Universidade do Minho
Escola de Engenharia
Departamento de Eletrónica Industrial

Luís Filipe Beites Alpoim

Study and application of computational techniques to identify anatomical back postures

December 2019

Universidade do Minho
Escola de Engenharia
Departamento de Eletrónica Industrial

Luís Filipe Beites Alpoim

Study and application of computational techniques to identify anatomical back postures

Master dissertation
Master Degree in Biomedical Engineering

Dissertation supervised by
Professor Alexandre Ferreira da Silva

December 2019

Acknowledgments

All the work carried out and described in the present document, wouldn't have been possible without the help and support from many people. Thus, in the present section, I will name the people that most contributed, directly or indirectly, to improve my dissertation.

Firstly, I would like to thank Professor Alexandre Ferreira da Silva and Professor Cristina Santos for all the support as well as the opportunity to do this work and to integrate a dynamic research group, named *BIRDLAB*.

Also, I am very thankful to my girlfriend and my family for their comfort and continued support during the entire of the time that took me to finish this dissertation.

Last but not the least, I would like to thank to all my laboratory colleagues for always make the lab such a wonderful environment to work.

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.

Resumo

Na atualidade, o número de ocorrências de doenças ocupacionais tem vindo a ser cada vez maior, tanto em postos administrativos, quanto em postos de produção, sendo as doenças musculo-esqueléticas, umas das mais comuns e que afetam maioritariamente a coluna.

Face ao problema supramencionado, torna-se importante desenvolver estratégias que possibilitem o reconhecimento e previsão das posturas desempenhadas pelos trabalhadores nos seus locais de trabalho. Para tal, esta tese propõe uma *pipeline* que tem como finalidade retornar um modelo de *Machine Learning* que poderá conseguir reconhecer em tempo-real as posturas da parte superior do corpo dos utilizadores.

A *framework* acima descrita permite obter um modelo capaz de reconhecer 6 posturas estáticas diferentes (flexão frontal do tronco, flexão lateral esquerda do tronco, flexão lateral direita do tronco, flexão frontal do pescoço, trabalhar com mãos acima da cabeça e, por fim, a postura neutra/corretamente em pé). Para além destas posturas estáticas, o modelo é também capaz de reconhecer as transições entre a postura neutra e cada uma das outras cinco.

Vários algoritmos de seleção de *features* foram testados de forma a encontrar as caraterísticas biomecânicas que melhor permitem distinguir as diferentes posturas/classes. Os algoritmos testados foram as técnicas de *Principal Component Analysis, Forward Sequential Selection, mRMR (minimum Redundancy Maximum Relevance)* e *ReliefF.*

Seguidamente, vários algoritmos de classificação foram testados (*Support Vector Machines, K-nearest Neighbors, TreeBagger, Discriminant Analysis, Convolutional Neural Network* e *Feed Forward Neural Network*). Estes algoritmos foram treinados e testados com dados de 50 sujeitos saudáveis que se voluntariaram a adquirir dados das posturas acima mencionadas.

Os melhores resultados foram obtidos usando o classificador *SVM* com o kernel quadrático e usando as caraterísticas selecionadas pelo algoritmo *mRMR.* O modelo de classificação mostrou resultados promissores durante a validação cruzada, mais especificamente, apresentou um valor de *MCC* de 0.973.

Como trabalho futuro, deverão ser integrados outros tipos de sensores, assim como extrair diferentes *features* a partir dos sinais.

Palavras-Chave: *Machine Learning*, *Feature Selection*, Doenças Ocupacionais, Classificação de Posturas

Abstract

Currently, the incidence of occupational diseases has been increasing, both in administrative and in production posts, being the musculoskeletal diseases, one of the most common and affect, mainly, the spine.

Considering the above-mentioned problem, it is important to develop strategies that enable the recognition and prediction of postures performed by workers in their workplaces. To this end, this thesis proposes a pipeline that aims to build a Machine Learning model that may be capable of recognizing, in real time, the upper body postures of the users.

The above described framework provides a model capable of recognizing 6 different static postures (back frontal bending, left lateral back bending, right lateral back bending, neck frontal bending, working overhead, and finally the standing posture). In addition to these static postures, the model is also capable of recognizing the transitions between the neutral posture and each of the other five.

Several feature selection algorithms have been tested to find the biomechanical features that best distinguish the different postures. The algorithms tested were the Principal Component Analysis, the Forward Sequential Selection, the Minimum Redundancy Maximum Relevance (mRMR) and the ReliefF.

Also, several classification algorithms were tested (Support Vector Machines, K-nearest Neighbors, TreeBagger, Discriminant Analysis, Convolutional Neural Network and Feed Forward Neural Network). These algorithms were trained and tested with data from 50 healthy subjects who volunteered to acquire data realizing the above-mentioned postures.

The best results were obtained using the SVM classifier with the quadratic kernel and using the characteristics selected by the mRMR algorithm. The classification model showed promising results during cross-validation, more specifically, it presented an MCC value of 0.973.

As future work, other types of sensors should be integrated, and other features should be tested to improve posture classifier performance.

Keywords: Machine Learning, Feature Selection, Occupational Diseases, Posture Classification

INDEX

1. Introduction 1
1.1 Motivation 1
1.2 Problem Contextualization 4
1.3 Main Goals and Research Questions 5
1.4 Methodology 6
1.5 Document Structure 7
1.6 Contributions 8
2. Theoretical Concepts 9
2.1 Instrumented Systems 9
2.2 Machine Learning 10
2.3 Feature Engineering 11
2.4 Data Pre-Processing 11
2.4.1 Data Normalization 12
2.4.2 Dimensionality Reduction 13
2.5 Learning Algorithms 17
2.6 Classification Model Validation 21
3. State of Art 25
3.1 Selection of Attributes and Sensors 26
3.2 Pre-Processing 28
3.3 Segmentation 28
3.4 Feature Calculation 29
3.5 Dimensionality Reduction Techniques 31
3.6 Classification Algorithms 32
3.7 Recognition System Issues 41
4. Methodology 43
4.1 Case studies 43
4.2 System Design 44
4.3 Data Collection Protocol 46
4.4 Human Posture Recognition Pipeline 49
4.4.1 Data Filtering 50

4.4.2 Feature Calculation ... 51
4.4.3 Feature Data Normalization ... 52
4.4.4 Dimensionality Reduction ... 52
4.4.5 Data Labeling ... 53
4.4.6 Model Building ... 54
4.4.7 Model Evaluation ... 56
5. Procedure Tests ... 58
5.1 Feature Calculation ... 59
5.1.1 Methods ... 59
5.1.2 Results ... 59
5.1.3 Discussion ... 60
5.2 Data Normalization ... 60
5.2.1 Methods ... 61
5.2.2 Results ... 61
5.2.3 Discussion ... 63
5.3 Feature Selection or Extraction ... 64
5.3.1 Methods ... 64
5.3.2 Results ... 65
5.3.3 Discussion ... 65
5.4 Classification Algorithm Selection ... 66
5.4.1 Methods ... 67
5.4.2 Results ... 67
5.4.3 Discussion ... 68
6. Human Posture Recognition Model Validation ... 69
6.1 Methods ... 69
6.2 Results and Discussion ... 70
6.2.1 10 Repeated 20-Fold CV ... 70
6.2.2 Test Activity ... 74
7. Conclusions ... 77
7.1 Final Considerations ... 77
7.2 Future Work ... 79
References ... 80

Appendix.. 84

Abbreviations and Acronyms

ACC - Accuracy

AUC – Area Under Curve

CV – Cross Validation

DA – Discriminant Analysis

FN – False Negatives

FP – False Positives

IMU – Inertial Measurement Unit

KNN – K-Nearest Neighbors

LDA – Linear Discriminant Analysis

LOO – Leave One Out

MCC – Matthews Correlation Coefficient

mRMR – minimum Redundancy Maximum Relevancy

MSD - Musculoskeletal Disorders

PA – Parallel Analysis

RF – Random Forest

ROC – Region of Convergence

SVM – Support Vector Machines

TN – True Negatives

TP – True Positives

List of Figures

Figure 1 - Different steps of the outlined methodology. 7

Figure 2 - Example of a decision tree and the respective training set. 18

Figure 3 - Example of SVM model's hyperplane optimization (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).. 19

Figure 4 - Example of a feed-forward ANN. 20

Figure 5 - Example of a CNN architecture.. 21

Figure 6 - Main stages of the human activity and posture recognition process. 26

Figure 7 - Examples of risk postures, developed in the construction environments............................. 43

Figure 8 - Main bones of the spine (Peppoloni et al., 2016). 45

Figure 9 - Representation of the six human postures studies (Stand – Standing, BFB – Back Frontal Bending, LBLB – Left Back Lateral Bending, RBLB – Right Back Lateral Bending, WO – Working Overhead, NFB – Neck Frontal Bending). 47

Figure 10 - Example of the data collection setup... 48

Figure 11 - Data collection protocol, followed by the subjects.. 49

Figure 12 - Pipeline flow chart.. 50

Figure 13 - CNN Architecture. 56

Figure 14 - Different blocks of the developed framework. 58

Figure 15 - Recognized posture versus the ground truth, for the test activity...................................... 75

List of Tables

Table 1 - Main conclusions of the above referred study (Da Costa & Vieira, 2010). ... 3
Table 2 - Example of a confusion matrix for a binary classifier ... 22
Table 3 - Most common used features. ... 30
Table 4 - Most important characteristics of the human motion recognition systems present in the literature. ... 34
Table 5 - Postures and motions to classify. ... 47
Table 6 - Main characteristics of the individuals that accepted to collect data. ... 48
Table 7 - Features extracted from the acceleration signals. ... 51
Table 8 - Different classes designation. ... 53
Table 9 - Average recognition model's performance with the different time windows. ... 60
Table 10 - Performance results of the SVM models, with the different normalization techniques. ... 61
Table 11 - Performance results of the TreeBagger model, with the different normalization techniques. 62
Table 12 - Performance results of the KNN models, with the different normalization techniques. ... 62
Table 13 - Performance results of the DA models, with the different normalization techniques. ... 62
Table 14 - Performance results of the implemented networks, with the different normalization techniques. ... 63
Table 15 - Best normalization technique, by learning algorithm. ... 64
Table 16 - Regular KNN model's performance, using the different dimensionality reduction techniques. ... 65
Table 17 - 10 repeated 2-fold CV results, for each learning algorithm. ... 67
Table 18 - Results and important parameters of the 10 repeated 20- fold CV. ... 70
Table 19 - Confusion matrix for the quadratic SVM model (St - Standing; BFB - Back Frontal Bending; NFB - Neck Frontal Bending; WO - Working Overhead; LBLB - Left Back Lateral Bending; RBLB - Right Back Lateral Bending) ... 72
Table 20 - Features extracted by the mRMR algorithm and the associated sensor's location. ... 73
Table 21 - Results of the test activity classification. ... 74

1. Introduction

The present work addresses the field of the human motion recognition and the main goal of this dissertation is to apply machine learning algorithms and obtain posture recognition models that will be applied to the upper-limb assistive devices. Additionally, these recognition models will be used to classify the worker's current upper-limb posture and posture transitions.

The major objective of the present chapter is to introduce and contextualize the work developed, and some challenges and goals of the present work.

1.1 Motivation

Nowadays, there is an increasing frequency of occupational diseases caused by many factors (such as poor posture, heavy physical work, overtime hours of work, low level of work satisfaction or high levels of distress), both at the administrative and manufacturing floor levels. Consequently, the aforementioned factors promote the propensity for a worker to suffer from some physical diseases, as the musculoskeletal disorders (MSD's).

In 1991, the European Foundation for the Improvement of Living and Working Conditions carried out the "First European Survey on the Work Environment", and it was used as a tool to help policy makers by providing a clearer picture of the working conditions in Europe. At the end of 1995 and the beginning of 1996, the Second Survey on Working Conditions was carried out on an extended basis, covering the main 15 countries with the addition of some other countries like Austria, Finland and Sweden. Regarding this second survey, Greece, Portugal and Spain presented a systematically higher proportion of workers exposed to many risk factors related to occupational diseases. Some examples are the noise, the vibrations, the extreme temperatures and the air pollution. In addition, Portugal appear as the second country, in which the percentages of operators with back pain was the highest (39%), only surpassed by the Greece (44%). Then, the same results were obtained with regard to headaches, since 24% and 21% of the Greece and Portugal workers, respectively, said to feel these symptoms (Paoli, 1996).

Later, on the fourth version of the same survey, published in 2007, it is important to note that the previous values have not improved significantly, since Portugal continues to present a higher value of workers that report backaches (30,7%), only surpassed by countries like Greece (47%) and Latvia (44,1%). With regard to headaches, the same scenario occurred, since the percentage of Portuguese workers that reported having headaches was around 23,9%, being one of the highest percentages of the 31 countries (A. Parent-Thirion, E. F. Macías, J. Hurley, 2007).

Concerned to the sixth and the most recent version of the related study, in 2015, it was concluded that the evolution of the physical environment indexes indicates improvements since 2005 for the majority of the European countries, except for France and United-Kingdom. It is also important to note that the most notable improvements were reported in Greece and Portugal (Agnès Parent-Thirion, Isabella Biletta, Jorge Cabrita, Oscar Vargas, Greet Vermeylen & Wilkens., 2017). This improvement can be justified by the specific legislation on the tasks involving the manual handling of loads and other demanding tasks.

Then, it is important to refer that the manual manipulation of loads, developed in occupational contexts, constitute one of the most frequent and riskier tasks, that could generate a WMSD (Work related Musculoskeletal Disorders). According to the National Research Council (NRC) and to the Institute of Medicine (IOM), the nature of the WMSD's is multifactorial, since other tasks realized in daily life, such as sports and housework, may have incorporated physical stresses to the musculoskeletal tissues. Additionally, it is also important to mention that, not everyone that develop a MSD was exposed to ergonomic risks at work, and not everyone exposed to ergonomic risks, developed a MSD (Da Costa & Vieira, 2010).

Regarding the Musculoskeletal Disorders (MSD), these include a wide range of inflammatory and degenerative conditions, affecting the tendons, ligaments, peripheral nerves, joints and supporting blood vessels. These disorders include clinical syndromes, such as tendon inflammations, nerve compression disorders (such as sciatica), osteoarthritis, and also standardized conditions like myalgia or low back pain. In addition, the body regions that are more involved are the low back, the neck, the shoulder, the forearms and the hands (Punnett & Wegman, 2004).

Therefore, the MSDs consist on one of the major problems in many countries, with substantial costs and impact on the worker's quality of life. Also, they constitute a higher proportion of all registered and compensable work-related diseases in many countries. It is also important to note that the MSDs represent the single largest category of work-related sickness, and the third or more of all registered occupational illnesses in the Nordic Countries, the United States and Japan. For example, in the United States, more than 600,000 employees reported having WSDSs. Consequently, the occurrence of these diseases results in many days away from work, every years (Da Costa & Vieira, 2010).

With regard to the high-risk sectors, these include nursing facilities, mining, food processing, leather tanning and, mainly, heavy and light manufacturing. Additionally, the most common risk factors, based on experimental investigations, includes repetitive motion patterns, non-neutral body-postures, insufficient recovery time, forceful manual activities, vibration (either segmental or whole-body), exposure to extreme temperatures, among others. Other suspect risk factors include smoking, obesity and lack of muscle strength (Punnett & Wegman, 2004). Nevertheless, it is important to note that the risk varies by age, ethnicity, gender, socio-economic status, weight, among many other aspects of work capacity.

An interesting work was developed by (Da Costa & Vieira, 2010), in which it was concluded, based on many other studies, the principal risk factors for the WMSDs, according to the different parts of the body (neck, low back, shoulder, wrist, elbow and others), being the main results presented in the Table 1.

Table 1 - Main conclusions of the above referred study (Da Costa & Vieira, 2010).

	Biomechanical Risk Factors	Psychosocial Risk Factors	Individual Risk Factors
Neck	Heavy physical work, awkward posture and frequent lifting	Low level of work satisfaction and support and high level of distress	Older age, female gender, sedentary lifestyle, smoking and co-morbidity
Low Back	Heavy physical work, awkward static and dynamic working postures and lifting	Negative affectivity, low level of job control and high work dissatisfaction	Younger age, female gender, black or African American race, smoking and co-morbidity
Shoulder	Heavy physical work and repetitive work	High levels of distress, performing monotonous work and low level of job control	Older age, sedentary life and high BMI (Body Mass Index)
Elbow/Forearm	Heavy physical work, awkward static and dynamic working postures, prolonged computer work and repetitive work	Negative affectivity, low level of job control and high work dissatisfaction	Older age, female gender, associate upper limb WMSD and high BMI
Wrist	Heavy physical work, repetitive work, prolonged computed work and awkward work postures	High level of distress	Older age, female gender, smoking, high BMI and co-morbidity

Summing up, the physical disorders are not the only type of the work-related diseases. Many etiologic researches have concluded that there is a strong relationship between workplace stressors and adverse health outcomes such as the cardiovascular diseases (CVD) and mental disorders (Belkic, K., & Savic, 2008). In this way, it had been talked about the "cognitive ergonomics", that intend to interpret how the workers process information, make decisions and carry out actions, for example.

1.2 Problem Contextualization

As it was previously mentioned, in some workplaces, the workers commonly develop incorrect postures, when performing their work tasks, being important to refer that the main diseases are associated to the upper body. In this way, the development of a system that could predict the human upper limb postures developed by the workers in their workplaces, is extremely important. The aforementioned system could provide important information, such as the time that the user developed the different high ergonomic risk postures or the number of consecutive risk postures developed by the workers. For such, a wide range of sensors could be used to extract information that allows the analysis of the ergonomic risk.. Therefore, many attributes can be measured, namely the environmental attributes, the dynamic attributes or the physiological attributes.

Then, the majority of the academic works that had, as the main objective the recognition of the different worker's postures, were based on the use of two or an array of many cameras (Gouiaa & Meunier, 2017; Juang, Member, & Wang, 2014; Wang, W. J., Chang, J. W., Haung, S. F., & Wang, 2016) or 3D body scanners (Werghi, N., & Xiao, 2002). In this context, the present dissertation intended to fill the gap in the literature, since the studied algorithms and techniques will be applied to data acquisition systems based on inertial sensors. These sensors are becoming increasingly important because they surpass the main disadvantage of the external sensors like the cameras, that is the lack of the individual's privacy and the need of a controlled environment. In addition, the inertial sensors processing technique are less computationally demanding, when compared to the cameras or the 3D scanners.

Since the different sensors will provide a huge data set (which could be constituted by biometric data, anthropometric data or biomechanical data, for example), it is important to use computational techniques that could process the entire data set in an automatized way. In this context, the human motion recognition systems, have an utmost importance in the prevention of the work related diseases since, these systems take, as input, a huge dataset of the measured attributes and, with the application of many computational algorithms, they are able to infer the postures or the tasks performed by the workers.

In a brief way, feature extraction and selection techniques are applied to the data coming from the sensors, selecting the characteristics that most contribute to distinguish the workers motion and postures (Avci & Bosch, 2010). Consequently, a *feature* vector is extracted from the raw data and, by the end, learning algorithms are used to generate a motion recognition model. Posteriorly, this model is used to recognize motions, from individuals that were not used in the training stage (D. Lara & Labrador, 2013).

1.3 Main Goals and Research Questions

In the present section, taking into account that the main goal of the present dissertation is the study and the application of different computational techniques to predict the worker's upper limb postures, some goals and research questions will be approached. Firstly, many characteristics of the systems have to be idealized, namely, the number of inertial sensors to be used and their locations on the body. Thus, the system should not be composed by many sensors, so as not to restrict the individual's movements and the system should be portable and inexpensive.

Therefore, it is also important to develop a research on the previous literature and to identify the shortcomings of the previous approaches and their solutions for the development of these systems and the computational algorithms.

After selecting all the characteristics related to the system design, it is important to select the computational algorithms to test and compare. With the research developed, it is possible to point out some research questions, which are expected to be answered with the developed work.

- **Research Question 1: Which are the best wearable sensor's locations to better distinguish the different human upper limb postures?** The selection of the sensor's location on the subject's body is a key factor to provide the best upper body posture classification performance. This importance could be explained due to the fact that the different upper body postures and movements are more felt in some specific locations. In addition, it is also important to have the minimum number of sensors that allow the correct posture classification, so as not to restrict the movements of the person.

- **Research Question 2: Which is the best data normalization method?** Some learning algorithms are sensitive to the data normalization technique, being important to select, for each learning algorithm tested, the respective best data normalization techniques.

- **Research Question 3: Which are the most promising dimensionality reduction algorithms?** The selection of the best dimensionality reduction technique is a key stage of the present dissertation due to the fact that, using these techniques, it is possible to obtain a better classification performance, using a less number of features/characteristics. Also, the training and the testing phases will be less time consuming and computationally demanding.

- **Research Question 4: Which is the best learning algorithm, with respect to the human upper limb postures classification?** Depending on the different characteristics of the features extracted from the sensor's signals, some learning algorithms could provide a better posture classification performance.

- **Research Question 5: Which is the best metric to compare the performance of the different classification models?** Some metrics do not describe exactly the fitness of a specific learning algorithm. Specifying, there are some metrics that do not provide good feedback about the classification performance, when the data are unbalanced, that is, when some classes/labels have less number of observations associated.

1.4 Methodology

As previously referred, the main objective of the present dissertation is to apply computational techniques in order to predict the worker's upper limb postures. Specifying, in order to construct the posture recognition models, the users wore some sensors, namely, IMUs (Inertial Measurement Units) and the recognition models learn, through the data provided by the sensors, how to distinguish the different human upper-limb postures.

With respect to the outlined methodology, a machine learning pipeline was developed, being an important tool with respect to the construction of the best upper limb posture classification model.

To accomplish such tool, the following goals were set, being next described:

1. Literature review related with the human motion recognition systems, with the objective to understand which are the most used and promising algorithms.
2. Development of a data collection protocol, that a high number of a healthy individuals with different physical characteristics should follow.
3. Calculation of different subsets of features, extracted from the sensor's signals.
4. Testing different data normalization algorithms, with the objective of improving the classification.
5. Implementation and the test of dimensionality reduction algorithms, in order to understand which are most important features.
6. Testing many different learning algorithms.

At the end, after testing and comparing all the different algorithms, the best upper body posture classification model will be applied to an activity that include the different postures selected to be analyzed. Next, Figure 1 illustrates the methodology steps followed, previously discussed.

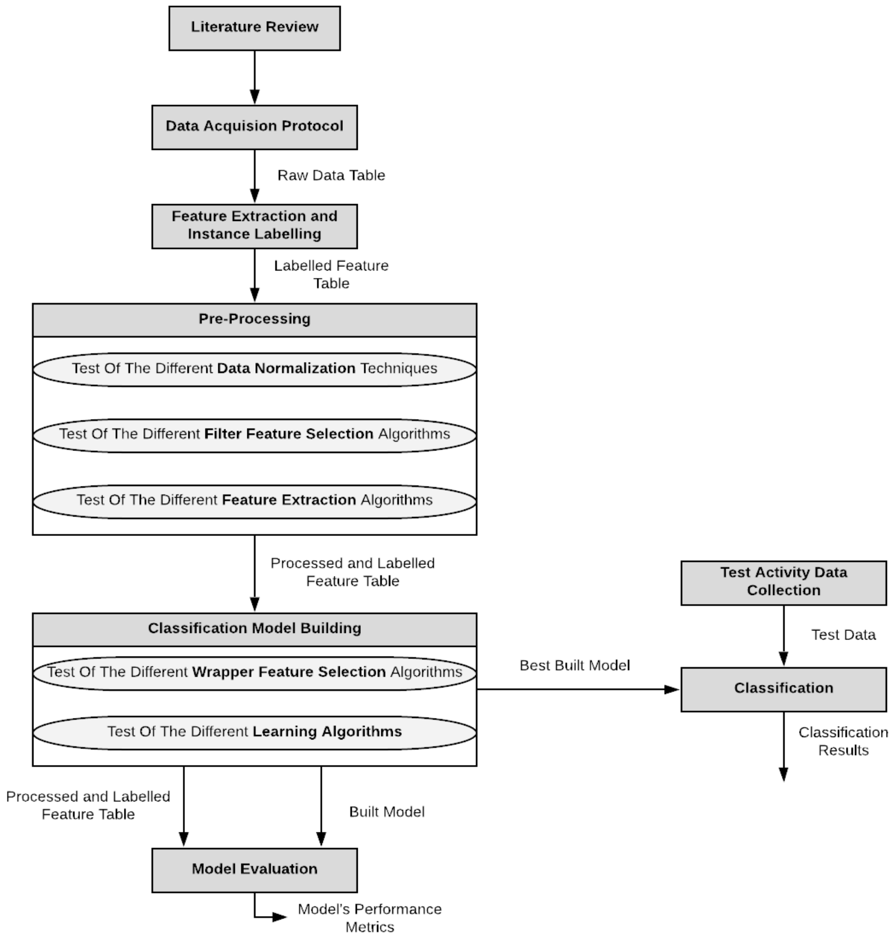


Figure 1 - Different steps of the outlined methodology.

1.5 Document Structure

Chapter 2 provides the reader with important definitions for the concepts required to understand the procedures described in the present dissertation.

Then, in the Chapter 3, information about the related studies developed by other researchers are approached and, in the Chapter 4, a detailed description about the methodology applied in the context of this study is presented.

In its turn, Chapter 5 approaches the tests performed using the different algorithms of data segmentation, data normalization, feature extraction and feature selection, and classification. Chapter 5 intends to find the combination of the algorithms that provides the best classification performance. In addition, it includes the discussion of the obtained results for each one of the stages.

Chapter 6 describes the performance results for the developed posture classification model, with regard for both the 20-fold repeated cross validation and activity test prediction.

Lastly, Chapter 7 reports the conclusions drawn from this work and some future challenges.

1.6 Contributions

The present dissertation contributes to the state of art of the machine learning techniques used for the prediction/classification of the individuals' specific postures and motions, more specifically, the upper limb postures. Firstly, with the work developed, it is possible to refer what are the body locations that provide the best information about the subjects' upper body postures, as well as, the characteristics extracted from the acceleration signal, that most contribute to distinguish the different postures.

Therefore, it is concluded based on several tests, which data normalization technique provides the best posture recognition performance, for each one of the different learning algorithms implemented. In addition, the present study contributes to the state of art of the different dimensionality reduction techniques, as well as, of the different learning algorithms. Among others, a comparison of the different performance metrics and validation approaches are reported. Summing up, this work contributes to the literature related to the computational techniques used to predict the human's upper body postures, using inertial sensors. As it was previously referred, the works using inertial sensors are less referred in the literature, when comparing with the external sensors like the cameras.

Concluding, based on the research developed, an article, named "Human Activity Recognition Systems: State of Art", was published on the 6th IEEE Portuguese Meeting in Bioengineering (Alpoim, L., da Silva, A. F., & Santos, 2019).

2. Theoretical Concepts

The present chapter contextualize the problem and provides the necessary theoretical concepts and definitions that allow a better understanding of the information described over the present document. In this context, many different definitions of data analysis or processing, and of machine learning will be explained, being all important for the development of the methodology followed in the present dissertation.

2.1 Instrumented Systems

In the first place, the development of an instrumented system that intelligently characterize the user's current posture and motion is of utmost importance. Thus, the motion recognition is very important in many computing applications, ranging from health care to rehabilitation (Paul, Dario, & Walls, 2012). There are many devices that can be used to construct a human motion recognition system, however the present dissertation will only focus on the use of inertial sensors.

Inertial sensor-based motion recognition integrates the emerging area of sensor networks, and focuses on other areas such as data mining and machine learning, in order to generate a model that characterize the different human motions and postures. More specifically, these systems are composed by a set of sensors (that could be accelerometers, gyroscopes and compasses, for example) attached to different points of interest on the individual's body. Thus, these devices enable a variety of applications like rehabilitation, sports medicine, geriatric care or health monitoring.

To sum up, in the present dissertation, many different computational approaches will be tested to obtain the best human posture recognition model that target the prediction of the posture developed by the workers. In addition, this model could be applied to the instrumented systems to predict, in real-time, the posture developed by the workers, and obtaining live information about the ergonomic risk that the workers are exposed.

2.2 Machine Learning

Machine Learning (ML) consists on the process that automatically learns from the past studies or experience, and acts without being explicitly programmed for. Machine Learning analyses the data to find patterns and construct models automatically, reliably and cost-effectively. There are four main categories in the ML field, namely, (i) the supervised learning, (ii) the unsupervised learning, (iii) the reinforcement learning and (iv) the semi-supervised learning (Kumar, Amgoth, Sekhara, & Annavarapu, 2019).

Firstly, supervised learning is one of the most important ML approaches. The data are composed by one or more predictor variables and followed by a response variable, generally named as label or class. In addition, during the training stage, the supervised learning algorithm finds the relationship between the predictor variables and the respective classes. Briefly, the supervised learning can be divided into regression and classification, being the second one the most promising for the present application. With regard to the classification goal, it can be divided into logic-based (Decision Tree and Random Forest algorithms), perceptron based (Artificial Neural Network – ANN, for example), statistical learning (Bayesian Networks or Support Vector Machines – SVM) and instance-based (*k*NN, for example) algorithms (Kumar et al., 2019).

With regard to the unsupervised learning approaches, there is no label/class associated at each example/input, being that, the model tries to extract the relationships between the data. The major contributions of the present approaches are to tackle various issues such as the connectivity problem, anomaly detection, routing and data aggregation. Summing up, the most well-known unsupervised learning approaches are the clustering techniques such as the *k*-means and the fuzzy-c-means.

Then, since the real-world application's data is the combination of labeled and unlabeled data, a semi-supervised learning approach emerged. This approach encompasses some applications like semi-supervised classification to perform classification on labelled data, constrained clustering to perform clustering with both labelled and unlabeled data, regression with unlabeled data and dimensionality reduction for labelled data. Briefly, semi-supervised learning can be divided into two main groups: *Transductive* learning and *Inductive* semi-supervised learning, being the first used to predict the exact labels for a given unlabeled dataset, whereas the second one, learns a function that is expected to be a good predictor on future data (Kumar et al., 2019).

Lastly, the reinforcement learning algorithm continuously learns trough the interaction with the environment and gathers information to take certain actions. This means that the system adapts to different circumstances by receiving inputs and performing actions that maximize its inherent reward.

2.3 Feature Engineering

The features extracted from the sensors can be characterized as one of the following three categories: (i) Relevant, (ii) Irrelevant and (iii) Redundant. In a brief way, the relevant features have an important influence on the output, whereas the irrelevant features consist of characteristics that do not have any influence on the output. In its turn, the redundant features are features that can take the role of another and, summing up, it is equal for the system performance using or not using them.

Based on the aforementioned, it is easily perceptible that the use of the irrelevant and redundant features could make the process of model building extremely complex and computationally demanding. Therefore, it is important to select and to retain the relevant features, while eliminating both the irrelevant and redundant features (Blum, A. L., & Langley, 1997).

Because of the above mentioned, a good feature engineering is probably the best tool to obtain a small subset of features that can be used to create a more accurate and simpler model. With respect to the motion recognition systems, the features extracted are generally quantitative characteristics from the time-domain of the acceleration and angular velocity signals, being important to note that these features could be highly correlated and redundant.

Summing up, it is usually that the feature selection occurs on the pre-processing stage, that will be better approached in the next section.

2.4 Data Pre-Processing

Posteriorly to extracting features from the raw data, it is important to apply some pre-processing techniques before using these characteristics in a machine learning algorithm.

With respect to the data pre-processing, there are many phases namely, the instance selection and outlier detection, the discretization, the data normalization, the dimensionality reduction (that includes the feature selection and the feature extraction techniques). Then, in the next sections, the most important phases of the data pre-processing are described.

2.4.1 Data Normalization

Data normalization consists of a "scaling down" transformation of the features, bringing the variables to a common scale, so they can be more fairly compared. Within a feature/variable there is often a large difference between the maximum and the minimum values. Thus, applying a normalization technique, the variable's values are scaled to appreciably low values (S. B. Kotsiantis, 2006).

Before the classification model building, the normalization is usually performed on the used data since many of these models are sensitive to features with very wide range, namely, many neural networks, the k-Nearest Neighbors algorithm or the SVM learning algorithm.

There are several normalization methods and the main four techniques will be next described.

Firstly, the centering technique consists in subtracting the mean of the feature vector from each value, making the average null and the data centered at zero. The aforementioned technique is represented in the Equation 1, where the x is the original value, μ is the feature vector mean and x' is the normalized value.

$$x' = x - \mu \tag{1}$$

Scaling is another well-known normalization method, that involves dividing the data by its standard deviation, turning each value, the deviation from the mean. The formula is next presented in the Equation 2, where the x is the original value, the σ is the standard deviation value of the feature/variable vector and the x' is the normalized value.

$$x' = \frac{x}{\sigma} \tag{2}$$

Then, Z-score technique is probably the most used normalization and consists on applying the centering method, followed by the scaling method. The present technique is very useful in algorithms that compare similarities between the variables based on distance measures (namely the SVM and the KNN). The Z-score formula is represented in the Equation 3, where x is the original value, μ is the mean value of the feature vector, σ is the standard deviation of the feature vector and the x' is the normalized value.

$$x' = \frac{x - \mu}{\sigma} \tag{3}$$

Finally, the min-max normalization technique is usually used for learning algorithms that only work with the data on a certain range of values, like the neural networks. The formula is represented in the Equation 4, where x is the original value, *min(x)* is the feature vector's minimum value, *max(x)* is the feature vector's maximum value and x' is the normalized value.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4}$$

2.4.2 Dimensionality Reduction

High dimensional data is a common problem when it is intended to use a learning algorithm, due to the high computational cost and memory usage (Khalid, S., Khalil, T., & Nasreen, 2015). The most common goals of these techniques is to reduce the number of features to be used, preventing the overfitting, improving the generalization of the models and providing a better understanding of the features and of their relationship with the response variable (also known as class or label). There are two dimensionality reduction groups, known as feature extraction and feature selection. Next, these two groups will be better approached.

- **Feature Extraction**

Feature extraction algorithms generally perform some transformation to the original features to generate other features that are more relevant. In a brief way, these algorithms try to find the optimal linear or non-linear combination of the original features in order to reduce the dimensionality of the features (Khalid, S., Khalil, T., & Nasreen, 2015).

One of the most common feature extraction technique is the Principal Component Analysis (PCA). PCA consists of a mathematical procedure that uses an orthogonal transformation to convert a set of possibly correlated observations into a set of linearly uncorrelated variables, called principal components. In addition, this transformation has the peculiarity that the first principal component has the largest possible variance, that is, it is responsible for the maximum variability of the data. Therefore, each next component, in its turn, has the maximum variance, with the restriction of being orthogonal to the previous components (Jolliffe, 2011). After the application of the PCA, the top components need to be selected based on one of the several different criteria. Some of these criteria will be next described.

Firstly, *Horn's Parallel Analysis* (PA) is one of the most used cut criteria, consisting on a Monte Carlo simulation process that compares the observed eigenvalues extracted from the correlation matrix with those obtained from uncorrelated normal variables. Then, a factor or component is retained only if the associated eigenvalue is bigger than the 95th percentile of the distribution of eigenvalues derived from the random data (Ledesma, R. D., & Valero-Mora, 2007). Like all Monte Carlo methods, the PA has the disadvantage of requiring an intensive computational process and, in addition, the PA criteria is sensitive to sample size in such way that, for large samples, the eigenvalues of random factors will be very small.

Another important criterion is the *Variance Explained* and consists of keeping the components to account for 90% of the total variation (Foerster, Smeja, & Fahrenberg, 1999).

Keiser criterion is another criterion being that it drops all components that have an eigenvalue less than 1.0 (corresponding to the eigenvalue equal to the information accounted by an average simple item). Despite of the high applicability of the aforementioned criterion, it is not advisable to use as the single cut-off method for the estimation of the number of components to retain, because it tends to over extract factors.

Concluding, it is important to note that the PCA is important in many machine learning applications. However, its application becomes unnecessary when the largest variance of the data does not represent the most important information to distinguish the different outputs/labels (Ledesma, R. D., & Valero-Mora, 2007).

- **Feature Selection**

High dimensional data consists on characteristics that can be relevant, irrelevant, misleading or redundant, which leads to the need of applying feature selection algorithms to rank the different features in order of their importance (Khalid, S., Khalil, T., & Nasreen, 2015). With respect to the aforementioned algorithms, there are three different families of these algorithms, namely, the *Filter*, the *Wrapper* and the *Embedded/Hybrid* methods.

Firstly, *Filter* methods use techniques that rank the total set of features in order of their weight/importance in distinguishing the different labels. Generally, the aforementioned methods use statistical tests like correlations tests that prevent the overfitting and are not computationally demanding. On the other hand, these algorithms have the disadvantage of selecting features that are redundant and, instead of the *Wrapper* methods, they don't use the classification model, which could lead to a worst performance (Chandrashekar, G., & Sahin, 2014). There are several filter feature selection algorithms and some will be next approached.

One of the most well-known method is the ANOVA (Analysis of Variance). This method provides a statistical test that indicates if the means of the different groups are significantly different or not. Regarding the classification context, the ANOVA is used to compare the total set of samples from each different class within a feature vector, and to verify if they are significantly different or not. The present method is better when it is intended to analyze more than 2 groups/labels, when compared with, for example, the *t-test*, that is less computationally demanding but is only useful if the number of labels is two.

The *Welch's t-test* consists on another *filter* method, being very similar to the common *Student's t-test* but with the peculiarity of being more useful to compare a variable in two different groups, in order to infer if their means are different or not. Summing up, this method is only applied when the test statistics follow a normal distribution and the scaling value in the test statistics is known. Another important test is the *Pearson's Correlation (r)* and its main objective is quantifying the linear dependence between two different variables/features, being that, the linear dependence value ranges from -1 to +1 (Chandrashekar, G., & Sahin, 2014). To sum up, the aforementioned test is not advisable when the relationships between the variables are not linear.

ReliefF algorithm is another filter approach that is mostly used to select the best subset of features. Basically, this algorithm estimates the importance of the different features analyzing the way that their values distinguish between instances that are near to each other. Firstly, *ReliefF* randomly selects an instance *i* from the class *C* and searches for *K* of its nearest neighbors from the same class (known as nearest hits – *H*) and for *K* of its nearest neighbors from the different classes (known as nearest misses – *M*). Then, the algorithm estimates the quality of the feature (*W*) by the following procedure. Briefly, if the example *i* and those in *M* have different values, the quality of the feature (*W*) is increased and, on the other hand, if the example *i* and those in *H* have different values, the quality of the feature (*W*) is decreased. Summing up, the process is repeated *n* times, value that is specified by the user. The present method has the disadvantage of does not reducing the redundancy of the features. Exemplifying, using the aforementioned algorithm, the first three features selected (most important) could be redundant and using the first one could possibly provide the same performance as using the first three (Chandrashekar, G., & Sahin, 2014).

Lastly, the mRMR *(*minimum Redundancy Maximum-Relevancy*)* is probably the most important *filter* method, capable of providing the best performance. This algorithm is similar to the ReliefF algorithm, with the advantage of selecting only the features that have the minimum redundancy. For such, mRMR algorithm is based on the principle of mutual information (MI) between the different features, selecting only the most important features that have the minimum redundancy (Ding, C., & Peng, 2005).

Summing up, the filter methods could provide good classification performances, being important to note that the mRMR algorithm is known to be the best method to rank the different features.

With regard to the *Wrapper* methods, they use the classification model as a black box and the classification model performance to evaluate the different subsets of features, being that the optimal subset is found by employing search algorithms, which work heuristically (Chandrashekar, G., & Sahin, 2014). Due to the fact that these methods use the classifier to test the different subset of features, they could lead to better performance, however they are more computationally demanding, when compared with the majority of the *filter* approaches. The *Wrapper* algorithms can be segregated into different groups, the Sequential Selection Algorithms and the Heuristic Search Algorithms. While the Sequential Search algorithms start with a list empty or full (in agreement if the type is forward backward) and add/remove features until the maximum objective function is reached, the heuristic search algorithms evaluate different subsets to optimize the objective function. The main disadvantage of the *wrapper* methods is their computational weight.

Firstly, with regard to the sequential selection algorithms, the forward SFS (sequential feature selection) and the backward SFS are probably the most well-known *Wrapper* methods. That said, in the forward SFS, the algorithm starts with an empty list of features and ads one feature each iteration, until the maximum value for the objective function is reached (Chandrashekar, G., & Sahin, 2014). The backward SFS is very similar to the forward SFS, with the difference that the algorithm starts with a full set of features and eliminates one feature at each iteration, until the maximum value for the objective function is obtained.

With respect to the heuristic methods, the GA (Genetic Algorithm) is probably the most well-known, being very similar to the brute-force search technique in the sense that they test several subsets of features obtained randomly and build models with each of those subset. Initially, this algorithm constructs a number (introduced by the user) of random subsets of features (population), where each subset (each one is a genome) is a binary vector that indicates if a feature is selected or not, and each one of the subsets is tested and the associated model's prediction accuracy is saved. Then, only the *n* best features (*n* is also introduced by the used) are passed to the following iteration and the selected genomes are used to generate the new population through mutations (switching randomly the values of the genome to 0 or 1). The algorithm stops when the maximum number of iterations (introduced by the user) is reached or if too many generations with the same best results pass (Chandrashekar, G., & Sahin, 2014).

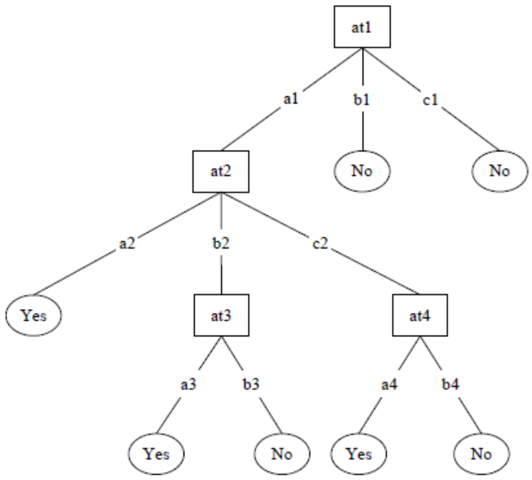
Lastly, regarding the *embedded* approaches, these intend to reduce the computation time for classifying different subsets of features, when comparing with the wrapper methods. The main approach is to incorporate the feature selection as part of the training process and try to combine the best parts of the *wrapper* and the *filter* approaches. Summing up, the most common type of these methods are regularization techniques that work by penalizing large coefficients in a model making it easier and simpler and decreasing the influence that some features have in the classification model (Chandrashekar, G., & Sahin, 2014).

2.5 Learning Algorithms

As described in the section 2.2, the Machine Learning approaches can take the supervised or the unsupervised types, depending if they take, as input, labeled data with a class representing the respective example or unlabeled data, respectively.

The Machine Learning is an emerging field that has been mostly used and improved in the last years. Next, only some supervised learning algorithms will be described due to their higher applicability when compared with the unsupervised learning algorithms. In a brief way, the supervised learning algorithms can be segregated into different families, namely, the logic based learning algorithms (such as, the decision trees), the perceptron based learning algorithms (such as, the artificial neural networks), the statistical learning algorithms (such as, the Bayesian Networks), the instance based learning algorithms (such as, the K-Nearest Neighbours) or the Support Vector Machines. Next, some of these will be approached.

Firstly, as above referred, the logic-based learning algorithm are generally decision trees being that, the C4.5 algorithm, is the most used decision tree algorithm. In a brief way, decision trees are trees that classify the different instances of data, by sorting them based on the feature's values. These trees are constituted by nodes and branches, being that each node represents an instance to be classified, and each branch represents a value that the node can take. With respect to the classification idealization, the different instances are classified starting at the root node and sorted based on their features values (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007). To better understand the way how the rules for tree construction are created, in the Figure 2, it is represented one decision tree example and the respective training set (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).



at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

Figure 2 - Example of a decision tree and the respective training set.

Due to the fact that individual decision trees tend to overfitting, it is common to apply Bootstrap-aggregated (bagged) decision trees. These algorithms combine the results of many decision trees, reducing the effect of overfitting and improving the model's generalization. For example, the TreeBagger algorithm, present in MATLAB software, grows the decision trees in the ensemble using bootstrap samples of data, and selects a subset of predictors to use at each decision split as in the random forest algorithm. To sum up, the number of trees consists of parameters inserted by the user.

With regard to the instance-based algorithms, the K-Nearest Neighbors (KNN) is the most used and simplest classification method of the present family. KNN learning algorithm is known to require less computational time during the training stage, when compared with other learning algorithms like the ANN, the Bayesian Networks or the Decision Trees, for example. On the other hand, they require more computational time during the testing stage (Aha, D. W., Kibler, D., & Albert, 1991). KNN is based on the principle that the instances will, mostly, exist in close proximity to other instances that have similar characteristics. In a brief way, this classification algorithm, locates the *k* nearest instances (neighbors) to the asking instance and determines its class/label by identifying the most frequent class (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

Also, it is possible to apply Weighted KNN being that, further the distance between an instance and its neighbor, less influence this neighbor has on the classification decision. Some authors defend that the weighted KNN is more robust and provide better results, when compared with the Regular KNN. Summing up, several distance metrics can be used, namely, the Euclidean metric (most used and, generally, the default), City block metric, Chebychev metric or Minkowski metric.

Therefore, the Support Vector Machine (SVM), is probably the most used learning algorithm. Basically, SVM algorithm intends to maximize the "margin" of a hyperplane that separates two classes.

In other words, the SVM classifier intends to create the largest possible distance between the hyperplane and the instances on either side of it (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

SVM can be tested with different kernels, namely, the linear kernel, the polynomial kernel (quadratic, cubic, for example) or the Gaussian kernel. Then, two important hyperparameters of the present algorithm are the Box Constraint (*c*) and the Kernel Scale, that is a specific parameter that only affects the performance of the Gaussian kernel. Briefly, *c* works a regularization parameter which signify that a larger *c* provides a model with lower bias but with a higher variance and the probability of the overfitting occurs is higher. On the other hand, if *c* is small, it is created a model with higher variance, lower variance and the probability of the underfitting occurs is higher. In its turn, the Kernel Scale parameter controls the way how the features smoothly vary being that, for a large sigma, the features will vary more smoothly, meaning higher bias and lower variance.

Summing up, the SVM was originally developed for binary classification. Consequently, for the multi-class applications, it is possible to use the "one-vs-one" or the "one-vs-all" techniques. In the first one, one model is created for each combination of two classes while, in the second one, a model is created for every class where the opposite class is composed by all the other classes joined together. An example of the hyperplane that an SVM algorithm tries to optimize is represented in the Figure 3 (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

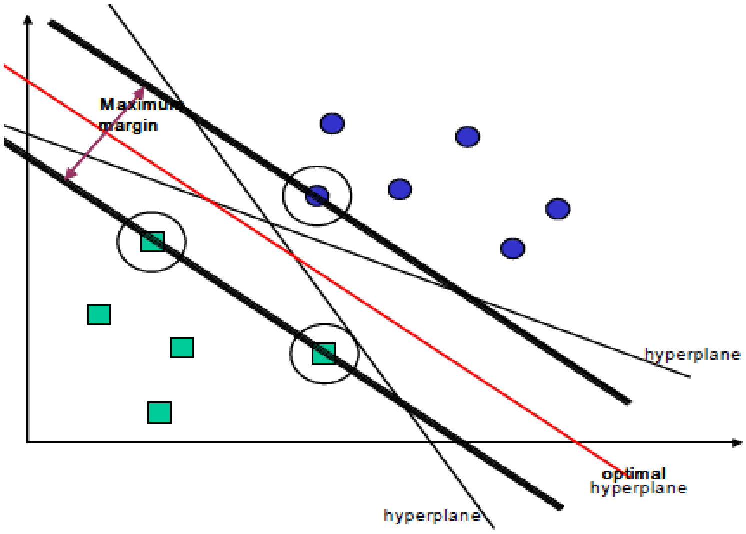


Figure 3 - Example of SVM model's hyperplane optimization (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

Linear Discriminant Analysis (LDA) and Fisher-LDA are also mostly used in some Machine Learning applications. These methods are simple statistical learning algorithms that intend to find the linear combination of features that best separate two or more classes. In a brief way, LDA only works when the features calculated for each observation are continuous variables. When working with categorical values, the equivalent technique is the Discriminant Correspondence Analysis (DCA).

Finally, the perceptron-based classification algorithms, are another alternative learning algorithm, being the multilayer perceptron (MLP) one of the most used. MLP are perceptron that have more than one layer of weights, which enables them to produce complex decision boundaries (Dybowski, R., & Gant, 2001). MLP with sigmoidal hidden node functions are the most commonly used Artificial Neural Networks (ANN). ANN consists of a large number of neurons joined together in a pattern recognition. Briefly, the network architecture includes three physical components: set of neurons, binding links (where each binding has an assigned weight), and processing elements (neurons) being that, the combination of these three, creates the neural network. Therefore, in the Figure 4, an example of a feed-forward ANN (information travels one way only, from input to output) is shown.

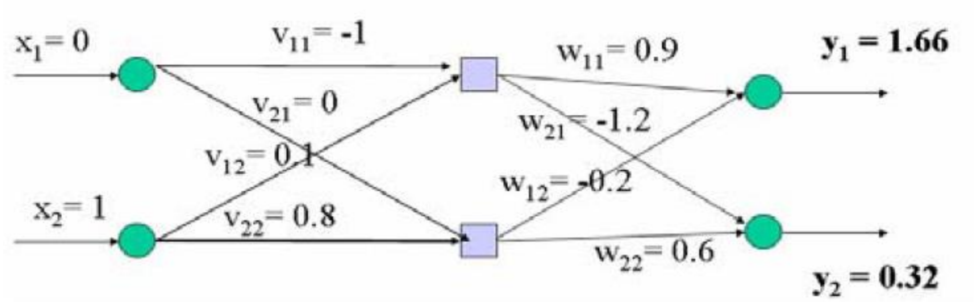


Figure 4 - Example of a feed-forward ANN.

In the last years, also the CNNs (Convolutional Neural Networks) have been used for many different Machine Learning applications. Briefly, CNNs intend to introduce a degree of locality in the patterns matched in the input data and to enable translational invariance, regarding the precision location of each pattern within a frame of movement data. With respect to the CNN architecture, each one contains at least one convolutional layer, followed by a pooling layer and at least a fully connected layer (with equal number of neurons to the number of different classes) prior to a top-level softmax-group. More specifically, each temporal convolution layer corresponds to a convolution of the input with n_f different kernels (feature maps) of width k_w. Then, the max pooling layer searches for the maximum within the specific regions of width m_w and corresponds to a sub-sampling. In addition, the previous mentioned layer introduces translational invariance to the system (Hammerla, N. Y., Halloran, S., & Plötz, 2016). Then, in the Figure 5, an example of a CNN is represented (Um, T. T., Babakeshizadeh, V., & Kulić, 2017).

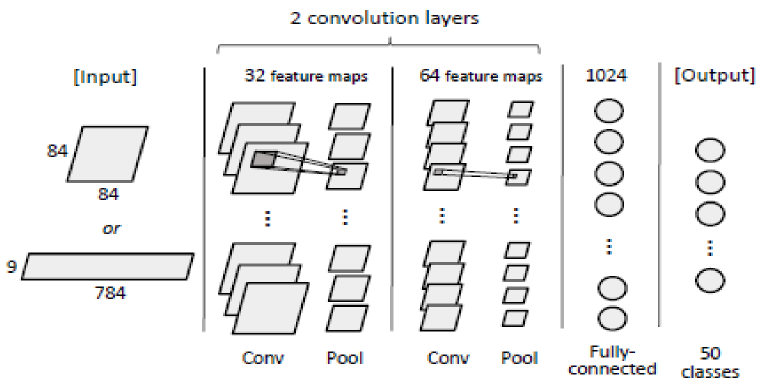


Figure 5 - Example of a CNN architecture.

Concluding, about Machine Learning field, it is important to note that the most widely known and used algorithms are the SVM and the KNN, both with advantages and disadvantages. Firstly, SVM models are complex and requires more computational resources for both the training and the testing stages but are much more embracing when compared with KNN learning algorithm. In its turn, KNN models are simple and easily interpretable, requiring less computational resources than the SVM algorithm.

2.6 Classification Model Validation

The choice of which classifier would be used is a critical step in every Machine Learning application. After a model being built, it is necessary to test its generalization performance. For such, the model is tested, predicting data that were not used (unseen data) to train the same, being this process, generally, name as model validation (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

- Validation Methods

Three of the most widely used validation techniques that will be next explained are (1) the Holdout validation, (2) the k-fold cross-validation and (3) the leave-one-out validation.

Firstly, with respect to the Holdout validation, the total set of data is divided into two different groups, the training dataset and the testing dataset, according to the percentages inserted by the user. Generally, it is advisable to select 70% of the data for the training stage and the remaining 30% to the validation stage. If it is intended to achieve more robust results, is advisable to use the repeated Holdout validation that consists in repeat the aforementioned process *n* times and average the results. However, the main disadvantage of this validation is still present since, repeating this process, some repeated observations will be selected to the testing stage while others will never be selected. Consequently, the errors made on those repeated observations have a higher impact on the total validation error (Arlot, S., & Celisse, 2010).

In its turn, the k-fold cross validation is the most used and recommend technique to compare the different classification models. This method consists in dividing the total dataset into *n* mutually exclusive and equal-size subsets, being *n*, a parameter defined by the user. Then, for each subset *n*, the model is tested with the subset *n* and trained with the union of the other $n - 1$ subsets. The average of the error for each one of the iterations consists of an estimate of the error rate of the classification model (D. Lara & Labrador, 2013). According to (Zhang, Y., & Yang, 2015), for predictive performance estimation, repeated 50-fold or 20-fold cross validation are best but, for model selection, repeated 2-fold cross validation is best. Summing up, it is suggested that the minimum number of repetitions should be between 10 and 20 times (Zhang, Y., & Yang, 2015). The repeated cross validation techniques are advantageous because in each cross-validation repetition, the data are randomized in a different way and, consequently, the observations used for the training and testing are different and the classification algorithm is evaluated in different scenarios.

Lastly, the leave-one-out validation is a special case of the cross validation. In this method, all test subsets represent a single instance. This makes the present validation more computationally demanding but more useful when it is intended to have a more accurate estimate of the classification model's error rate (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

- **Performance Metrics**

There are several performance metrics used to evaluate the fitness of a specific mode, being important to note that the majority of them are obtained from the confusion matrix. The confusion matrix is a quadratic matrix of order *n*, being *n*, the number of classes/labels. Usually, the rows of the matrix represent the true classes, while the columns represent the predicted classes. However, the confusion matrix could also be programed to have the predicted classes in the rows and the true classes in the columns. In the Table 2, one example of a confusion matrix for a binary classifier is presented.

Table 2 - Example of a confusion matrix for a binary classifier.

	Predicted: NO	Predicted: YES
Actual: NO	TN	FP
Actual: YES	FN	TP

Analyzing the Table 2, it is possible to understand all the designations of the matrix, that are used to calculate some performance metrics, namely, the true positives (TP), the true negatives (TN), the false positives (FP) and the false negatives (FN).

Some of the most commonly used metrics are the Accuracy, the Precision, the Specificity, the Sensitivity, the F1-score, the Matthews Correlation Coefficient (MCC) and the Area Under the Curve (AUC). It is important to note that, with the exception of the AUC metric, all of the other metrics are calculated only using the four designations obtained through the confusion matrix (TP, TN, FP and FN). Next, these metrics will be described and compared.

Firstly, the Accuracy metric is probably the most used and well-known metric, consisting in the percentage of the correct classified feature vectors. However, the present metric should not be used to evaluate the performance of a specific model when the different classes have different number of observations, that is, when the dataset is unbalanced. The Accuracy formula is described in Equation 5.

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

The Precision metric represent the ratio between the true positives and all positives predictions, both true or false and, its formula is presented in Equation 6.

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

Sensitivity is the proportion of positive cases that are correctly classified as such. It is usually named as the true positive rate and its formula is expressed in Equation 7.

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

In its turn, Specificity is the proportion of negative cases that are correctly classified as such and it is usually name as the true negative rate. Its formula is represented in Figure 8.

$$Specificity = \frac{TN}{TN+ FP} \tag{8}$$

F1-score metric consists of the harmonic mean of the precision and the sensitivity. Its main drawback consists in the fact that the metric ignores the true negatives and its formula is described in the Equation 9.

$$F1\ score = \frac{2 * TP}{2*TP+FP+FN} \tag{9}$$

Another important metric is the Matthews Correlation Coefficient (MCC). It is a correlation-based metric and it is known to be the best metric when evaluating classification models in which the number of observations in the different classes are not equal (unbalanced data). Other metrics, in the cases of unbalanced data, could provide improper values. The MCC formula is expressed in Equation 10.

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{10}$$

Finally, area under the curve (AUC) is another metric used to evaluate the performance of a classification model. The ROC space is defined by the true positive rate (benefits, plotted in the y axis) and by the false positive rate (costs, plotted in the x axis).

3. State of Art

Taking into account that the main goal of the present work consists on the implementation of computational techniques to classify human upper body postures, a literature review was carried out on the subject. The first human motion recognition systems date back to the late '90s, as example, the work developed by (Foerster et al., 1999), in which four piezo resistive accelerometers were placed in sternum, wrist, thigh and lower leg and the motions analyzed included the standing posture, the sitting posture, the lying supine posture, operating PC keyboard, among others

However, with the appearance of the first works, certain limitations were found, motivating the development of new techniques, in order to increase the performance under more realistic conditions. These new approaches had, as the main challenges: (D. Lara & Labrador, 2013):

- The selection of the attributes to be measured.
- The construction of a portable, unobtrusive and inexpensive data acquisition system.
- The collection of data under more realistic conditions.
- The selection of the best data normalization method.
- The design of the feature extraction and selection methods, as well as the learning algorithms.
- The system flexibility to support new users without the need of re-training it.

Initially, these systems have been based in two different ways: *external* sensors and *wearable* sensors. With regard to the first approach, sensors are placed at predetermined points of interest. Consequently, the analysis of the subject's motion depends entirely on the interaction that the subject may or not have with these sensors. On the other hand, with regard to the *wearable* sensors, these are placed on individual's body. Intelligent homes are an example of applications that rely on *external* sensors, being able to recognize complex activities and postures such as eating, taking a shower, sleeping, sitting or washing machine, for example. Exemplifying, the data are acquired from many sensors placed in some target objects which people are supposed to interact, like the stove, the faucet or the washing machine. However, the applicability of these systems decreases if the individual is out of the reach of the sensors or if the user is performing tasks that do not require the interaction with the aforementioned objects (D. Lara & Labrador, 2013).

In this way, cameras are the most used external sensors. In fact, the recognition of different tasks and postures, from video sequences, has been focus of extensive research, as the case of the research developed by (Turaga, Chellappa, Subrahmanian, & Udrea, 2008). On the aforementioned study, the authors affirm that recognizing human motions from video sequences, is one of the most promising applications of computer vision (Turaga et al., 2008). However, video sequences have certain limitations related to their applicability in these recognition systems. Firstly, the most important issue is the *lack of privacy*, as not everyone is willing to be permanently monitored and recorded by the cameras. The second one is *pervasiveness,* since video recording devices are difficult to attach to the target individuals. Moreover, systems based on video sequences also have the disadvantage of their computationally heavy processing techniques (Turaga et al., 2008).

With all the above presented limitations, the *wearable* sensors have become preferential for the development of the human motion recognition systems. It is also important to note that, most of the measured attributes are related to the user movements (which can be measured by accelerometers or GPS's), the environmental attributes (such as temperature or humidity) or the physiological signals (for example, heart rate monitors) (D. Lara & Labrador, 2013).

Then, the main stages of the motion recognition process will be approached, as well as, the most used algorithms and methods present in the literature.

The first step of any motion recognition system is (1) the selection of the sensors and the attributes to be measured. Then, previous to any system be able to distinguish the different human motions, the data coming from the different sensors are manipulated and processed in different stages. Thus, the main processes can be segregated into the following groups, that are: (2) Pre-processing, (3) Segmentation, (4) Feature Calculation, (5) Dimensionality Reduction and (6) Classification. Next, to better understand the above mentioned, the main stages of the mentioned systems are represented in the Figure 6 (Avci & Bosch, 2010).

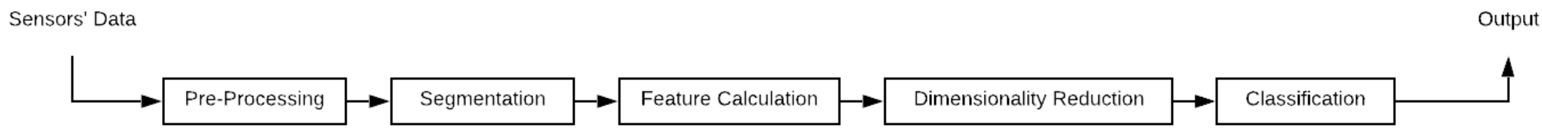


Figure 6 - Main stages of the human activity and posture recognition process.

3.1 Selection of Attributes and Sensors

A wide range of attributes have been monitored in the human motion recognition systems, namely, the environmental attributes, the dynamic attributes and the physiological attributes. With respect to the environmental attributes, many studies have measured attributes like the temperature, the audio level, the humidity or the luminosity, with resource to sensors like microphones, photodiodes, humidity sensors or thermometers (Maurer, Smailagic, Siewiorek, & Deisher, 2006; Pärkkä et al., 2006; Pirttikangas, Fujinami, & Nakajima, 2006).

With regard to the dynamic attributes, tri-axial accelerometers are probably the most used sensors to evaluate human motions or postures like walking, running, jogging, descending or ascending stairs, standing, sitting, back forward bending, among others (He & Jin, 2009; Nath, Chaspari, & Behzadan, 2018; Nurhanim, Elamvazuthi, Izhar, & Ganesan, 2017).

To exemplify the applicability of the accelerometers in these recognition systems, in the study developed by (Khan, Lee, Lee, & Kim, 2010), the authors constructed a system based on a single accelerometer, placed on the individual's chest. It is important to note that this system was able to recognize fifteen different motions and postures, with an average accuracy of about 99%. Also, (Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018) used a single accelerometer to classify seven cervical postures and the overall system's accuracy rounded the 100%. Alternatively, other devices that have been used in some works, with great results, are the Inertial Measurement Units (IMU's) being some of these sensors able to provide acceleration data, angular velocity data and magnetic data (Attal et al., 2015).

Lastly, physiological signals like the heart or respiratory rate, skin temperature or conductivity, were used to classify the physical motions, in some works (Pirttikangas et al., 2006; Tapia et al., 2007). However, some researchers, concluded that physiological signals like heart rate are not useful in the present context, as when an individual is performing a demanding task, its heart rate will remain at a high level for some time. In this way, this signal will introduce errors when the system is recognizing the following activity, due to the fact that the heart rate is still influenced by the previous activity. In addition, these attributes are not expected to be useful in the present dissertation because the motions analyzed are not physically demanding and some of them consist of static postures.

3.2 Pre-Processing

As it is known, due to the nature of the inertial sensors, the acquired data should firstly pass through a pre-processing phase. Almost always, the data provided by the accelerometers, contains some noise that is important to filter out, before using them to recognize the subject's motions (D. Lara & Labrador, 2013). Therefore, many filters can be applied to reduce the impact that this noise has in the classification model's performance.

For example, with the research developed, it is possible to refer a wide range of filters that were previously used. Some examples are a 3rd order moving median average filter (Khan et al., 2010), a low-pass filter (Bidargaddi, N., Sarela, A., Klingbeil, L., & Karunanithi, 2007), a Laplacian filter, a Gaussian filter, a median filter (Karantonis et al., 2006) or a 4th order Butterworth low pass filter, for example (Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018).

Then, the representation of the raw data while preserving important information is generally the key to make efficient and effective solutions, providing a better recognition performance. On the other hand, these techniques could make the system more weighted, computationally speaking (Avci & Bosch, 2010). Two examples of these techniques consist of the Fourier Transform and its derivations, and the Wavelet Transform and its derivations, and have been used in some motion recognition related studies (Attal et al., 2015; Mantyjarvi, J., Himberg, J., & Seppanen, 2001). The aforementioned techniques appear as powerful methods for recognizing transitions between some motions and eliminating noise during activities like walking or running.

3.3 Segmentation

Due to the fact that it is more difficult to recognize human activities and posture transitions only by the analysis of individual samples, it is of extreme importance to apply techniques that allow the analysis of a set of samples (Mukhopadhyay, 2015).

Thus, a key factor of the human motion recognition systems, consists on the correct selection of the time window's size, being important to consider the trade-off between the computational weight and the amount of information provided. In other words, having short windows, the feature calculation is not computationally demanding but may not provide sufficient information to fully describe the performed motion or posture. In contrast, using windows that are too large, the information provided will be enough to describe and recognize the human motions, but the feature extraction will become computationally demanding and, in addition, it could be possible to have more than one motion within the same time window (D. Lara & Labrador, 2013).

Thus, many different time windows can be found in the motion recognition systems: 1 second (Fs = 25 Hz), equal to 25 samples (Attal et al., 2015); 3.2 seconds (Fs = 20 Hz), equal to 64 samples (Khan et al., 2010); 4 seconds (Fs = 20 Hz), equal to 80 samples (Maurer et al., 2006); 5 seconds (Fs = 50 Hz), equal to 250 samples (Ó. D. Lara, Prez, Labrador, & Posada, 2012); 10 seconds (Fs = 20 Hz), equal to 200 samples (Kwapisz, Weiss, & Moore, 2011).

3.4 Feature Calculation

Feature extraction techniques are applied to each time window, filtering the relevant information and obtaining quantitative measures that allow the signals to be compared. Additionally, these techniques provide the transformation of a large input data, into a reduced set of features, generally named as *feature vector* (D. Lara & Labrador, 2013). Then, the most common extracted features can be segregated into three different groups: The Time-Domain (TD) Features, the Frequency-Domain (FD) Features and the Heuristic Features.

Firstly, the TD features consist of basic waveform characteristics and signal statistics, being directly derived from a data segment. Some authors extracted from the signals many TD features such as the mean, the maximum, the minimum or the variance (Ó. D. Lara et al., 2012; Nath et al., 2018; Nurhanim et al., 2017; Tapia et al., 2007). The variance is mostly used to capture the signal variability, while the mean, represents for example, the continuous component (DC component) of the acceleration data. In addition, other TD features that are usually extracted are the interquartile range (IQR) (Attal et al., 2015; Nath et al., 2018; Nurhanim et al., 2017) and the root mean square (RMS) (Bayat, Pomplun, & Tran, 2014; Ghasemzadeh, Loseu, Guenterberg, & Jafari, 2009; Maurer et al., 2006).

With respect to the FD features, these are focused on the periodic structure of the signal, being extracted from the signals obtained with the Fourier Transform, for example (Avci & Bosch, 2010). One of the most commonly extracted FD features, in the human motion recognition context, is the spectral energy, being this feature used to obtain the periodicity of the acceleration data (Attal et al., 2015; Ó. D. Lara et al., 2012; Nath et al., 2018). Spectral entropy was also extracted in many works (Attal et al., 2015; Khan et al., 2010). By the end, the skewness and the kurtosis of the frequency domain are another characteristics that intend to represent the different human motions (Attal et al., 2015; Nurhanim et al., 2017).

Lastly, heuristic features consists on characteristics that derive from a fundamental understanding of how a specific activity would generate a distinguishable sensor signal (Avci & Bosch, 2010). Therefore, the signal magnitude area (SMA) was calculated in many studies, as well as the signal magnitude vector (SMV) (Karantonis et al., 2006; Khan et al., 2010). For example, in (Khan et al., 2010), the system detects a fall when, at least, two consecutive peaks in the SMV are recorded above a defined threshold. Finally, another heuristic feature that was used to classify motions, is the mean absolute value (MAV) (Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018).

To sum up, in the Table 3, are represented the most widely used features in the human activity and posture recognition systems, according to their different types.

Table 3 - Most common used features.

Type	Most Common Features	References
Time-Domain	Mean, Maximum, Minimum, Variance, Standard Deviation	(Ó. D. Lara et al., 2012; Nath et al., 2018; Nurhanim et al., 2017; Tapia et al., 2007)
	Interquartile Range (IQR)	(Attal et al., 2015; Nurhanim et al., 2017)
	Root Mean Square (RMS)	(Bayat et al., 2014; Ghasemzadeh et al., 2009; Maurer et al., 2006)
Frequency-Domain	Spectral Energy	(Ó. D. Lara et al., 2012; Nath et al., 2018)
	Spectral Entropy	(Attal et al., 2015; Khan et al., 2010)
	Skewness and Kurtosis	(Attal et al., 2015; Nurhanim et al., 2017)
Heuristic	SMA (Signal Magnitude Area) SMV (Signal Magnitude Vector)	(Karantonis et al., 2006; Khan et al., 2010)
	MAV (Mean Absolute Value)	(Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018)

3.5 Dimensionality Reduction Techniques

The dimensionality reduction techniques are generally used due to their capacity of reducing the computational complexity of the system and increasing the system performance (Avci & Bosch, 2010). Relatively to the computational complexity reduction, it can be explained as less features involved in the classification process, less computational memory and time are needed to perform the classification. Additionally, these techniques are advantageous because many irrelevant features are discarded, being the processes of training and testing, more efficient and less time consuming (D. Lara & Labrador, 2013). Then, these techniques can be segregated into two different groups: (i) Feature Selection Methods and (ii) Feature Transform Methods.

With regard to the former group, as its name indicates, the main objective is selecting the features that most contribute for discriminating the different activities or postures and, consequently, for the improvement of the classifier performance. With the use of these techniques, a new set of features is created (Avci & Bosch, 2010). Thus, (Wang, S., Yang, J., Chen, N., Chen, X., & Zhang, 2005) used the *SVM-based feature selection method* to sort their 19 features, according to their relevance order. Consequently, the researchers found the most powerful features.

Another well-known and most used feature selection technique is the *Forward/Backward Sequential Search,* that consists of a wrapper method. For example, (Pirttikangas et al., 2006) used this method to select the best features, resulting on a subset of 19 features, related to the accelerometer and heart-rate data. By the end, the *ReliefF* method is an alternative feature selection method found in the literature, belonging to the Filter feature selection family. As example, in the work developed by (Nath et al., 2018), the *ReliefF* technique was used with the objective of ranking the 288 original features, in order of their effectiveness. .

Finally, the *mRMR* algorithm is one of the most used feature selection algorithm, belonging to the filter family. This algorithm is very similar to the *ReliefF* algorithm with the advantage of eliminating the redundant features and it has been used in some machine learning related works (Ding, C., & Peng, 2005).

In its turn, the feature transform techniques intend to map the high-dimensional feature space into a much lower dimension space, generating a subset that is the combination of the original features with a fewer number of features (Avci & Bosch, 2010). These techniques have some associated advantages, being one of the most important, the fact that they handle the situation in which multiple features collectively provide good discrimination, while they provide poor discrimination individually.

One of the most used feature transform technique, in the motion recognition studies, is the *Principal Component Analysis* (PCA) (He & Jin, 2009; Mantyjarvi, J., Himberg, J., & Seppanen, 2001). Also, the *Independent Component Analysis* (ICA) is another most used technique to reduce the dimensionality of the feature vector. For example, (Mantyjarvi, J., Himberg, J., & Seppanen, 2001), used both the PCA and ICA techniques with the Wavelet Transform and they concluded that the difference between PCA and ICA is negligible and both of these two provided good recognition performances.

3.6 Classification Algorithms

A very critical step of the machine learning applications consists on the choice of the most appropriate learning method. It is important to refer that these techniques differ from their nature, being that they can be logical algorithms, perceptron-based algorithms, statistic algorithms, among others (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007).

Firstly, logic-based classification algorithms are, generally, decision trees, Therefore, the C4.5 decision tree algorithm has provided a great recognition performance, in some motion recognition related studies. More specifically, the system developed by (Wang, S., Yang, J., Chen, N., Chen, X., & Zhang, 2005) recognized the different motions with an average accuracy of 98.24%, using the C4.5 algorithm. In addition, using the same classifier, (Maurer et al., 2006), recognized the physical postures and activities, with an average accuracy of 92.8 %.

Next, perceptron-based classification algorithms, are another alternative learning algorithm, being the multilayer perceptron one of the most used. ANNs have been used in many activity and posture recognition related works. For example, in the work developed by (Wang, S., Yang, J., Chen, N., Chen, X., & Zhang, 2005), this learning algorithm provided an average accuracy of the motions recognition of 89.45% but the authors concluded that other classification algorithms, such as the SVM and the C4.5 algorithm, performed better than the ANN. Additionally, (Pirttikangas et al., 2006), tested the ANN to classify 9 different postures and they concluded that, for example, the KNN classifier (average accuracy of 92.89%) performed better than the ANN (average accuracy of 89.76%).

Alternatively, statistical learning algorithms as the Naïve Bayes (NB) networks and the Bayesian Networks (BN) can be used to classify the different human motions. These two algorithms are not most used in the academic works that developed a motion recognition system. Even so, (Maurer et al., 2006), tested both the NB classifier and the BN classifier, and they concluded that NB classifier perform better that the NB classifier being that, the average accuracy of the system rounded the 98%, being that the main objective of the study was to classify between six different motions, including the standing posture, walking, ascending and descending stairs, among others. In addition, in the work developed by (Ó. D. Lara et al., 2012), the Bayesian Network learning algorithm provided an average accuracy of the activity recognition of 90.57%, but it was surpassed by other learning algorithms like the Additive Logistic Regression (ALR).

Also, instance-based algorithms could be used in the human posture and motion recognition systems. These learning algorithms are known to require less computational time during the training stage, when compared with other learning algorithms like the ANN, the BN or the decision trees, for example. On the other hand, they require more time during the classification stage (Aha, D. W., Kibler, D., & Albert, 1991). KNN is the one of the most well-known instance-based learning algorithm and it has been used in some motion recognition related studies (Attal et al., 2015; Maurer et al., 2006; Pirttikangas et al., 2006).

More specifically, in the study developed by (Attal et al., 2015), the KNN classifier provided the best recognition performance (average accuracy of 99.25%), surpassing other learning algorithms like the SVM or the Random Forest classifiers. Also, as previously mentioned, (Pirttikangas et al., 2006), tested the KNN and the ANN classifier and the authors affirmed that, the KNN classifier performed better than the ANN classifier, resulting on an average accuracy of 90.61%, with regard to the posture recognition.

Therefore, the Support Vector Machine (SVM), is probably the most used learning algorithm in the motion recognition related works. Hence, this classifier provided a good recognition performance in many studies, being an example, the work developed by (He & Jin, 2009), in which the SVM classifier with the OVO (one-vs-one) strategy, provided an average accuracy of 97.5%, when recognizing four different daily activities. Another study in which the SVM classifier provided great results, was developed by (Nurhanim et al., 2017). In this work, the authors used the SVM classifier with OVA (one-versus-all) strategy to classify six motions and the system provided an average accuracy of 98.95%.

Lastly, it is possible to refer that the LDA learning algorithm was used in many studies (Sarcevic, Kincses, & Pletl, 2014; Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018). More specifically, (Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018), used the LDA classifier and recognized seven cervical postures, with an overall accuracy of approximately 100%.

Therefore, in the Table 4, some human motion recognition systems are described. More specifically, are approached the main objectives of the work, the data acquisition system, the feature extraction and selection methods used, the learning method used, the processing approach and the most important conclusions about the related work.

Table 4 - Most important characteristics of the human motion recognition systems present in the literature.

Reference	Work Objectives	Acquisition System and Measured Variables	Feature Calculation and Dimensionality Reduction	Learning Algorithms and Performance Metrics	Processing Approach	Main Conclusions
(Mantyjarvi, J., Himberg, J., & Seppanen, 2001)	Recognize four motions: (1) standing, (2) walking, (3) walking upstairs and (4) walking downstairs.	Two sets of accelerometers (ADXL202): one placed in the left side of the hip and other in the right side.	Windows of 256 samples, equal to 1 second (Fs = 256 Hz). Two different datasets: (1) Using PCA and (2) Using ICA. Both with the wavelet transform.	Multilayer Perceptron (MLP) classifier with ten-folded cross-validation. Best accuracy rounds 83-90%.	Acceleration vectors are processed offline, in the MATLAB software.	Differences between ICA and PCA turned out to be negligible.
(Wang, S., Yang, J., Chen, N., Chen, X., & Zhang, 2005)	Recognize three different postures: (1) Drinking, (2) Phoning and (3) Writing.	Three accelerometers (KXP74) used placed on: (1) telephone receiver, (2) base of the cup and (3) top of the pen.	Windows of 64 samples, equal to 2 seconds (Fs = 32 Hz). Features: mean, standard deviation, energy, among others. SVM-based feature selection method was used.	(1) C4.5 – average accuracy of 98.24%. (2) SVM: average accuracy of 90.74%. (3) MLP – average accuracy of 89.45%.	Sensor Data are transmitted, via RF (radiofrequency) signal to the base station, which is connected to the serial port of a laptop.	Best results obtained with the cross-validation, surpassing the leave-one-subject-out validation. Decision Tree performed better than the other two classifiers.
(Maurer et al., 2006)	Recognize in real time, six primary activities/postures: (1) sitting, (2) standing, (3) walking, (4) ascending stairs, (5) descending stairs and (6) running.	Authors propose the eWatch: device based on the Philips LPC2106 ARM7 TDMI microcontroller and four sensors: a dual axes accelerometer, a light sensor, a temperature sensor and a microphone. Device placed on the waist.	Windows of 4 seconds (80 samples due to the Fs of 20 Hz). Time-Domain Features: RMS, STD, variance, mean absolute deviation,	C4.5 algorithm – average accuracy of 92.8%. It was concluded that the NB and C4.5 classifier performed better than the other learning algorithms (K-NN and BN).	The classification run in the *e-Watch* (that has incorporated a microcontroller), with resource to a decision tree classification algorithm.	Between C4.5 and NB classifier, C4.5 was chosen due to its good balance between accuracy and computational complexity.

			IQR, mean crossing rate, zero crossing rate.			
(Karantonis et al., 2006)	Recognize in real-time 12 different postures transitions (sitting-to-stand, stand-to-sit, lying, sit-to-lying, walking, fall, standing, among others).	TA unit that includes two orthogonally mounted dual axis accelerometers (MXR7210GL, MEMSIC, Inc., North Andover, MA). The device was placed on the individual's waist. One microcontroller (MSP430F149, Texas Intruments, Dallas, TX). One ZigBee module (CC2420EM, Chipcon, Oslo, Norway).	Windows of 45 samples, equal to 1 second (Fs = 45 Hz) . Features: SMA (signal magnitude area), tilt angle (angle defined between the positive z-axis and the gravitational vector *g)* and SMV (signal magnitude vector)	Researchers implemented a simple threshold-based algorithm. System average accuracy of 90.8%.	Acceleration data are sampled and classified, in real-time, in a microcontroller within the TA unit. Then, relevant data are transmitted from the TA unit to a host PC, where the classification takes place.	The microcontroller memory is the main disadvantage because it forced, for example, the IIR filter to be a maximum of third-order. In addition, the FFT calculation is impossible, using the microcontroller.
(Pirttikangas et al., 2006)	Recognize 17 different activities and postures (Clean Whiteboard, sit and read, drink, lay down, walk, descend stairs, among others).	Authors used their own device, named *Cookie*, constituted by a Bluetooth module, some accelerometers, a compass, an ambient light sensor and a heart rate sensor. Four *Cookies*: (1) right wrist, (2) left wrist, (3) right thigh and (4) necklace.	Window sizes tested: 0.1 s, 0.2 s, 0.5 s, 0.7 s, 1 s and 1.5 seconds (Sampling Frequency of 10 Hz). Features: Acceleration mean, acceleration STD, acceleration correlation coefficients between x and y axis, acceleration mean crossing and heart rate mean. Forward feature selection algorithm was applied.	KNN and MLP with 4-fold cross-validation. KNN has provided a better average accuracy (90.61) than the MLP.	Raw data were acquired and then sent to a data collecting terminal (Sony VAIO computer), using a Bluetooth module.	Best results obtained with the windows of 1 second (10 samples) and with the reduced feature vector, obtained with the Forward-backward search algorithm.

(Tapia et al., 2007)	Recognize thirty different activities, being that some of them are the same but with different intensities. Some examples are lying down, walking, cycling, ascend stairs, among others.	Five tri-axial accelerometers, a wireless HR (heart rate) monitor and a laptop with a wireless receiver. Accelerometers locations: (1) dominant waist, (2) dominant ankle, (3) dominant upper arm, (4) dominant thigh and (5) dominant hip.	Windows of 126 samples, equal to 4.2 seconds (Fs = 30 Hz). Features: From the accelerometers (AUC, variance, mean, entropy, correlation coefficients). From the HR monitor (the number of heart beats above the resting HR value). Forward Sequential Selection, to select the most relevant features.	NB algorithm and the decision tree algorithm C4.5. Accuracy of C4.5: 98.7%. Only results for the C4.5 were presented due to its less computationally weight.	Data from the accelerometers are transmitted to a laptop, via wireless. The classification is done remotely, in real-time, in a laptop, with resource to WEKA toolkit.	Best performance of C4.5 algorithm was obtained with the elimination of the features AUC and mean (because their sensitivity to accelerometer calibration and orientation).
(Gyorbíró, Fábián, & Hományi, 2009)	Recognize six different activities that are: (1) resting, (2) typing, (3) gesticulating, (4) walking, (5) running and (6) cycling.	Windows of 200 samples, equal to 2 seconds (Fs = 100 Hz). A smartphone, a MotionBand device (placed on the individual's wrist) that contains an accelerometer, a gyroscope and a magnetometer.	Only the intensity of motion is calculated from the acceleration data, for each instant *t:* $I(t) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{s(t-1)-s(t-i-1)}{\Delta x(t-i)}$	Six feed-forward ANN were trained, one for each different activity. The average F-measure of overall system recognition rounds the 79,76%.	Classification occur, in real-time, with resource to a neural network, that is implemented in the smartphone.	Authors have also tested the decision tree C4.5 algorithm but the recognition rate greatly degraded during live experiments.
(He & Jin, 2009)	Recognize 4 different activities: (1) running, (2) still, (3) walking and (4) jumping.	A single tri-axial accelerometer (ADXL330 produced by Analog Devices) placed in the subject's trousers pocket.	Features: DCT coefficients. PCA applied to reduce the feature vector dimension and select the most relevant coefficients.	SVM with OVO (one-versus-one) strategy for classification. System accuracy rounds the 97.5%.	The data generated by the accelerometer were transmitted to a personal computer (via Bluetooth), where the activity classification is done, in real-time.	Best performance obtained with 48 DCT coefficients and 20 PCA coefficients.

(Curone, Bertolotti, Cristiani, Secco, & Magenes, 2010)	Classify the activity type (for example, upright standing, upright mild activities, upright intense activities) and also classify the individual's postures transition (lying-down to standing and vice-versa and falling to the ground).	Tri-axial accelerometer (ADXL330, Analog Devices, Inc, USA) placed on the individual's upper trunk. ADUC7027 microcontroller (by Analog Devices, Inc). Bluetooth module (F2M03GLA-S01 by Free2Move, Halmstad, Sweden).	Windows of 50 samples, equal to 1 second (Fs = 50 Hz). For the activity classification, was used the SMA (signal magnitude area) feature. For the posture classification, average low-frequency acceleration was used, to obtain the sensor orientation.	Threshold based algorithm. System accuracy rounded the 96.2% (for the intensity activity recognition) and the 89.6% (posture transition recognition).	Data processing and classification occurs in real-time in the microcontroller and the outputs are transmitted to a PC, with resource to the Bluetooth module.	Main error occur in the posture transition classification, in a "climbing wall bars" activity, due to the fact that the individual's trunk is bended more than the 60° threshold.
(Khan et al., 2010)	1°: recognize the state related to a specific activity (static, transition or dynamic). 2°: recognize the 15 different activities and postures (lying, sitting, walking, stand-to-lie, walk-to-stand, running, among others.	A single accelerometer (*Witilt V2.5* from Sparkfun) placed on the individual's chest.	Windows of 90 samples, equal to 1 second (Fs = 90 Hz). Features for the 1° task: mean, STD, spectral entropy and correlation were extracted from the acceleration signals. For the 2° sub-work: AR (autoregressive) coefficients, SMA (signal-magnitude area) and TA (tilt angle).	For the 1° case, a single ANN was trained to recognize the three different states and it has provided an accuracy of about 99%. For the 2° case, to recognize the 15 activities, one ANN was constructed for each group of activities. System accuracy rounded the 97,7 %.	Data from the accelerometer are transmitted, with resource to a Class 1 Bluetooth link, to a universal serial bus dongle attached to a computer. Data are stored and processed, in real-time, in this computer.	Best performance obtained using the total feature vector that includes the AR coefficients, the SMA and the TA.
(Kwapisz et al., 2011)	Recognize six different activities: (1) walking, (2) jogging, (3) ascending stairs, (4) descending stairs, (5) sitting and (6) standing. Twenty-nine users have performed the above mentioned activities.	The system only uses a phone-based accelerometer to perform activity recognition. Smartphone placed on the individual's tight.	Windows of 10 seconds (Sampling Frequency of 20 Hz) Features: Mean, standard deviation and time between peaks in the signals, for each one of the three axis of accelerometer. Also, was extracted the average resultant acceleration as follows:	Three different classifiers: (1) J48, (2) Logistic Regression and (3) Multilayer Perceptron. J48 have provided an average accuracy of 85.1%, Logistic Regression of 78.1% and, the Multilayer perceptron classifier, of 91.7%.	The data are collected with resource to a smartphone application. Authors do not present information about the place where the task classification occurs. Probably, the processing was done offline.	Authors have concluded that the algorithms failed principally when were intended to distinguish between the activities ascending stairs and descending stairs.

			$\sqrt{x_i^2 + y_i^2 + z_i^2}$ Windows of 10 seconds.			
(Gjoreski, H., & Gams, 2011)	Recognize seven different motions: (1) standing, (2) sitting, (3) lying, (4) on all fours, (5) sitting on the ground, (6) standing up and (7) going down.	Three wearable Xsens-MTx sensors (includes an accelerometer, a gyroscope and a compass), placed on: (1) Chest, (2) Right thigh and (3) Right ankle.	Windows of 6 samples, equal to 1 second (Fs = 6 Hz) Features from the accelerometer (for each axis): module of the acceleration vector, mean value, standard deviation and root mean square. Posteriorly, Euler angles and quaternions were calculated from the gyroscope and the magnetometer.	Three learning algorithms tested (SVM, Random Forest and J48). Random Forest performed better than the others, resulting on an overall recognition accuracy of 90.5%. Leave-one-subject out approach was used.	No information about the place where the processing is done.	Best results obtained with the data from the 3 sensors and with the integration of the data from the gyroscopes and compasses. Main misclassifications were related to the activities “Going down” and “Standing up”.
(Ó. D. Lara et al., 2012)	Recognize 5 different activities: (1) walking, (2) running, (3) sitting, (4) ascending and (5) descending.	*BioHarness™ BT chest sensor strap* allow the measurement of acceleration data and vital signals such as the heart rate, respiration rate, ECG amplitude, among others.	Window sizes tested: 5 s, 10 s, 15 s. 8 statistical features extracted from acceleration data (for example, mean, variance, IQR from time domain (TD) and spectrum energy from FD). +9 structural and two transient features extracted from vital signs.	ALR (Additive Logistic Regression): average accuracy of 95.7%. BN (Bayesian Network): average accuracy of 90.57%.	Mobile application receives the data coming from the sensors, in real-time, via Bluetooth. Then, data are sent to an application server and the classification occur offline.	The best time window was 12 seconds and best results were achieved taking into account both the acceleration data features and the vital signals features.
(Bayat et al., 2014)	Recognize 6 different activities: (1) running, (2) slow-walk, (3) fast-walk, (4) dancing, (5) stairs-up and (6) stairs down.	Phone-based tri-axial accelerometer. Smartphone locations tested: (1) subject’s hand and (2) subject’s pants pocket.	Time windows of 128 samples, equal to 1.28 seconds (Fs = 100 Hz). Features extracted: the mean value of each window, the average of peak frequency, the RMS, the STD, the	All classifiers evaluated with 10-fold cross validation. 1° MLP with an average accuracy of 89.72%.	Data were processed offline, with resource to the WEKA toolkit.	Authors concluded that the two different smartphone locations (hand and pocket) produced similar results.

	Four subjects (2 male and 2 female), performed the activities.		correlation between different axes, among others.	2° SVM with an average accuracy of 88.76%. 3° Random Forest with an average accuracy of 87.55%.		
(Sarcevic et al., 2014)	Recognize specific arm movements in both stationary and dynamic positions (for example, standing without arm movement, raising ald lowering the left arm during standing, among other nine).	One IRIS mote (contains an ATmega 1281L 8-bit microcontroller and a FR231 radio transceiver). One Wireless IRIS sensors, mounted on the subject's wirst (contains 1 ADXL345 accelerometer, 1 ITG3200 gyroscope and 1 HMC5883 magnetometer).	Time windows tested: 10, 25, 50, 100 samples, equal to 80, 200, 400 and 800 ms (Fs = 125 Hz). Features: MAV (Mean Absolute Value), Number of Zero Crossing (NZC), number of slope sign changes (NSSC), waveform length (WV), among others.	LDA classifier. The highest system accuracy rounded the 96.33%.		Best recognition performance obtained with the data from the gyroscope sensor. Recognition performance could be improved, by increasing the time of the processed windows, but the microcontroller restricts the window's size.
(Attal et al., 2015)	Recognize twelve different activities such as standing, sitting, lying down, among other nine.	Xbus Kit from XSens (Enschede, Netherlands) constituted for 3 IMU's. . IMU's locations: chest, right thigh and left ankle.	Windows of 25 samples, equal to 1 second (Fs = 25 Hz). 11 TD features were extracted (mean, variance, IQR, skewness, among others), 6 FD features (DC component in FFT spectrum, energy spectrum, among others).	1) K-NN with an average accuracy of 99.25% 2) RF with an average accuracy of 98.95% 3) SVM with an average accuracy of 95.55%.	Collected data are transmitted from the XBus Master to a host PC, using a Bluetooth link. Classification occurs, offline, in the aforementioned host PC.	Authors concluded that the supervised learning algorithms are advantageous when compared to the unsupervised learning algorithms.
(Nurhanim et al., 2017)	Study the different kernels of SVM. 6 activities: (1) standing, (2) laying, (3) sitting, (4) upstairs, (5) downstairs, (6) walking.	An accelerometer and a gyroscope incorporated in the smartphone Samsung S2, placed on the waist of the individuals.	17 features: minimum, maximum, standard deviation, IQR from the time domain. Kurtosis and skewness of the frequency domain, among others.	SVM with a OVA (one-versus-all) strategy. System accuracy rounds the 98.95%.	The authors use the UCI Machine Learning Respiratory database. Processing and activity classification occur in offline.	Best kernel for the activity recognition is the polynomial, with the exception of the activity of lying. In this case, the Gaussian kernel is advantageous.

(Nath et al., 2018)	Main goal: classify 3 different categories: category-0 (no risk activities), category-1 (lift, lower, carry) and category-2 (push and pull)	Accelerometer and a gyroscope, both incorporated in a smartphone. Two smartphone locations tested: upper arm and waist.	Features: mean, min, max, standard deviation, IQR, 4th-order autoregressive coefficients, among others. Was used a feature selection method known as RELIEF.	SVM with quadratic kernel and OVO (one-versus-one) strategy. System accuracy rounds the 92.07%.	All the classification approaches are tested offline, with resource to the MATLAB software.	Best performance obtained with the smartphone placed in the upper arm, windows size of 2 seconds and using the features obtained through the *ReliefF* method.
(Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018)	Main goal: classify seven different neck postures: (1) Normal pose, (2) Neck flexion, (3) Neck extension, (4,5) Neck inclination to both shoulders and (6,7) Neck rotation for both sides (right and left).	A single tri-axial accelerometer (MPU9250), placed on the individual's forehead. Microcontroller (STM8L151) and a Bluetooth module (HJ580).	Time Windows of 250 seconds, 40% overlapped, equal to 2 seconds (Fs = 125Hz). Only two features were extracted from the acceleration signals: MAV (Mean Absolute Value) and RMS (Root Mean Square).	LDA (Linear Discriminant Analysis) method as a learning algorithm with a 5-folded cross validation. System accuracy rounds 100%.	The recorded acceleration data were pre-processed and analysed offline, with resource to the MATLAB software.	Another important system characteristic is its capacity of recognize four levels of the cervical inclination.

3.7 Recognition System Issues

With respect to the motion recognition systems, there are many issues that need to be approached. In this way, will be explained some of these issues, namely, the system usability, the data collection protocol, the flexibility of the system and the processing approaches.

Firstly, with the interest of acquiring different types of attributes and having them combined into the same system, creates a trade-off situation with the system usability. Any system should be designed to be non-intrusive, comfortable and usable, and this only becomes possible with less number of sensors and low complexity systems (D. Lara & Labrador, 2013). As example, there are systems that recognize the physical postures and activities, with a good performance, requiring the user to wear only a cell phone (Nath et al., 2018; Nurhanim et al., 2017).

On the other hand, the more sources of data, the richer the information that can be extracted from the measured attributes. But, in contrast, with a higher number of sensors, the system could be uncomfortable, invasive, expensive and could restrict the user movements, as example of the work developed by (Tapia et al., 2007), in which the user has to wear more than five sensors. Summing up, the minimization of the number of sensors required to recognize the activities, is advantageous not only for comfort, but also to reduce the energy consumption and the complexity of the system.

With respect to the data collection protocol, it is important to note that factors like the procedure followed by the subjects while collecting data, the number of subjects that acquired data and their physical characteristics, could affect the recognition performance of the system (D. Lara & Labrador, 2013). In this way, to construct a robust system, it is advisable to consider a large number of subjects with different characteristics with respect to factors like the gender, the age, the height or the weight. Thus, the system will be more flexible to support new users without the need of re-training the system.

Additionally, there are many other factors that could affect the recognition performance of the motion recognition systems. Some examples of these factors are the postures and motions set to be measured, the quality of the training set, the features calculated through the data coming from the sensors, the selected learning algorithm and the way how the data is processed (on board or remotely, for example).

As above mentioned, the way how the system processes the data could also affect the recognition performance. In this way, there are systems that process the data in real-time and others that process the data offline. Within the real-time systems, there are two different approaches that are commonly used to process the data: (i) locally and (ii) remotely.

With regard to the former approach, generally, the systems make the processing with resource to a microcontroller, that is embedded in the system, as the example of the works developed by (Curone et al., 2010; Karantonis et al., 2006; Maurer et al., 2006). On the other hand, the processing could occur remotely. More specifically, the system could process the data, for example, on a laptop (Tapia et al., 2007) or on a personal digital assistant (PDA) (Pärkkä et al., 2006). Thus, it is important to establish a communication protocol, that allow the information to be well transmitted, between the sensors and the device where the processing take place.

In contrast to the aforementioned, many systems collect and process the data offline, making the system less complex and, generally, providing better recognition performances, as the example of the study developed by (Turaga et al., 2008).

Finally, is important to note that the processing approach selected could provide some limitations related to the motion recognition systems. Thus, when it is intended to implement a system that recognize the motions with an on-board real-time approach, some problems appear. As example, (Karantonis et al., 2006) and (Curone et al., 2010), developed posture recognition systems, in which the processing is done in a microcontroller embedded in a portable system and worn by the individuals. As it is known, the microcontrollers memory and their processing capabilities are limited, when compared with a computer or a PDA. Consequently, the aforementioned limitations, generally, restricts the set of the postures and activities to be classified, as well as, the length of the time windows or the algorithms to be applied. For example, according to the authors (Karantonis et al., 2006), the Fourier Transform only becomes advantageous when applied to time windows that have, at least, 3 seconds of data (equal to 135 samples, due to the system's sampling frequency of 45 Hz), which could be a problem using a certain microcontrollers.

4. Methodology

As it was previously referred, the main objective of the present dissertation consists on the construction of a machine learning classification model that could classify the different individual's upper limb postures and provide information about the worker's ergonomic risk, when realizing their workplace tasks. Firstly, in the present section, it will be referred some case studies, in which the posture recognition systems could be advantageous in the prevention of the work related illnesses. Posteriorly, many steps have to be idealized, namely, the sensor locations on the subject's body, the selection of the postures to be measured and the consequent data collection protocol, the treatment and labeling, the feature calculation, the feature selection or extraction, and the model building and evaluation.

4.1 Case studies

As it is known, a wide range of workers are exposed to some ergonomic risks in their workplaces. Thus, it is possible to refer some careers, in which a system that could recognize the different postures, would be advantageous.

Firstly, the construction environments involve complex activities and substantial amounts of labor, being important to note that this type of work normally requires intense physical movements of the worker's body parts. In addition, awkward postures in construction activities could cause substantial hazards in both instantaneous injuries and long-term work-related musculoskeletal disorders (WMSDs). Specifying, the most common awkward postures developed in the construction context include the following postures: working with the hands above the head (working overhead), kneeling, Back bending forward or backward, Back lateral bending, Squatting, Neck bending forward or Working with the elbows above the shoulders and with hands lower than the head (Reaching). Some of the above mentioned postures are represented in the Figure 7.

Figure 7 - Examples of risk postures, developed in the construction environments.

In addition, the nursing often involve extended work shifts (longer than 8 hours per day), long work hours (greater than 40 hours per week), on-call work and compulsory overtime and shift work (work times other than 7:00 A.M. to 6:00 P.M.). In this way, the nursing profession is established as a physically and psychological demanding profession with high prevalence rates of work-related musculoskeletal disorders (WMSDs). Exemplifying, the nurses commonly realize the back forward bending posture, mostly when they are making beds, helping the patient to dress or helping the patient to move from the bed to the wheelchair. In addition, the nurses often done other risk postures such as the neck bending forward or the shoulder abduction (Goswami, Haldar, & Sahu, 2013).

Lastly, with regard to the Hairdressing profession, it is important to note that hairdressers state that regularly suffer from long-term complaints as a result of their work. More specifically, more than 40 % of these workers suffer from neck complains/pain and approximately more than 30 % suffer from back complaints. Some examples of incorrect postures practiced by the hairdressers are the back forward flexion when they are washing the costumer's hair and the chair is not height adjustable, the back frontal and lateral bending when they are cutting long hairs, and by the end, working with elbows above the shoulders, mainly when the hairdressers are blow-drying the costumer's hair.

4.2 System Design

The first step is to idealize the sensor's position in the subject's body, taking into account that the main structures to be analyzed are the back, the shoulders and the neck.

With regard to the back analysis, as in the work developed by (Umer et al., 2017), three sensors will be placed in the individual's back, more specifically in the S1 bone, in the T4 bone and in the T12 bone, which are represented in the Figure 8. It is important to note that, placing these three sensors in the aforementioned locations, the back will be segregated into two segments. The first segment (thoracic spine) is defined between the T4 and T12 bones, while the second one (lumbar spine) is defined between the T12 and S1 bones.

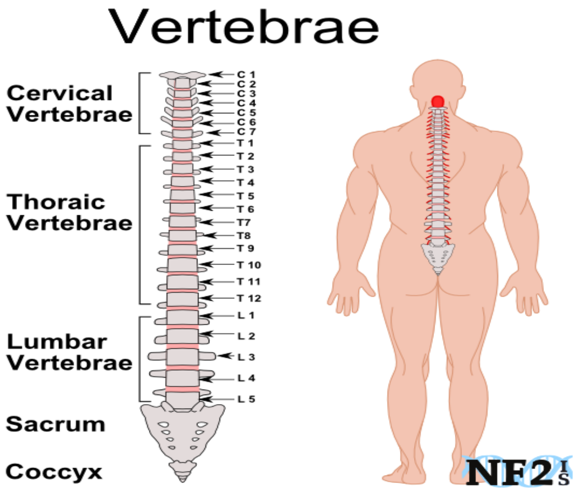


Figure 8 - Main bones of the spine (Peppoloni et al., 2016).

Then, with regard to the shoulders analysis, as in the systems developed by (Taunyazov, Omarali, & Shintemirov, 2016) and (Peppoloni, Filippeschi, Ruffaldi, & Avizzano, 2016), one sensor will be placed in each subject's upper arm, defining each shoulder as a spherical joint.

Lastly, according to (Jasiewicz, Treleaven, Condie, & Jull, 2007), to measure the cervical movements and postures, usually one tracking device is placed on the subject's forehead, generally, using a lightweight adjustable headband centered on the forehead, while a second one is placed over the C7 spinous bone. Additionally, both the studies developed by (Cuesta-vargas & Williams, 2014) and (Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018), placed their system's sensing device on the individual's forehead, being important to refer that the second referred work achieved an average accuracy of the seven different neck postures of about 100%. It is also important to take into account the disadvantages of the aforementioned locations, since placing the sensor on the individual's forehead will not be suitable for the connection of the different devices and, on the other hand, placing the sensor on the C7 bone, could possibly disturb the user, when realizing specific motions like the neck extension. To overcome the above disadvantages, the sensor could be placed in the individual's nape.

Concluding, the proposed system will be composed, in the total, by 6 IMUs (with an integrated accelerometer and gyroscope), placed on the following locations:

- S1 bone
- T4 bone
- T12 bone
- Left upper-arm

- Right upper arm
- Nape.

4.3 Data Collection Protocol

In order to find the postures that may cause greater discomfort to the workers, the work developed by (Kee & Karwowski, 2001) was analyzed, in which it is proposed a technique for postural loading on the upper body assessment (LUBA), more specifically for the hands, arms, neck, back and shoulders. Twenty male subjects participated in the aforementioned experiment, in order to measure perceived joint discomforts. Thus, the authors classify the relative discomfort score of many postures (as example, flexion, extension, radial deviation, abduction, among others) associated to many body structures (as above mentioned, back, neck, shoulders, among others) and with different angles.

After analyzing the classification of the different movements in agreement of their discomfort score, it is important to note that many postures provide a high discomfort, namely, the back frontal bending, the back extension, the back lateral bending, the back rotation, the neck extension, the neck flexion, the neck lateral bending, the shoulder abduction, the shoulder flexion, the working overhead, among others.

Based on the aforementioned, six human postures were selected to be inferred by the classification model constructed. Firstly, the neutral/correct posture will be analyzed as the reference, being commonly named as “standing”. In the second place, “back frontal bending” is another posture that will be classified by the system, being a task that is commonly done by the construction workers, by the hairdressers and by the nurses, for example. Then, the third and fourth postures to classify will be the “back right lateral bending” and the “back left lateral bending" that are commonly done by the construction workers and by the hairdressers. By the end, the fifth and the sixth human postures that will be analyzed are the “neck forward bending” and the “working overhead”, being these two postures mostly done by the construction workers. In addition, all the transitions between the different postures will be analyzed and classified, being some examples, the transitions between the postures Standing-Back Frontal Bending and the opposite or the transitions between the postures Standing-Working Overhead and the opposite. For a better analysis, in the Figure 9 and Table 5, the postures and motions that will be classified are represented and described, respectively.

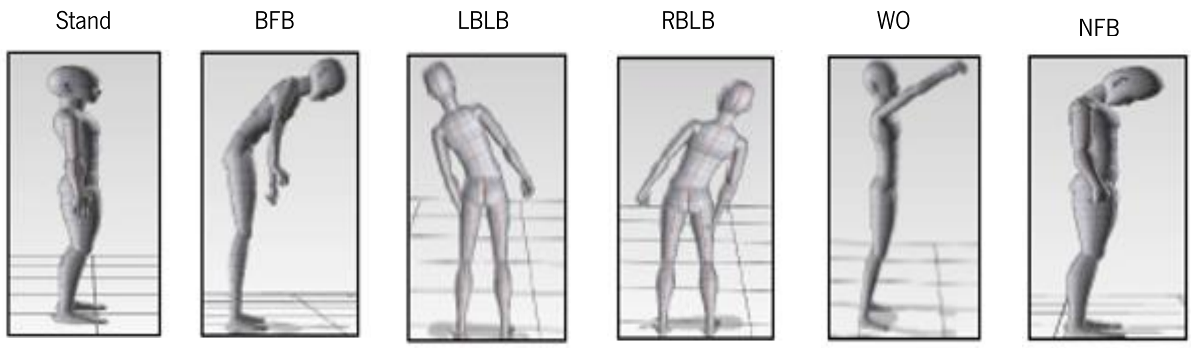


Figure 9 - Representation of the six human postures studies (Stand – Standing, BFB – Back Frontal Bending, LBLB – Left Back Lateral Bending, RBLB – Right Back Lateral Bending, WO – Working Overhead, NFB – Neck Frontal Bending).

Table 5 - Postures and motions to classify.

Postures	
Standing	Static Postures
Back Frontal Bending	
Neck Frontal Bending	
Working Overhead	
Left Back Lateral Bending	
Right Back Lateral Bending	
Standing – Back Frontal Bending	Posture Transitions
Back Frontal Bending – Standing	
Standing – Left Lateral Back Bending	
Left Lateral Back Bending – Standing	
Standing – Neck Frontal Bending	
Neck Frontal Bending – Standing	
Standing – Working Overhead	
Working Overhead – Standing	
Standing – Right Lateral Back Bending	
Right Lateral Back Bending	

In order to build a robust model, the acceleration data from different individuals realizing the six different postures should be collected and labeled. For such, 50 individuals accepted to acquire data for the present dissertation, being some physical characteristics of them described in the Table 6.

Table 6 - Main characteristics of the individuals that accepted to collect data.

Number of Females	Number of Males	Mean Age	Mean Weight	Mean Height
20	30	22.27 Years	65.91 Kg	170.42 cm

All the data were acquired using the *InertialLab* system, being an example of the setup represented in the Figure 10. Some of the most important characteristics of the above-mentioned system are described in the Appendix VIII and Appendix IX (Costa, L. O. L., 2018).

Figure 10 - Example of the data collection setup.

Then, each individual that accepted to collect data was asked to perform each one of the pretended postures. In each trial, the individual was asked to stay in the standing posture for 10 seconds, do the respective posture during 20 seconds, stay in the standing posture for more 10 seconds and, by the end, do the respective posture during more 20 seconds. The procedure during each trial was always the same, the only aspect that change was the posture that the individuals were asked to perform. Next, in the Figure 11, it is represented the above described data collection protocol.

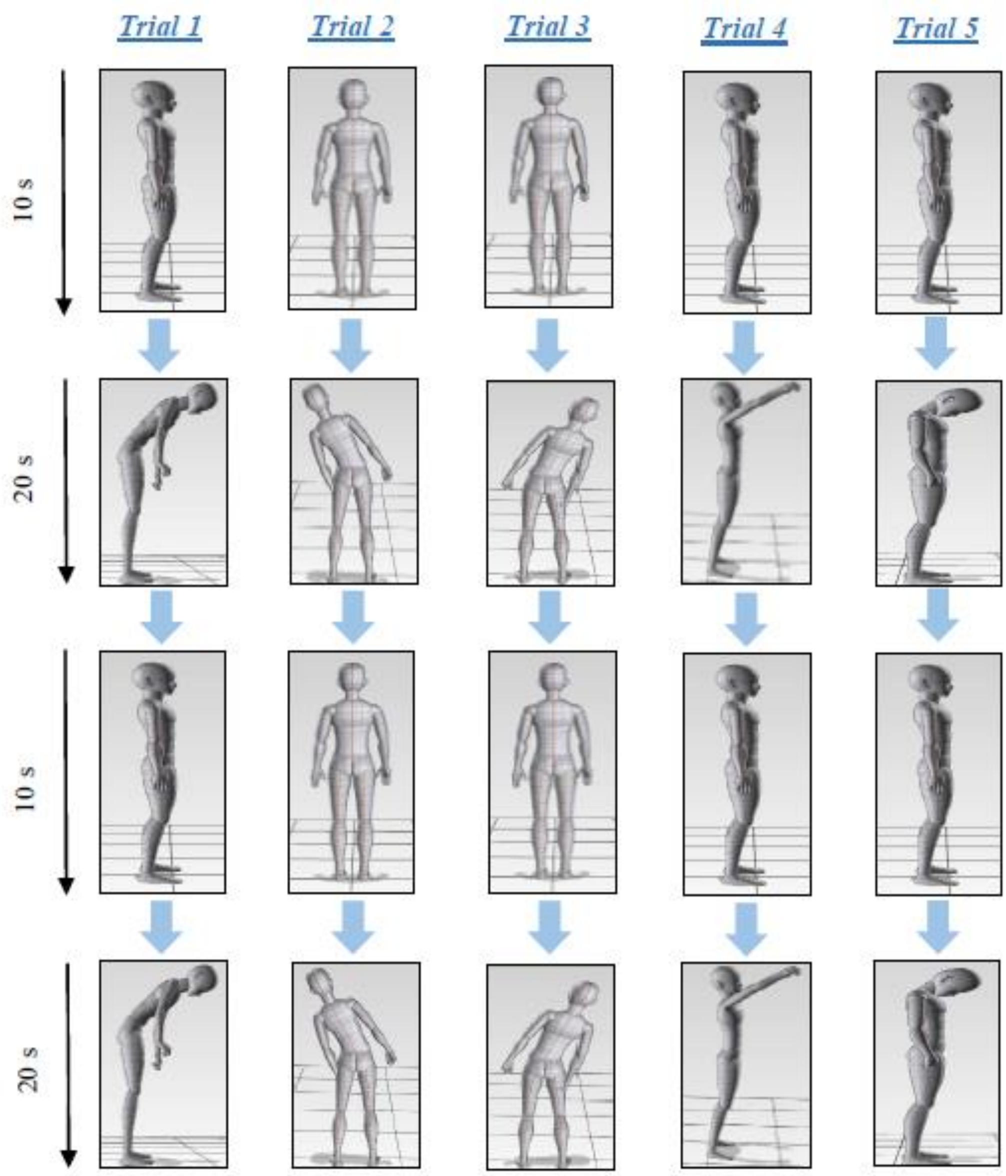


Figure 11 - Data collection protocol, followed by the subjects.

4.4 Human Posture Recognition Pipeline

In the present section, it will be described the methods and the protocols used for the development of the human posture recognition models. For such, a pipeline/framework for the development of the different models was designed to provide an interface that allow the user to test many different machine learning techniques. This pipeline makes use of many already implemented machine learning algorithms on the Mathworks MATLAB (2018b), being the associated schematic represented in the Figure 12.

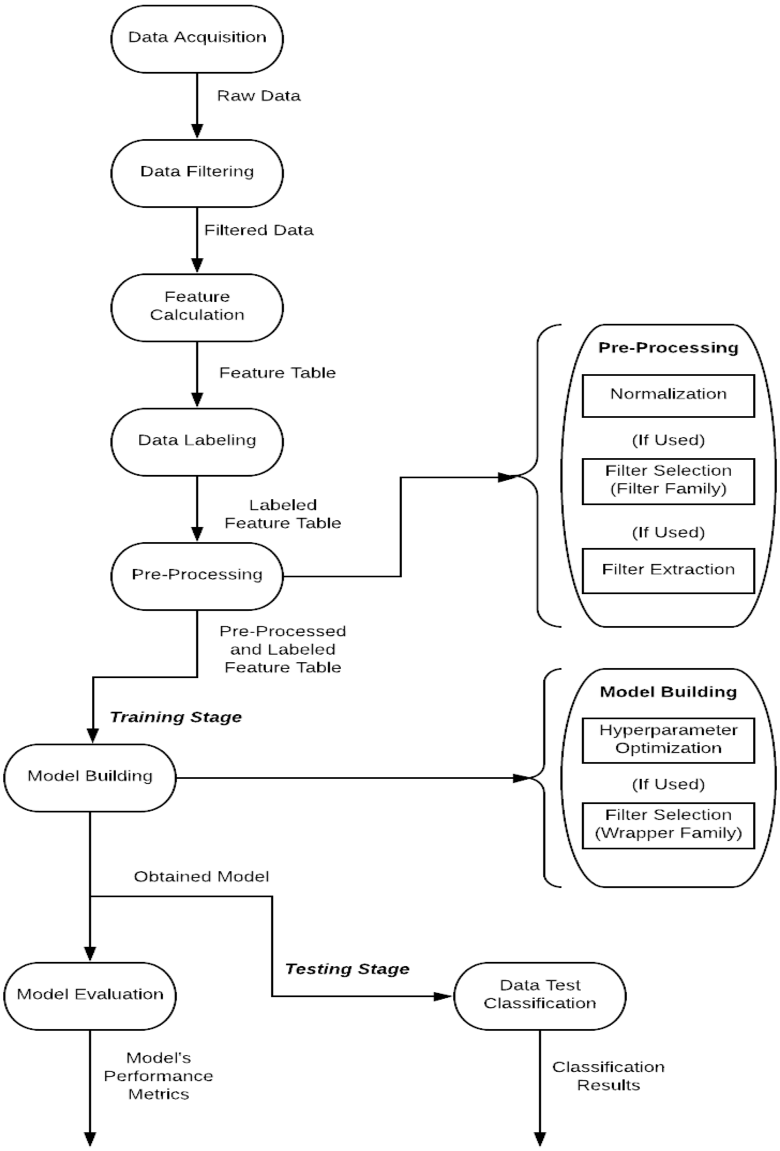


Figure 12 - Pipeline flow chart.

4.4.1 Data Filtering

The inputs of the present pipeline consist of 16 databases, each one with only data related to each different posture or posture transition. For such, each set of the data acquired from the 50 subjects, was manually segmented in order to save in each one of the different 16 databases the information related with the different postures.

After the user select the aforementioned 16 databases to be loaded, the framework shows the *Filtering* Panel. In this block the users could select the option of filtering all the data related to the different postures with resource to a button. Also, the users have the option to filter manually some posture data or posture transitions data. For such, a new window is opened, where it is presented one of the acceleration signal that better represent the performed posture. Also, the FFT of the signal is plotted in order to provide a better information about the cut-off frequency. After the user analyzed the aforementioned information, it is necessary to fill to numeric box with the filter order and the cut-off frequency of the low pass filter. The manually filtering window of the pipeline is shown in the Appendix I.

If the user accepted to filter all data with predefined parameters, a low-pass Butterworth filter is applied to the different postures and postures transitions data, with the filter order of 2 and the cut-off frequency of 0.5 Hz as in the work developed by (Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, 2018).

4.4.2 Feature Calculation

Then, after importing the data related to the different motions and filtering them, it is necessary to extract characteristics/features from these data. These features will be used to train and test the different models and each one of these characteristics should be meaningful of the total dataset.

The *Feature Calculation* block has, as main function, to transform the raw data (input) into a feature vector representative of the respective motion (output). Thus, the user has to fill a box with the number of samples that each time window will contain. Despite of the fact that both the acceleration and angular velocity data were acquired, only features from the 3 axis of the six IMUs are extracted, being them represented in the Table 7.

Table 7 - Features extracted from the acceleration signals.

Feature Designation	IMUs	Axis	Number of Feats
Mean	6	x,y,z	18
Variance	6	x,y,z	18
Standard Deviation (STD)	6	x,y,z	18
Interquartile Range (IQR)	6	x,y,z	18
Root Mean Square (RMS)	6	x,y,z	18
Acceleration Module	6	-	6
Euler Angles (Roll, Pitch, Yaw)	6	-	18
Acceleration Slopes	6	x,y,z	18

With respect to the window's length, it is important to refer that the selection of this parameter is very important, since a very high value could provide more than one posture within the same window and a very low value could not provide the sufficient information to characterize the posture developed by the individuals. Summing up, the output of the present block is a matrix of *n*f*, of which *n* is the number of observations/time windows and *f* is the number of features extracted from each window.

It is also important to refer that, in the present block, the user has the option to use the raw data from the six accelerometers. This option has to be used when it is intended to use, as learning algorithm, the Convolutional Neural Network (CNN) network. In this case, for each time sample, 18 measurements are saved, that consist of the accelerations of the six accelerometers through the 3 axes (x, y and z). Finally, the feature calculation toolbox is shown in the Appendix II.

4.4.3 Feature Data Normalization

After extracted the features in agreement with the window's length chosen by the user, the feature matrix is the input of the *Feature Normalization* block, shown in the Appendix III. The present block has, as the main objective, bring the feature values to a common scale, providing these to be better compared. In this block, the user has the hypothesis of chosen between four of the most well-known data normalization techniques that are, (i) the centering normalization, (ii) the scaling normalization, (iii) the z-score normalization and (iv) the min-max normalization. Also, none normalization technique could also be used. Summing up, the output of the block is the same feature matrix but with all the feature vectors normalized in agreement with the chosen method.

4.4.4 Dimensionality Reduction

Posteriorly to the feature matrix normalization, this matrix is the input of the *Dimensionality Reduction* block, where the main objective is to reduce the complexity of the system, increase the system performance, eliminate redundant and irrelevant features and, consequently, select only the most important characteristics/features.

In the present block, the user can select between three different feature selection algorithms (*ReliefF*, *mRMR* and *SFS*) and one feature extraction algorithm (*PCA*). With respect to the SFS and the PCA algorithms, they calculate and infer how many best features have to be select. In this way, SFS algorithm stops searching for more features when the maximum value for the optimization function is reached. In the PCA case, the algorithm is applied to the original feature data, extracting the most important components and then the Horn's Parallel Analysis is applied as the cut-off criteria, to select only the more powerful coefficients and also the features that explain the most variance in the dataset.

On the other hand, for the mRMR and the ReliefF algorithms, it was necessary to implement manually the code that runs these two algorithms with an incremented number of features to select and the algorithms stop when the classification performance was not improved since the last iteration. The dimensionality reduction block is represented in the Appendix IV.

4.4.5 Data Labeling

With respect to the pretended supervised classification and in order to the model classify the data more accurately, the batch of data should be labeled in agreement with the posture or the posture transition associated. Thus, these labeled data become the ground truth on which the model bases its decisions for the posterior posture classifications. Each sample will have an associated label, as represented in the Table 8.

Table 8 - Different classes designation.

Label	Designation	
1	Standing	Static Postures
2	Back Frontal Bending	
3	Neck Frontal Bending	
4	Working Overhead	
5	Left Back Lateral Bending	
6	Right Back Lateral Bending	
7	Standing – Back Frontal Bending	Posture Transitions
8	Back Frontal Bending – Standing	
9	Standing – Left Lateral Back Bending	
10	Left Lateral Back Bending – Standing	
11	Standing – Neck Frontal Bending	
12	Neck Frontal Bending – Standing	
13	Standing – Working Overhead	
14	Working Overhead – Standing	
15	Standing – Right Lateral Back Bending	
16	Right Lateral Back Bending	

4.4.6 Model Building

After calculating the most important features in the *Dimensionality Reduction* stage, the user has to fill the parameters of the *Learning Algorithm* block. Firstly, the user has to select one of the following classification algorithms present in the framework:

- Support Vector Machines (SVM)
- Ensemble of Decision Trees (TreeBagger)
- K-Nearest Neighbours (KNN)
- Discriminant Analysis (both Linear and Quadratic algorithms)
- Feedforward Neural Network (FNN)
- Convolutional Neural Network (CNN)

All the aforementioned learning algorithms were chosen due to their prevalence in the previous motion recognition studies (see Chapter 3).

With regard to the SVM learning algorithm, this can be tested with three different kernels: (i) Linear, (ii) Quadratic and (iii) Gaussian. In addition, the SVM learning algorithm was applied using the OVO (one-vs-one) strategy, where a model is trained for each pair of labels/classes. This strategy is much less sensitive to imbalanced datasets but is much more computationally demanding.

Therefore, regarding the KNN, this algorithm can be tested with the neighbor distances equal weighted (Regular KNN) or with the neighbor distances squared inverse weighted (Weighted KNN).

After selecting the pretended classification algorithm, the users have to fill a check box if they want that the framework applies the hyperparameter optimization. Therefore, all classification algorithms could pass through the tuning of the hyperparameters, with the exception of the DA models, due to the robustness of the default parameters provided by the MATLAB software. Also, the CNN models does not pass through the hyperparameters optimization.

Next, when the KNN is used, the number of nearest neighbors (*k*) is tuned. The *k* value is incrementally increased, starting with $k = 1$, and the algorithm only saves the *k* value that provided the best model's performance. In the case of the SVM learning algorithm with the linear or the quadratic kernels, only the Box Constraint (*c*) hyperparameter is optimized. For such, the model is evaluated using every different *c* value, defined as 2^X, where X varies from [-10:10]. When the Gaussian kernel of the SVM model is used, both the Box Constraint (*c)* and the Kernel Scale are tuned, with resource to a gridsearch approach, where each possible pair of these two hyperparameters is tested and the one that yields the best performance is chosen.

When the Ensemble of Decision Trees (*TreeBagger*) is used, the number of trees is tuned by testing the model performance with the values ranging from 10 to 300, with an increment of 10. This range of values was chosen since, as present in literature, when the dataset is small, it is commonly to use between 10 and 100 trees and, when the dataset is large, it is possibly to use up until 300 trees (Sagi, O., & Rokach, 2018).

In the case of the FNN, it consists of three layers (input, hidden and output) and the number of neurons in the hidden layer is tested, always with the number of epochs equal to 30. The number of neurons tested vary from 1 to 100 neurons with an increment of 5. Also, it is important to refer that the transfer function chosen, for both the input and the hidden layers, was the Logarithmic Sigmoid. Additionally, for the output layer, the transfer function chosen was the Linear. To sum up, the training function used to update the weights and the bias states was the Levenberg-Marquardt method.

With regard to the CNN, the input has to be a matrix with 18 (N_0) rows (corresponding to the accelerations of the 3 axes of each one of the 6 accelerometers) and 25 (L_0) columns, that represents the number of samples to describe the different motions. The 25 number was chosen due to the fact that it is possible that the individuals perform some postures transitions in the corresponding time ($^1/_4\,s$). With respect to the CNN architecture, it consists of two convolutional layers, each one followed by a Max-Pooling Layer, in order to downsample the feature maps by a factor (*r*) *of* 2. Then, a fully connected layer with 16 neurons (same number of classes to classify) is applied and next, a Dropout Layer with $p = 0.5$, a Soft-Max Layer and a Classification Layer are applied. With respect to the convolutional layers, the first one has 32 (N_1) features maps, while the second one has 64 feature maps (N_2).

For a better analysis, Figure 13 represents the CNN architecture.

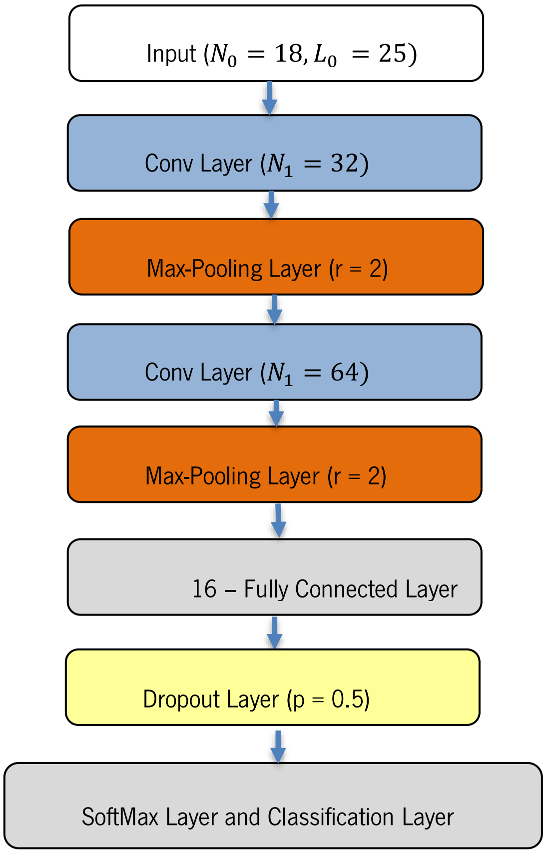


Figure 13 - CNN Architecture.

To sum up, in the *Learning Algorithm* block (see Appendix V), the user has to select the number of folds (*k*) that will be used in the 10 repeated k-fold cross validation. Consequently, for example, if the user chose a k = 2, two classification models will be evaluated and the mean performance will be calculated, for the repeated 10 times.

4.4.7 Model Evaluation

After the classification model is created, the cross validation has an important contribution since, it allows the calculation of a performance metric that evaluate the model's generalization capacity. Therefore, the cross validation is important for the comparison of the resulting models that have been built using different methods on any of the pipeline's phases. It is also important to compare models that were built with different hyperparameters.

Summing up, if the intention is to report the final model's performance, performance is evaluated using 10 repeated 20-fold cross validation. On the other hand, if the goal is to compare the different classification models obtained with the application of different algorithms, the evaluation is performed with resource to a 10 repeated 2-fold cross validation. After the CV is performed, the MCC and the ACC metrics are presented in the build pipeline, as well as, the possibility of visualizing one confusion matrix that resulted from the best iteration of the CV (see Appendix VI and VII).

5. Procedure Tests

The build posture recognition model construction framework is organized in several blocks/stages that are sequentially followed, as represented in the Figure 14.

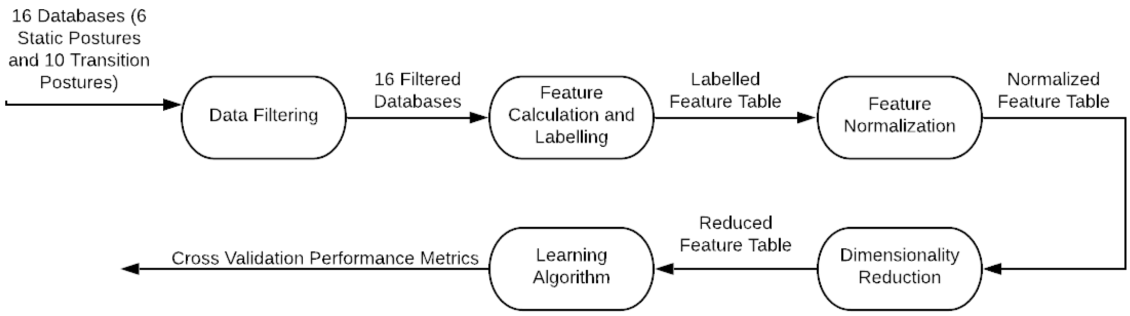


Figure 14 - Different blocks of the developed framework.

In each of these blocks, many algorithms or parameters can be tested to achieve different machine learning models. In this way, the present chapter has, as the main objective, the comparison between the model's performance obtained with the different algorithms/parameters.

Before presenting the results, it is important to note that each model evaluation was done using a 10 repeated 2-fold CV, and the MCC metric was chosen as the representative metric, due to the fact that the data are unbalanced (some classes/labels have fewer number of observations, namely the posture transitions). To exemplify the usability of the MCC metric, if the posture transitions (that are in less number) are incorrectly classified and the static postures (that are in higher number) are correctly classified, the accuracy metric will remain with a high value that will not allow the comparison of the different models. On the other hand, the MCC metric will be affected by the above mentioned, due to its good reliability when the dataset is unbalanced.

5.1 Feature Calculation

After filtering all the data, the pipeline shows the *Feature Calculation* block. In this stage, the user has to fill a box with the number of samples present in each time window. In this block, it is very important to select the most appropriate size of the time windows, in order to better represent the static postures and the posture transitions.

5.1.1 Methods

With regard to the window's sizes, six different numbers of samples were tested: (i) 10 samples, equal to 100 milliseconds, (ii) 15 samples, equal to 150 milliseconds, (iii) 25 samples, equal to 250 milliseconds; (iv) 50 samples, equal to 500 milliseconds; (v) 75 samples, equal to 750 milliseconds; (vi) 100 samples, equal to 1 second. All the previous conversions are based on a 100 Hz sampling rate, used for the data collection. Regarding the feature calculation, all the features (described in the section 4.4.2) were extracted from the six accelerometers and the normalization technique chosen was the z-score method (described in the section 4.4.3).

Regarding the *Dimensionality Reduction* stage, no feature selection algorithm was applied due to the fact that in these tests the main objective is to compare the impact of the different window's lengths in the classification performance. Finally, the learning algorithm selected to test the different lengths of the time windows was the KNN with the default parameters, such as, $k = 1$, equal distance weight between neighbors or *Euclidean* as the distance metric. KNN was chosen because it is trained and tested quickly, and provides reasonable results.

As mentioned before, the MCC metric was used to compare the impact of the different window's length and a 10 repeated 2-fold CV was also used.

5.1.2 Results

The performance results of the models obtained with the different time windows are represented in Table 9. The values next presented represents the MCC mean value and the error range that is obtained adding the standard deviation value and subtracting the same value, respectively. The results are segregated by the different length of the time windows.

Table 9 - Average recognition model's performance with the different time windows.

Window's Length (Samples)	MCC Mean Value and Error
10	0.978 ± 0.03
15	0.975 ± 0.05
25	0.971 ± 0.06
50	0.940 ± 0.19
75	0.934 ± 0.09
100	0.913 ± 0.32

5.1.3 Discussion

Firstly, it is important to note that, since only features based on mathematical operations were extracted from the acceleration signals, this task was not highly computationally demanding. Then, the results of the posture recognition models show that segregating the data into windows with 10 samples, 15 samples or 25 samples are the better options to correctly classify the postures and the posture transitions, being that, each one of the three aforementioned window's length provide similar recognition performances. Through the analysis of the Table 9, it is also possibly to infer that, segregating the data into time windows with 50 samples, 75 samples or 100 samples, the performance of the classifier decreases. These results could be explained by the fact that some individuals could develop the posture transitions in less than half a second, more specifically, the transition between the standing and the neck frontal bending postures. Consequently, segregating the data into time windows of more than 50 samples will not be the best option because it is possible to have in the same time window more than one posture or posture transition.

Thus, the better option is to segregate the data into time windows of 25 ($^1/_4$s) samples because it is not important to classify the individual's posture at each 100 milliseconds or 150 milliseconds.

Summing up, in the further tests, features will be calculated from time windows with 25 samples.

5.2 Data Normalization

As referred in the section 2.4, the data normalization is a key step in the data pre-processing stage. Data normalization techniques bring the variables to a common scale, allowing them to be more fairly compared and diminishing the time required to the classification model be trained and tested.

Therefore, there are several normalization techniques that, depending on the data and on the classification model, could provide a better or a worse performance.

Due to the aforementioned, tests had be done to find the most proper method to be used on the building of the different classification models. The following sections describe the tests developed, as well as, the different classification model's performance associated at each different normalization technique.

5.2.1 Methods

In order to analyze the effects of the data normalization on the performance of the different classification models, these were trained and evaluated with data from all subjects normalized with the four different normalization techniques. With the exception of the CNN learning algorithm, all features were extracted from time windows of 25 samples, and no hyperparameter optimization was done, so the default values provided by the MATLAB software were used. With respect to the CNN classifier, the feature matrix (that goes through the learning algorithm panel) consisted of a matrix of 18 rows (each one of the three axis of the six accelerometers) and 25 columns that represent the number of samples to describe the performed motion (in this case, 25 samples its equal to 250 milliseconds.

The analyzed normalization techniques were the (i) Scaling normalization, the (ii) Z-Score normalization, the (iii) Centering normalization and the (iv) Min-Max normalization, all described in the section 2.5. Also, the results with no normalization applied are next reported.

5.2.2 Results

Firstly, Table 10 shows the performance results for the SVM classification models with the Linear kernel, the Quadratic kernel and the Gaussian kernel.

Table 10 - Performance results of the SVM models, with the different normalization techniques.

	MCC Mean Value and Error		
Normalization Technique	Linear SVM	Quadratic SVM	Gaussian SVM
None	0.915 ± 0.11	0.949 ± 0.09	0.917 ± 0.12
Centering	0.913 ± 0.09	0.9451 ± 0.09	0.918 ± 0.14
Scaling	0.948 ± 0.15	0.959 ± 0.173	0.808 ± 0.053
Z-Score	0.948 ± 0.13	0.971 ± 0.06	0.808 ± 0.08
Min-Max	0.930 ± 0.09	0.963 ± 0.11	0.955 ± 0.11

Then, Table 11 shows the performance results for the TreeBagger model, with the number of trees equal to 50.

Table 11 - Performance results of the TreeBagger model, with the different normalization techniques.

	MCC Mean Value and Error
Normalization Technique	TreeBagger
None	0.960 ± 0.12
Centering	0.960 ± 0.10
Scaling	0.959 ± 0.15
Z-Score	0.959 ± 0.09
Min-Max	0.961 ± 0.10

Next, Table 12, reports the performance results of the Regular KNN model and the weighted KNN model (Squared Inverse weighted).

Table 12 - Performance results of the KNN models, with the different normalization techniques.

	MCC Mean Value and Error	
Normalization Technique	Regular KNN	Weighted KNN
None	0.921 ± 0.14	0.919 ± 0.12
Centering	0.921 ± 0.17	0.920 ± 0.11
Scaling	0.967 ± 0.07	0.966 ± 0.08
Z-Score	0.966 ± 0.06	0.966 ± 0.05
Min-Max	0.943 ± 0.08	0.943 ± 0.12

Table 13 shows the performance results for the Linear DA and the Quadratic DA, using all the default parameters provided by the MATLAB software.

Table 13 - Performance results of the DA models, with the different normalization techniques.

	MCC Mean Value and Error	
Normalization Technique	Linear DA	Quadratic DA
None	0.904 ± 0.09	0.936 ± 0.10
Centering	0.903 ± 0.10	0.933 ± 0.09
Scaling	0.902 ± 0.11	0.935 ± 0.08

Z-Score	0.903 ± 0.09	0.935 ± 0.09
Min-Max	0.902 ± 0.09	0.933 ± 0.13

Finally, Table 14 reports the performance results of the Feedforward Neural Network (FNN) and the Convolutional Neural Network (CNN).

Table 14 - Performance results of the implemented networks, with the different normalization techniques.

	MCC Mean Value and Error	
Normalization Technique	FNN	CNN
None	0.818 ± 0.24	0.911 ± 0.34
Centering	0.842 ± 0.21	0.912 ± 0.29
Scaling	0.808 ± 0.25	0.937 ± 0.25
Z-Score	0.803 ± 0.32	0.938 ± 0.21
Min-Max	0.804 ± 0.35	0.901 ± 0.29

5.2.3 Discussion

In a generalized way, analyzing the results, it is possible to note that, the Min-Max and the Z-Score normalization are the best techniques. This is in agreement with earlier researches that have used the Z-Score normalization (Begg, R., & Kamruzzaman, 2005) or the Min-Max normalization (Novak, D., Goršič, M., Podobnik, J., & Munih, 2014).

Due to the fact that the data normalization techniques are not very referred in the majority of the motion recognition studies, the comparison between the results obtained in the present section becomes a difficult task.

Nonetheless, with respect to the obtained individual performance results, it is possible to infer that the performance of the DA models (both linear and quadratic) is not affected by the different normalization techniques. The aforementioned could be explained by the fact that the DA models work by reorienting the data in agreement with the feature's axis (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007). Also, regarding the TreeBagger model, it is possible to infer equally that its performance varies very little with the different normalization techniques.

Therefore, with respect to the Regular and the Weighted KNN models, the Z-Score and the Scaling techniques provided the best performances. On the other hand, using no normalization, using the Centering normalization and using the Min-Max normalization, the performance of the models is worst.

With respect to the SVM models, using the Linear and the Quadratic kernels, the most promising normalization technique is the Z-Score while, using the Gaussian kernel, the model's performance with the Z-Score normalization is worse. Thus, using the Gaussian kernel, the best normalization technique is the Min-Max. Regarding the FNN performances, it is possible to note that it is not very affected by the different normalization techniques. However, the normalization technique that provided the best results was the Centering method.

In the case of the CNN learning algorithm, it is possible to conclude that applying the Min-Max or the Centering normalizations, the classifier's performance is similar to the test where none normalization was applied. On the other hand, using the Scaling or the Z-Score normalizations, the CNN performance is increased.

Summing up, Table 15 reports the conclusions about the best normalization technique for each different learning algorithm. For the remainder tests, data normalization will be applied in agreement with the Table 15.

Table 15 - Best normalization technique, by learning algorithm.

Learning Algorithm	Best Normalization Technique
Linear and Quadratic DA	Min-Max
TreeBagger	Min-Max
SVM with Gaussian Kernel	Min-Max
SVM with Quadratic Kernel	Z-Score
SVM with Linear Kernel	Z-Score
Regular and Weighted KNN	Z-Score
CNN	Z-Score
FNN	Centering

5.3 Feature Selection or Extraction

As referred in the section 2.6, the datasets can be constituted by relevant, irrelevant and, also, redundant features. In this way, feature selection and extraction algorithms have an utmost importance since, they reduce the number of features to be used, prevent the overfitting and improve the classification model's generalization. Due to the fact that there is no perfect feature selection or extraction algorithm for all the applications, different algorithms should be tested in order to determine what is the best dimensionality reduction technique to use in the present dissertation.

5.3.1 Methods

All the following tests were performed using the Regular KNN classifier with $k = 1$, with all the features extracted from the time windows of 25 samples and the dataset normalized by the Z-Score technique.

Three different feature selection algorithms were tested including the *mRMR* algorithm, the *ReliefF* algorithm and the sequential forward selection (*SFS*) algorithm. In addition, the feature extraction algorithm *PCA* and none dimensionality reduction algorithm, were also tested.

5.3.2 Results

Table 16 reports the model's performance using each one of the different dimensionality reduction techniques.

Table 16 - Regular KNN model's performance, using the different dimensionality reduction techniques.

Dimensionality Reduction Technique	Number of selected features	MCC Mean Value and Error of the Regular KNN model
None	132	0.966 ± 0.10
ReliefF	50	0.961 ± 0.06
mRMR	50	0.969 ± 0.05
SFS	42	0.972 ± 0.08
PCA	21	0.942 ± 0.11

It is important to note that the *PCA* and the *SFS* algorithms stop when the best performance is obtained. On the other hand, with respect to the *mRMR* and *ReliefF* feature selection algorithms, the user has to indicate to the algorithm, how many best features it should select. In this way, these two algorithms were applied with different number of features, starting with 5 features and ending with 132 features (total number of features), with an increment of 5 in each iteration. In addition, the aforementioned algorithm also stops if the performance of the classifier does not improve since the previous iteration, that is, when a local maximum is reached.

5.3.3 Discussion

Analyzing the Table 16, the Relief algorithm and the PCA method, provided the worst classification performances of the Regular KNN. The aforementioned results are in agreement with the theoretical concepts, since the ReliefF algorithm does not reduce the redundancy of the features and, with respect to the PCA algorithm, if the most important characteristics are not related with the variance between the variables, this algorithm could not be a very useful algorithm. It is also important to note that PCA was applied to the dataset composed by all the features, extracting the most important components. Therefore, the Horn's Parallel Analysis was applied as the cut-off criteria, selecting the most powerful coefficients and also the features that explain the most variance in the dataset. Also through the analysis of the Table 16, the PCA algorithm with the Horn's PA as the cut-criteria, selected the 21 features that most contribute to explain the variance in the dataset, that is, with respect to the first principal component.

Then, with respect to the mRMR algorithm, it provided the second best model's performance, only surpassed by the SFS feature selection method. This is in line with the theoretical expectation, given that this algorithm is very similar to the ReliefF algorithm, with the advantage of eliminating the features that are redundant. For such, the algorithm is based on the principle of mutual information (MI) between the different features, selecting only the most important features that have the minimum redundancy.

After mRMR, the dimensionality reduction technique that provided the best recognition performance, was the SFS feature selection algorithm. This is also in line with the theoretical concepts since, the *SFS* algorithm uses the classification model to obtain the best subset of features. In addition, the performance of the classifier was improved when compared to the test where all the features were used, with the advantage of making the processes of training and testing, less time consuming. On the other hand, the only disadvantage of the present method is related with its high computationally weight.

Summing up, the *SFS* and mRMR are the most promising dimensionality reduction techniques, due to the fact that were the feature selection algorithms that provided the best classification model's performances. Thus, the mRMR will be used for the construction of the best classification model due to its much less computationally weight, when compared with the SFS algorithm.

5.4 Classification Algorithm Selection

As previously referred in the present dissertation, there are two main families of the classification algorithms, namely, the supervised algorithms and the unsupervised algorithms. However, the present research only focuses on the supervised learning algorithms being the K-Nearest Neighbors, the Support Vector Machines, the ensemble of Decision Trees or the feed-forward Neural Networks, some examples.

The present chapter has, as the main function, report the results of the tests performed with the different learning algorithms used in the present study and, the main goal, is to select the learning algorithm that better distinguish the different postures and posture transitions.

5.4.1 Methods

Using the all the data collected by the different 50 subjects, obtained with time windows of 25 samples and normalized in agreement with the pretended classification algorithm (as concluded in the section 5.2), several learning algorithms were tested, namely, the Linear and Quadratic Discriminant Analysis, the Regular and the Weighted KNN, the Linear, Quadratic and Gaussian SVM, the TreeBagger, the Feedforward Neural Network (FNN) and the Convolutional Neural Network (CNN).

Therefore, none dimensionality reduction technique was applied and, consequently, the total data set of features was used to train and evaluate the model's performance.

With the objective of increasing the model's performance, with the exception of the DA models, all the other learning algorithms went through the hyperparameters optimization. As explained in the section 4.4.6, depending on the classification model, the hyperparameters are tuned with an iterative approached, that is, the model's performance is tested using the different hyperparameter values, within a pre-defined range. Summing up, only the values that provide the best performance are saved.

5.4.2 Results

Table 17 reports the performance results of each different learning algorithm, resulting from the 10 repeated 2-fold CV, as well as the average amount of time, in milliseconds, that each different learning algorithm takes to classify a single observation/time window. .

Table 17 - 10 repeated 2-fold CV results, for each learning algorithm.

Learning Algorithm	MCC Mean Value and Error	Classification Time (ms)
Linear SVM	0.946 ± 0.09	9.2
Quadratic SVM	0.973 ± 0.11	9.9
Gaussian SVM	0.967 ± 0.11	10.7
TreeBagger	0.962 ± 0.10	0.3
Regular KNN	0.966 ± 0.07	1.4
Weighted KNN	0.966 ± 0.08	1.5
LDA	0.904 ± 0.10	0.4
QDA	0.934 ± 0.15	0.4
FNN	0.828 ± 0.25	4.6
CNN	0.935 ± 0.18	0.3

5.4.3 Discussion

Through the analysis of the above presented results, it is possible to refer that the SVM learning algorithm with the Quadratic kernel and with the Box Constraint parameter equal to 0.5, performed better than the other learning algorithms. Due to the aforementioned, the SVM model with the Quadratic kernel will be used for the final model building. The only disadvantage associated to this learning algorithm, is the fact that it takes more time for the classification task (9.9 ms / observation), when compared to the other studied classifiers, with the exception of the Gaussian SVM (10.7 ms / observation).

The obtained results are in agreement with the previous literature since, for example, (Nurhanim et al., 2017) studied the effect of the different SVM kernels in the performance of their motion recognition system and they concluded that the polynomial kernel with order 2 (Quadratic kernel) is advantageous to predict five of the six postures and motions analyzed.

With respect to the LDA and the FNN learning algorithms, these were the models that had provided the worst results. With respect to the LDA, the weak results could be related to the simplicity of this classifier.

Also in agreement with the theoretical expectation with the information presented in the section 2.7, KNN models provided a good posture recognition performance and they had taken less time for the classification stage. However, these models provided a little bit worst results, when compared with the Quadratic SVM and the Gaussian SVM. It is also important to refer that for both the Regular and Weighted KNN models, the best number of neighbors, obtained with the hyperparameter optimization, was 1.

Also, the TreeBagger and the CNN learning algorithms provided good performance results. With respect to the TreeBagger learning algorithm, 110 trees were used, being this number the result of the hyperparameter optimization block. Then, instead the fact that the *CNNs* were initially developed for image applications, with the obtained results, it was proved that the *CNNs* could be applied for many different machine learning applications.

6. Human Posture Recognition Model Validation

The mail goal of the constructed pipeline/framework was the building and evaluation of different human posture recognition models. After performing tests, with the objective of concluding which techniques are advantageous in each of the phases of the pipeline, the present section reports the performance results for the obtained final posture recognition models to evaluate its robustness and fitness of use.

Firstly, the best final model, in the next section described, was evaluated using a 10 repeated 20-fold CV and the MCC metric was chosen to be the representative performance result of this evaluation. In second place, the best obtained model was evaluated when predicting data from an individual that performed an activity that includes the different postures used for the model's training.

Also, an example of the confusion matrix associated at each test will also be analyzed, providing important information like the classes/postures that were more difficult to classify.

6.1 Methods

Posteriorly to the calculation of the best techniques to be used in each phase of the constructed framework, the final human posture recognition model was built and tested. For such, firstly, features were calculated from time windows of 25 samples (as concluded in the section 5.1), feature vectors were normalized by the Z-Score normalization (as concluded in the section 5.2), *ReliefF* was used to select the most important features (as concluded in the section 5.3) and regarding the learning algorithm, SVM was used with the quadratic kernel (as concluded in the section 5.4). Then, the final model also passed through the hyperparameter optimization block. In this case, since the quadratic SVM model was used, only the Box Constraint (*c*) was tuned. Summing up, the model was built using data from all subjects.

As above mentioned, in addition to evaluate the performance of the model with the 10 repeated 20-fold CV, the classification model was also used to test its fitness when predicting the data obtained from an individual performing the test activity that includes all the postures used for the training stage. For such, important parameters like the mean or the standard deviation value of each feature vector and the features selected by the *mRMR* algorithm, were saved, in order to apply the same proceedings to the data coming from the individual that developed the test activity. The test activity protocol is next described.

1. 10 seconds in the **Standing** posture → 30 seconds in the **Working Overhead** posture → 10 seconds in the **Standing** posture.
2. 30 seconds in the **Back Frontal Bending (BFB)** posture → 10 seconds in the **Standing** posture.
3. 30 seconds in the **Left Back Lateral Bending (LBLB)** posture → 10 seconds in the **Standing** posture.
4. 30 seconds in the **Right Back Lateral Bending (RBLB)** posture → 10 seconds in the **Standing** posture.
5. 30 seconds in the **Neck Frontal Bending (NFB)** posture → 10 seconds in the **Standing** posture.

Due to the different tests developed, the results' section will be divided into two different sub-sections.

6.2 Results and Discussion

As explained previously, the evaluation of the best model was done in two different stages. For such, the present section will be divided in two different sub-sections. The first one reports the results obtained with the 10 repeated 20-fold CV, while the second one reports the classification model's performance when it is used to predict the data from individuals that developed the test activity.

6.2.1 10 Repeated 20-Fold CV

Table 18 shows the performance metrics obtained with the evaluation of the quadratic SVM model, resulting from the 10 repeated 20-fold CV. In addition, the Table 18, reports the most important parameters of the quadratic SVM model.

Table 18 - Results and important parameters of the 10 repeated 20- fold CV.

Model	Number of Used Features	Best Box Constraint	Mean ACC Value (%)	Mean MCC Value
Quadratic SVM	50	0.5	99.76	0.978

Since it was applied a 10 repeated 20-fold CV, 200 pairs of training data and test data were used to evaluate the classification model, and the average performance was calculated.

Therefore, 200 confusion matrixes were calculated being important to note that only the confusion matrix that had associated the better MCC value was saved and it is next presented in the Table 19.

Table 19 - Confusion matrix for the quadratic SVM model (St - Standing; BFB - Back Frontal Bending; NFB - Neck Frontal Bending; WO - Working Overhead; LBLB - Left Back Lateral Bending; RBLB - Right Back Lateral Bending).

Ground Truth / Predictions	St	BFB	NFB	WO	LBLB	RBLB	St to BFB	BFB to ST	St to LBLB	LBLB to St	St to NFB	NFB to St	St to WO	WO to St	St to RBLB	RBLB to St
St	1.0	0	0	0	0	0	0	0	0	0.059	0	0	0	0	0	0
BFB	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NFB	0	0	0.988	0	0.008	0	0	0	0	0	0	0	0	0	0	0
WO	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
LBLB	0	0	0.012	0	0.992	0	0	0	0.04	0	0	0	0	0	0	0
RBLB	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
St to BFB	0	0	0	0	0	0	0.963	0	0	0	0.053	0.0625	0	0	0.25	0
BFB to St	0	0	0	0	0	0	0.037	0.962	0	0	0.053	0	0	0	0	0
St to LBLB	0	0	0	0	0	0	0	0	0.88	0.118	0.053	0	0	0	0	0
LBLB to St	0	0	0	0	0	0	0	0	0.08	0.824	0	0	0	0	0	0
St to NFB	0	0	0	0	0	0	0	0	0	0	0.841	0.0625	0	0	0	0
NFB to St	0	0	0	0	0	0	0	0.038	0	0	0	0.875	0	0	0	0
St to WO	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
WO to St	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
St to RBLB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0
RBLB to St	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Then, the fifty best features selected by the *mRMR* algorithm are described in the Table 20.

Table 20 - Features extracted by the mRMR algorithm and the associated sensor's location.

Features	Sensor Location	Number of Features
Accel_Mod, Mean_y	S1 bone	2
Mean_y, Mean_x, RMS_y	T12 bone	3
RMS_y, IQR_y, Mean_x, Mean_y, Slope_x, Std_y, Roll, Pitch, Accel_Mod, Var_y	T4 bone	10
Mean_y, IQR_z, Mean_x, Yaw, Slope_y, RMS_y, Pitch, Slope_z, Mean_z, Std_x, Roll	Left Upper Arm	11
Std_z, Pitch, Accel_Mod, Slope_y, RMS_x, Std_x, Mean_y, Mean_z, RMS_y, IQR_y, Slope_x, Mean_x	Right Upper Arm	12
Roll, Std_y, Std_x, Mean_y, Accel_Mod, Std_z, RMS_y, IQR_x, Mean_x, IQR_y, Pitch, Slope_x	Nape	12

Analyzing the Tables 18 and 19, it is possible to refer that the recognition model built with the best algorithms tested in the section 5, classified the different postures and posture transitions with a very good performance. Also, it is important to note that only using the best 50 features from the total set of features, obtained with the application of the *mRMR* feature selection algorithm, the Quadratic SVM model recognized the different postures with an extremely high performance. The only errors of the model occur mainly when it is predicting the posture transitions, namely, the transitions between the standing and the left back lateral bending postures. Also, with respect to the transitions between the standing and the neck frontal bending postures, the Quadratic SVM model failed in some observations.

Then, regarding the features selected by the *mRMR* algorithm and analyzing the Table 20, it is possible to conclude which are the sensors that most contribute to distinguish the different classes, more specifically, the six postures and the 10 posture transitions. So, the most important sensors are, firstly, the sensor placed on the individual's nape and its importance could be explained by the fact that the postures and the transitions associated to the neck, are more felt in the nape accelerometer's data. Then, the sensors placed on the right and left upper arm were also important in the posture classification process, probably, due to the fact that the left and right lateral back bending postures produce significantly changes in these sensors data.

With respect to the three sensors placed on the individual's back (namely, on the S1 bone, T12 bone and T4 bone), analyzing the Table 20, it is possible to infer that the sensor placed on the T4 bone is the one that provides the most important information. The aforementioned is in line with the theoretical expectation since the back frontal and lateral bending postures are more felt on the uppermost part of the back. Concluding, if in a future application it is necessary to reduce the number of sensors placed on the individual's body, it is expected that the performance of the classification model will not be reduce without the use of the data provided by the sensors placed on the S1 and T12 bones.

6.2.2 Test Activity

In order to validate the proposed classification model, an activity that contain multiple sequencing postures was captured to examine the performance of the classifier in more real conditions. The activity developed by an individual was also video-recorded and based on this video, all observations/time windows are labeled. Consequently, the assigned postures and posture transitions serve as the ground truth for the validation test.

Since the best classification model was the Quadratic SVM, it was also possible to extract the posterior probabilities, that consist on the probabilities of each observation/time window belonging to the different classes. In the present validation, posterior probabilities consist of a table with dimensions n_o * n_c, being the first parameter the number of observations while the second one is the number of classes, in this case, 16, as explained in the section 4.4.5.

Therefore, in the Table 21, the results of the test activity classification, using the Quadratic SVM model, are represented.

Table 21 - Results of the test activity classification.

Class Type	Task	Number of Time Windows	Overall Correct	Overall Incorrect	ACC (%)	Overall ACC (%) and MCC value
Postures	Stand	219	217	2	99.09	
	BFB	104	104	0	100	
	NFB	112	112	0	100	
	WO	104	104	0	100	
	LBLB	98	98	0	100	
	RBLB	103	103	0	100	
Posture Transitions	Stand - BFB	14	8	6	57.14	98.77
	BFB - Stand	13	8	5	61.54	0.8944
	Stand - LBLB	11	2	9	18.18	
	LBLB - Stand	15	8	7	53.33	
	Stand - NFB	16	4	12	25	
	NFB - Stand	10	4	6	40	
	Stand - WO	15	7	8	46.67	
	WO - Stand	14	6	8	42.76	
	Stand - RBLB	10	6	4	60	
	RBLB - Stand	12	0	12	0	

Through the analysis of the Table 21, it is possible to refer that the classification model provided a good recognition performance, when predicting the data of the activity containing the different postures. Regarding the prediction of the static posture, the overall accuracy of the model rounds the 100%, showing that, the classification model very rarely fails when faced with time windows associated with static postures. On the other hand, the classification model has a worse performance when it is predicting the posture transitions. These results could be explained by the fact that the classification models were trained with much more static postures data, when compared with the posture transitions data.

To better examine the performance of the classification model when predicting the test activity data, Figure 15 represents, along the y-axis, both the predicted (color red) and the ground truth (color green) classes. If the predicted class is the same as the ground truth class, the bar will be green. Otherwise, if the predicted and the ground truth classes are not the same, the ground truth class will appear in the green color and the predicted class will appear in the red color. In addition, along the x-axis, it is represented the consecutive time windows, being important to note that, each one of them represents 25 samples ($^1/_4$s).

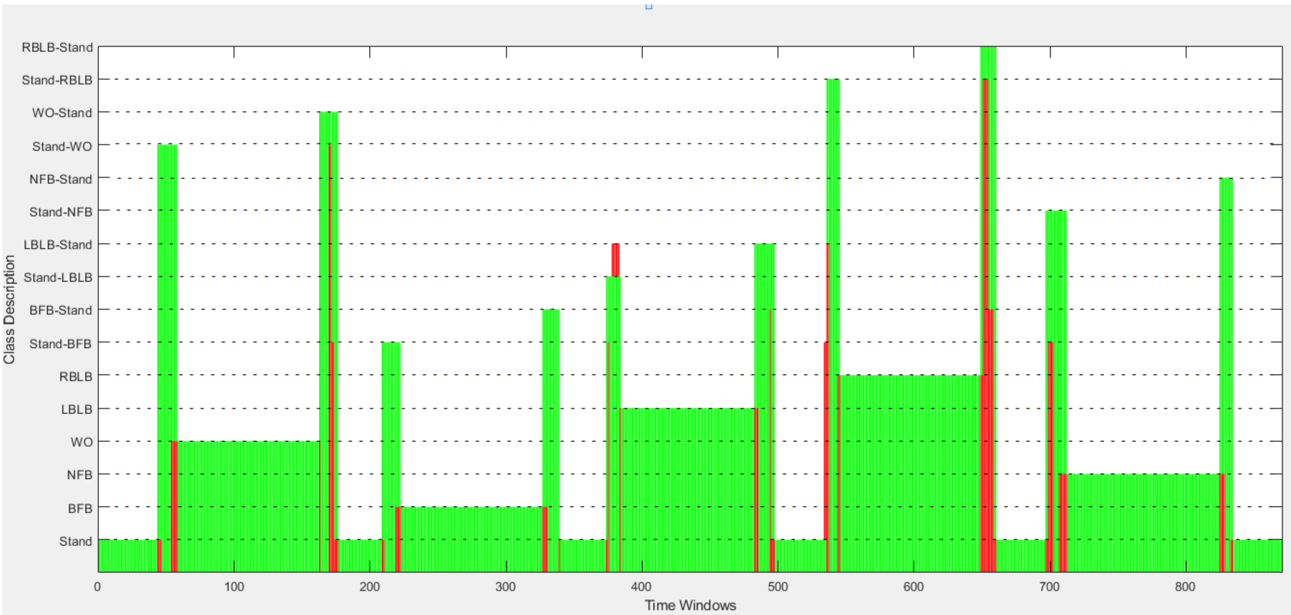


Figure 15 - Recognized posture versus the ground truth, for the test activity.

Based on the Figure 15 and on the Table 21, it can be verified that, for static postures, the classification model provided an accuracy of about 100%. Also with the analysis of the above present Figure, the classification model only fails when it is predicting the posture transitions, and most of the time, the main mistake of this is to classify the previous posture when we are at the beginning of the transition and to classify the next posture when we are in the final phase of the transition. The aforementioned classification errors were expected since the classifier, at a specific time, is close to the threshold that separates two different but similar classes. In addition, there are posture transitions that are worst classified by the model, namely, the transition between the right back lateral bending and the standing postures and the transition between the standing and the neck frontal bending postures.

7. Conclusions

In the present dissertation, several techniques and learning algorithms were tested with resource to the framework constructed, being the main goal of the aforementioned framework, the creation of classification models that are intended to be used to predict the human upper limb postures. The most important conclusions of the above mentioned procedures are next presented as well as some considerations to take into account in a future work.

7.1 Final Considerations

In order to obtain well-performing classification models, several algorithms were tested and compared in the different blocks of the constructed framework. Thus, with respect to the Feature Calculation block, it was concluded that segregating the acceleration data into time windows of 25 samples $({}^{1}/{}_{4}\, s)$ is advantageous to classify the different human upper limb postures and posture transitions.

In the normalization block, generally, the min-max technique demonstrated to be the most promising normalization technique. However, there are exceptions because some classification models perform better when the data is normalized with the Z-Score or the Centering techniques, for example.

In what concerns the dimensionality reduction block, it was concluded that the mRMR and the SFS algorithms are the most promising. Between the two algorithms, the mRMR was selected due to its less time taken to select the best features.

Next, the selected classification algorithm for the final model building was the Quadratic SVM. The final model constructed performed well when predicting the test activity data, more specifically, with an overall accuracy that rounds the 100% with respect to the static postures classification. On the other hand, the performance of the classification model decreases when it is predicting time windows that represents the posture transitions.

With respect to the best subset of features selected by the mRMR algorithm, it is important to note that it consists of 50 features and the majority of them are related to the sensors placed on the T4 bone, on the left and right upper arm and on the nape. Summing up, it is possible to infer that the sensors placed on the S1 and T12 bones don't provide important information about the human upper limb postures.

Summing up, compared to the static postures, the posture transitions are worst classified probably due to the fact that the classification model was trained with much more data related to the static postures, comparing to the posture transitions.

Research Question 1: Which are the best wearable sensor's locations to better distinguish the different human upper limb postures?

From the six locations tested, sensors on the T4 bone, on left and right upper arms and on nape, are the best. Sensors on the T12 and S1 bones provide weak information. Since the postures studied in the present dissertation are present in the majority of the different workplace tasks, the best aforementioned sensor locations could be used in further studies.

Research Question 2: Which is the best data normalization method?

Generally, the Min-Max technique provided the best results. However, there are some classification algorithms that present best results with other data normalization technique like the Z-Score, for example. All conclusions are presented in the section 5.2.3.

Research Question 3: Which are the most promising dimensionality reduction algorithms?

mRMR and SFS provided the best recognition performance. mRMR was chosen due to its less computationally weight.

Research Question 4: Which is the best learning algorithm, with respect to the human upper limb postures classification?

SVM with the Quadratic kernel provided the best results, followed by the SVM with the Gaussian kernel.

Research Question 5: Which is the best metric to compare the performance of the different classification models?

Due to the fact that the data are unbalanced with respect to the different classes, the MCC (Mathews Correlation Coefficients) metric is used.

7.2 Future Work

In the future work, it could be important to test the integration of other sensory information like the EMG sensors, the gyroscopes or the magnetometers. With respect to the data acquisition system, in a future work, its robustness should be improved in order to improve the performance of the computational techniques applied to the data provided by the system. Therefore, different data collection protocols followed by the volunteers should also be tested as well as extract different features from the sensors.

Regarding the prediction of the posture transitions, a deeper literature research about the techniques and the features calculated from the sensors should also be done, due to the fact that the main errors occurred when the classification model was predicting the posture transitions data.

Finally, the testing of the classification model in real-time with resource to an inertial system that has incorporated a microcontroller should also be done, in order to understand if the constructed classification model could be used to give feedback to the workers in real-time.

References

A. Parent-Thirion, E. F. Macías, J. Hurley, G. V. (2007). *Fourth European Working Conditions Survey. Context* (Vol. 2008). https://doi.org/www. eurofound.eu.int/ewco/surveys/EWCS2005/index.htm.

Agnès Parent-Thirion, Isabella Biletta, Jorge Cabrita, Oscar Vargas, Greet Vermeylen, A. W. and, & Wilkens., M. (2017). *6th European Working Conditions Survey*.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37–66.

Alpoim, L., da Silva, A. F., & Santos, C. P. (2019). Human Activity Recognition Systems : State of Art. In *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG) (pp. 1-4). IEEE.* (pp. 1–4).

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*, 40–79. https://doi.org/10.1214/09-SS054

Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2015). Physical human activity recognition using wearable sensors. *Sensors (Switzerland)*, *15*(12), 31314–31338. https://doi.org/10.3390/s151229858

Avci, A., & Bosch, S. (2010). Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. *23rd International Conference on Architecture of Computing Systems (ARCS)*, (January), 1–10. https://doi.org/10.1007/978-3-319-13105-4_17

Bayat, A., Pomplun, M., & Tran, D. A. (2014). A Study on Human Activity Recognition Using Accelerometer Data from Smartphones. *Procedia - Procedia Computer Science*, *34*, 450–457. https://doi.org/10.1016/j.procs.2014.07.009

Begg, R., & Kamruzzaman, J. (2005). A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data. *Journal of Biomechanics*, *38*(3), 401–408. https://doi.org/10.1016/j.jbiomech.2004.05.002

Belkic, K., & Savic, C. (2008). The occupational stress index–An approach derived from cognitive ergonomics applicable to clinical practice. *Scandinavian Journal of Work, Environment & Health*, *34*(6), 169–176.

Bidargaddi, N., Sarela, A., Klingbeil, L., & Karunanithi, M. (2007). Detecting walking activity in cardiac rehabilitation by using accelerometer. In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information, IEEE, Melbourne, Australia* (pp. 555–560). https://doi.org/10.1109/ISSNIP.2007.4496903

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*(1–2), 245–271.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Costa, L. O. L. (2018). Wearable sensory systems for real-time human gait analysis (Doctoral dissertation, Universidade do Minho).

Cuesta-vargas, A. I., & Williams, J. (2014). Inertial sensor real-time feedback enhances the learning of cervical spine manipulation : a prospective study, 1–5.

Curone, D., Bertolotti, G. M., Cristiani, A., Secco, E. L., & Magenes, G. (2010). A Real-Time and Self-Calibrating Algorithm Based on Triaxial Accelerometer Signals for the Detection of Human Posture and Activity. *IEEE Transactions on Information Technology in Biomedicine*, *14*(4), 1098–1105.

Da Costa, B. R., & Vieira, E. R. (2010). Risk factors for work-related musculoskeletal disorders: A systematic review of recent longitudinal studies. *American Journal of Industrial Medicine*, *53*(3), 285–323. https://doi.org/10.1002/ajim.20750

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*(2), 185–205.

Dybowski, R., & Gant, V. (Eds. . (2001). *Clinical applications of artificial neural networks*. (Cambridge

University Press., Ed.). https://doi.org/10.1017/CBO9780511543494

Foerster, F., Smeja, M., & Fahrenberg, J. (1999). Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, *15*(5), 571–583. https://doi.org/10.1016/S0747-5632(99)00037-0

Ghasemzadeh, H., Loseu, V., Guenterberg, E., & Jafari, R. (2009). Sport training using body sensor networks: a statistical approach to measure wrist rotation for golf swing. *Proceedings of the 4th International ICST Conference on Body Area Networks*, (April). https://doi.org/10.4108/ICST.BODYNETS2009.6035

Gjoreski, H., & Gams, M. (2011). Activity/Posture recognition using wearable sensors placed on different body locations. In *Proceedings of (738) Signal and Image Processing and Applications, Crete, Greece, 2224, 716724*.

Goswami, S., Haldar, P., & Sahu, S. (2013). An ergonomic study of postural stress of nurses working in orthopedic wards. *International Journal of Occupational Safety and Health*, *3*(1), 26–31.

Gouiaa, R., & Meunier, J. (2017). Learning cast shadow appearance for human posture recognition Learning cast shadow appearance for human posture recognition. *Pattern Recognition Letters*, 54–60. https://doi.org/10.1016/j.patrec.2017.06.023

Gyorbíró, N., Fábián, Á., & Hományi, G. (2009). An activity recognition system for mobile phones. *Mobile Networks and Applications*, *14*(1), 82–91. https://doi.org/10.1007/s11036-008-0112-y

Hammerla, N. Y., Halloran, S., & Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *ArXiv Preprint ArXiv*.

He, Z., & Jin, L. (2009). Activity recognition from acceleration data based on discrete consine transform and SVM. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, (October), 5041–5044. https://doi.org/10.1109/ICSMC.2009.5346042

Jasiewicz, J. M., Treleaven, J., Condie, P., & Jull, G. (2007). *Wireless orientation sensors : Their suitability to measure head movement for neck pain assessment*. *Manual therapy* (Vol. 12). https://doi.org/10.1016/j.math.2006.07.005

Jolliffe, I. T. (2011). *Principal Component Analysis, Second Edition*. *In International encyclopedia of statistical science*. https://doi.org/10.2307/1270093

Juang, C., Member, S., & Wang, P. (2014). An Interval Type-2 Neural Fuzzy Classifier Learned through Soft Margin Minimization and its Human Posture Classification Application. *IEEE Transactions on Fuzzy Systems*, *23*(5), 1474–1487. https://doi.org/10.1109/TFUZZ.2014.2362547

Karantonis, D. M., Member, S., Narayanan, M. R., Mathie, M., Lovell, N. H., Member, S., & Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, *10*(1), 156–167.

Kee, D., & Karwowski, W. (2001). LUBA : An Assessment Technique for Postural Loading on the Upper Body Based on Joint Motion Discomfort and Maximum Holding Time LUBA : an assessment technique for postural loading on the upper body based on joint motion discomfort and maximum holding time. *Applied Ergonomics*, *32*(4), 357–366. https://doi.org/10.1016/S0003-6870(01)00006-0

Khalid, S., Khalil, T., & Nasreen, S. (2015). A survey of feature selection and feature extraction techniques in machine learning. In *In 2014 Science and Information Conference IEEE* (pp. 372–378).

Khan, A. M., Lee, Y. K., Lee, S. Y., & Kim, T. S. (2010). A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE Transactions on Information Technology in Biomedicine*, *14*(5), 1166–1172. https://doi.org/10.1109/TITB.2010.2051955

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160*(1), 3–24. https://doi.org/10.1007/s10751-016-1232-6

Kumar, D. P., Amgoth, T., Sekhara, C., & Annavarapu, R. (2019). Machine learning algorithms for wireless sensor networks : A survey. *Information Fusion*, *49*(April 2018), 1–25. https://doi.org/10.1016/j.inffus.2018.09.013

Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, *12*(2), 74. https://doi.org/10.1145/1964897.1964918

Lara, D., & Labrador, M. a. (2013). A Survey on Human Activity Recognition using Wearable Sensors, *15*(3), 1192–1209.

Lara, Ó. D., Prez, A. J., Labrador, M. A., & Posada, J. D. (2012). Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, *8*(5), 717–729. https://doi.org/10.1016/j.pmcj.2011.06.004

Ledesma, R. D., & Valero-Mora, P. (2007). Determining the Number of Factors to Retain in EFA : an easy-to- use computer program for carrying out Parallel Analysis. *Practical Assessment, Research & Evaluation*, *12*(2), 1–11.

Mantyjarvi, J., Himberg, J., & Seppanen, T. (2001). Recognizing human motion with multiple acceleration sensors. *In Systems, Man, and Cybernetics, 2001 IEEE International Conference*, *2*, 747–752.

Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. (2006). Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, 113–116. https://doi.org/10.1109/BSN.2006.6

Mukhopadhyay, S. C. (2015). Wearable Sensors for Human Activity Monitoring : A Review. *IEEE Sensors Journal*, *15*(3), 1321–1330. https://doi.org/10.1109/JSEN.2014.2370945

Nath, N. D., Chaspari, T., & Behzadan, A. H. (2018). Automated ergonomic risk monitoring using body-mounted sensors and machine learning. *Advanced Engineering Informatics*, *38*(June), 514–526. https://doi.org/10.1016/j.aei.2018.08.020

Novak, D., Goršič, M., Podobnik, J., & Munih, M. (2014). Toward real-time automated detection of turns during gait using wearable inertial measurement units. *Sensors*, *14*(10), 18800–18822. https://doi.org/10.3390/s141018800

Nurhanim, K., Elamvazuthi, I., Izhar, L. I., & Ganesan, T. (2017). Classification of human activity based on smartphone inertial sensor using support vector machine. *2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation, ROMA 2017*, *2017–Decem*, 1–5. https://doi.org/10.1109/ROMA.2017.8231736

Paoli, P. (1996). *Second European Survey on Working Conditions*.

Pärkkä, J., Ermes, M., Korpipää, P., Mäntyjärvi, J., Peltola, J., & Korhonen, I. (2006). Activity classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine : A Publication of the IEEE Engineering in Medicine and Biology Society*, *10*(1), 119–128. https://doi.org/10.1109/TITB.2005.856863

Paul, J., Dario, V., & Walls, T. A. (2012). Human motion recognition using a wireless sensor-based wearable system. *Personal and Ubiquitous Computing*, *16*(7), 897–910. https://doi.org/10.1007/s00779-011-0455-4

Peppoloni, L., Filippeschi, A., Ruffaldi, E., & Avizzano, C. A. (2016). International Journal of Industrial Ergonomics (WMSDs issue) A novel wearable system for the online assessment of risk for biomechanical load in repetitive efforts. *International Journal of Industrial Ergonomics*, *52*, 1–11. https://doi.org/10.1016/j.ergon.2015.07.002

Pirttikangas, S., Fujinami, K., & Nakajima, T. (2006). Feature selection and activity recognition from wearable sensors. *In International Symposium on Ubiquitious Computing Systems*, 516–527. https://doi.org/10.1007/11890348_39

Punnett, L., & Wegman, D. H. (2004). Work-related musculoskeletal disorders: The epidemiologic evidence and the debate. *Journal of Electromyography and Kinesiology*, *14*(1), 13–23.

https://doi.org/10.1016/j.jelekin.2003.09.015

S. B. Kotsiantis, D. K. and P. E. P. (2006). Data Preprocessing for Supervised Leaning. *International Journal of Computer Science*, *1*(2), 111–117.

Sagi, O., & Rokach, L. (2018). Ensemble learning : A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1249. https://doi.org/10.1002/widm.1249

Sarcevic, P., Kincses, Z., & Pletl, S. (2014). Wireless sensor network based movement classification using wrist-mounted 9DOF sensor boards. In *Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium* (pp. 85–90).

Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A., & Friedman, R. (2007). Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor.

Taunyazov, T., Omarali, B., & Shintemirov, A. (2016). A Novel Low-Cost 4-DOF Wireless Human Arm Motion Tracker. In *In Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE International Conference* (pp. 157–162).

Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine Recognition of Human Activities: A Survey BT - Circuits and Systems for Video Technology, IEEE Transactions on, *18*(11), 1473–1488. Retrieved from http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4633644&url=http%253A%252F%252Fieeexplore.ieee.org%252Fstamp%252Fstamp.jsp%253Ftp%253D%2526arnumber%253D4633644%5Cnhttp://ieeexplore.ieee.org/stampPDF/getPDF.jsp?arnumber=4633644&url=http%253A%252

Um, T. T., Babakeshizadeh, V., & Kulić, D. (2017). Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2385–2390.

Umer, W., Li, H., Ph, D., Pui, G., Szeto, Y., Ph, D., ... Ph, D. (2017). Low-Cost Ergonomic Intervention for Mitigating Physical and Subjective Discomfort during Manual Rebar Tying. *Journal of Construction Engineering and Management*, *143*(10). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001383.

Wang, S., Yang, J., Chen, N., Chen, X., & Zhang, Q. (2005). Human Activity Recognition with User-Free Accelerometers in the Sensor Networks. *In Neural Networks and Brain, 2005. ICNN&B'05. International Conference On*, *2*, 1212–1217. https://doi.org/10.1080/10192557.2018.1496641

Wang, W. J., Chang, J. W., Haung, S. F., & Wang, R. J. (2016). Human Posture Recognition Based on Images Captured by the Kinect Sensor. *International Journal of Advanced Robotic Systems*, *13*(2). https://doi.org/10.5772/62163

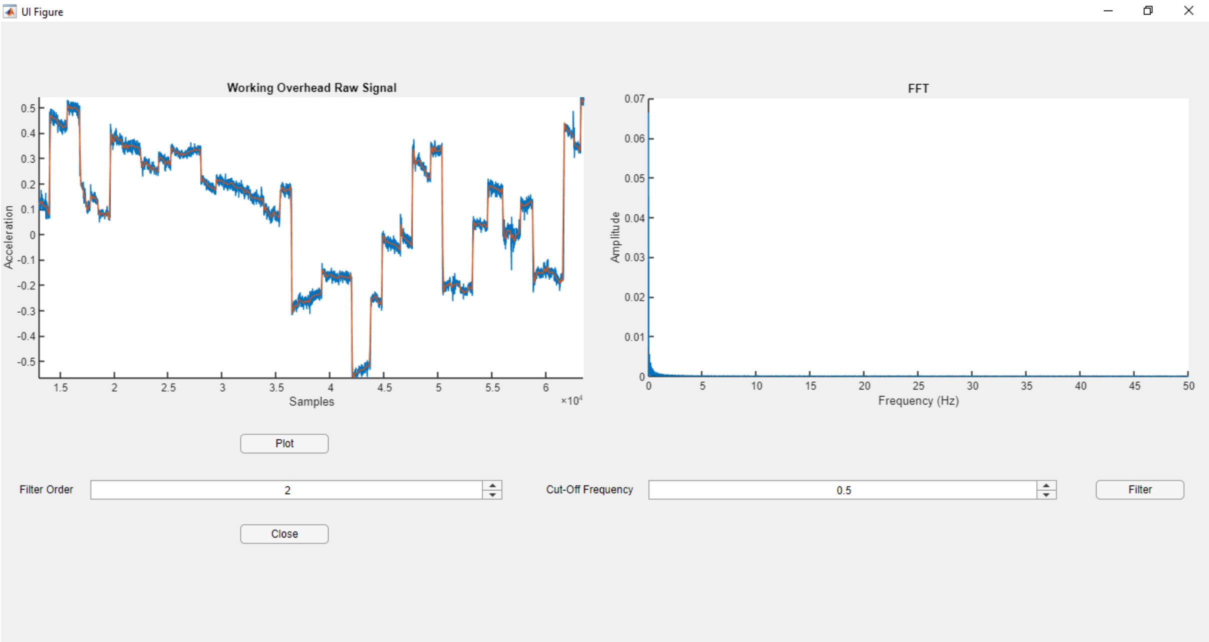
Wang, Y., Zhou, H., Yang, Z., Samuel, O. W., Liu, W., Cao, Y., & Li, G. (2018). An intelligent wearable device for human's cervical vertebra posture monitoring. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 3280–3283).

Werghi, N., & Xiao, Y. (2002). Recognition of human body posture from a cloud of 3D data points using wavelet transform coefficients., (May).

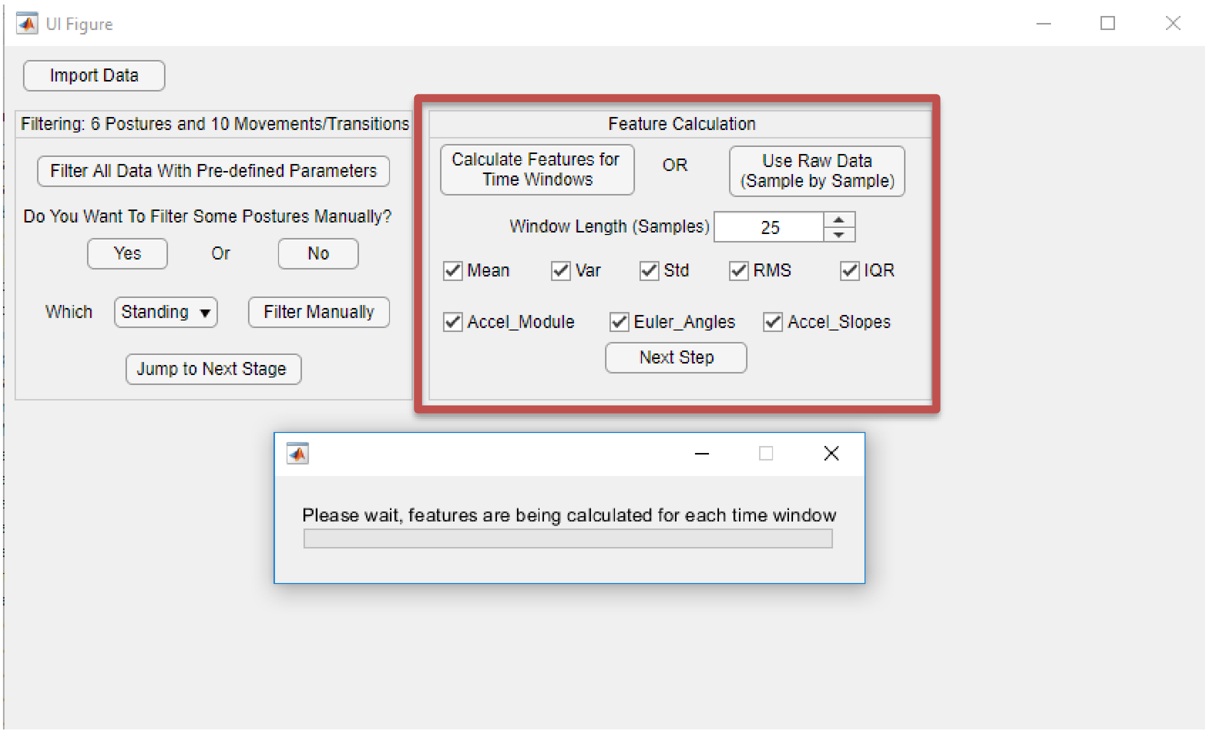
Zhang, Y., & Yang, Y. (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics*, *187*(1), 95–112. https://doi.org/10.1016/j.jeconom.2015.02.006

Appendix

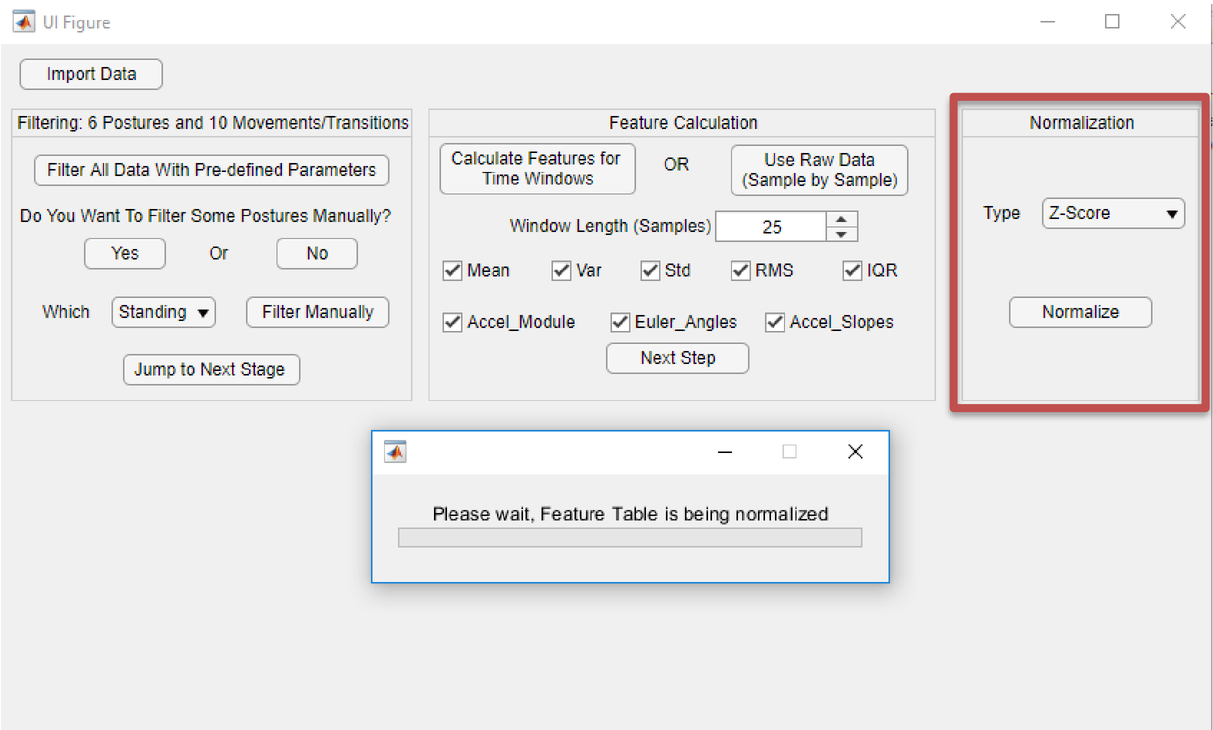
Appendix I: Data manually filtering window of the pipeline



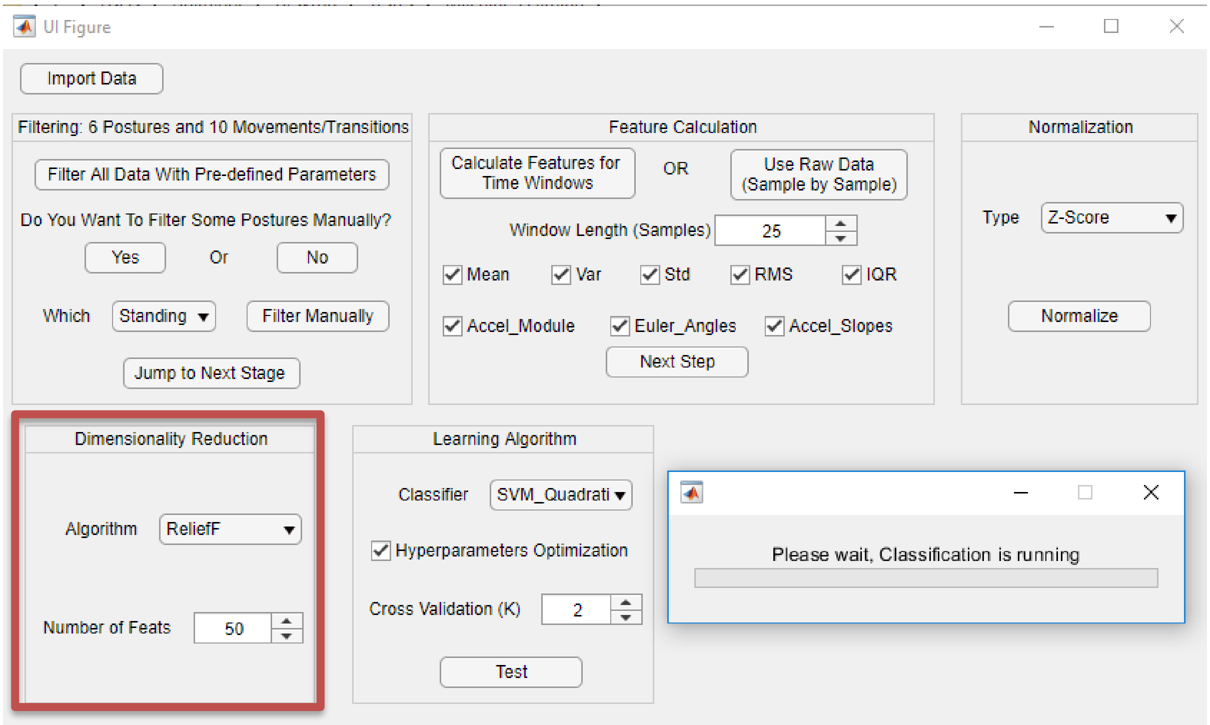
Appendix II: Feature calculation block of the pipeline



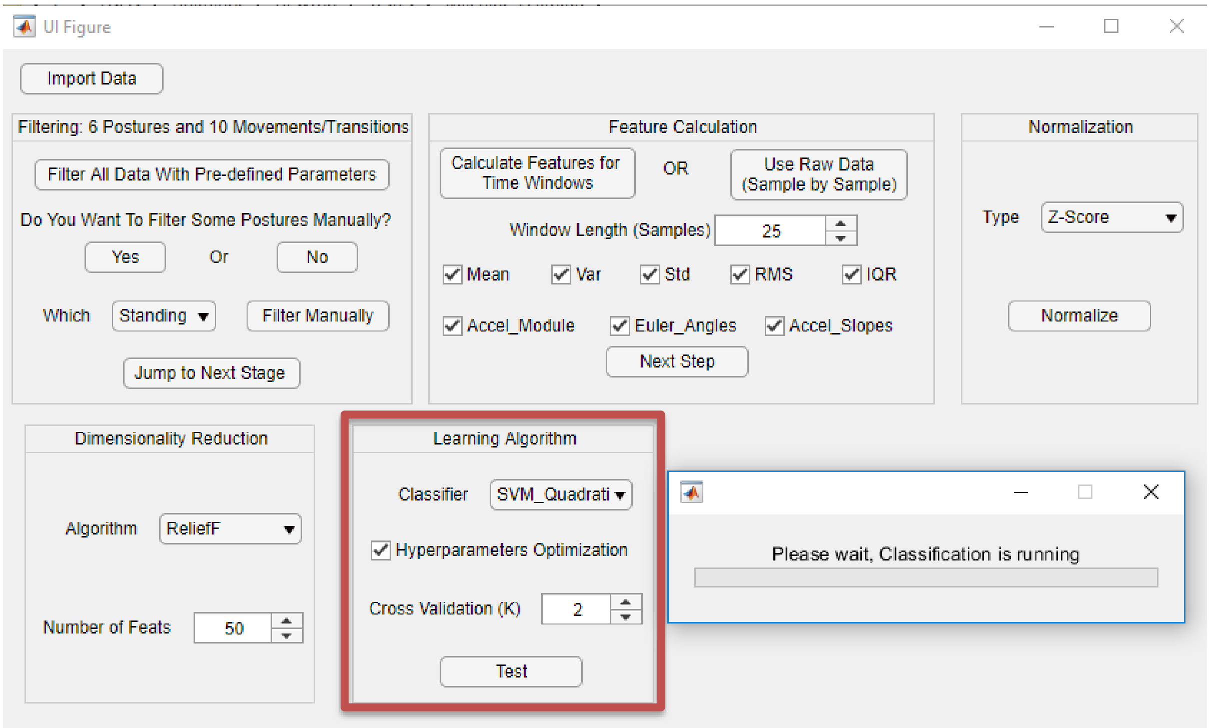
Appendix III: Data normalization of the pipeline



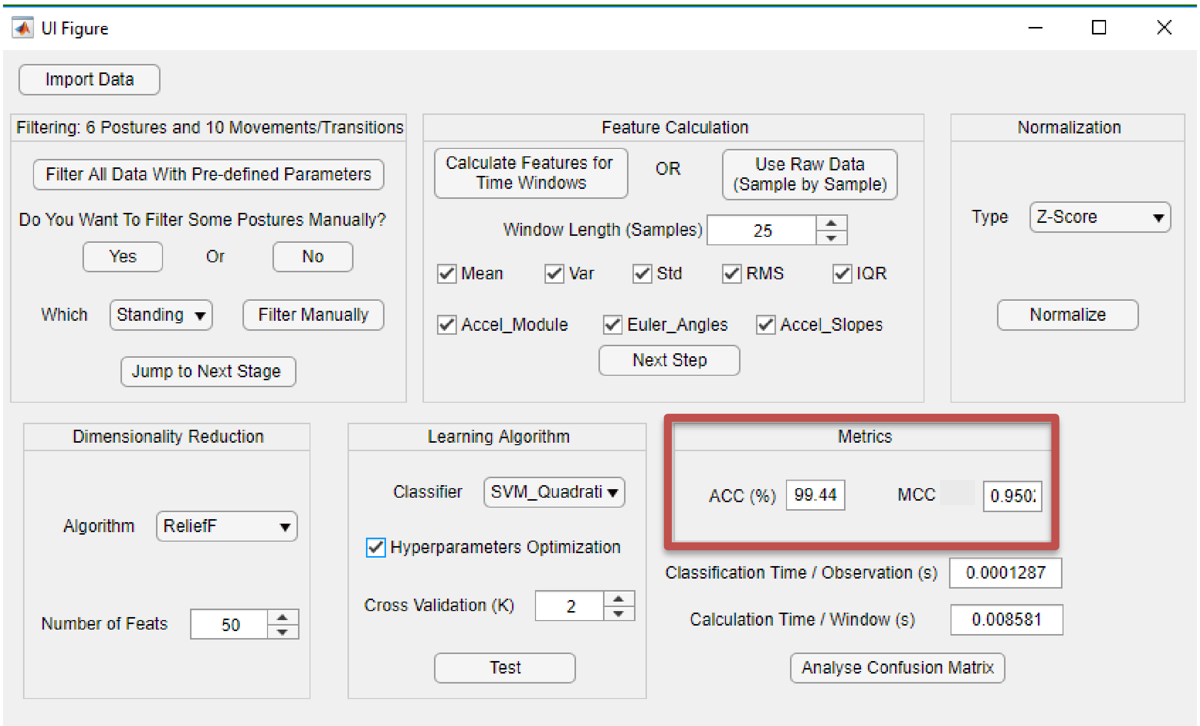
Appendix IV: Dimensionality reduction block of the pipeline



Appendix V: Learning algorithm block of the pipeline



Appendix VI: Metrics block of the build pipeline



Appendix VII: Confusion matrix visualization window of the pipeline

2464	0	78	0	1	0	2	0	1	0	3	3	1	0	0	0
0	2514	1	0	0	0	2	4	0	0	0	1	0	0	0	0
53	23	2381	0	26	1	2	1	0	0	14	7	0	0	0	0
0	2	0	2496	0	0	0	0	0	0	1	1	5	2	0	0
0	0	34	1	2464	3	0	1	4	10	3	2	1	0	0	0
1	2	0	4	0	837	0	0	0	0	0	1	0	0	2	0
5	1	1	0	0	0	221	12	0	0	3	0	0	0	0	0
5	2	0	0	1	2	28	215	1	1	8	9	0	0	0	0
0	0	0	0	2	0	0	0	230	17	5	1	0	0	1	0
0	1	0	3	4	1	0	0	10	190	0	2	1	0	1	1
2	0	4	0	0	2	0	0	4	0	144	5	0	0	0	0
3	4	5	0	0	0	0	3	0	1	32	133	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	0	204	15	0	0
0	0	0	0	0	0	4	2	0	0	0	0	22	179	0	0
0	0	0	0	0	2	2	1	0	3	1	0	0	1	56	10
0	2	2	0	0	0	2	0	5	3	3	2	0	0	10	43

Appendix VIII: Table representing the main specifications of the *InertialLab* system

Component	Quantity	Consumption	Autonomy	Charging Time
Microcontroller STM32F4DISCOVERY	1	220 mA 5 V	6 h	4 h
IMU 9250	7			
Multiplexer	1			
USB mini-cable	7			
USB Pen Drive	1			
PowerBank	1			

Appendix IX: Overview of the *InertialLab* system