. In signal processing, one of the most

Physical Activity Detection using Smartphones



(a) Acceleration signal when walking.



(b) Walking signal after rotation.

Figure 5.5: Comparison of the acceleration signal before and after a rotation.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Figure 5.6: Transform matrix.

used is the Autoregressive model. Depending on the order of the model, a set of coefficients are calculated representing the original signal. For instance, if the model is of order 3 it has 3 coefficients that can be used to obtain the original signal. These values can be crucial when facing a problem of pattern classification like physical activity recognition, because they can be used as the input of the classifier instead of all the values of the signal. The signal representation is important because instead of sending to an ANN 600 values only 9 values are used. The learning process is fastest and the generalization is easier.

An important issue is to decide how many taps, the order of the model, to use in order to extract coefficients representative of 5 seconds of a physical activity, for the acceleration in each axis. The process of calculating the coefficients has always an error associated. Increasing the order of the model minimizes that error, but having too many taps also increasing the processing time of the application that will calculate to coefficients. The strategy used was to start with a model of order

1 and verify the value of error. After that, the model of the error was increased until the error reach a constant value. Figure 5.7 shows a graph representative of this iterative process. By analyzing the plot one can conclude that using 3 taps a reasonable error is obtained, and when increasing the number of taps the error does not change.

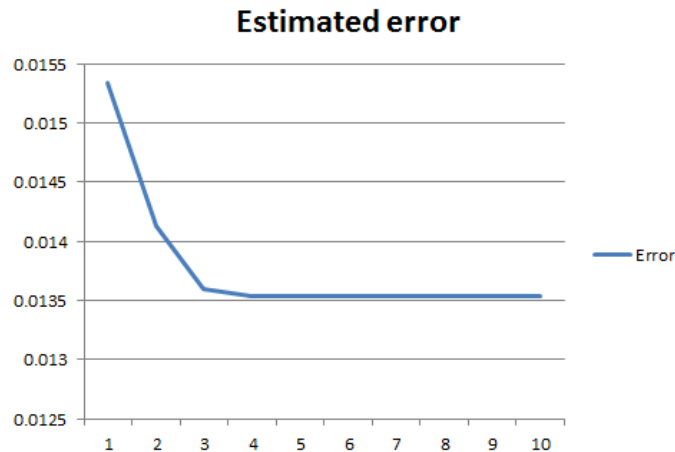


Figure 5.7: Evolution of the error of the autoregressive modeling when increasing the order.

In order to calculate the coefficients of the model using different taps, the software program R was used. R or GNU S, is an open source programming language and software for statistical purposes. It is available a library that fit an autoregressive time series model to a specified data. This library enables the programmer to specify the desired method, order of the model and the data representative of the discrete time series.

Since the Burg's method is one of the methods that achieve better results [RKEV03] [MDHD96], it was chosen to be used in this thesis. After selecting the order of the model to be used using the R software, it was implemented an algorithm based on the Burg's method using the Java programming language. The code developed was adapted from an open source program in C, developed by Paul Bourke [Bou98].

5.4.3 Extraction of velocity and displacement

Another useful information to distinguish between different physical activities is the median velocity of that activity and also the total displacement traveled. For instance, it is expected that a person when running will be changing is position with an higher velocity than when walking. Moreover, to distinguish between walking upstairs and walking downstairs, the displacement and velocity in the vertical position can be used to easily differentiate these two physical activities. It is known that the velocity is the integral of the acceleration and the displacement is the double integral of the acceleration, or the integral of the velocity. So, in order to obtain these to values based on the signal of the acceleration one must proceed to an integration, using a method of numerical integration like the Trapezoidal rule. This technique is based on the assumption that the integral

of a function is approximate by the area of the region bellow the graph, and can be mathematically expressed by the following formula:

$$\int_a^b f(x)d(x) = (b - a) * ((f(a) + f(b))/2) \tag{5.4}$$

To obtain the function of velocity this formula must be applied more than one time, depending on the step pretended, and then sum all of the areas sequentially. If an high value for the step is chosen, a lot of integrals must be calculated but the result will be a function representing the integral of the original function with a low error. Figure 5.8 shows at the left side three different signals representing the acceleration on each axis. In the middle is present the respective signals after integrating the original signals of the acceleration, showing the instant velocity. Finally, the third column presents the displacement, obtained by the integration of the previous signals of the velocity.

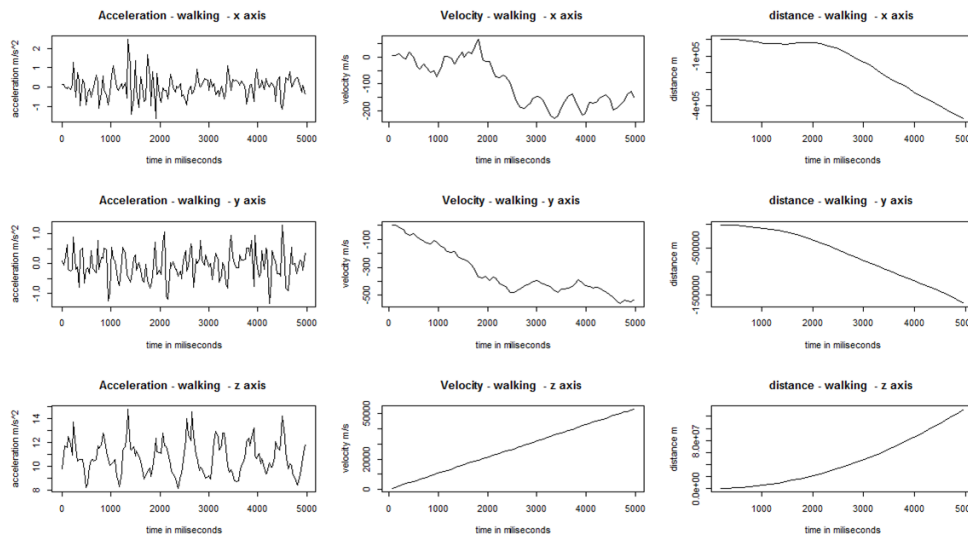


Figure 5.8: Plot of the original signal, velocity signal after integration and the displacement after double integration.

The signals of velocity and displacement in each axis are useful information about the physical activity performed. However, these data will not be used directly as an input to the classifier. In the case of velocity it is not necessary to know the instant of velocity at every time, it is more important to obtain a median velocity along all the physical activity. The evolution of the displacement is also not relevant, being more important to know the total displacement traveled. Moreover, if all these values were sent to a classifier, the input would be huge.

Firstly, the median of the velocity in each axis is calculated using the various signals of the velocity. After that, the module of the vector constituted by the sum of the three components of the median velocity in each axis is calculated. This module will be one of the features extracted to

be sent to the classifier, constituting important information indicating the velocity along the three axis.

The signals of displacement will also be used in order to extract useful information. The most important information that can be extracted is the total displacement traveled along the physical activity performed, being present in the last point of the discrete signal of the displacement. Using the values corresponding to the total displacement traveled in axis X and Y, the module of the sum of the vectors will be calculated. This value will be used also as an input to the classifier and it is important to distinguish between walking and running, for example, since the last one will have a displacement much higher in this direction.

Finally, the value of the total displacement along Z axis will also be used as an input to the classifier, since it is very useful to distinguish between walking upstairs and walking downstairs. It is expected that walking downstairs will have positive values of the displacement in the vertical direction and walking upstairs will hold negative values.

5.5 Classification

The general purpose of the system developed is to recognize five different physical activities: walking, standing, running, walking upstairs and walking downstairs. The recognition is performed using the data from the embedded smartphone sensors. This problem belongs to the group of problems of pattern classification. In this case, the classifier will identify the pattern of the acceleration along each absolute axis, obtained for the embedded accelerometer and rotation sensors, and then will be able to recognize physical activities.

The paradigm of supervised learning be used, giving to the classifier a dataset of inputs and the corresponding ideal output. The classifier will be able, after the learning phase, to generate an output based on the input [Bis06]. So, an initial phase of training will be applied, whereas the features of the physical activity and the correct output is given to the classifier. .

Note that the input of the classifier will not be the raw acceleration data returned by the accelerometer of the smartphone. The list of features explained in Section 5.4 will be used: 9 coefficients calculated by the autoregressive modeling technique that represents acceleration and the values of the median velocity and the total displacement.

Artificial Neural Networks can be used as a classifier, and the literature indicates that in the field of human activity recognition it achieves an high accuracy since its learning system is adaptive and robust, even when fuzzy data is presented [KLLK10] [KLLK08]. So, Artificial Neural Networks will be the focus of this thesis. Additionally, Support Vector Machines will also be used as a term of comparison.

5.5.1 Artificial Neural Network

The architecture of an Artificial Neural Network (ANN) is crucial to obtain good results in the classification process. Figure 5.10 shows a standard ANN with multiple parameters. The first layer is a set of input neurons, in this case there will be a neuron for each feature.

The last layer is the output of the system, so it will have 5 different neurons, one for each physical activity to be recognized. In the initial phase, the supervised training consists of giving to the network values to the input and output neurons. The value of an output neuron is 0 or 1 depending on the physical activity performed. So the values of the output neurons will be set on the training phase to the values present in the following table:

Physical Activity	neuron 1	neuron 2	neuron 3	neuron 4	neuron 5
Standing	1	0	0	0	0
Walking	0	1	0	0	0
Running	0	0	1	0	0
Walking upstairs	0	0	0	1	0
Walking downstairs	0	0	0	0	1

How many hidden layers and the number of hidden neurons present in each layer is not a straightforward decision. It depends on the noise and amount of data used as input and also if the function to learn is linear or nonlinear [Bis06]. One hidden layer may be enough to generalize the model. Since the literature does not indicate a specific value for these parameters, an iterative approach starting with a small network increasing the number of hidden neurons and layers was taken. On each iteration the total mean square error was calculated, when this value does not decreases from one iteration to another the process stops. Figure 5.9 documents the iterative process taken along with the evolution of the error, being possible to conclude that 5 hidden neurons with just 1 hidden layer do not decreases the error. In section 5.5.3 the total mean square error and other metrics used to evaluate the accuracy of the ANN will be explained.

Finally, the learning rate established to train the network is also crucial. Learning too fast can be bad, because the network can lead to a divergent result, and so will not detect the desired patterns. Dynamically changing the learning rate during the training is also an option, and can be a good option when the network is not achieving the accuracy desired. A learning rate of 0.2 was used in the scope of this thesis, since it achieved good results as explained on section ??, it was not necessary to try other values.

5.5.2 Standardization

Standardizing is the process of subtracting a set of values by a measure of location and dividing by a measure of scale. For instance, subtracting the mean and divide by the standard deviation is one of the methods of standardization. This concept can be applied to the input data that is going to be sent to the input neurons. However, it is not always necessary to use this method, mainly depends on the activation function that is being used on the hidden neurons. Moreover, when the input data has values with very distinct ranges, it can be crucial to do this process. For instance, if the values on the first input neuron have a range between 0 and 10, and the values on the second neuron range from 0 to 1000, they will have a different variability and this will have an high impact on the training phase, since the importance of each neuron is well represented. Standardization is crucial also because of weights initialization, if the values are centered to zero local minima will probably

Hidden neurons	Layers	Total Mean Square Error
1	1	0.148021796
3	1	0.148346444
5	1	0.146076122
10	1	0.146120166
15	1	0.146753679
20	1	0.200551794
30	1	0.147644133
50	1	0.175461936
1	2	0.146224782
3	2	0.143944384
5	2	0.14284034
10	2	0.142545276
15	2	0.146923452
20	2	0.146923452
30	2	0.173182163
50	2	0.178254922
5	3	0.14284034
10	3	0.142545276
10	4	0.142545276

Figure 5.9: Iterative process to decide the number of hidden layers and neurons.

be avoided.

Because our data is as mentioned, standardization was done using the mean and standard deviation. Firstly, using the dataset to be used in the training phase the mean and standard deviation are calculated. After computing this two values, all the values from the training dataset will suffer a standardization, applying the following transformation to each value:

$$x'_i = \frac{x_i - \text{mean}}{\text{standarddeviation}} \quad (5.5)$$

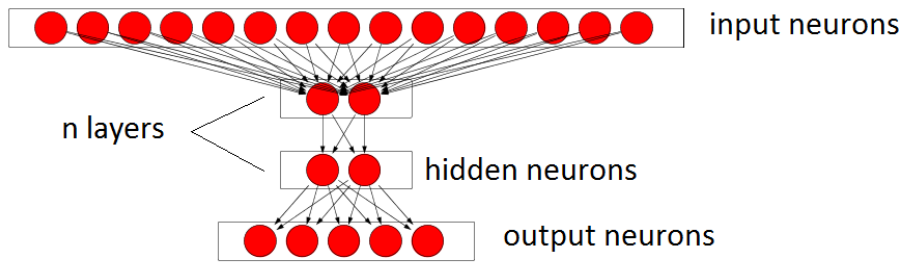


Figure 5.10: Example of ANN with 2 layers each one with 2 neurons, 15 input neurons and 5 output neurons.

When testing the network with a new dataset, before sending the data to the network it is also necessary to proceed to a standardization using the mean and standard deviation calculated using the training data.

5.5.3 Metrics

Validation is an important step when constructing a model and defining the architecture of an Artificial Neural Network. A concrete theory to analyze the efficiency and correctness of the network does not exist, but there are metrics to evaluate the performance of the network [Gra07]. For instance, the mean squared error (MSE) is often used to compare different architectures and decide which one is the best using this value. MSE is expressed by the following formula:

$$MSE = \frac{\sum_{i=1}^n \sum_{j=1}^m (o_{ij} - t_{ij})^2}{n} \quad (5.6)$$

Where n is the number of input neurons, m is the number of neurons in the output layer, o is the output of the output neuron j , t is the ideal value of the output neuron j , and the input neurons are contained in vector i . MSE is used to evaluate the network not the classification process.

The confusion matrix is a well known table used in the area of machine learning, specially in supervised learning. It can be used to validate if the network is solving a specific classification problem. The columns represents the predicted class, and the rows represents the desired output or class. An example of a confusion matrix used in the scenario of physical activity recognition is present in Figure 5.11.

Analyzing the Figure it is possible to conclude that, for instance, standing was predicted 10 times, and those predictions always met the expected values. However, the algorithm predicted 1 time running, when the actual activity was walking. So, using this kind of table is helpful to understand which classes the system is predicting well. Moreover, as the name indicates, it gives information about the confusion between two classes, for instance if the system is often predicting running instead of walking or vice-versa, it indicates that the system does not easily distinguish between these two activities.

Physical Activity Detection using Smartphones

Predicted	Actual class				
	Standing	Walking	Running	Walking upstairs	Walking downstairs
Standing	10				
Walking		10			
Running		1	1		
Walking upstairs				10	
Walking downstairs					10

Figure 5.11: Example a confusion matrix applied to the classification of physical activities.

Besides the confusion matrix, another useful information is the percentage of success on the recognition of each activity. This information can also be extracted from table 5.11 dividing the total of the correct outputs by the size of the dataset. It can be useful to easily identify the activities that are not being recognized with success, and quantify that situation with a percentage.

5.5.4 Training and testing

Classification always involves two distinct phases: training and testing. At the first stage, the classifier will learn the patterns based on a training dataset. The paradigm used is the supervised training, giving to the classifier the input data and the corresponding class. After training the network another dataset will be used to test, and a confusion matrix is generated in order to analyze the performance of the network.

In order to train and test the classifier features data from the acceleration and rotation sensor are necessary. The collection of data was done using the application described in Section 5.3 generating files contained a dataset with the raw data of the accelerometer and the orientation of the device. After that, a java program developed will import that files and extract the features described in Section 5.4 and proceed to standardization. After the feature extraction, it will shuffle the dataset with the features, and separate the dataset in two different datasets with the same size, one to be used in training and another to be used in testing. The files will contain multiple lines and each one will contain new input to the network and have the following format: "feature 1, feature 2, ..., feature n, output value 1, output value 2, output value 3, output value 4, output value 5". Note that the output values depends on the physical activity corresponding to the input values. Finally, two files are available, one containing the dataset to be used in training and the other to be used in testing.

To train and test the network, a software was created that use Neuroph framework [ZS12] to create an ANN using the Java programming language, using the datasets computed before. Furthermore, it generates a confusion matrix and the percentage of success in detecting each physical activity.

Neuroph also has a tool based on Netbeans entitled "Neuroph Studio" that enables the creation of an ANN using a visual menu. It is also possible to import the files with the datasets to train and test the network. In the training phase, Neuroph Studio shows a graph of the network, with the weights between each neuron visible (Figure 5.12).

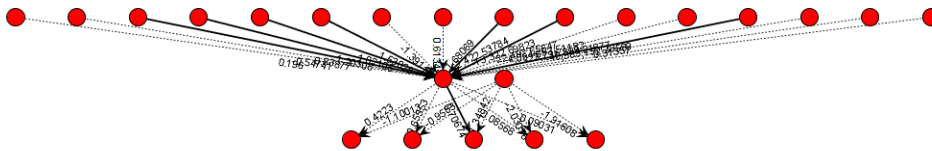


Figure 5.12: Example an ANN with one layer and one hidden layer generated by Neuroph Studio.

Moreover, it is possible to analyze in runtime the evolution of the error of the network, since the Neuroph Studio generates a plot of the error like Figure 5.13 shows. This plot is useful to conclude how many iterations of training is necessary to achieve the error desired, and it is also possible to stop the training manually.

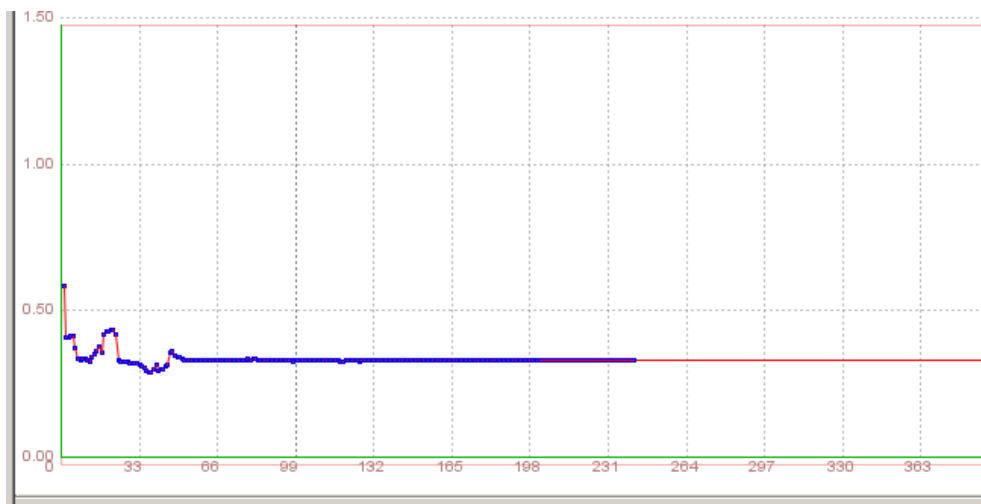


Figure 5.13: Plot of the evolution of the error in the training phase generated by Neuroph Studio.

After training the network using Neuroph studio, the framework enables the user to import a dataset to test the network. Testing will generate a file exporting the output calculated for each input and the total mean square error obtained.

The software program R was also used to compare and confirm the results obtained by the java program developed using the framework Neuroph. R has a library for Artificial Neural Networks [?], enabling training and testing using external files containing the corresponding datasets. It also generates the total mean square error and a confusion matrix.

Physical Activity Detection using Smartphones

To validate the network and the features chosen a person using the smartphone LG Maximo Black collected data using the application described on Section 5.3. The data correspondent to 7 minutes of each physical activity was collected and sent to a server. Part of the data was used to train, and the rest to test the network.

The java programa developed using the Neuroph Framework was used to training the network with the training dataset, and then the network was tested with the other dataset obtaining the results present in Figure 5.14. Analyzing the table it is possible to conclude that standing, walking and running are easily recognized by the classifier, obtaining an accuracy near 100 per cent. Note that the success rate is the percentage of correctly detected events. However, the classifier does not distinguish accurately between walking upstairs or downstairs. The last row shows that the classifier predicted 10 times walking upstairs, when the actual activity being performed was walking downstairs. Although, the classifier presents this confusion pattern, walking downstairs and upstairs achieve also reasonable results, with a percentage of success 75 and 97.5 respectively .

In order to compare the results obtained using the Neuroph Framework, the same datasets were used to train and test an ANN using R. The confusion matrix obtained was similar and is present in Figure 5.15. The confusion between walking upstairs and downstairs is also present. Note that the fact that the results are slightly different is because the random weight initialization. Even using the same program, parameters and input data, the results are always slightly different.

Finally, Support Vector Machines were also used in order to compare the accuracy with an Artificial Neural Network. The confusion matrix is present in Figure 5.16, and is similar to the tables using ANN. According to the results, using Support Vector Machines can be an alternative, but does not bring better results.

		Actual Class					Success
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	38	1	0	0	0	95.0%
	Walking	0	39	0	0	1	97.5%
	Running	0	0	40	0	0	100.0%
	W. Downstairs	2	0	0	30	0	75.0%
	W. Upstairs	0	0	0	10	39	97.5%

Figure 5.14: Confusion matrix of the classification process an Artificial Neural Network developed on Neuroph Framework.

		Actual Class					Success
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	40	0	0	0	0	100.0%
	Walking	0	38	0	2	0	95.0%
	Running	0	0	40	0	0	100.0%
	W. Downstairs	0	0	0	32	8	80.0%
	W. Upstairs	0	5	0	0	35	87.5%

Figure 5.15: Confusion matrix of the classification process using an Artificial Neural Network on R

Physical Activity Detection using Smartphones

		Actual Class					Success
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	40	0	3	0	0	100.0%
	Walking	0	38	0	0	8	95.0%
	Running	0	0	36	0	0	90.0%
	W. Downstairs	0	1	1	34	0	85.0%
	W. Upstairs	0	1	0	6	32	80.0%

Figure 5.16: Confusion matrix of the classification process using Support Vector machines on R

Chapter 6

Results

6.1 Remote M2M Application

A system to enable the remote monitoring of vital signs was developed, following the M2M paradigm. The framework MyContext implements the XMPP protocol and was used to develop the applications. This chapter will describe the result obtained and how the system was tested. Figure 6.1 shows multiple devices that are all interconnected exchanging information. Three mobile phones have installed the application Vital Tracker, each one representing a different user as the legend suggests.

The user with the device logged in as "anaaguiar" was using the Vital Tracker application to monitor herself. The heart rate sensor was on her chest and connected with the device. The application was publishing the position, acceleration data and heart rate.

Another user was using the Vital Tracker application in a different device, also with the purpose of monitor himself, being logged in as "dlucani". This user was publishing only the acceleration data and position.

The third mobile phone was used by the user "jprud" aiming to monitor heart rate, physical activity and geographic location of the user "anaaguiar".

Finally, the tablet application has installed a different version of the Vital Tracker, which enables a user to monitor multiple persons at the same time, as Figure 6.2 shows. It is possible to see that the application is showing the context data of the two users that are publishing information. Moreover, a geographic barrier is defined in the map, and a notification is being received, indicating that the user "anaaguiar" has left the area defined by the barrier.

Multiple providers were developed, with the purpose of publishing context to the context broker. These providers consume a type of context (heart rate, features of physical activity or GPS data), do specific calculations and then publish again another type of context. Each provider is a Java application that must be running in order to disseminate the information between mobile devices, so the laptop present in Figure 6.1 is running all the providers developed.

Results

In order to test the provider responsible of detecting an heart rate abnormality, the user "anaaguiar" performed physical activities like running and walking upstairs and downstairs, with the purpose of augmenting the heart rate. When the heart rate of the user reached 100 bpm, a notification was received on the device of the user and the other devices that were monitoring this user: the tablet and the mobile phone logged in as "jprud".

While performing physical activities the user "anaaguiar" was also able to see the on the device the label of the physical activity being changed, since the provider to detect the physical activity was running on the laptop. The user perceived that the detection was having a good accuracy, however errors occurred. Most of the times, the errors consisted of sporadic false detections, for instance when standing sometimes the system recognized that the user was walking for a moment, but quickly returned to the correct value. Furthermore, a delay was detected of more or less 8 or 9 seconds. The reason for this delay is because of the distributed solution, since the acceleration data is sent to the context broker at PT Inovação, then it is consumed by a source provider on the laptop at FEUP. After that, the source provider sends the information back to PT Inovação and it returns to the composite provider responsible of detecting the physical activity located at FEUP. After the recognition, then the recognized activity is sent to the context broker again that forwards that information to each mobile phone that had subscribed to this context. Besides the time lost in processing and sending the data, the mobile phone is collecting the acceleration data only sends the data relative to the physical activity after a time frame of 5 seconds, proceeding to a process of features extraction before sending it.

The functionality of geo fencing was tested using the application Fake GPS installed on the mobile phones. This applications enables the user to change the geographic location perceived by the mobile phone, without the need to move to a different place. This is useful to change the geographic position of this user, firstly placing the user inside the geographic barrier defined and then placing the user outside it. The tablet received a notification sent by the geo fencing provider notifying that the user "anaaguiar" is outside the recommend area.

After proceeding to the tests explained before, it is possible to conclude that the system has the functionalities that were defined, being possible to monitor important information, using a smartphone or a tablet. A problem detected is that the application Context Agent installed on the mobile phones, that was developed by PT Inovação and is responsible of the communication between the Vital Tracker application and the context broker, is that it shuts down when the Internet connection is lost, being a serious problem for an application designed for a mobile environment. This is an issue that is not in the scope of this thesis, but will be solved by the developers of this application.

6.2 Physical Activity Detection using Smartphones

The results relative to the M2M application were presented in Section 6.1. The functionalities of the system are described, and the tests that were done. Since a service to recognize physical



Figure 6.1: Test of the system developed, using two mobile phones as context sources, and another mobile phone and a tablet as context consumers.

activities is integrated in this system, this module was also tested but not using metrics to evaluate the accuracy of the network. So, this Chapter aims to present the results in the area of physical recognition as an independent module. In Section 5.5.4 the process to train an Artificial Neural Network to recognize physical activities using the accelerometer data was described. The classifier was trained using the data extracted when one individual was doing multiple activities, using only one smartphone. The dataset that was used for the training is constituted by 3 minutes and

Results

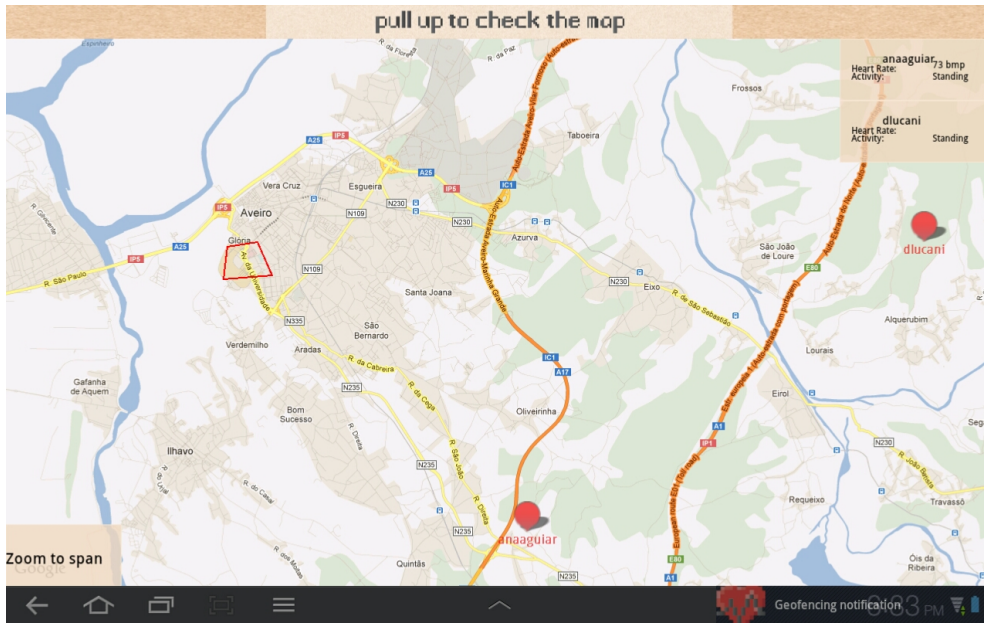


Figure 6.2: Vital Tracker application in a tablet, monitoring two users and showing a geo fence alarm.

20 seconds of each activity, representing 40 training sets, constituting 5 seconds of an activity each one. The network was tested using accelerometer data from the same subject, obtaining an accuracy of near 100 per cent for walking, running and stand, and 80 per cent for walking upstairs and downstairs (Figure 6.4).

However, it is not known how the network will behave in natural conditions when the user may change from one physical activity to another very often. Moreover, if another smartphone or another person is used to test the network the results are unpredictable analyzing only the mentioned test.

So, a route at the Faculty of Engineering of University of Oporto was defined, including all physical activities. Figure 6.3 illustrates this route, where the person must follow the course. Note that, this route includes doors and other obstacles that can interfere with the physical activity that need to be executed at each phase of the course.

Five individuals volunteered to travel along the route, using the application to collect data described at Section 5.3. While executing the physical activities, the individual was also indicating the transition from one physical activity to another using the application. The population of the individuals is constituted by two women and three men, and one of the women is the subject used to train the network. Moreover, each individual travelled along the route 4 times, 2 times using the smartphone LG Maximo Black, used for training, and 2 times using another smartphone, a Samsung device.

The results that were obtained testing the network with this data is presented in a confusion

Results

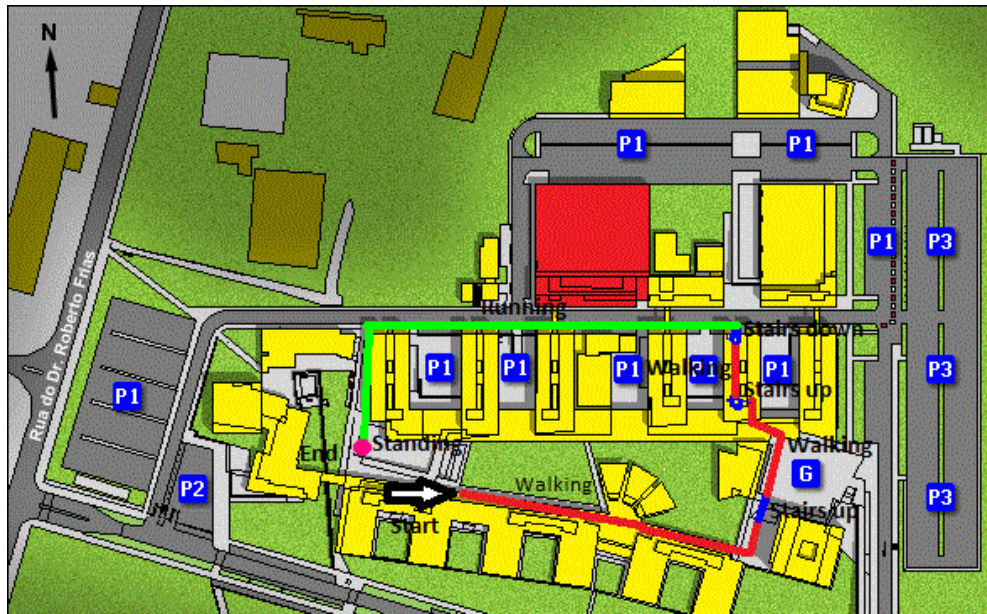


Figure 6.3: Route at FEUP.

matrix (Figure 6.4). Analyzing the table is possible to conclude that standing and walking are recognized with an accuracy lower than 15 per cent, and they are often confused with walking upstairs and downstairs. Running has a higher accuracy, reaching 45 per cent, however the confusion with physical activities practiced on stairs is also present. Finally, walking upstairs and downstairs are the activities with an higher accuracy, 60 and 51 respectively, but are often confused.

		Actual Class					Success
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	15	6	5	1	0	11.0%
	Walking	3	126	3	3	9	15.0%
	Running	3	3	150	1	0	45.0%
	W. Downstairs	76	406	87	102	98	60.0%
	W. Upstairs	43	281	87	63	114	51.0%

Figure 6.4: Confusion matrix of the test using all the data collected by the 5 individuals.

The results obtained for all the individuals using both the devices were not good results, since most of the activities were not recognized more than a half of the times. In order to understand why this is happening, the network was tested using only the data from the individual and the smartphone used for training. Figure 6.5 illustrates the confusion matrix obtained in this test, where all physical activities achieve an accuracy higher than 70 per cent, except running. The reason why the accuracy to recognize running is so low is due to the fact that the individual in the training phase used the smartphone on his pocket while running and in this test the user was grabbing the smartphone on his hand since it was necessary to interact with the application to

Results

indicate a transition from one activity to another. Considering this, the results obtained for this individual are good, since there are obstacles in the course that decrease the accuracy of the network.

		Actual Class					Success
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	8	4	0	0	0	80.0%
	Walking	0	90	1	0	0	90.0%
	Running	0	0	5	0	0	11.0%
	W. Downstairs	2	3	6	10	3	71.0%
	W. Upstairs	0	3	33	4	15	83.0%

Figure 6.5: Confusion matrix of the test using the data from the subject that trained the network, using the same smartphone that was used in training too.

Since the route is known, as well as the physical activities performed on each stage of the route, it is possible to know when in time the network had failed to recognize the activity. Figure 6.6 illustrates the physical activities predicted by the ANN along the route. Moreover, it shows the correct activity that the user was executing, being possible to analyze when the ANN had committed an error. Although in second 195 the ANN detected that the user was walking upstairs and the correct output was walking, the ANN is correct since there are stairs on this area, but since they only have 5 steps these stairs were ignored for the test. Another important situation is in second 220 when the ANN recognizes that the user is standing, when the correct output is walking. However, what is really happening is that the user stopped for a moment to open the door that exists on this area.

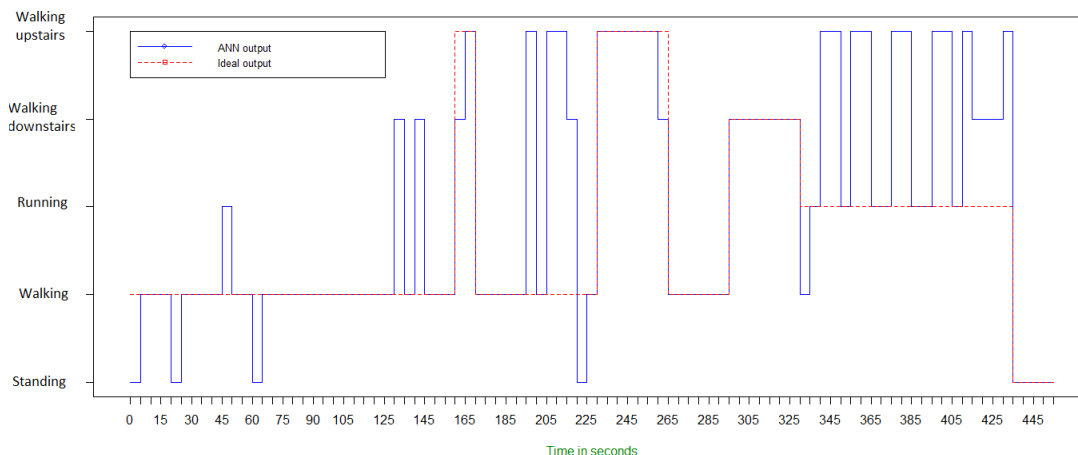


Figure 6.6: Evolution of the physical activities along the route for the subject and smartphone used in training.

Results

In order to understand if the hardware is important, the data from the same individual on the previous test was used, but only the data collected using the second smartphone. Figure 6.5 presents the confusion matrix of this test. Analyzing the table is possible to conclude that the recognition of the physical activities is highly dependent on the smartphone used, since the accuracy is very low, almost all activities are confused with walking downstairs.

		Actual Class					Success
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	7	1	16	0	0	58.0%
	Walking	1	0	0	0	0	0.0%
	Running	1	0	1	0	0	2.0%
	W. Downstairs	3	84	10	14	15	100.0%
	W. Upstairs	0	0	17	0	1	6.0%

Figure 6.7: Confusion matrix of the test using the data from the subject that trained the network, using a different smartphone.

Figure 6.8 shows a plot of the evolution of the recognition of the physical activities along the route, for the individual that trained the network using another smartphone. Analyzing the plot is possible to conclude that the classifier almost always predicts that the user is walking downstairs.

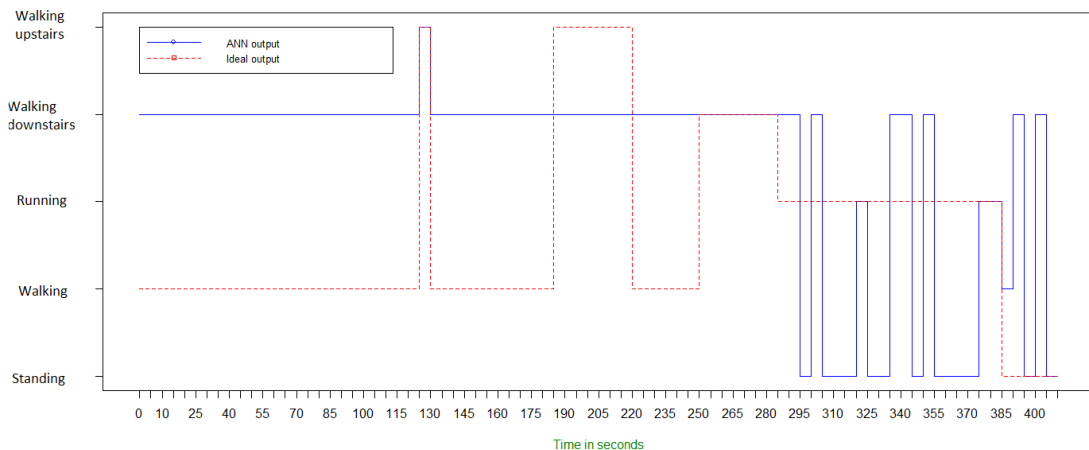


Figure 6.8: Evolution of the physical activities along the route for the subject used in training, with a different smartphone.

In order to compare the results obtained with the subject that trained the network and another subject, Figure 6.9 shows the result for another person using the smartphone using in training. The network does not recognize easily any of the physical activities, and the confusion that is always present is with walking upstairs.

Results

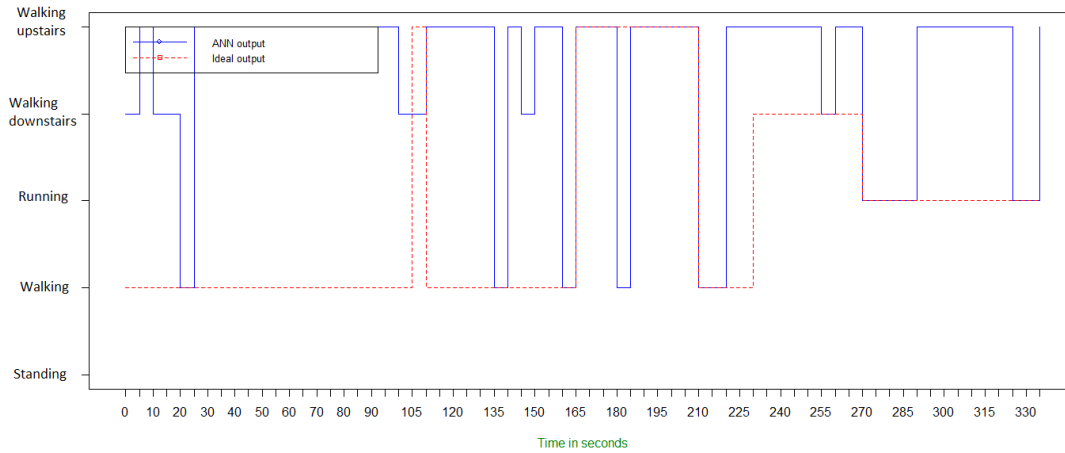


Figure 6.9: Evolution of the physical activities along the route for a subject that was not used in the training phase.

To understand if the low accuracy obtained when a user that had not collected data from his physical activity to train the network before testing, another test was done. Part of the data obtained by the tests of the 5 individuals using only one smartphone were used to train the network, and the rest was used to test. The results are presented in the confusion matrix present in Figure 6.10. An accuracy near 60 per cent is obtained for all activities. This is an important result since it proves that if the network is trained before for a specific individual before testing, the accuracy is high. A similar test was done using another smartphone and results obtained are identical (Figure 6.11).

		Actual Class					Sucess
		Standing	Walking	Running	W. Downstairs	W. Upstairs	
Predicted Class	Standing	19	2	1	2	1	59.0%
	Walking	6	149	1	8	12	62.0%
	Running	2	2	62	1	0	91.0%
	W. Downstairs	4	35	4	21	10	56.0%
	W. Upstairs	1	51	0	5	34	59.0%

Figure 6.10: Confusion matrix obtained after training the network with the data from the 5 individuals traveling along the defined route using LG smartphone.

Analyzing this results, it is possible to conclude that to achieve a good accuracy on the recognition of the physical activities it is crucial that the individual used in training has the smartphone on the same position than the individual that is testing the network. Moreover, if a different user is testing the network, it is probably that the patterns of his physical activity is distinct from the one used in training, so it is not guaranteed that the results will be good. Finally, it is also important to use the same smartphone at the training phase and when testing the network.

The features chosen to train the network were proven to be good to achieve an high accuracy detecting the physical activities. The only requirement is that an initial training phase for the

Results

	Actual Class					Success
	Standing	Walking	Running	W. Downstairs	W. Upstairs	
Standing	30	3	0	0	2	69.0%
Walking	1	32	0	9	14	51.0%
Running	1	5	36	1	0	81.0%
W. Downstairs	10	21	8	28	5	62.0%
W. Upstairs	1	1	0	5	34	62.0%

Figure 6.11: Confusion matrix obtained after training the network with the data from the 5 individuals traveling along the defined route using the Samsung smartphone.

individual and smartphone to be used must be done.

Results

Chapter 7

Conclusions

This thesis studied the problem of the creation of a remote application in an healthcare scenario, using the paradigm of M2M. A functional system was developed constituted by an application for smartphones that enables the monitoring of the geographic position, heart rate and physical activity being performed by the user. Moreover, the same application can be used to monitor the information of another person. Finally, a similar application was developed for a tablet, being possible to monitor the data of multiple users. Remote services were developed to receive and detect special conditions like an heart abnormality, a user passes a virtual geographic fencing or to recognize the physical activity being performed. All these applications were developed following an M2M architecture using a framework developed by PT Inovação.

A service that was developed is responsible of detecting the physical activity using the acceleration data returned by the mobile phone. This study was done without using the remote application, collecting acceleration data from different users and using a paradigm of supervised learning to train the classifier used - an Artificial Neural Network. After obtaining a good accuracy in the detection of the activities, the classifier was integrated in a service, being able to monitor the activity being performed in real time using the application Vital Tracker.

The main challenge in the physical activity recognition was to decide which features use to process the raw acceleration data. It was decided to use an Autoregressive model to filter the signal and proceed to a numerical integration of this signal to obtain the velocity and displacement. Moreover, the architecture of the network and the number of datasets to train the network were also important decisions. Initially the network was trained with the acceleration data of a user using always the same smartphone. After testing the network with data from the same user the accuracy of the network was near 100 per cent for standing, walking and running and near 80 per cent for walking upstairs and downstairs. However, after testing the network with other users and other smartphones the results were bad, since the network almost all the times detected the activity walking upstairs. So, the network was training with the data from all the users and the network was tested again, obtaining an accuracy near 60 per cent for all activities.

Conclusions

The system developed to recognize physical activities was proven to obtain good results when the user that is testing the network had previously trained it, and the smartphone used is the same in both phases.

As future work, a study of the generalization for multiple users and devices could be done. Furthermore, more activities could be included in the classifier, like getting up or laying down (transition physical activities). Besides that, testing could be done with another classifier and vote from multiple classifiers.

A study of M2M infrastructures could be performed, instead of using MyContext framework, Smack or other libraries could be used and the performance of each one should be compared. Finally, a study of energy optimization could be done, deciding where to proceed to the calculation of features of the physical activity: on the device or on the remote service.

References

- [And04a] P. Saint Andre. Extensible messaging and presence protocol: Core. ietf rfc 3920. Technical report, IETF, October 2004.
- [And04b] P. Saint Andre. Extensible messaging and presence protocol: Instant messaging and presence. ietf rfc 3921. Technical report, IETF, October 2004.
- [And10] P. Saint Andre. This document defines the standards process followed by the xmpp standards foundation. Technical report, XMPP Standards Foundation, March 2010.
- [And12] Android. Android sdk, 2012. Available at <http://www.developer.android.com>, accessed on 9 of Februray of 2012.
- [ANMF10] N. Armstrong, C.D. Nugent, G. Moore, and D.D. Finlay. Developing smartphone applications for people with Alzheimer’s disease. *Information Technology and Applications in Biomedicine (ITAB), 2010 10th IEEE International Conference on*, pages 1 –5, November 2010.
- [ATKI08] M. Ahad, J.K. Tan, H.S. Kim, and S. Ishikawa. Human activity recognition: Various paradigms. *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pages 1896 –1901, October 2008.
- [BA11] H.T. Bilgic and A.Z. Alkar. A secure tracking system for GPS-enabled mobile phones. *Information Technology and Multimedia (ICIM), 2011 International Conference on*, pages 1 –5, November 2011.
- [BGYW09] Alberto Bonomi, Annelies Goris, Bin Yin, and Klaas Westerterp. Detection of Type, Duration, and Intensity of Physical Activity Using an Accelerometer. *Journal of Medicine and Science in Sports and Exercise*, page 1770–1777, August 2009.
- [Bis06] C. M. Bishop. *Patter recognition and machine learning*. Springer, forth edition, 2006.
- [Bou98] Paul Bourke. Autoregression analysis (ar), November 1998. Available at <http://paulbourke.net/miscellaneous/ar/>.
- [CV95] C. Cortes and V. Vapnik. Support-vector networks. , pages 273–297, 1995.
- [ETS11] ETSI. Machine to machine communications (m2m). use cases of m2m applications for ehealth. Technical report, ETSI, March 2011.
- [Fau93] V. Laurene Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. Prentice Hall, first edition, 1993.

REFERENCES

- [fMS11] Centers for Medicare and Medicaid Services. National Health Expenditures, 2011. <http://www.cms.gov/>.
- [Fra06] W. Randolph Franklin. Point inclusion in polygon, 2006. Available at http://www.ecse.rpi.edu/Homepages/wrf/Research/Short_Notes/pnpoly.html.
- [Gra07] D. Graupe. *Principles of Artificial Neural Networks*. World Scientific, second edition, 2007.
- [Hoa12] Ahmad N. Pham C. Hoang, D. An Active Approach in Health Monitoring. *Biomedical Engineering Systems and Technologies, 5th International Joint Conference on*, February 2012.
- [Hop88] J.J. Hopfield. Artificial neural networks. *Circuits and Devices Magazine, IEEE*, pages 3–10, September 1988.
- [IET08] IETF. Routing over low power and lossy networks <http://datatracker.ietf.org/wg/roll/charter/>. Technical report, IETF, February 2008.
- [IET10a] IETF. Constrained application protocol (coap) <http://tools.ietf.org/html/draft-ietf-core-coap-06>. Technical report, IETF, March 2010.
- [IET10b] IETF. Constrained restful environments (core) <http://datatracker.ietf.org/wg/core/charter/>. Technical report, IETF, March 2010.
- [JG11] Edgar Domingues João Gonçalves. Context management. Technical report, PT Inovação, December 2011.
- [JKC07] Do-Un Jeong, Se-Jin Kim, and Wan-Young Chung. Classification of Posture and Movement Using a 3-axis Accelerometer. *Convergence Information Technology, 2007. International Conference on*, pages 837–844, November 2007.
- [JMM96] A.K. Jain, Jianchang Mao, and K.M. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, pages 31–44, March 1996.
- [JZ11] Wu Jian and Cai Zongyan. A method for the decision of a point whether in or not in polygon and self-intersected polygon. *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, pages 16–18, July 2011.
- [KLK08] A. M. Khan, Y. K. Lee, and T.-S. Kim. Accelerometer signal-based human activity recognition using augmented autoregressive model coefficients and artificial neural nets. *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 5172–5175, August. 2008.
- [KLLK10] A.M. Khan, Young-Koo Lee, S.Y. Lee, and Tae-Seong Kim. A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer. *Information Technology in Biomedicine, IEEE Transactions on*, pages 1166–1172, September 2010.
- [KNM⁺06] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G.m Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *Information Technology in Biomedicine, IEEE Transactions on*, pages 156–167, January 2006.

REFERENCES

- [LLL⁺11] Rongxing Lu, Xu Li, Xiaohui Liang, Xuemin Shen, and Xiaodong Lin. GRS: The green, reliability, and security of emerging machine to machine communications. *Communications Magazine, IEEE*, pages 28 –35, April 2011.
- [MDHD96] H. Schoonewelle M.J.L. De Hoon, T.H.J.J. Van Der Hagen and H. Van Dam. Why Yule-Walker should not be used for autoregressive modelling. *Annals of Nuclear Energy (1996)*, pages 1219–1228, 1996.
- [Mil10] Saint-Andre P. Meijer R. Millard, P. Xep-0060: Publish-subscribe. Technical report, XMPP Standards Foundation, July 2010.
- [Mon08] C. Jennings M. Kulahci Montgomery, C. Douglas. *Introduction to Time Series Analysis and Forecasting*. John Wiley and Sons. Inc., first edition, 2008.
- [MTIPL06] Evangelia Micheli-Tzanakou, Robert S. H. Istepanian, Costantinos S. Pattichis, and Swamy Laxminarayan. Ubiquitous M-Health Systems and the Convergence Towards 4G Mobile Technologies. *Topics in Biomedical Engineering. International Book Series*, pages 3–14, 2006.
- [Mur11] Glenn Murray. Rotation about an arbitrary axis in 3 dimensions, July 2011. Available at <http://inside.mines.edu/~gmurray/ArbitraryAxisRotation/>.
- [Org04] World Health Organization. Towards Age-friendly Primary Health Care, 2004. <http://whqlibdoc.who.int/publications/2004/9241592184.pdf>.
- [PM92] S.K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets, and classification. *Neural Networks, IEEE Transactions on*, pages 683 –697, September 1992.
- [RKEV03] K. Roth, I. Kauppinen, P.A.A. Esquef, and V. Valimaki. Frequency warped Burg’s method for AR-modeling. *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 5 – 8, October 2003.
- [RPB10] Yonglin Ren, R.W.N. Pazzi, and A. Boukerche. Monitoring patients via a secure and mobile healthcare system. *IEEE Wireless Communications*, pages 59 –65, February 2010.
- [SA05] P. Saint-Andre. Streaming XML with Jabber/XMPP. *Internet Computing, IEEE*, pages 82 – 89, 2005.
- [SA06] P. Saint-Andre. Xep-0119: Extended presence protocol suite. Technical report, XMPP Standards Foundation, August 2006.
- [SA09] P. Saint-Andre. XMPP: lessons learned from ten years of XML messaging. *Communications Magazine, IEEE*, pages 92 –96, April 2009.
- [SAP09] K. Tronçon R. Saint-Andre P., Smith. *XMPP: The Definitive Guide*. O’REILLY, first edition, 2009.
- [Ser09] S. Sergey. Programming the algorithm learning neural network back propagation. *Perspective Technologies and Methods in MEMS Design, 2009. MEMSTECH 2009. 2009 5th International Conference on*, pages 77 –78, April 2009.
- [Shi11a] F.Y.; Arash; Wang M.D. Shieh, Y.Y.; Tsai. Mobile Healthcare: Opportunities and Challenges. *International Conference on the Management of Mobile Business*, page 50, July 2011.

REFERENCES

- [Shi11b] Donghoon Shin. M-healthcare revolution: an e-commerce perspective. *First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, pages 23–25, May 2011.
- [TR11a] 3GPP TR. Service requirements for machine-type communications (mtc). Technical report, 3GPP, June 2011.
- [TR11b] 3GPP TR. System improvements for machine-type communications. Technical report, 3GPP, June 2011.
- [TS10a] ETSI TS. M2m communications; functional architecture. v1.1.1 www.etsi.org/m2m. Technical report, ETSI, August 2010.
- [TS10b] ETSI TS. Machine-to-Machine communications (M2M); M2M service requirements V1.1.1 www.etsi.org/m2m. Technical report, ETSI, August 2010.
- [VBdV⁺96] P.H. Veltink, HansB.J. Bussmann, W. de Vries, WimL.J. Martens, and R.C. Van Lummel. Detection of static and dynamic activities using uniaxial accelerometers. *Rehabilitation Engineering, IEEE Transactions on*, pages 375–385, December 1996.
- [WL90] B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: perceptron, Madaline, and backpropagation. *Proceedings of the IEEE*, pages 1415–1442, September 1990.
- [WTJ⁺11] Geng Wu, Shilpa Talwar, Kerstin Johnsson, Nageen Himayat, and Kevin Johnson. M2M: From mobile to embedded internet. *IEEE Communications Magazine*, page 36–43, April 2011.
- [WWL⁺09] A.K.-S. Wong, Tim Kam Woo, A.T.-L. Lee, Xiaoming Xiao, V.W.-H. Luk, and Kwok Wai Cheng. An AGPS-based elderly tracking system. *Ubiquitous and Future Networks, 2009. ICUFN 2009. First International Conference on*, pages 100–105, June 2009.
- [Xu07] G. Xu. *GPS, Theory, Algorithms and Applications*. Springer, second edition, 2007.
- [ZS12] Jon Tait et al Zoran Sevarac, Ivan Goloskokovic. Neuroph, 2012. Available at <http://neuroph.sourceforge.net/>, accessed on 9 of Februray of 2012.