Universidade do Minho
Escola de Engenharia
Departamento de Informática

PG35401 Paulo Alexandre Azevedo Ferreira

A context-aware system architecture to assist workout plans

October 2019

Universidade do Minho
Escola de Engenharia
Departamento de Informática

PG35401 Paulo Alexandre Azevedo Ferreira

A context-aware system architecture to assist workout plans

Master dissertation
Master Degree in Computer Science

Dissertation supervised by
Paulo Novais, Universidade do Minho
Juan Pavón, Universidad Complutense de Madrid

October 2019

DESPACHO RT - 31 /2019

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ACKNOWLEDGEMENTS

I would first like to thank the professors Paulo Novais and Juan Pavón from Escola de Engenharia Informática, Universidade do Minho and Facultad de Informática, Universidad Complutense de Madrid, respectively, for being my tutors. Also, I am truly grateful for the help of the professor Leandro Freitas, Centro Algoritmi, Escola de Engenharia Informática, Universidade do Minho and Marlon Cárdenas, Facultad de Informática, Universidad Complutense de Madrid, for the huge help provided throughout the development of this work. Besides them, I want to address my special thank you for all the people who were willing to perform physical activities so I could build a data set in order to train the system.

Finally, I must express my huge gratitude to my family and friends for all the support that was given during not only the development of this thesis but also throughout the past 5 years. This accomplishment would not have been possible without them.

ABSTRACT

This thesis has the aim to correlate context awareness and physical workout. Nowadays the body's health is a true concern to most of the people and working out is one of the ways to go in order to improve health in general. Although, the lack of knowledge can lead to unwanted endings such as contracting an injury. The aim of this work is to develop a *Context-Aware (CA)* architecture that intends to help the user with the workout according to the context that person is inserted in. Accelerometer data, air temperature and humidity are collected in order to infer the context. This work compares three different algorithms, *Support Vector Machine (SVM)*, *k-Nearest Neighbor (k-NN)* and *Random Forest (RF)* for the recognition of the activity. From those three, SVM was the algorithm that presented a better performance overall, since it presented better accuracy, precision and recall. Therefore, it is used to recognise activities and give the suggestions about the way that activity is being executed.

The system was validated and tested getting an overall accuracy of 96.4% with the SVM algorithm. The activities *Standing Still*, *Walking* and *Running* presented a precision of about 99%. While the *Squat*, got a precision of about 74% and the *Wrong Squat* a precision of 61%. Moreover, since data for the activity *Running* was collected through a simulation, a test using real life data for *Running* was made in order to validate the training with simulation generated data. A precision of about 94% was achieved with the SVM algorithm.

Keywords: Context Awareness, Physical Activity, Machine Learning, Human Activity Recognition

RESUMO

Esta tese tem o objetivo de relacionar context awareness e exercício físico. Hoje em dia a saúde representa uma verdadeira preocupação para a maior parte das pessoas, e o exercício físico é o caminho a seguir quando o indivíduo quer melhorar a sua saúde no geral. Contudo, quando o conhecimento não é suficiente, a prática de exercício pode facilmente causar algum dano irreversível no nosso corpo. Como tal o objetivo é criar uma arquitetura CA que ajude o utilizador com o seu plano de exercício físico de acordo com o contexto que o rodeia. Os dados recolhidos para inferir o contexto do utilizador são acelerometro, temperatura ambiente e humidade relativa. Três algoritmos são comparados neste trabalho, SVM, k-NN e RF e o algoritmo que apresentou melhor performance, SVM, foi o escolhido para o sistema final. Uma vez que apresentou melhor accuracy, precision e recall. Este algoritmo é utilizado para reconhecer atividades e dar sugestões ao utilizador baseado na forma como essas atividades estão a ser feitas.

O sistema desenvolvido obteve uma eficácia geral de 96.4% com o algoritmo SVM. As atividades *Parado*, *Andar* e *Correr* obtiveram uma precisão de cerca de 99%. Enquanto que o *Agachamento* obteve uma precisão de cerca de 74% e o *Agachamento Errado* uma precisão de cerca de 61%. Para além disso, um teste, com dados reais da atividade *Correr*, foi efetuado com o objetivo de validar o sistema, uma vez que este foi treinado, para esta atividade em específico, com dados gerados por uma simulação. Uma precisão de cerca de 94% foi atingida com o algoritmo SVM.

Palavras Chave: Context Awareness, Physical Activity, Machine Learning, Human Activity Recognition

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

C

CA Context-Aware.

CNDS Chronic Noncommunicable Diseases.

K

K-NN k-Nearest Neighbor.

P

PCC People Centric Computing.

R

RF Random Forest.

S

SVM Support Vector Machine.

U

UI User Interface.

1

INTRODUCTION

This document presents a Master Thesis in Informatics focused on context awareness and its influence on workouts. It was accomplished at Universidade do Minho, Portugal.

The world is constantly developing and health related solutions is not an exception. Throughout the past decades, it was possible to witness an enormous evolution regarding human health and methods to improve it. However, some diseases have increased in an alarming rhythm during the past few years. Those are the *Chronic Noncommunicable Diseases (CNDs)*, and physical inactivity is one of the main factors that lead to such increasing statistics Beaglehole and Yach (2003). The group of CNDs includes diseases such as cardiovascular diseases, cancers, chronic respiratory diseases and diabetes. They tend to be of long duration and they are a result of a combination of genetic, physiological, environmental and behaviours factors [1] and according to Ermes et al. (2008) *"CNDs cause 60% of global deaths and the figure is expected to rise to 73% by 2020"*.

Therefore it is fundamental to find a solution in order to improve these statistics. Physical activity plays a huge role on preventing this kind of diseases. For instance, it is known that physical inactivity increases the risk of cardiovascular diseases once it is related to factors such as obesity and high blood pressure Rothenbacher et al. (2003).

However it is not always easy for a common individual to know what or how to do it and this lack of knowledge can cause irreparable damages to the human body. Considering that, there is a need to encourage physical activity and that is what this work has the aim to do. The purpose of this thesis is to develop a CA architecture, that gives suggestions to the user according to the context, so a more effective and healthier workout becomes possible for the common user. This architecture intends to be an aggregation to a professional personal trainer or help the user in case of the non existence of one for the most various reasons such as money, time or any other reason.

A system based on this architecture will take in consideration the context of the user, for example, temperature, humidity and other parameters which are further detailed. Those parameters are collected through several sensors, and they are used to infer the context of the user. However, to accomplish this it is needed to recognise the activity being performed

1 *http://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases*

by them. This stage includes three main steps which are distribution of sensors, data pre-processing and data classification Attal et al. (2015).

Regarding the sensors' placement a study was performed in order to define how the sensors should be placed. Considering the complexity of the subject, first of all there was a need to understand the places where more accurate results can be collected. Besides that, those devices need to be inobtrusive, the sensors are supposed to get data in order to help the user and not stop them from doing their normal movements while performing an activity as mentioned in D. Lara and Labrador (2013) *"To be successful in practice, HAR (Human Activity Recognition) systems should not require the user to wear many sensors nor interact too often with the application."*. Even though this sentence is a bit outdated, once nowadays devices such as smartphones or smart watches carry several sensors inside it, it can still be applied since using more sensors and more devices that are spread on the body will increase the accuracy of a context-aware system.

While for data classification there are two main approaches, *supervised* and *unsupervised*. Usually, for physical activity, the most used ones are the supervised approaches which are the most efficient and accurate Attal et al. (2015). More specifically one of the most used techniques is SVM, as can be seen in the studies Pernek et al. (2015) and Ravi et al. (2005). Although, Attal et al. (2015) used four supervised approaches such as k-NN, SVM, Supervised Learning Gaussian Mixture Models (SLGMM) and RF in order to compare them and verified that k-NN was the best among the four. A discussion amongst SVM, k-NN and RF is also done on this document.

A system based on this architecture acts during the workout and there is no difference whether the person is indoor or outdoor. It is what really differentiates this work from others, it tries to interrupt or change a current workout plan based on the user's state. For instance, if during a cardio exercise the environment temperature is too high, a message is displayed to the user suggesting him to slow down. However, the parameters are later discussed on this document. The architecture uses Machine Learning techniques in order to learn what is the correct way of doing an exercise, so a suggestion can be given to the user in case the activity is being wrongly executed, to define what is wrong, some professional personal trainers were consulted as well as what is said in the literature.

This document is organised as follows. Chapter 2 contains the state of the art which discusses about important concepts to this work such as Machine Learning, Context Awareness, People Centric Computing and Wearable Devices. Chapter 3 covers related work. In chapter 4 the research proposal is presented. The main and specific objectives are covered in chapter 5. The chapter 6 covers details about physical activity and the activities recognised by this system. The chapters 7 describes in detail the architecture as well as some screenshots of the application. The chapter 8 covers the methodology. And the document finishes in chapter 11 with the conclusion which contextualises concerning the document as well as future work.

2

STATE OF THE ART

The aim of this chapter is to provide some background and contextualise the reader on the concepts related to this thesis such as Context Awareness, Machine Learning, People Centric Computing and Wearable Devices.

With the increasing number of sensors available and the "omnipresence" of the internet, it is possible to witness the growth of the concept *Smart Environments*. According to Weiser (1991), *Smart Environments* are composed by multiple computers embedded in physical objects. Those computers communicate between each other and the whole system acts according to a specific event. They need to understand what is happening in the environment and take action depending on the situation, it leads to what is called *Context awareness*.

2.1 CONTEXT AWARENESS

Context awareness was first introduced in Schilit and Theimer (1994), it was defined as the ability of a mobile application to collect data and react according to that data. It refers to the identification of the current state of entities and how they influence the system. However, to better comprehend context awareness a notion of what context is is needed. Therefore, this section discusses a definition of context and some other aspects about context awareness.

2.1.1 *Context*

There are several definitions about the term context in the literature. Some consider it as the environment or situation, others consider it to be the user's environment while others consider it as the environment of the application. According to Abowd et al. (1999), context is any information that can be used to describe the current situation of an entity, where an entity can be a person, place, object or anything that can be relevant to the interaction between a user and an interface, it is important to note that it mentions that context consists only of implicit information.

According to Alegre et al. (2016), context can fit in two categories, *operational* and *conceptual*. On one hand the former helps to understand the issues and challenges of data acquisition techniques while on the other hand the latter helps to better understand the relationship between different contexts.

Another way to look at context is by the perspectives of active context, which can be described as *discovered context* and *passive context* where the application presents the context to the user on the precise moment and/or store it for the user to retrieve later as defined in George Wamamu Musumba (2016). However, according to Abowd et al. (1999) CA applications take in consideration the *who's*, *where's*, *when's* and *what's* of entities and use that information to determine why the situation is occurring. But they are not enough and there are certain types of context more important than others such as **location**, **identity**, **activity** and **time**. With the former ones (who, where, when and what) the application can know information about location and identity. However, in order to characterise a situation activity and time information are needed. These context types not only answer the questions of who, what, when, and where, but also act as indices into other sources of contextual information.

Considering that, a CA application needs to use the context it is inserted in in order to provide information or automatically execute a service. For instance, playing a song when the user arrives at a certain place or as presented in the case study Oliveira Freitas (2018), when it is time to take medicine, a message is displayed on his smart phone, for instance. According to Schilit and Theimer (1994), a CA application adapts itself to the context. Another interesting and more general definition given at Abowd et al. (1999) says that a CA application uses context to not only adapt its behaviour according to it but it also uses context to display relevant information. For example, if an application that only shows the temperature of a room is considered, it is not modifying its own behaviour but it still is a CA application. Although, the definition given by Schilit and Theimer (1994) gives the idea that every CA application is reactive such as an application that turns on the air conditioner when the room is too hot.

This shows how CA computing displays a huge role on a daily basis and it can be applied literally to anything that comes to one's mind.

2.1.2 *Context-aware computing*

According to George Wamamu Musumba (2016), CA computing is a mobile computing paradigm in which applications can discover and take advantage of contextual information from the user such as his location, time of the day, devices which are connected to the user's device and so on.

The context categories presented in George Wamamu Musumba (2016), *active* and *passive context*, leads CA computing to be defined in two perspectives:

- Active context awareness - Which automatically adapts to discovered context by changing the application's behaviour;
- Passive context awareness - Which presents the new or updated context to an interested user or saves the context to make it later available as information for the user.

2.2 PEOPLE CENTRIC COMPUTING

CA computing has its foundation on a concept called *People Centric Computing (PCC)* once it relies on the behaviour of an user to take action. According to Delmastro et al. (2016), in PCC, a user not only represents the final user of the application but they also contribute to its behaviour either by acting as humans by sharing contents on social media, for example, or acting as virtual sensors while walking around with their own smart phones full of sensors.

There are 3 paradigms in PCC that are used to design efficient and personalised mobile applications in several domains (health, urban monitoring, etc). Those paradigms are *participatory sensing* Burke et al. (2006) which refers to where the users explore their sensing devices and share the information with other users through the web; *opportunistic sensing* Conti and Kumar (2010) in which a mobile application exploits all the sensing technologies in the environment and; the last paradigm is *opportunistic mobile social networks* D. Lane (2008) in which users directly generate and share all types of contents with nearby users in real time. The paradigm that was exploited on this work was the *opportunistic sensing*, since the developed application only takes advantage of the sensors devices and there is no shared data.

2.3 MACHINE LEARNING

Over the past decade it was possible to witness a huge development in what concerns Machine Learning, even if unperceived, it has become an important part of human's life. It is being gradually implemented in every aspect of people's daily routine, from the keyboard on the smartphones to autonomous driving.

It can be defined as a set of methods that can automatically detect patterns in data and then use the uncovered patterns to make predictions about the near future data or to perform other kinds of decision making under uncertainty Smola and Vishwanathan (2008). Thus, based on these patterns the system would be able to improve autonomous decision making. For example, when the user has 6 hours of sleep and he can still perform a determined task, that task will start to be suggested to him more often. The system should contain a knowledge base with the data required to properly suggest changes to the current physical activity. That data will be processed and therefore interpreted in order to better adapt to the one using it.

Regarding the learning problems the system faces, it is important to characterise them according to the type of data they use Smola and Vishwanathan (2008), once similar problems can be solved using similar techniques and therefore ending up saving a lot of time. One example given at Smola and Vishwanathan (2008) is that *"natural language processing and bioinformatics use very similar tools for strings of natural language text and for DNA sequences"*. To solve those problems, a machine learning system uses a wide variety of algorithms. There are two main approaches when it comes to algorithms, *supervised* and *unsupervised* D. Lara and Labrador (2013), and the vast majority of human activity recognition applications uses the *supervised* one. The former deals with labeled data, in other words, all the data that is used has an identification, for example, if a person is running, the data which is passed to the model should contain that information. While *unsupervised* deals with unlabeled data and since a human activity recognition application needs to return a label it is not widely used.

There are a lot of algorithms which belong to each of these categories, the most used ones in human activity recognition according to D. Lara and Labrador (2013) are decision trees, k-NN, SVM and Naive Bayes. It is possible to reduce these set of algorithms only to SVM and k-NN, according to Pernek et al. (2015), Ravi et al. (2005) SVM is the most used one and according to Attal et al. (2015) k-NN is the one which is used the most. Regarding the accuracy of those algorithms in Attal et al. (2015) the authors made a comparison between several algorithms, one of them was SVM and the other one k-NN, it concluded that k-NN was more efficient for the specific purpose of recognising physical activity. Also in Ravi et al. (2005) several algorithms are compared, in this work there were four considered settings. There are two of those four which are particularly relevant for this work which are setting 3, *"Data collected for a single subject on one day used as training data, and data collected for the same subject on another day used as testing data"* and setting 4 *"Data collected for a subject for one day used as training data, and data collected on another subject on another day used as testing data."*. While on the setting 3 and 4, the SVM, got an accuracy of 68.78% and 63%, respectively, the k-NN algorithm got 72.93% and 49.67%, respectively. An algorithm which stood out from the others on the setting 3 was the Naive Bayes with 89.96%.

2.4 WEARABLE DEVICES

Due to the huge development of wearable devices, this platform uses them as one of the ways to get data and therefore process it. These kind of devices allows to sense data unobtrusively, in other words, it is possible to get data without disturbing the user's actions. The user can wear it for instance in his shoes, as a wristband, on the jersey, etc, and that is a big advantage over other kinds of sensors such as cameras. The main goal of this devices is to gather data in order to process it and use it as context for the architecture.

There have been several studies regarding them, the work El-Amrawy (2015) evaluates the accuracy of devices used to count steps or measure the heart rate and concludes whether their application on health or sports is worth it or no. Devices such as Apple Watch, Samsung Gear Fit, Samsung Gear 1, Samsung Gear 2, Samsung Gear S, Mi Band and many more were used in it, having in consideration factors such as price, popularity and consumer surveys it was concluded that overall this kind of devices are quite reliable and are a good way to encourage physical exercise.

3

RELATED WORK

Currently there are a lot of studies going on regarding context awareness and how it can be implemented on the daily routine. For instance, wearable devices with sensors are able to assist users considering the context they are inserted in Abowd et al. (1997). The paradigm of computation is changing, the days where a computer needed to be used in a desk are long gone now, nowadays it is even possible to wear the computer.

Considering that fact, it is natural that many technologies on this field are rising. One of the first projects which tries to motivate an user simulating the job of a personal trainer is the Philips Virtual Coach IJsselsteijn et al. (2004). It is meant to be used at a pre established place in static bike. While the user is using it, a 2D animated personal trainer is projected on a screen as well as a simulation of an outside environment. However, after a study on 24 users was conducted , it was concluded that the presence of the animated personal trainer was not as effective as it was expected before hand. User's would not train harder with the presence of it and it is thought that one of the reasons was the fact that it would only show the user's heart rate rather than other information that might have a bigger impact on the motivation such as the burnt calories. However, the used approach on this work, obligates the user to stick in a specific place. Which might not be appropriate for everybody due to several factors such as the needed space to put the bike at home or the lack of freedom to choose the place to practice the activity.

Also the work D. Lara and Labrador (2013) surveys the state of the art on Human Activity Recognition. It researched aspects such as activities that can be recognised with state of the art technology, the most used algorithms for the effect, problems that appear on the development of such application. It also presents a typical architecture of a human activity recognition system that partly inspired the creation of the architecture used on this work as well.

Another tool that is worth mentioning, even considering its differences to this study is SensVest Knight et al. (2005). It is a wearable device developed to be used on outdoor activities. It proposes to measure, record and transmit physical activity performance metrics, such as heart rate, temperature and movement. It proposes to quantify the quantity of energy that was spent while executing a specific task as well as the force applied. Following there are two sections describing two applications which have the most similarities to this work.

3.1 PHAROS PHYSICAL ASSISTANT ROBOT SYSTEM

PHAROS Angelo Costa and Julian (2018) is an interactive robot that intends to help elderly people, i.e., 60 years old and above, in their daily physical activities. There are some requirements that need to be met such as being "...friendly, intuitive and proactively assistive. In addition, it should be accepted by the elderly..." and those requirements were fundamental when choosing the robot itself once elderly people usually tend to be quite sceptical towards technology. That's why the robot is a human-shaped robot "with high levels of acceptance by the elderly", named "Pepper robot".

It starts with an interaction with the user in order for him/her to be identified using a camera. Once identified, the robot tries to find the most suitable exercises for that user according to his/her physical limitations. After that a daily physical series is scheduled so the system can continue to adapt itself to the user's needs.

Its architecture, figure 1, is divided in two modules, the "Human Exercise Recognition" and the "Recommender". The "Recommender" recommends a determined physical exercise which the user enjoys and is capable of doing, at a scheduled time and the "Human Exercise Recognition" verifies if the exercise is being executed in a correct form.

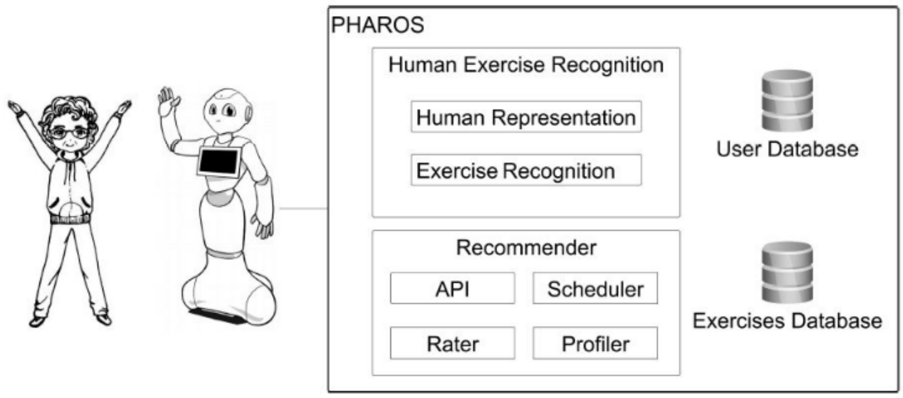

Figure 1: Pharo's architecture

The recommendations are based on a rating system given accordingly to the user's execution of an exercise so the architecture perceives his/her evolution on a determined exercise. A lower rating might mean some kind of problem with her/his health once the exercise was not executed as well as previews times. This rating system was built upon Glicko2 which is a rating system usually used to measure the ability of a player in an ability game such as chess or Go.

Another part of this relies on the recognition of the exercise being executed. It uses a computer vision approach in order to achieve the recognition. Deep learning techniques are used Recurrent Neural Networks and Convolutional Neural Network more specifically.

However, this work uses a different approach for the activity recognition. It uses cameras, which limits its use to a pre determined space. Whereas, the sensors approach allow the user to use it everywhere.

3.2 MOPET: A CONTEXT-AWARE AND USER-ADAPTIVE WEARABLE SYSTEM FOR FITNESS TRAINING

MOPET Buttussi and Chittaro (2008), is a mobile personal trainer with the goal to train and motivate the user. It was designed for running and walking and it should be used anywhere a user can do those activities if that place is outdoors. A heart rate sensor with accelerometer and a PDA with a GPS are used on the chest and wrist respectively.

Figure 2: MOPET 3D Animation

It is mainly composed of two parts. The training part and the motivation part. Regarding the training part of the application, it shows a 3D animation on the PDA of the activity, figure 2. It is important to help the user understand how an activity should be correctly performed and avoid injuries. For the motivation part, its strategy is to use audio and video. It also separates speed in four groups, slow walking, fast walking, moderate running and fast running. The system uses those four groups to motivate the user by audio. For instance, if the user is moderately running, the system will tell the person that the current speed is moderate and if not tired, an increase on the speed should be tried. When giving this audio feedback, a positive approach is preferred over an aggressive one. The best parts of the current situation are highlighted, because according some prior studies, the feedback on more aggressive approaches was negative. A big proportion of this work has some similarities to the work proposed on this document. For example, the live recommendations when the user is performing an activity. However, it lacks the suggestion for bad executed physical activities.

The aforementioned works have some things in common to the documented application. For instance, they all have the goal to encourage and monitorise physical activity, they also give real time feedback during the execution of the activity in order to motivate and/or correct the subject. However, this work have some differentiators, for example, the first presented work captures the movement with a camera rather than sensors. And the second one, focus on giving real time feedback in order to motivate the user but it does not provide feedback about how the current context can affect performance, for example.

4

RESEARCH PROPOSAL

This study aims to propose a solution to identify when an activity is not being well executed, for now it will only focus on the activity *Squat*. It is done by capturing data for what is considered a well executed activity and a bad executed one. The validation of the system is done by using three popular metrics, accuracy, precision and recall. This metrics are calculated using the values of the confusion matrix of each algorithm, which contain values such as true positives, true negatives, false positives and false negatives. After the calculation of these metrics, a comparison is made between three machine learning algorithms SVM, k-NN and RF. Besides that, two validations for the activity *Running* are made, since the system is trained with data generated by a simulation there is a need to validate it both with data also generated by the simulation and data generated in real life by collecting that data from the sensors.

Apart from identifying a well executed or not well executed exercise, it is intended that the system gives the user advises or warnings taking in consideration the context. For instance, tell to the user to lower the pace of a run if the temperature is above a certain threshold.

5

OBJECTIVES

5.1 MAIN OBJECTIVES

The main goal of this Master's thesis is to develop a CA architecture and compare the best machine learning techniques for activity recognition. This architecture should give its users a suggestion while executing a physical activity that should help on preventing injuries and increase performance. Those suggestions vary according to the activity that is being performed. For some activities, only tips regarding the current context will be given. While for others, a suggestion regarding the actual activity will be made. These suggestions are based on data collected from sensors such as accelerometer data, environment temperature and relative humidity in air. The system, will detect the movement that is being performed and provide a feedback to the user. For example, if the user makes wrong movements while performing an exercise, a warning will be given to that person so the activity is well executed and injuries are less likely to happen. Also, suggestions based on the environment temperature and relative humidity will be made. For example, *running* under a 35° C temperature will trigger a warning, advising the user to stop running as it can be dangerous to the body.

The physical activities that are recognised by the system are *Standing Still*, *Walking*, *Running* and *Squat*.

5.2 SPECIFIC OBJECTIVES

To reach the main objective, the following specific objectives should be achieved:

- To define which data should be collected from sensors to be used as parameters, in order to ensure the quality of the suggestions;
- To define what suggestion should be made based on the data;
- To define the necessary architecture components, given the initial research;
- To define the most appropriate machine learning techniques that fit this architecture;

- To test and validate the final architecture.

6

METHODOLOGY

Aiming to achieve the objectives described in chapter 5 it is intended to perform a qualitative research through an investigation and understanding of the object of study (context awareness and machine learning in workout plans).

According to Alegre et al. (2016) there is no specific methodology regarding the development of a CA system. Once context is a very complex concept and as already mentioned previously, literature does not have a straight forward definition to it. However the same study outlined a typical life cycle for context information before being delivered to the CA system Freitas (2018).

- *Acquisition* - Information need to be gathered. Usually it comes from sensors but it can be any kind of source. In the proposed architecture in the current document data is taken mainly from wearable sensors. Three sensors were placed in total, two on the user's wrists and one on the chest. The data gathered from the sensors was accelerometer data which is used to feed the model and then recognise the performed activity and temperature and humidity data which were used in order to warn the user whether it is a good condition or not to perform such activity;
- *Modelling* - After being sensed, data needs to be translated from real world concepts into modelling constructs;
- *Reasoning* - According to Perera et al. (2014) this stage typically has three steps. The first one which is *Context pre-processing* has the aim to clean the sensor data since data may not be accurate or even missing. Consequently missing values need to be filled and outliers, the values which are outside the range of what is expected, removed. The second step is *Sensor data fusion* which is the process of combining data from several sources in order to get more accurate data. And last but not least there is *Context inference* which is defined by Perera et al. (2014) as "Generation of high-level context information using lower-level context. The inferencing can be done in a single interaction or in multiple interactions.";

- *Dissemination* - In which both high-level and low-level context need to be distributed to the consumer. The context must have high-availability in order to be used.

Following this steps it is believed that it is now possible to develop an architecture specifically for CA systems and use data in a more dynamic way.

Therefore, this steps were taken in order to develop this system. It first started by collecting data. The data went through a process of *fusion*, from the three different sensors. Then, that data was pre-processed, *null* values were replaced and data whose label did not exist deleted. The process of collecting data is highly detailed on the next section.

This thesis, was developed in two different universities, Universad Complutense de Madrid and Universidade do Minho. The practical part, such as data collection and the code development was accomplished in Madrid. While the written part was left for Braga.

6.1 DATA SET

Data was collected from 9 subjects with ages ranging between 12 and 46 years old, averaging 26.8 years old. Both male and female participated on it, more details about the subjects can be seen in table 1. People with different backgrounds on physical exercise were also considered. The reason why people with such a wide variety of ages and different backgrounds on exercise were chosen was to reduce the *Flexibility* problem as much as possible.

Subject	Genre	Age	Experience
1	M	23	Yes
2	M	22	No
3	F	42	No
4	M	46	No
5	M	12	No
6	M	23	Yes
7	F	24	Yes
8	M	27	Yes
9	F	23	Yes

Table 1: Subjects who participated

The sensor, figure 3, produced about 11 or 12 registers per second, which means it had a frequency around 11 Hz.

There are many types of physical activities that a system can recognise. However, it can become quite overwhelming to build a system that covers many of those activities. Moreover, it is a process that takes time, it is needed to find subjects who agree on participating on data collection, collect the data for each activity and train the model. Therefore, it was decided that the set of activities recognised by the system on an initial version would be the following ones.

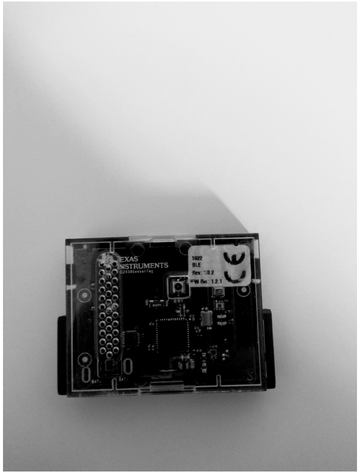

Figure 3: Sensor

- **Standing Still** - This activity is the base activity of the system, when the user is not executing any activity and standing still, the system shows the temperature, humidity and the corresponding image for the activity still. The process to obtain this data was simple. The subjects were asked to stand still figure 4 for 5 minutes while the script would collect data and print it to a file;
- **Squat** - One of the main activities which are recognised by the system. Initially, the capture of this activity lasted 7 seconds. However, as the *squat* has a duration of about 2 or 3 seconds, more than half of the data for that activity would be standing still. Then, it would create a confusion with the actual activity standing still. Therefore, the time of capture was changed to 4 seconds. The subjects were asked to start as indicated in the figure 4 and to do the movement of a proper squat as indicated in the figure 16. After going down, the subject should return to the start position and wait until the scripts stops running. This activity separates in two, the right way and the wrong way, details on how a *squat* was defined as right or wrong will be further discussed;
- **Walking** - This is yet another activity that is recognised by the system. The subjects were asked to walk for 10 minutes on a thread mil. The thread mil was necessary part since the sensors were being paired to a computer. So the possibility of letting the computer in a place was a proper fit for this activity;
- **Running** - Regarding running, the data for this activity was generated using the mentioned simulation tool developed by the team *GRASA* from Universidad Complutense

Figure 4: Initial Position

de Madrid. Running can be quite demanding physically therefore not every individual who participated in this study would be prepared to generate enough data for the system, that is why this was the chosen activity whose data would be generated by the tool. Figure 5 describes the virtual scenario.

6.1.1 *Virtual Living Labs*

Ambient assisted living is a branch of the ambient intelligence and it has the goal to make elderly or disabled people independent for as long as possible by using ambient intelligence techniques [1].

The ability to simulate an environment and a physical activity significantly eases the process of collecting data. There were some problems which led to the use of the *SociAAL* tool developed by the *GRASIA* team from Universidad Complutense de Madrid [2].

First of all, data can be generated whenever it is necessary. This way, the process of collecting data does not depend on the subject's schedule or other problems that can occur. Such as the sensor running out of battery or some injury that might prevent the user to perform an exercise.

The second problem was the subject's fitness. *Running* is a quite demanding physical activity. A person needs to be in shape to be able to run for a while. And, as this data was used to train a machine learning architecture, a significant amount of data is needed. So, it would take a long time to capture *Running* accelerometer data. Figure 5 represents the virtual scenario that was created for the activity *Running*.

1 *https : //www.igi − global.com/dictionary/privacy − data − protection − towards − elderly/*33084
2 http://grasia.fdi.ucm.es/sociaal/

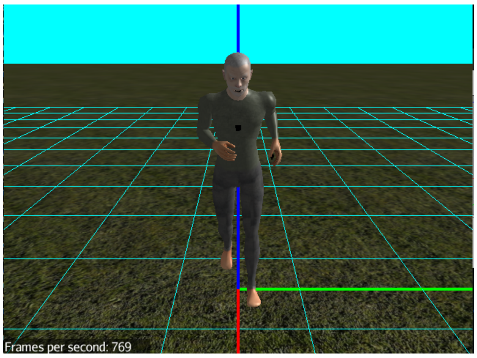

Figure 5: Virtual Scenario

For the collection of the data from the simulation. There is one thread per sensor. The data is printed in a file and then joined. It follows the same principle of the collection of data from the real sensors.

6.2 SQUAT CORRECTNESS

In order to determine whether an exercise is being correctly executed or not, it is needed to define what is wrong and right so the system can advise the user. As previously mentioned, in this work *squat* is one of the chosen activities and it is the activity which is evaluated whether the individual is executing it right or not. Besides reading the literature to define how the *Squat* is wrong, also two personal trainers gave their opinion on this matter. Following it is explained what is considered a wrong *Squat*.

Squat is one of the most important exercises. According to Escamilla (2001), there are three main variations of a *squat*, *bodyweight squat*, *barbell squat*, and *machine squat*. This work focus on the *bodyweight squat*. Due to the importance of it, it is a part of many workout programs for athletes. The hip, thigh and back become stronger if *squat* is a regular exercise on an athlete's workout program Escamilla (2001). By consequence, this will naturally protect athletes from injuries. However, if it is not well executed, especially without the follow up of a personal trainer, the individual is very prone to contract an injury. Especially on the knees which are a really sensitive part of the human body. The stability of the knee is key to perform a perfect stable *squat* and avoid injuries. The article Myer et al. (2014), analyses the proper way of performing a *squat* and improve the efficiency of it. According to it, when performing a *squat*, the feet should be flat on the ground and when going down, the thigh should be parallel to the floor. The subject's back should also be as straight as possible as shown in figure 8. Another important factor are the knees, they should follow the line of the feet as can be seen in figure 16 and should not be knuckled in as can be seen in figure 6. Besides that, the knees should not pass the line of the toes 8.

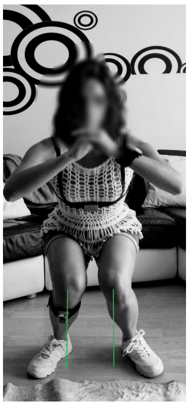

Figure 6: Wrong squat

Considering all of this, two forms of the *squat* were recorded with the sensors to train the model. The well executed *squat*, figure 16 and the wrong way figure 6. Given that this is the body weight *squat*, the back positioning is not as important as in back *squats*, where the subject carries a bar with weights and naturally the person is way more prone to contract a back injury compared to the normal *bodyweight squat*.

Figure 7: Squat by front

Figure 8: Squat by side

7

PHYSICAL ACTIVITY

In this chapter, aspects related to human activity recognition are described in details. Characteristics of sensors are presented, as well as other parameters like temperature and relative humidity, which are collected by the architecture and how it influences physical exercise.

In order to make a new recommendation the developed architecture needs to recognise the activity being executed by the user. One of the ways to do that is by using wearable devices. The work Attal et al. (2015) classifies the physical activity of elderly while inertial wearable sensors are used by them. Besides the fact that it only classifies every day activities such as standing, stair descent, sitting, sitting down, sitting on the ground, lying, walking, stair ascent, standing up and so on. It is still relevant to learn how to classify more complex physical activities such as physical activities, focusing on specific muscles.

In the literature there are two approaches to recognise activities using *external* and *wearable* sensors D. Lara and Labrador (2013). The former is characterised by devices which are fixed in predefined points of interest. For instance, cameras installed in specific places Chaquet et al. (2013), Angelo Costa and Julian (2018) which confines the subject to that area. The latter uses sensors which are attached to the user D. Lara and Labrador (2013), 157 (2006), Chernbumroong et al. (2011). However, there are some problems with the camera approach according to D. Lara and Labrador (2013). *Privacy* is one of them, as the subject might not agree to be recorded all the time while that subject is performing an activity. The other one is *pervasiveness* because the subject needs to stick around the boundaries defined by the camera's scope and it is not either comfortable nor convenient. There is also the *complexity*, since image contains way more information than text, it can cause processing problems. However, it can not be really considered as a problem since in a CA system, the more information there is the better.

By using *wearable sensors* the only problem which can persist, from the aforementioned problems, depending on the type of sensor, is *privacy*. However, a new problem is carried by them. *Obtrusiveness*, in other words, the application should contain the least amount of sensors possible and with it comes the decision of where to place them.

The placement of the wearable sensors is something that should be carefully considered. According to Attal et al. (2015) they are usually placed on the sternum, lower back and waist. Even though these placements are mainly used to recognise daily activities as mentioned above, they can also be used to detect running which is useful for this architecture. Also, according to Attal et al. (2015), there was a study comparing accelerometer based multi-sensor versus single-sensors in activity recognition. Its conclusion is that the former gives the highest recognition rate. In what concerns the classification itself, classification techniques such as k-NN, SVM and RF are used, which will be discussed more in depth later in this document.

Not only it is necessary to detect daily activities, it is also necessary to detect when an individual is doing physical exercise. For instance, a person with certain diseases needs to follow a protocol of exercises. By detecting those activities it is possible to provide feedback to the caregiver or to the person who is executing it. There are also the cases where the user is monitored in order to track the evolution of a disease and keep the person safe from risks. For example, the work Capecci et al. (2016) has the goal to "detect and quantify the freezing of gating (FOG)", FOG is a motor symptom that affects people who suffer with this disease, in patients with Parkinson's disease. It uses both video and accelerometer data for the detection. In the case of this work it is fundamental to detect the movements and give the user feedback about the correctness of the exercise. For instance, if the person is performing a *squat* and puts the knees in a wrong position a warn should be given in order to avoid future injuries.

In Pernek et al. (2015), the authors propose a solution to recognise the exercise itself and to predict other exercise correctness metric which is intensity with a network of wearable accelerometers with different groups of acceleration features. It uses 5 sensors, one on the chest, two on both left and right wrist and two on left and right upper arm. This document will further get more in detail about it. In Karaman et al. (2010), wearable cameras are used to keep track of patients with dementia. However, it has its differences when compared to this work since it has the goal to detect daily activities and not sports specifically.

According to D. Lara and Labrador (2013), using the current state-of-art on human activity recognition it is already possible to detect sports such as running, rowing, lifting weights, spinning, nordic walking, push ups and cycling.

7.1 DATA COLLECTION

The data set to train the architecture was obtained using two different approaches. The first one was using sensors and the second one using the tool *SociAAL* which is a simulation tool developed by the lab *Grasia* from Universidad Complutense de Madrid [1].

1 http://grasia.fdi.ucm.es/sociaal/

7.1.1 *Sensors*

There are mainly two factors which should be taken into consideration when choosing the sensors for a Human Activity Recognition system: *selection of attributes* and *obtrusiveness*. The former, resides on choosing the right attributes which will dictate the context of the user. While the latter, addresses the problem that comes with using too many sensors.

Regarding the *obtrusiveness*, the application needs to use the least amount of sensors possible so it does not become an obstacle to the user rather than aggregating. Moreover, a high number of sensors will also be costly. Therefore, it is needed to find the smaller number of sensors and still be able to keep a relatively high accuracy.

However, using just one sensor might be inefficient and, as pointed in Bulling et al. (2014), Maurer et al. (2006) the combination of sensors allows to improve the recognition accuracy when compared to individual sensors. For example, Bao and Intille (2004) uses 5 biaxial accelerometers, one on the right hip, one on the dominant wrist, one on the non-dominant upper arm, one the dominant ankle and one on the non-dominant thigh. Trabelsi et al. (2013) uses 3 sensors, placed at the chest, the right thigh and the left ankle of the subject. However it may become quite uncomfortable to the user giving the high number of sensors. There are also studies who recognise activities just using a smartphone Wannenburg and Malekian (2017), Capecci et al. (2016).

Considering this, another important factor that should be considered is the place where the sensor should be. According to Chernbumroong et al. (2011), the recognition of the activity achieved 94.13% accuracy just using one sensor on the wrist. Considering this work and many others which use sensors on the wrist Maurer et al. (2006), Wang et al. (2012), on the chest Trabelsi et al. (2013) and knee Huynh and Schiele (2006) and given its accuracy it was decided that this work would use three sensors one on the left hand, one on the chest and one on the right knee as can be seen in figure 9.

Moving on to the *selection of attributes*, the attributes which are measured need to be chosen. According to D. Lara and Labrador (2013) there are four groups of attributes measured by sensors in a Human Activity Recognition application and they are *environmental*, *acceleration*, *location* and *physiological signals*:

- ***Environmental*** - This group contains attributes such as temperature, humidity, light intensity, audio level, etc. and they are used to provide contextual information about the place the user is currently in. The combination of those attributes can be used on an application to predict the place the user is in. For example, if the audio level is relatively high and the light intensity is low it can indicate that the user is probably in a cinema or a club;

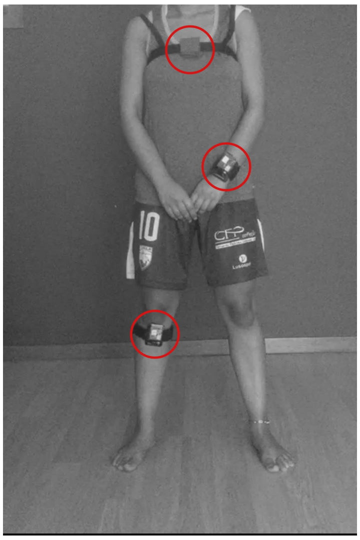

Figure 9: Sensor's placement

- ***Acceleration*** - According to 157 (2006) accelerometers are one of the most used sensors for activity recognition. They are inexpensive and can be found relatively easy as they are embedded on our phones. Besides that, they present high recognition accuracy;

- ***Location*** - A location system such as the GPS is also a big part of our every day lives, as the Acceleration group we can also find it on our pockets as they are present on most of the phones sold nowadays. It can be used to infer some activities as well, for example, if an user is at the beach it is likely that the person is not brushing their teeth. However, there are some downsides, according to Reddy et al. (2010) they consume a relatively high amount of energy especially when it comes to real time tracking applications. Also, it does not work when the user is indoor, it is only useful on outdoor environments and it has some issues concerning the user's privacy as they might not be willing to share their location every time;

- ***Physiological signals*** - This group includes all of the attributes which are related with the body. For example, heart rate, respiration date, etc. however the work Tapia et al. (2007) presents an activity recognition system which combines accelerometer data and heart rate monitor. It was shown that the heart rate is not useful for the recognition after an intense activity such as running because once an user stops running he can sit down but the heart rate will still be really high.

Considering not only the advantages and disadvantages to each one of the mentioned group of attributes. But also the sensors that were available. The attributes which are collected are *acceleration*, *temperature* and *humidity*. *Acceleration*, allows the application to detect the movements of the user while performing a physical activity. While *temperature* and *humidity* have a great impact on an individual during a physical activity. This impact will be later discussed on this document.

The sensors which are used in this work to collect data are *SensorTag CC2650STK* with a sampling rate of 12 Hz. The sampling rate is the average number of entries produced in one second.

7.1.2 *Temperature and Humidity*

Temperature and relative humidity are the two other parameters which were chosen besides accelerometer data to be collected by the architecture. Human body can be hugely altered by the environment conditions, therefore these two factors need to be taken in consideration since they are two of the most important parameters related to the impact environment has on the body El Helou et al. (2012). High environmental temperatures and a high humidity percentage provoke a significant level of stress on athletes who train and compete under such conditions Nybo (2010). First of all, there is a concept that should be understood before getting in depth so further explanation can be better assimilated. VO_2max is one way to measure the physical condition of an individual, it represents the maximum value of oxygen that the body consumes during physical exercise and the higher the better condition the individual has Hardman (1999).

The work Zhao et al. (2013), has three case studies, the first one is made under the conditions of 21°C/20% RH (relative humidity) which is the control case, the second one was 33°C/20% RH, classified as hot-dry and the third one 33°C/80% RH, classified as hot-wet. It concluded that the VO_2max is affected on both hot-dry and hot-wet conditions when compared to the control case. Although, there was no difference between the hot-dry and hot-wet conditions. Being that an indicator that the environmental temperature has a bigger impact on the subject comparing to the relative humidity. Although, not only high temperatures can affect one's performance, the study Galloway and Maughan (1997) reported that the range to exhaustion was between 47.8-136.1 min under the temperature of 4°C which is substantially lower comparing to 11°C whose time to exhaustion was 70.7-121.2 min.

However, one important thing that should be considered regarding temperatures, humidity and its effects on the human body is whether the individual is indoor or outdoor. External factors can help cooling down the human's body, which will reduce the risk of dehydration, such as wind and rain, on contrary to the indoor activities in which rain and wind will not play a role, at least not like ondoor activities. The study Nybo (2010) focused on cycling, however, those results can be extrapolated, at some extent, to other physical exercises such as running. It concluded that on a flat terrain, the body's temperature can be balanced because of the wind opposing to hill terrains if the individual is climbing it. However, it only happens if the relative humidity is not too high which can make the subject become hyperthermic. The effect of hyperthermia, according to the same study, impairs the athlete's performance. Still regarding the matter of indoor vs outdoor, the study Galloway and Maughan (1997) reported a high drop on the performance when the room's temperature was raised from 21°C to 31°C and that the difference on the temperature can lead to severe hyperthermia. It also observed that the longest time to exhaustion occurred under the temperature of 11°C, with 31°C everyone gave up before the 60 mins mark and with the temperatures 4°C and 21°C before the mark of 90 min. As can be seen it is not as straight forward as one might think, all those factors have an impact on the human body. Other variable that should be considered is the hydration, as long as the individual is be able to keep hydrated, the impact of high air temperature is not the same as if there was not any hydration at all. However, this was a bit simplified for the sake of this work, since the used sensors do not allow to detect whether an individual is indoor or outdoor neither if he is getting hydrated or not.

The study El Helou et al. (2012), focused on the impact of environmental parameters on running. It studied six marathons Paris, London, Berlin, Boston, Chicago and New York during 9 years (2001-2010), 4 environmental parameters were studied, temperature, humidity, dew point and atmospheric pressure. However, only temperature and humidity are considered for this work since they are the only parameters out of the 4 which are provided by the sensors. One of its conclusions was that air temperature was the parameter which had the most impact on the individual's performance and humidity the second one. After the analysis of those 60 races, it was concluded that around 10°C the velocity would start to lower gradually, it also says that "The optimal temperatures to run at maximal speed for men and women, varied from 3.8°C to 9.9°C. The American College of Sports Medicine (ACSM) claims that the acceptable maximum temperature is a WBGT of 28°C. However, in 2004 Boston Marathon with a temperature of 22.5°C around 300 emergency calls were made and in 2007 London Marathon with a temperature of 19.1°C, 73 people went to the hospital and one death was registered El Helou et al. (2012). But what does actually happen when the temperature is not in that ideal range, in other words, when it is cold or hot? In Galloway and Maughan (1997), there was an alteration in CHO oxidation and oxygen consumption under the temperature of 4°C, however, it says that there was no conclusion on why it happened being further investigation needed.

The table 2 presents some studies and ideal temperatures for activity as well as some thresholds.

Regarding the relative humidity, there are not many studies which specifically study the impact of the relative humidity alone on the human body. Most of them are like the ones aforementioned which correlate temperature and humidity. However, the work Zhao et al. (2013) reported a difference on the skin temperature in the hot-dry condition (33°C/20%

Study	Ideal temperatures	Thresholds
El Helou et al. (2012)	3.8°C - 9.9°C	15°C [a]
Ely et al. (2007)	10.1°C - 15°C [b]	
Suping et al. (1992)	8°C - 15°C [c] / 12°C - 24°C [d]	20°C [e]
Galloway and Maughan (1997)	11°C	4°C and 21°C
Chmura et al. (2017)	Up to 22°C and humidity below 60% [f]	

a Marathon dropouts begin to raise
b Reports that most world records were beaten in between this range
c Ideal range for marathons/high intensity running
d Ideal range for recreational activities
e Above 20, there is an increasing risk of having a stroke or dehydrate
f Best performance by the athletes

Table 2: Reported temperatures in the literature

RH), where the skin temperature was actually lower after the exercise than before the exercise, whether in the hot-wet condition (33°C/80% RH) it was reported that the skin temperature was higher after than before the exercise. The IFA (International Fitness Association) recommends an ideal range of 40% and 60% [2] and the United States Department of Labor [3] recommends an humidity between the range of 20% and 60% for indoor establishments, i.e., gyms.

Considering all of the mentioned works there were established three limits for temperature and two for relative humidity levels.

- < 5°C - Temperature is too low, warn the user;
- > 20°C (Running) / 24°C (Walking) - Temperature is getting a bit high, might be good to keep hydrated;
- > 31°C (Running) - The temperature is too high, might be a good idea to pause for a bit;
- < 30% humidity and > 60% humidity

These values were chosen considering the information in the literature. As can be seen in the table 2, the lower thresholds vary between 3°C and 10°C, so a more or less intermediate value was chosen. Regarding the intermediate value, 20°C, there seems to be an agreement on this regard. For instance, the studies Galloway and Maughan (1997), Suping et al. (1992) agree that above 20°C or 21°C, the performance starts to be impaired. Regarding the higher threshold, 31°C, the study Galloway and Maughan (1997) states that at 31°C there was a

2 *https : //www.ifafitness.com/health/temperature.htm*
3 *https : //www.osha.gov/dts/osta/otm/otm_iii/otm_iii_2.html5*

significant increasing on the time for exhaustion. Regarding the humidity, 30% and 60% were considered because of the literature already provided. For instance, Zhao et al. (2013) says that a combination that a high temperature, above 31°C and 80% of humidity highly increases the risk of dehydration.

7.2 ISSUES

When designing a human recognition system there are issues which need to be taken in consideration, according to D. Lara and Labrador (2013), such as:

- **Data collection protocol** - The accuracy of a model depends on how the data is collected. Data collected in a controlled environment, such as a laboratory, tends to have a higher accuracy than data collected in a natural environment. In Foerster et al. (1999), posture and motion detection showed a high accuracy for data recorded in a laboratory while for data recorded on outside environments the model had more trouble on identifying them;

- **Flexibility** - One of the most known problems in a human activity recognition is *flexibility* Ponce-Espinosa et al. (2016), there is no consensus whether a model gets restricted to a person when it is trained using data from an individual in particular. Since every human has its own way of doing a certain activity, such as walking, an elderly has a different way of running than a teenager or a young adult. So a solution that might lower this problem is to collect data from people with the most varied characteristics such as different ages, weights, height. Either this or train the model for each subject in particular, however, it would become a rather impossible work if there are many activities recognised by the system and would present a lot of scalability problems;

- **Recognition Performance** - Another issue pointed by D. Lara and Labrador (2013) is the recognition perfomance. It points out that the accuracy of a system depends on the activities recognized by the system, the quality of the dataset used for training, the feature extraction and the algorithm used to classify the activity. Besides that, enough data should be provided for training and testing the model;

- **Uncertainty** - It is a problem that affects every CA system. The data collected by the sensors are sometimes misleading and do not reflect the actual context of the user. Therefore, the data set which is used to train the model should be checked in order to find any irregularity. An usual solution for this problem is user mediation, i.e, provide the user with the information and let it be corrected by them Dey et al. (2002).

In order to avoid this issues some measures were taken. For instance, to avoid the *flexibility* problem, people from both genders, with a wide range of ages and with different fitness background participated in the collection of data. This way, it is expected that the system is more flexible and is be able to recognise the physical activity for many types of people more easily. Moreover, it was tried to vary as much as possible the source of the data. For example, for the activity *walking*, some of the data was collected in a controlled environment, a thread mil, while other data was collected outdoors. Also, the use of a simulation environment for *running* and an usual environment for *squats*. This approach was taken to avoid the first mentioned problem, *Data collection protocol*.

8

ARCHITECTURE

This chapter describes in details the architectures of the proposed system. This includes an overview of it through an workflow, the extraction of features, machine learning algorithms and implementation.

8.1 WORKFLOW

The workflow of this application, figure 10, is based on a client-server architecture. The client gathers the data and sends it to the server, which then responds with the prediction of the activity.

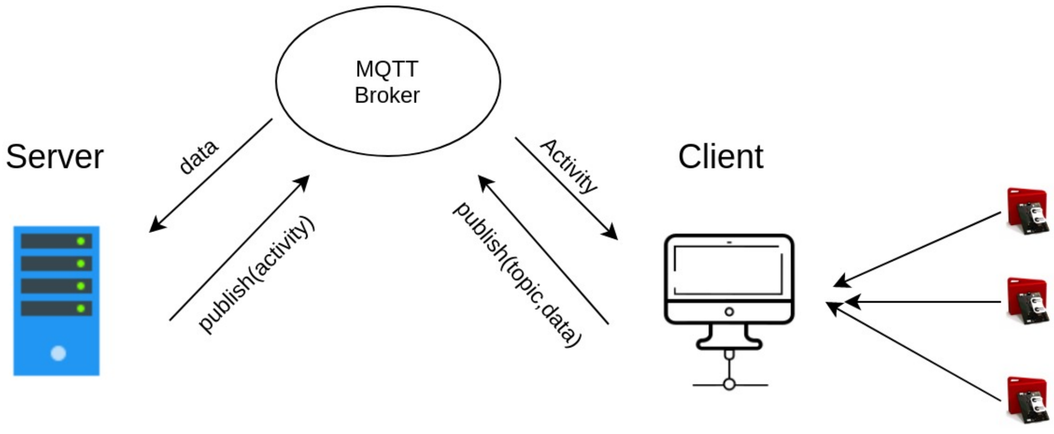

Figure 10: System architecture

Both components communicate using the MQTT protocol. Usually, applications which use micro services that need to comunicate with each other, do it through REST APIs and they are based on HTTP as well. However, when it comes to Internet of Things it certainly is not the most appropriate protocol out there. The work Yokotani and Sasaki (2016), compares the performance of both HTTP and MQTT on Internet of Things applications. It concluded that MQTT protocol performs better than HTTP in this kind of situation. HTTP has a fairly bigger header than MQTT with information that might be relevant for the internet, however it is not so relevant for an internet of things application. It causes what is called of overhead Yokotani and Sasaki (2016), which results in a lot of delays and a high network's resources consumption when compared to MQTT. Besides that, it operates under the protocols TCP/IP which means that it has a reliable communication. However, in order to provide a reliable communication, it needs to be connection oriented. This means that every time data is sent, a connection needs to be established. This implies the increasing of the delay time. On the other side, the MQTT protocol allows to choose the reliability of its connection. There are three types, QoS0 (Non assured transmission), QoS1 (Assured transmission) and QoS2 (Assured service on applications) Yokotani and Sasaki (2016). Therefore, in a system where reliability is not a top priority, the faster option can be chosen, which makes the exchange of data way faster.

8.1.1 *Server's architecture*

According to the literature there is a pattern regarding the architecture of a system when it comes to human activity recognition. Both D. Lara and Labrador (2013) and Martiskainen et al. (2009), present activity recognition architectures and they were an inspiration for the creation of the architecture on the figure 11. It first starts by pre processing the data. This is the step where outliers, nulls or values which had the label equal to zero were removed. After that, a data splitting is performed. Similarly to other machine learning applications, a human activity recognition application goes through two stages, training and testing D. Lara and Labrador (2013). Data is always separated in two, usually on a ratio of 75%/25% as mentioned in D. Lara and Labrador (2013). One part of it is used to train the classifier while the other one is used to test the classifier after the training is done. This happens because if the same data is used to train and test, the system will "remember" it and a phenomenon called overfitting can happen. Overfitting means that a model fits too accurately for the data of a particular dataset and not so well for new data. However, in this kind of applications it is needed to generalise, i.e., the system should be able to predict correctly or with a relatively high accuracy when new data is presented Guido (2017).

Then, features are extracted either for the training data set and the testing data set. Once the features are calculated, the system starts to train the model using a machine learning technique. More details about this in section 7.3 of this same chapter. Once the model is trained, the features of the test data set is passed to it and an evaluation of the model is made in order to check the accuracy of the model. If the model is approved it is ready to receive data from external environment gathered by the sensors.

Both the feature extraction and model evaluation will be discussed in detail further on this document.

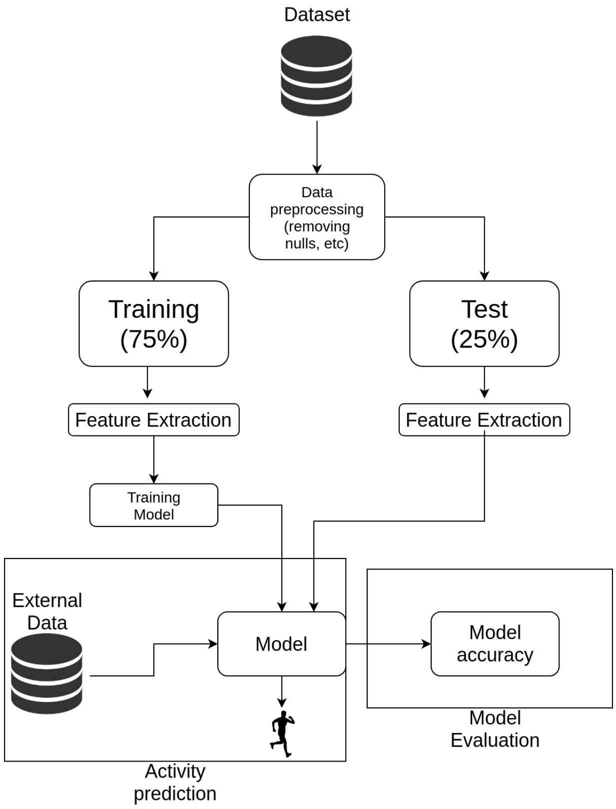

Figure 11: Human Activity Recognition server architecture

8.2 FEATURE EXTRACTION

The accuracy of an activity recognition system is based on feature extraction. Since there is not enough information on the raw accelerometer for the system to identifty an activity, it is impossible to ask for a prediction solely based on values as they come out from the sensor. Besides that, computing every and each single row of data would be too heavy computational wise. Therefore there is a need to do the feature extraction which allow signals to be compared D. Lara and Labrador (2013). Features are calculated in chunks. A time window which varies according to the activities. Bigger time windows tend to be better. However, it should not be bigger than the duration of an activity. Since it would include two different activities in the same window Twomey et al. (2018). This subject is discussed later on the subsection 7.2.1.. According to Olszewski (2001), there are two ways to extract features from time series data, *statistical* and *structural*. While the statistical use quantitative characteristics, the structural approach considers the relationship between data. Usually there are two types for features, the time domain features and the frequency domain features Twomey et al. (2018), features such as *mean*, *standard deviation*, *variance*, etc. belong to the time domain while on the frequency domain side *Fourier Transform* is the most famous one. The features should be carefully chosen. A good amount of features can improve the accuracy of the system, but if the number of features is too large, it will impair the accuracy of the model, especially if the data set is not too big Mannini and Sabatini (2010). Each category of data has its most common set of features, the table 3 sums up the features that can be found on the literature for human activity recognition and which are reportedly the most efficient ones for that same purpose.

Study	Features
D. Lara and Labrador (2013)	*mean*, *standard deviation*, *variance*, *mean absolute deviation*, *fourier transform*, *correlation between axes*
Bao and Intille (2004)	*mean*, *energy*, *frequency-domain entropy* and *correlation*
Zhen-Yu He and Lian-Wen Jin (2008)	*mean*, *standard deviation*, *energy* and *correlation between axes*
Martiskainen et al. (2009)	*mean*, *standard deviation* and *kurtosis*

Table 3: Feature selection in the literature

Considering all of the aforementioned, the chosen features for this work are the following: For a given signal $X = \{x1 \ldots, x_n\}$

Root Mean Square (rms)

As the name suggests, it is the square root of the mean square. It takes all of the axis from a sensor, computes the average of the square of each element and then the square root is calculated, as shown in equation 1.

$$RMS(X) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \tag{1}$$

Fast Fourier Transform

This equation computes a signal from the time domain to frequency domain [1]. It was calculated using the method **fft** of the library *Scipy*. It returns an array *y*, which *y[0]* represents the zero-frequency term, *y[1:n/2]* the positive term and *y[n/2:]* the negative term. Each of these were calculated for each sensor.

Standard Deviation

This equation calculates the amount of variation of a set of values as shown in equation 2.

1 https://en.wikipedia.org/wiki/Fast$_{F}ourier_{t}ransform$

$$\sigma_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2)} \tag{2}$$

Variance

Variance allows to measure how spread out are the values, of a random set, from their average. It is a bit similar to the *Root Mean Square*, however, it does not calculate the square root of the result.

$$\sigma_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{3}$$

Mean

This is the standard average of a set of values from the three axis of a sensor. It sums every element and then it is divided by the number of elements that were summed, 4

$$\boldsymbol{mean} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{4}$$

8.2.1 *Time window and Overlap*

According to Baños et al. (2014), there are three types of data segmentation:

- Activity-defined windows - It has two limits which refer to the beginning and the end of an activity;
- Event-defined windows - The proposal of this approach is to find specific occurrences of events to define the window;
- Sliding windows - The most used approach for data segmentation and it is also known as *windowing*. A window is defined with an unchangeable size and no gap between two windows which is called, *overlap*. Considering the scope of this work, this is the most appropriate approach to be used.

It is extremely important to choose the right time window so the signal segmentation can be properly done and the model's accuracy maximised. The process of segmentation varies on the type of application. If the system is specifically designed to recognise a pre determined activity, it is easier to choose this parameter since it is only needed for that activity. However, if the system is designed to recognise multiple activities, the size of the window should fit as many activities as possible which makes it difficult to choose the right size Baños et al. (2014).

The sliding window method is simple and there is no need to preprocess data, it is what makes windowing ideal for applications which need to deliver a response in real time. In the literature there is a wide range of window sizes, 5 seconds Zhen-Yu He and Lian-Wen Jin (2008), 6.7 seconds Bao and Intille (2004), Pirttikangas et al. (2006) compared several time windows 1, 2, 5, 7, 10 and 15 and the one in which the accuracy was higher was the 1 second window combined with the algorithm kNN, it had an accuracy of about 90% and the study Mantyjarvi et al. (2001) used a window size of 2 seconds. From the literature it is apparent that smaller window sizes are more accurate and it is confirmed by Baños et al. (2014) which reports that a size between the range of 1 second and 2 seconds is the best compensation between the speed of activity prediction and accuracy.

The other parameter which should be taken in consideration is the overlap. There seems to be a consensus on this regard as the majority of works that use this parameter uses a 50% overlap window Chernbumroong et al. (2014), Chernbumroong et al. (2011), Wannenburg and Malekian (2017), Mannini and Sabatini (2010), Bao and Intille (2004).

Considering that and the activities that are recognise by the system, *Standing Still*, *Walking*, *Running* and *Squat*, all of these activities are activities which usually take some time, apart from the *Squat* which has usually a duration of about 2 to 3 seconds. So, a time window of 2 seconds is an appropriate window to cover the duration of all the activities on the system. Since it has the duration of the *Squat* and the other exercises usually occur for more than 2 seconds so they can be segmented in chunks of 2 seconds.

Taken all the aforementioned in consideration, it was decided that a time window of 2 seconds and an overlap of 50% would be used.

8.3 ALGORITHMS

Machine learning has a wide variety of algorithms depending on the problem it needs to solve. Those algorithms fit in two different groups, *supervised* and *unsupervised*. The *supervised* approach deals with labeled data while *unsupervised* approach deals with unlabeled data D. Lara and Labrador (2013). Inside the *supervised* learning there are the *classification* and the *regression* problems. Human activity recognition is a classification problem since it needs the system to predict an activity or to give a label to the data that is provided. That is why the *supervised* approach is often used for such problems.

For *classification* problems there are several algorithms such as SVM, k-NN, RF, etc.. According to D. Lara and Labrador (2013), which gathers all the state of art techniques to human activity recognition, two of the most used algorithms are SVM and k-NN, Twomey et al. (2018) mentions RF, Zhen-Yu He and Lian-Wen Jin (2008) uses SVM. So it is clear that these three are the ones that are most common for activity prediction. Therefore, those three will be compared in this work to evaluate which one is better for its purpose.

8.4 CODE IMPLEMENTATION

This section contains detailed information about the implementation of the application. Aspects such as technologies, implementatiion of the client and server are explained in detail.

The server was built in Python (version 3.7.1), using the library Pandas for data manipulation and Scikit-learn for machine learning. Regarding the client, it is splitted in two parts, the *User Interface (UI)* and the part responsible to connect to the sensors and the server. The UI was built using the Javascript's framework VueJS, while the other part was also built in Python. It uses bluepy which is an interface for the Bluetooth LE, responsible to connect with the sensors.

8.4.1 *Data Collection*

For the creation of the data set a Python script was created. The figure 12 represents a diagram of the script. The script pairs with the three sensors and gather all the data into a CSV file. So once the script is launched three different threads are created, one per sensor. Once a thread starts, it tries to pair with the sensor, which is made through bluetooth, and when the pairing is completed it checks whether the other threads have already paired with their sensor, if yes then it is ready to proceed, otherwise it will be put asleep and be awaken whenever a new thread is paired to the sensor. For this effect, a global variable which is shared by the three threads is incremented inside a lock block in order to avoid concurrency problems. Once this variable reaches the value 3 it means that every thread has paired with their respective sensor and the script is ready to advance to the next stage. The sensors, Sensortag CC2650STK, are composed of several sensors, IR Temperature, Movement (accelerometer, gyroscope and magnetometer), Humidity (both relative humidity and temperature), Pressure and Optical. For this work, only Humidity and Movement are collected so this sensors need to be turned on. They are turned on by the following process, according to the official website [2]

```
sensorOn = struct.pack("BB", 0x7F, 0x02) #Movement sensor
humSensor = struct.pack('B',0x01) #Humidity sensor

  config_uuid = TI_UUID(0xAA82) #Movement sensor
```

[2] 3

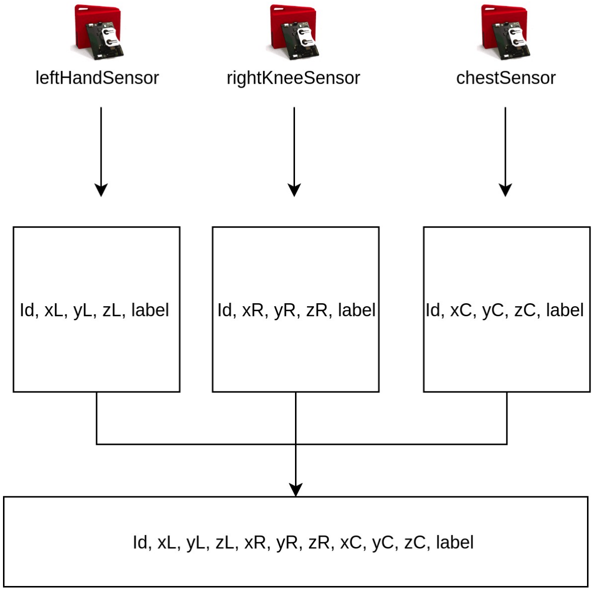

Figure 12: Dataset Creation

```
5     data_uuid = TI_UUID(0xAA81) #Movement sensor

7     hum_uuid = TI_UUID(0xAA21) #Humidity sensor
      hum_config = TI_UUID(0xAA22) #Humidity sensor
9
      sensor = Peripheral(sensorMAC)
11    sh = sensor.getCharacteristics(uuid=config_uuid)[0] #Movement sensor
      sh.write(sensorOn, withResponse=True) #Movement sensor
13
      sh_hum = sensor.getCharacteristics(uuid=hum_config)[0] #Humidity sensor
15    sh_hum.write(humSensor, withResponse=True) #Humidity sensor
```

Listing 8.1: Sensor's configuration

Line 1 and 2 are the bytes which give the indication to turn on the movement sensor and humidity sensor respectively. The lines 4 and 5 are the bytes of configuration and data respectively for the movement sensor. The configuration one is the one responsible to turn on and off while the data one is the responsible to collect data. The same happens on the line 7 and 8, but for the humidity sensor. Lines 10, 11 and 12 is where effectively is given the instruction to turn the movement sensor on. While for the humidity sensor, the lines 14 and 15 have that task.

After everything is set up it is time to start the data collection. For this effect, three python dictionaries were created, one for each thread. The keys for each one of these dictionaries is a standard index which is incremented every time the sensor provides data. Each thread runs for the time that was passed when the program was executed, once this timer finishes the script goes through the three dictionaries and joins the data with the same index on a CSV file. The reason why the join is made using the index was because using time frames would be too complex, since time frames differ in milliseconds and maybe a time frame from 5 milliseconds ago could be joined to a different register than the supposed one, so it is assumed that the registers with the same index correspond more or less to the same exact moment during the activity. The format of the CSV file is the following *Id, xL, yL, zL, xR, yR, zR, xC, yC, zC, label*, where the variables with an L correspond to the left hand sensor, with an R to the right knee sensor and the C to the chest sensor. The label represents the activity that is being recorded, there are 5 possible values for the label, 1 for *Standing Still*, 2 for *Walking*, 3 for *Running*, 4 for *Squat* and 5 for *Wrong Squat*, in other words, a squat having the knees buckled in. The script's duration depends on the activity which is being recorded.

8.4.2 *Client*

The Client starts a connecction with the MQTT broker in order to communicate with the server side. The broker that was used was ***test.mosquitto.org***. MQTT works with a topics system. A message is sent to a specific topic and, in order to receive that message, the other part needs to subscribe to that topic. This way it is possible to exchange messages without a verification of what it is. Therefore, after the connection is established, it subscribes to the topic *"Server"*, so the client can receive messages from the server. Once this part is done, the next part follows the principle used on the implementation of the script which creates the data set. It first starts to create three threads, one for each sensor, each thread tries to pair with its correspondent sensor. Once every sensor is paired, the systems starts to collect data. While it collects the data, the server is already waiting for a message from the client. The client sends data to the server every 22 registers, about each 2 seconds. It was considered that 22 registers was the minimum amount of registers that would compose an activity. That is the duration of the *squat*, the faster activity in the system. The server responds with an activity and it is displayed to the user as shown in the rectangle 3 on the figure 13.

All of this is controlled by a simple UI, the user should switch a toggle button to "on". Then, it will be indicated by a green light when the sensors are fully paired, otherwise those indicators show up as red as can be seen on the rectangle 1 from the figure 13.

Internally, the data is collected the following way.

```python

rawVals = sh.read() #read raw values from sensor
(accX, accY, accZ) = struct.unpack('<hhh', rawVals)
```

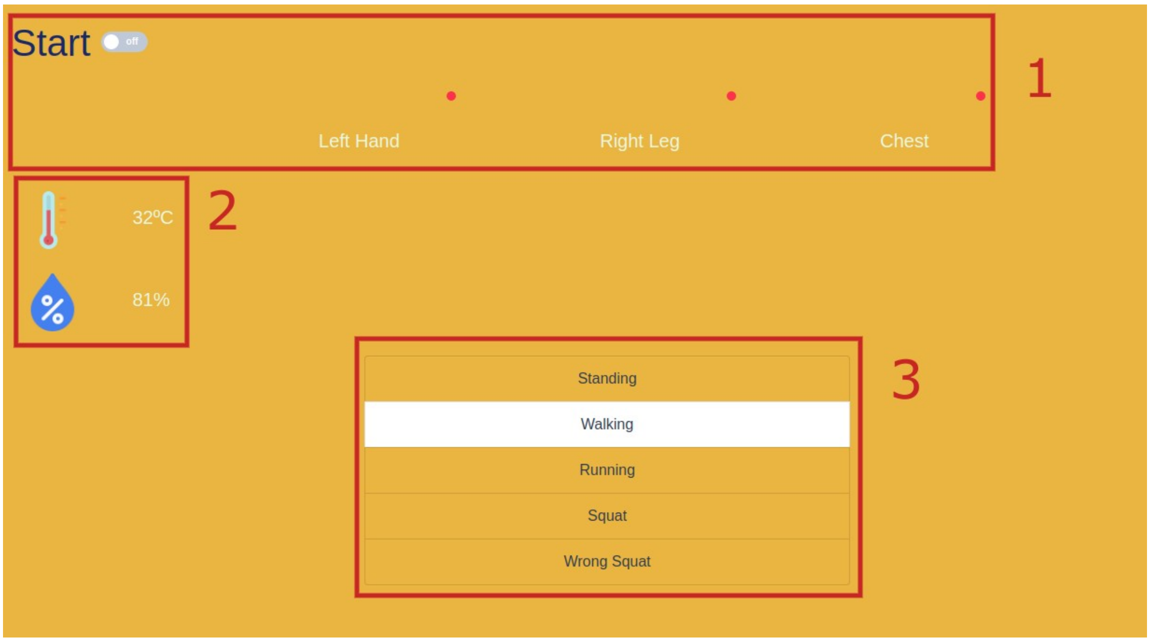

Figure 13: Application's UI

```
5   scale = 32768/8

7
    data += '{\"x\": %f, \"y\": %f, \"z\": %f, \"sensor\": \"%s\"},' % (accX /
      scale, accY / scale, accZ / scale, sensorName)
9
```

Listing 8.2: Python example

Initially, it reads the raw values from the sensors (line 2), which are purely hexadecimal values. Then, those values are unpacked to several variables, one for each axis (line 3). These values are divided by scale, as indicated in [4] concatenated in a JSON format string and sent to the server (line 8).

Moreover, besides the accelerometer data, temperature and relative humidity should be taken into account, they are also indicated to the user in the UI, rectangle 2, figure 13. The first solution would be to constantly collect these two parameters. However, it decreased the amount of registers per second. Considering that they are not parameters that vary in a short time period, they are only updated every two minutes. The temperature and humidity are taken from the three sensors, then the average of the three is calculated and presented to the user.

According to the sensor's official website 4, the temperature and relative humidity are captured the following way.

4 http://processors.wiki.ti.com/index.php/CC2650$_{S}ensorTag_{U}ser's_{G}uide$

```
  rawVals_hum = sh_hum.read()
2
  (temp,hum) = struct.unpack('hh',rawVals_hum )
4
  temp = (temp/65536)*165-40
6 hum = (hum / 65536)*100
```

Listing 8.3: Temperature Relative Humidity capture

It starts by reading the raw values from the sensor (line 1). Then, those values are extracted to two separate variables, *temp* and *hum*, one for temperature and one for humidity respectively (line 3). At last, line 5 and 6 calculates temperature and humidity as indicated on the sensor's documentation 4.

Besides of what is shown in figure 13, the UI also displays several alerts to the user according to the current context. The figures 14, 15 and 16 represent the alerts that are showed for low temperature, high temperature and high humidity and a wrong squat, respectively. These alerts are explained in detail on the chapter 7.

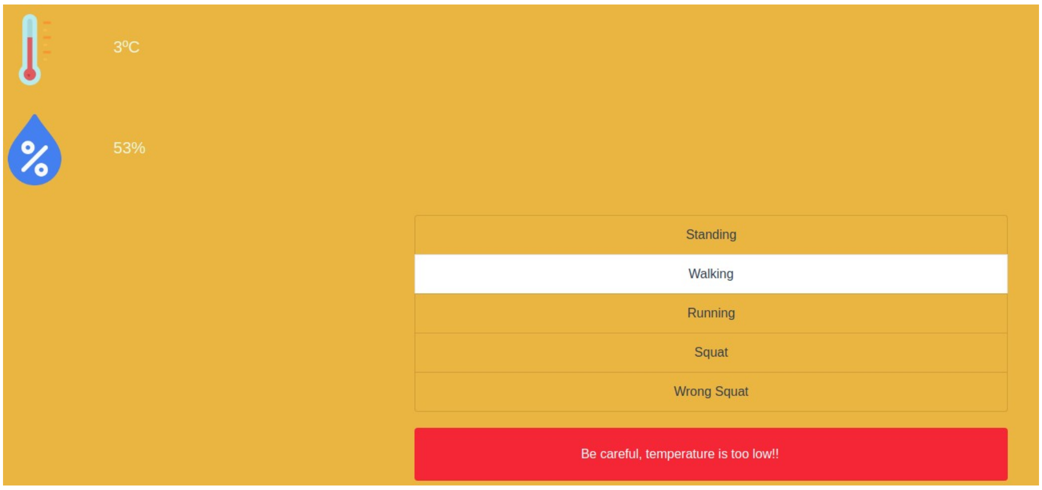

Figure 14: Alert low temperature

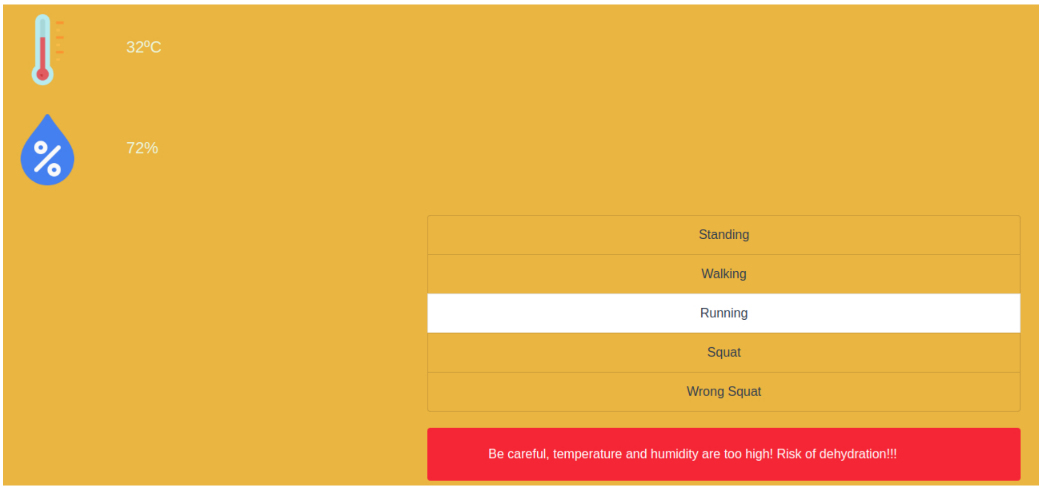

Figure 15: Alert high temperature

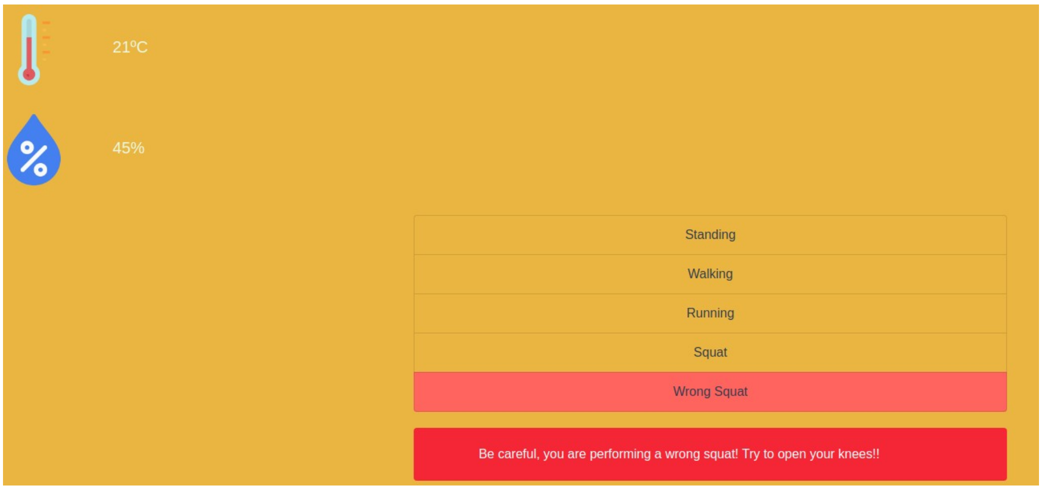

Figure 16: Alert Wrong Squat

8.4.3 *Server*

The server starts by reading the CSV file which contains all the activities data. The data is inserted on a pandas data frame with time frame as an index and a time difference between registers of 9 milliseconds. At the end of the reading not only ***NULL*** values are dropped but also lines in which the label is undefined for some reason.

The program proceeds to split the data, the *Scikit-learn* function *train_test_split* is used for this effect.

```
X_train, X_test, y_train, y_test = train_test_split(data, y, test_size
=.25, random_state=0,shuffle=False)
```

Listing 8.4: Split Data

This method has 5 parameters, **data** contains the accelerometer data, **y** contains all the labels for each row which is present in **data**. The third one, *test_size* $= .25$ indicates that the method will split 25% of the provided rows for testing purposes. The parameter *random_state* assures that the splitted data is the same no matter what, ie, the training data and test data are always the same regardless the number of times this function is applied on the same data set. The last parameter, **shuffle**, is set to false, since accelerometer data can only be translated into an activity if that data refers to the sequence of movements of that specific activity. If every row gets shuffled, features will be calculated with rows from completely different activities, therefore, the accuracy will substantially drop.

Subsequently, the application proceeds to extract the features of both, the training data set and the test data set. Features are calculated with a window size of 2 seconds, already discussed previously on this document.

Once the features are calculated, the model is ready to be trained. Three algorithms are compared in this study, k-NN, SVM and RF. There is a general method that trains the model which is the following.

```
1  def train_model(X, y, est, grid):
       print('::::Train Model::::')
3      gs = GridSearchCV(estimator=est, param_grid=grid, scoring='accuracy',
   cv=5, n_jobs=-1)
       gs = gs.fit(X, y.values.ravel())
5
   return (gs.best_estimator_, gs.best_params_)
7
```

Listing 8.5: Train Model

The method *GridSearchCV* finds the best parameters for a determined estimator, i.e., an algorithm (line 3). Aftwerwards, *fit* is done and the best estimator and best parameters are returned (line 4). The training for each one of the three algorithms is done the following way.

```
param_range = [0.0001, 0.001, 0.01, 0.1, 1, 10 ,100]

print('Support Vector Machine')
svm_model, params = train_model(X_train, y_train,
                    est=SVC(probability=True),
                    grid={'C': param_range, 'gamma': param_range, 'kernel': ['
linear']})
```

```
8      print('K-Nearest Neighbor')
       knn_model, params = train_model(X_train, y_train,
10                         est=KNeighborsClassifier(),
                           grid={'n_neighbors':[5, 8, 10, 12], 'weights':['uniform',
       'distance']})
12
       print('Random Forest')
14     model, params = train_model(X_train, y_train,
                           est=RandomForestClassifier(n_jobs=-1, criterion='entropy')
       ,
16                         grid={'n_estimators':[10,30,100]})

18
```

Listing 8.6: Algorithms

Line 1 describes the parameters for the SVM model *C* and *Gamma*. The *C* parameter trades off an accurate classification of training examples against maximization of the decision function's margin. While *gamma*, defines the influence of a single training example [5]. These values were chosen because those two parameters usually vary between the range of $10^{-}3$ to 10^{3}, which is proven to be enough 5. The lines 5, 10 and 15 correspond to the estimator, i.e, the classifier, SVM (line 5), k-NN (line 10) and RF (line 15). The grid has a range of value for each parameter of each classifier, on the $train_{m}odel$ method, *GridSearchCV* uses cross-validation to find the best parameters from the correspondent grid (lines 6, 11 and 16).

Afterwards, the server stays on listening to MQTT messages. There are three topics to which data will be sent, *Left*, *Right* and *Chest*. Each one of these topics corresponds to a sensor. After receiving data from these three sources, they are arranged in an array. Afterwards, that data is put in a *pandas* data frame and the features calculated. Those features, are then passed to the models which return an activity prediction, this prediction is sent to the client to be shown to the user.

5 *https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html*

9

RESULTS

This chapter discusses the obtained results of the evaluation of the application. That evaluation is made through metrics such as: *accuracy*, *precision* and *recall*. All of this metrics are compared according to different parameters for each algorithm.

When evaluating a classifier, data is usually presented on a *confusion matrix* $M_{n \times n}$ for n classes D. Lara and Labrador (2013). It is a matrix where the value M_{ij} means the amount of activities i, that were classified as j. Therefore, this matrix contains the following values:

- *True Positives* (TP): The number of instances that were correctly classified;
- *True Negatives* (TN): The number of instances that were classified as another class, correctly;
- *False Positives* (FP): Number of instances that were classified as positive but they are negative.
- *False Negatives* (FN): Number of instances that were wrongly classified as belonging to a different class.

A standard confusion matrix has the following format:

		Predicted class	
		1	2
Classes	1	TP	FN
	2	FP	TN

Table 4: Standard Confusion matrix

From the creation of a confusion matrix, there are some metrics that can be calculated to evaluate the overall performance of the system. The work D. Lara and Labrador (2013), describes the following evaluation metrics:

- Accuracy: It is useful to measure the performance of the system as a whole, equation 5;

- Precision: Ratio of the correctly predicted instances to the total of instances that were classified as positive, equation 6;
- Recall: It is the same as the precision, but instead of ratio to total of instances that were classified as positive, it is the ratio of positives to the total number of actually positive instances, equation 7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

A confusion matrix was created for each of the three algorithms SVM 5, k-NN 6 and RF 7, using cross-validation, with every activity recognised by the system.

Actual activity	**Predicted activity** Standing Still	Walking	Running	Squat	Wrong Squat
Standing Still	449	0	3	5	1
Walking	1	1013	0	0	2
Running	0	0	850	2	0
Squat	10	0	2	95	30
Wrong Squat	6	4	0	27	51

Table 5: SVM algorithm's confusion matrix

Actual activity	**Predicted activity** Standing Still	Walking	Running	Squat	Wrong Squat
Standing Still	454	3	0	0	1
Walking	1	1012	3	0	0
Running	0	2	850	0	0
Squat	25	3	1	72	36
Wrong Squat	7	4	0	41	36

Table 6: k-NN algorithm's confusion matrix

Actual activity \ Predicted activity	Standing Still	Walking	Running	Squat	Wrong Squat
Standing Still	452	2	0	2	2
Walking	0	1012	4	0	0
Running	0	0	850	2	0
Squat	11	3	2	97	24
Wrong Squat	10	7	0	34	37

Table 7: RF algorithm's confusion matrix

It can be seen that almost every instance of *Standing Still*, *Running* and *Walking* got correctly classified. For *Walking*, the tables present a number ranging between 449 and 454 instances as true positives. Regarding the false negatives, there are 9, 4 and 6 for the tables 5, 6 and 7, respectively. Concerning the false positives, it ranges between 17 and 33 instances. The activity *Walking* even presents better results by having true positives ranging between 1012 and 1013, false negatives between 3 and 4 and false positives are either 4 or 12. The same can be applied for *Running*, regarding the true positives every table has the same number of instances, 850. While false negatives have 2 instances and false positives are 4, 5 and 6 for SVM, k-NN and RF, respectively. Now, the *Squat* and the *Wrong Squat* are the activities that got less correctly classified. k-NN was the algorithm that predicted less *Squat* and *Wrong Squat* instances as true positives, 72 and 36. While SVM classified 95 instances as true positives for the *Squat* and 51 instances as true positive for *Wrong Squat*. And, RF 97 and 37 for *Squat* and *Wrong Squat* respectively. However, the problem resides on the false negatives and false positives. Many instances of the *Squat* are classified as *Standing Still* or even as a *Wrong Squat*. For instance, 27, 41 and 34 are classified as a *Squat* when they are a *Wrong Squat*, tables 5, 6 and 7, respectively. Whereas, many *Wrong Squat* instances are mixed up by *Squat*, as can be seen on the confusion matrices. There are two possible reasons for this to happen:

- A *Squat* or a *Wrong Squat*, starts and finishes at the same position. The same position that was considered when recording the activity *Standing Still*. Therefore, during the training phase, the defined window could have split a *Squat* in two different parts. Where the second part, would be precisely the *Standing Still* activity but the label attached to it would be for *Squat*, since it belonged to it;
- The other problem, where the *Wrong Squat* gets classified as *Squat* or the other way around is way more evident. Even though, the subjects were asked to knuckle the knee in, those two movements are still very similar. And, sometimes it is hard for the system to differentiate it due to the similarities.

A possible solution for the first problem can be restricting the data to the movement of the *Squat* itself. This way, no matter how that data would be split for the calculation of features, it would never have similar values to the *Standing Still* activity. Regarding the second problem, a possible solution for it could be the addition of another sensor on the left knee. Since both knees are involved on the execution of this activity and it would add more accuracy to the system. This could worsen the problem of *obtrusiveness* described on chapter 7. However, the trade off could be advantageous if the accuracy improved with it. Another simpler solution that could be done with the material already used would be to collect more data for both of these activities. It would probably increase the accuracy.

The compilation of these results got SVM an overall accuracy of 96.4%, k-NN an overall accuracy of 95.2% and RF an overall accuracy of 96.1%. The accuracy, precision and recall for each activity are presented on the tables 8, 9 and 10.

By analysing thoroughly each matrix, it is evident that there was no big problem in predicting the activities *Standing Still*, *Walking* and *Running*. For the *Standing Still* activity, in all algorithms the accuracy and recall was within a range of of 98% to 99%, table 8 and 10. Regarding the precision, it lowered a bit, being the SVM algorithm the best one with 96.3%, table 9. The activity *Walking*, is around 99% for every metric, accuracy, precision and recall. The same happens for *Running*.

		Activities			
Algorithm	Standing Still	Walking	Running	Squat	Wrong Squat
SVM	98.9%	99.7%	99.7%	97%	97.2%
k-NN	98.5%	99.6%	99.8%	95.8%	96.5%
RF	98.9%	99.3%	99.6%	96.9%	97.2%

Table 8: Accuracy

		Activities			
Algorithm	Standing Still	Walking	Running	Squat	Wrong Squat
SVM	96.3%	99.6%	99.4%	73.6%	60.7%
k-NN	93.1%	99.6%	99.5%	63.7%	49.3%
RF	95.6%	98.8%	99.3%	71.8%	58.7%

Table 9: Precision

Algorithm	Standing Still	Walking	Running	Squat	Wrong Squat
	Activities				
SVM	98%	99.7%	99.8%	69.3%	58.0%
k-NN	99.1%	99.6%	99.7%	52.6%	40.9%
RF	98.6%	99.6%	99.7%	70.8%	46.8%

Table 10: Recall

As can be seen from the results, there is no big difference between these three activities regardless the algorithm. It goes to the encounter of many studies where ambulation activities such as *walking* and *running* have a high recognition percentage. For instance, Óscar D. Lara et al. (2012) got an accuracy of 100% for *running*. Zhen-Yu He and Lian-Wen Jin (2008), presented an accuracy of 98.29% for *walking*.

However, the problem relies on the detection of *Squat*. Regarding the accuracy, both *Squat* and *Wrong Squat*, ranges between 95% and 97% 8. By analysing the precision and recall, it can be seen that it drops substantially. For the *Squat*, the precision varies between 63.7% (k-NN) to 73.6% (SVM), table 9. This means, that the best algorithm for this activity, SVM, only predicted 73.6% of *Squat* correctly. The activity *Wrong Squat*, also performed better on the algorithm SVM and worse on the algorithm k-NN. SVM only predicted a *Wrong Squat*, 60.7% of the times.

Regarding the recall, the algorithm SVM also performed better for both *Squat* and *Wrong Squat* with 69.3% and 58%, respectively. The algorithm, k-NN presented worse performance with 52.6% and 40.9% for *Squat* and *Wrong Squat*, respectively. This means that using the SVM algorithm, from all the *Squats*, the system labeled 69.3% as a *Squat*. And from all the *Wrong Squats*, the system labeled 58% as a *Wrong Squat*.

These values agrees with many of the accuracies presented in the literature for a *Squat*. For instance, Olguin presents an accuracy of 75.8%, Adaskevicius (2014) got an accuracy of about 77% and Margarito et al. (2016) has an accuracy of 73% on normal weight population. Therefore, the values obtained on this work do not diverge from the values presented on the literature.

Considering all of this, the best algorithm to be used on this system is the SVM. It presented an overall performance better than the other two. Moreover, it was the algorithm that performed the better on the two specific activities, *Squat* and *Wrong Squat*. This tests were key to complete one of the specific objectives that was to define the most appropriate machine learning technique. Also, the discrimination between *Squat* and *Wrong Squat* allows to give a personalised suggestion to the user regarding the form of the exercise. Which belongs to the specific objectives that is, to define what suggestion should be made based on data.

Another validation that was made was using real life *Running* data on the system trained with the simulation generated data for *Running*. Since, this application is made to be used in real life, it is necessary to validate it for real life. The performance of the system, with real *Running* data, is represented in the table 11.

	Predicted activity		
	SVM	k-NN	RF
Accuracy	98.9%	99.6%	99.5%
Precision	94.3%	100%	97.9%
Recall	94.3%	96.1%	97.4%

Table 11: System performance for real *Running* data.

By analysing the table 11, it can be seen that the values dropped slightly. However, they still keep a relatively high performance not too far away from the values presented in the literature and in the validation with data from the simulation. It is natural to have a drop on these values since the machine, while running keeps doing the same movements over and over again. Whereas a person can do unexpected movements when an obstacle comes on the way for instance. Therefore, it is considered that the system is prepared to receive data from the real world and the last specific objective was achieved, i.e, to test and validate the architecture.

10

CONCLUSION

This document described in detail the development of a CA system architecture that collects data from the user such as accelerometer, environment temperature and air humidity and gives the user suggestions based on that data. It contains the state of the art that introduces the reader to the most recent studies as well as some definitions about important concepts to this thesis such as context-awareness, people centric computing, machine learning and wearable devices. It also contains some related work that was found during the research with one of them being Angelo Costa and Julian (2018) which is a robot that intends to help elderly with their workout. Besides that, this document covered the methodology that was used for the development of this work, the final architecture and a chapter that gets in depth on physical activity.

The main objective of this work is to develop an architecture that gives the user suggestions while performing a workout. A system based on this architecture will be able to work either indoor and outdoor. And to accomplish this main objective, smaller specific objectives were created such as to define which parameters should be sensed, the development of the architecture, to define the machine learning techniques that fit the architecture and finally test and validate the architecture.

The results obtained by this architecture were very satisfactory. The system had no problem in recognising the activities *Standing Still*, *Walking* and *Running*. The SVM was the algorithm that performed the better with an accuracy of 99% for *Standing Still* and 99.7% for both *Walking* and *Running*. Regarding the *precision* and *recall*, the SVM was the best one overall as well, those values were within the range of 96% and 99%. For the other test that was made, i.e., with real *Running* data, the system also presented a good overall performance with the values ranging between 94% and 99% for *accuracy*, *precision* and *recall*. Whereas the system could not recognise so well the *Squat* and *Wrong Squat*. The SVM, got an *accuracy* of about 97% for both activities. While the *precision* and *recall* dropped to about 73% and 70% respectively for the *Squat*. While the *Wrong Squat* had values close to 60% for both metrics.

Regarding the future work, firstly it should be implemented in wearable devices since it is easier to be used and does not need any extra device. Another thing that could be done would be to improve the way the warnings are provided to the user, for example, voice warnings instead of the current warnings on the display. Audio warnings allows the user to focus totally on the execution of the exercise. Therefore, the exercise can be better performed and lower the odds of injury even more. Also, more data could be sensed to improve the quality of the suggestions. For instance, on the outdoor activities the use of GPS could help in the process of detecting an activity or the kilometers that were ran could be provided. The heart rate and sleep detection would also be very useful to provide quality suggestions. Regarding the suggestions, apart from giving suggestions during the exercise, suggestions could also be given before the exercise. For example, if the user slept 6 hours instead of 8, the system would recommend a run of 3 kilometers instead of 5. Another thing that will improve the system is the implementation of physical attributes such as age, gender, weight and height. Those parameters influence physical activity and adding it to the system would improve the quality of the recommendations overall.

BIBLIOGRAPHY

Activity classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):119–128, Jan 2006. ISSN 1089-7771. doi: 10.1109/TITB.2005.856863.

G. D. Abowd, A. K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. In *Digest of Papers. First International Symposium on Wearable Computers*, pages 179–180, Oct 1997. doi: 10.1109/ISWC.1997.629943.

Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In Hans-W. Gellersen, editor, *Handheld and Ubiquitous Computing*, pages 304–307, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-48157-7.

Rimas Adaskevicius. Method for recognition of the physical activity of human being using a wearable accelerometer. *Elektronika ir Elektrotechnika*, Vol.20:127–131, 01 2014. doi: 10.5755/j01.eee.20.5.7113.

Unai Alegre, Juan Carlos Augusto, and Tony Clark. Engineering context-aware systems and applications. *J. Syst. Softw.*, 117(C):55–83, July 2016. ISSN 0164-1212. doi: 10.1016/j.jss.2016.02.010. URL https://doi.org/10.1016/j.jss.2016.02.010.

Miguel Cazorla Angelo Costa, Ester Martinez-Martin and Vicente Julian. Pharos—physical assistant robot system. 2018. doi: 10.3390/s18082633.

Ferhat Attal, Samer Mohammed, Mariam Dedabrishvili, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. Physical human activity recognition using wearable sensors. *Sensors*, 15(12):31314–31338, 2015. ISSN 1424-8220. doi: 10.3390/s151229858. URL http://www.mdpi.com/1424-8220/15/12/29858.

Oresti Baños, Juan Manuel Gálvez, Miguel Damas, Héctor Pomares, and Ignacio Rojas. Window size impact in human activity recognition. In *Sensors*, 2014.

Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, pages 1–17, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

R Beaglehole and D Yach. Globalisation and the prevention and control of non-communicable disease: the neglected chronic diseases of adults. *The Lancet*, 362(9387):

903 – 908, 2003. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(03)14335-8. URL http://www.sciencedirect.com/science/article/pii/S0140673603143358.

Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.*, 46(3):33:1–33:33, January 2014. ISSN 0360-0300. doi: 10.1145/2499621. URL http://doi.acm.org/10.1145/2499621.

J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *In: Workshop on World-Sensor-Web (WSW'06): Mobile Device Centric Sensor Networks and Applications*, pages 117–134, 2006.

Fabio Buttussi and Luca Chittaro. Mopet: A context-aware and user-adaptive wearable system for fitness training. *Artificial Intelligence in Medicine*, 42(2):153 – 163, 2008. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2007.11.004. URL http://www.sciencedirect.com/science/article/pii/S0933365707001480. Wearable Computing and Artificial Intelligence for Healthcare Applications.

Marianna Capecci, Lucia Pepa, Federica Verdini, and Maria Ceravolo. A smartphone-based architecture to detect and quantify freezing of gait in parkinson's disease. *Gait Posture*, 50, 08 2016. doi: 10.1016/j.gaitpost.2016.08.018.

Jose M. Chaquet, Enrique J. Carmona, and Antonio Fernández-Caballero. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633 – 659, 2013. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2013.01.013. URL http://www.sciencedirect.com/science/article/pii/S1077314213000295.

S. Chernbumroong, A. S. Atkins, and H. Yu. Activity classification using a single wrist-worn accelerometer. In *2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings*, pages 1–6, Sep. 2011. doi: 10.1109/SKIMA.2011.6089975.

Saisakul Chernbumroong, Shuang Cang, and Hongnian Yu. A practical multi-sensor activity recognition system for home-based care. *Decision Support Systems*, 66:61 – 70, 2014. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2014.06.005. URL http://www.sciencedirect.com/science/article/pii/S0167923614001791.

Paweł Chmura, Marek Konefał, Marcin Andrzejewski, Jakub Kosowski, Andrzej Rokita, and Jan Chmura. Physical activity profile of 2014 fifa world cup players, with regard to different ranges of air temperature and relative humidity. *International Journal of Biometeorology*, 61(4):677–684, Apr 2017. ISSN 1432-1254. doi: 10.1007/s00484-016-1245-5. URL https://doi.org/10.1007/s00484-016-1245-5.

M. Conti and M. Kumar. Opportunities in opportunistic computing. *Computer*, 43(1):42–50, Jan 2010. ISSN 0018-9162. doi: 10.1109/MC.2010.19.

Shane Musolesi Mirco Miluzzo Emiliano T. Campbell Andrew D. Lane, Nicholas Eisenman. Urban sensing systems: Opportunistic or participatory? pages 11–16, 2008. doi: 10.1145/1411759.1411763.

Oscar D. Lara and Miguel Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys Tutorials, IEEE*, 15:1192–1209, 01 2013. doi: 10.1109/SURV.2012.110112.00192.

F. Delmastro, V. Arnaboldi, and M. Conti. People-centric computing and communications in smart cities. *IEEE Communications Magazine*, 54(7):122–128, July 2016. ISSN 0163-6804. doi: 10.1109/MCOM.2016.7509389.

Anind Dey, Jennifer Mankoff, Gregory Abowd, and Scott Carter. Distributed mediation of ambiguous context in aware environments. pages 121–130, 01 2002. doi: 10.1145/571985.572003.

Nounou M. I. El-Amrawy, F. Are currently available wearable devices for activity tracking and heart rate monitoring accurate, precise, and medically beneficial?.h. 2015.

Nour El Helou, Muriel Tafflet, Geoffroy Berthelot, Julien Tolaini, Andy Marc, Marion Guillaume, Christophe Hausswirth, and Jean-François Toussaint. Impact of environmental parameters on marathon running performance. *PLOS ONE*, 7:1–9, 05 2012. URL https://doi.org/10.1371/journal.pone.0037407.

Matthew Ely, Samuel Cheuvront, and Scott Montain. Neither cloud cover nor low solar loads are associated with fast marathon performance. *Medicine and science in sports and exercise*, 39:2029–35, 12 2007. doi: 10.1249/mss.0b013e318149f2c3.

M. Ermes, J. PÄrkkÄ, J. MÄntyjÄrvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):20–26, Jan 2008. ISSN 1089-7771. doi: 10.1109/TITB.2007.899496.

R. F. Escamilla. Knee biomechanics of the dynamic squat exercise. *Med Sci Sports Exerc*, 33 (1):127–141, Jan 2001.

F Foerster, M Smeja, and J Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, 15(5):571 – 583, 1999. ISSN 0747-5632. doi: https://doi.org/10.1016/S0747-5632(99)00037-0. URL http://www.sciencedirect.com/science/article/pii/S0747563299000370.

Leandro Oliveira Freitas. *Uncertainty and Incompleteness Handling in Context-Aware systems*. PhD thesis, Escola de Engenharia Departamento de Informática, Universidade do Minho, March 2018.

Stuart D. R. Galloway and Ronald John Maughan. Effects of ambient temperature on the capacity to perform prolonged cycle exercise in man. *Medicine and science in sports and exercise*, 29 9:1240–9, 1997.

Henry O. Nyongesa George Wamamu Musumba. Context awareness in mobile computing: A review. Vol 2, No 1, 2016.

Andreas C. Müller Sarah Guido. *Introduction to Machine Learning with Python, A GUIDE FOR DATA SCIENTISTS*. 2017. ISBN 9781449369415.

A. E. Hardman. Accumulation of physical activity for health gains: what is the evidence? *Br J Sports Med*, 33(2):87–92, Apr 1999.

Tâm Huynh and Bernt Schiele. Towards less supervision in activity recognition from wearable sensors. pages 3–10, 10 2006. doi: 10.1109/ISWC.2006.286336.

Wijnand IJsselsteijn, Yvonne de Kort, Joyce Westerink, Marko de Jager, and Ronald Bonants. Fun and sports: Enhancing the home fitness experience. In Matthias Rauterberg, editor, *Entertainment Computing – ICEC 2004*, pages 46–56, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-28643-1.

S. Karaman, J. Benois-Pineau, R. Megret, V. Dovgalecs, J. Dartigues, and Y. Gaestel. Human daily activities indexing in videos from wearable cameras for monitoring of patients with dementia diseases. In *2010 20th International Conference on Pattern Recognition*, pages 4113–4116, Aug 2010. doi: 10.1109/ICPR.2010.999.

James F. Knight, Anthony Schwirtz, Fotis Psomadelis, Chris Baber, Huw W. Bristow, and Theodoros N. Arvanitis. The design of the sensvest. *Personal and Ubiquitous Computing*, 9 (1):6–19, Jan 2005.

A. Mannini and A. M. Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors (Basel)*, 10(2):1154–1175, 2010.

J. Mantyjarvi, J. Himberg, and T. Seppanen. Recognizing human motion with multiple acceleration sensors. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, volume 2, pages 747–752 vol.2, Oct 2001. doi: 10.1109/ICSMC.2001.973004.

J. Margarito, R. Helaoui, A. M. Bianchi, F. Sartor, and A. G. Bonomi. User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach. *IEEE Transactions on Biomedical Engineering*, 63(4):788–796, April 2016. doi: 10.1109/TBME.2015.2471094.

Paula Martiskainen, Mikko Järvinen, Jukka-Pekka Skön, Jarkko Tiirikainen, Mikko Kolehmainen, and Jaakko Mononen. Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Applied Animal Behaviour Science*, 119(1):32 – 38, 2009. ISSN 0168-1591. doi: https://doi.org/10.1016/j.applanim.2009.03.005. URL http://www.sciencedirect.com/science/article/pii/S0168159109000951.

U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 4 pp.–116, April 2006. doi: 10.1109/BSN.2006.6.

Uwe Maurer, Anthony Rowe, Asim Smailagic, and Daniel Siewiorek. *Location and Activity Recognition Using eWatch: A Wearable Sensor Platform*, pages 86–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-37788-7. doi: 10.1007/11825890_4. URL https://doi.org/10.1007/11825890_4.

G. D. Myer, A. M. Kushner, J. L. Brent, B. J. Schoenfeld, J. Hugentobler, R. S. Lloyd, A. Vermeil, D. A. Chu, J. Harbin, and S. M. McGill. The back squat: A proposed assessment of functional deficits and technical factors that limit performance. *Strength Cond J*, 36(6): 4–27, Dec 2014.

Lars Nybo. Cycling in the heat: performance perspectives and cerebral challenges. *Scandinavian journal of medicine science in sports*, 20 Suppl 3:71–9, 10 2010. doi: 10.1111/j.1600-0838.2010.01211.x.

Daniel and Olguin.

Pedro Novais Paulo Oliveira Freitas, Leandro Rangel Henriques. Uncertainty in context-aware systems: A case study for intelligent environments. 2018. doi: 225-231.10.1007/978-3-319-77703-0_22.

Robert Olszewski. Generalized feature extraction for structural pattern recognition in time-series data. 03 2001.

C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014. ISSN 1553-877X. doi: 10.1109/SURV.2013.042313.00197.

Igor Pernek, Gregorij Kurillo, Gregor Stiglic, and Ruzena Bajcsy. Recognizing the intensity of strength training exercises with wearable sensors. *Journal of Biomedical Informatics*, 58: 145 – 155, 2015. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2015.09.020. URL http://www.sciencedirect.com/science/article/pii/S1532046415002142.

Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature selection and activity recognition from wearable sensors. volume 4239, pages 516–527, 10 2006. doi: 10.1007/11890348_39.

Hiram Ponce-Espinosa, Luis Miralles Pechuán, and María de Lourdes Martínez-Villaseñor. A flexible approach for human activity recognition using artificial hydrocarbon networks. In *Sensors*, 2016.

Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'05, pages 1541–1546. AAAI Press, 2005. ISBN 1-57735-236-x. URL http://dl.acm.org/citation.cfm?id=1620092.1620107.

Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2): 13:1–13:27, March 2010. ISSN 1550-4859. doi: 10.1145/1689239.1689243. URL http://doi.acm.org/10.1145/1689239.1689243.

Dietrich Rothenbacher, Albrecht Hoffmeister, Hermann Brenner, and Wolfgang Koenig. Physical Activity, Coronary Heart Disease, and Inflammatory Response. *JAMA Internal Medicine*, 163(10):1200–1205, 05 2003. ISSN 2168-6106. doi: 10.1001/archinte.163.10.1200. URL https://doi.org/10.1001/archinte.163.10.1200.

B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32, Sept 1994. ISSN 0890-8044. doi: 10.1109/65.313011.

Alex Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Press Syndicate of the University of Cambridge, 2008.

Zhang Suping, Meng Guanglin, Wang Yanwen, and Li Ji. Study of the relationships between weather conditions and the marathon race, and of meteorotropic effects on distance runners. *International Journal of Biometeorology*, 36(2):63–68, Jun 1992.

E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 37–40, Oct 2007. doi: 10.1109/ISWC.2007.4373774.

D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. An unsupervised approach for automatic activity recognition based on hidden markov model regression. *IEEE Transactions on Automation Science and Engineering*, 10(3):829–835, July 2013. ISSN 1545-5955. doi: 10.1109/TASE.2013.2256349.

Niall Twomey, Tom Diethe, Xenofon Fafoutis, Atis Elsts, Ryan McConville, Peter A. Flach, and Ian Craddock. A comprehensive study of activity recognition using accelerometers. *Informatics*, 5:27, 2018.

Zhelong Wang, Ming Jiang, Yaohua Hu, and Hongyi Li. An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16:691–699, 2012.

J. Wannenburg and R. Malekian. Physical activity recognition from smartphone accelerometer data for user context awareness sensing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(12):3142–3149, Dec 2017. ISSN 2168-2216. doi: 10.1109/TSMC.2016.2562509.

Mark Weiser. The computer for the 21 st century. *Scientific American*, 265(3):94–105, 1991. ISSN 00368733, 19467087. URL http://www.jstor.org/stable/24938718.

T. Yokotani and Y. Sasaki. Comparison with http and mqtt on required network resources for iot. In *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, pages 1–6, Sep. 2016. doi: 10.1109/ICCEREC.2016.7814989.

Jiexiu Zhao, Santiago Lorenzo, Nan An, Wenping Feng, Lili Lai, and Shuqiang Cui. Effects of heat and different humidity levels on aerobic and anaerobic exercise performance in athletes. *Journal of Exercise Science Fitness*, 11(1):35 – 41, 2013. ISSN 1728-869X. doi: https://doi.org/10.1016/j.jesf.2013.04.002. URL http://www.sciencedirect.com/science/article/pii/S1728869X13000087.

Zhen-Yu He and Lian-Wen Jin. Activity recognition from acceleration data using ar model representation and svm. In *2008 International Conference on Machine Learning and Cybernetics*, volume 4, pages 2245–2250, July 2008. doi: 10.1109/ICMLC.2008.4620779.

Óscar D. Lara, Alfredo J. Pérez, Miguel A. Labrador, and José D. Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, 8(5):717 – 729, 2012. ISSN 1574-1192. doi: https://doi.org/10.1016/j.pmcj.2011.06.004. URL http://www.sciencedirect.com/science/article/pii/S1574119211000794.