ESCUELA DE INGENIERÍA DE FUENLABRADA

Degree in Biomedical Engineering

Bachelor's Thesis

Prediction of gait kinematics with inertial sensors using machine learning algorithms

by

Daniel Real Sánchez

Tutor: Sara García de Villa
Co-tutor: David Casillas Pérez

Academic Year 2023/2024

"You are the owner of your life and your emotions, never forget it.
For better and for worse."
The Little Prince

Acknowledgements

After many years of effort, I am proud to be able to close my university cycle with this project.

First of all, I would like to thank my tutors Sara Garcia de Villa and David Casillas Perez for their guidance throughout these months. Your help and commitment have been essential for the development of this project.

I also want to thank all the people who have passed through my life in my university stage. Thank you for having shared the best and worst of these years and for helping me in my moments of weakness. In particular, I would like to thank my Erasmus classmates Luis, Ferrán and Costa, who have become brothers to me.

Lastly, I would like to thank my lifelong friends, both those who four years ago decided to leave town to study and those who did not, for being a constant source of support.

Abstract

The analysis of the human gait is a research area of great importance due to the impact on the autonomy of people. This project proposes an alternative to gait analysis based on data provided by inertial sensors for the prediction of lower limb angles. The decrease in the number of sensors and their position for gait analysis is evaluated.

This work focuses on the application of Machine Learning (ML) models for the prediction of lower extremity angles during gait. It is proposed as an alternative to the use of the Kalman Filter (KF). These predictions are made from signals recorded by four Inertial Measurement Units (IMUs), taking advantage of its low cost and its ease of use on a day-to-day basis.

To carry out this study, five healthy volunteers data is used during gait in a controlled environment. It is recorded with four sensors made up of accelerometers and gyroscopes. They are located on each shin and thigh. Pitch and Roll angles are predicted. A Unscented Kalman Filter (UKF) provides reference data, commonly used in gait monitoring. For this, more or less data is used depending on the experiment to be carried out.

In the first two experiments, the quality of the prediction of the movements, that is, the Euler angles from the inertial signals, is analyzed. The IMU Left Shin (LShin) and Left Thigh (LThigh) angles are predicted separately knowing volunteer's data. These experiments are called Baseline experiments and are the starting point for the following ones. The objective of the next three experiments is to optimize the number and position of the IMUs to estimate the angles. The angles of an IMU knowing the data are predicted using three, two and one IMU data. The last three focus on extrapolating ML models to data from new volunteers. The results are validated with the UKF reference and compared with the baseline.

The ML models that are used are Multiple Linear Regressor (MLR), Support Vector Regressor (SVR) with Linear, Gaussian and Polynomial kernel, Treebagger (TB) and Multilayer Perceptron (MLP). Furthermore, to overcome the limitation of having little amount of data, a signal segmentation method called Temporal Window (TW) is applied.

This method enlarges the available database. The TW sizes applied are 5, 10, 20, 40 and 60 samples.

The results obtained confirm that the number of IMUs and their position can be optimized. Besides, knowing a person's gait data improves its prediction. The smallest error is obtained by predicting the angles of an IMU using the data from two IMUs. Specifically, an Root Mean Square Error (RMSE) of 0.039 rads and 0.032 rad is obtained for the Pitch and Roll angles respectively, for the IMU LShin; and RMSE of 0.034 rad and 0.029 rad for the Pitch and Roll angles respectively, for the IMU LThigh using data from the IMUs on the right leg. This is obtained with the TB algorithm and 60 samples of TW size.

These results suggest that ML techniques have great potential to be applied in various areas from clinical diagnosis to assistance and monitoring devices. It is worth mentioning that in the *state-of-the-art* methods there is no study in which the aforementioned algorithms are used as a contrast to the KF for the prediction of gait angles. These results open new possibilities in the area of gait pattern generation, which can be used in mechanized and intelligent prostheses. The objective of optimizing the number of IMUs leads to a reduction in the number of these, which reduces the amount of resources needed and, therefore, reduces the cost.

Index of Contents

List of Figures

List of Table

Abbreviations

1DLX 1 Device for a IMU. 45, 60, 67

2DLX 2 Devices for a IMU. 45, 59, 62, 65, 69

3DLS 3 Devices for LShin. 44, 45, 57, 59, 60, 65–67

AI Artificial Intelligence. 6, 13, 20, 21

ANN Artificial Neural Network. 13, 14

BLEEX Berkeley Lower Extremity Exoskeleton. 7

BPNN Back Propagation Neural Network. 17, 18

CNN Convolutional Neural Network. 14, 16

CoM Center of Mass. 14

CSR Classification Success Ratio. 17

DL Deep Learning. 6, 15, 19, 21, 22

DNN Deep Neural Network. 21

DoF Degrees of Freedom. 15

DT Decision Tree. 17, 18, 26

EEG Electroencephalogram. 6, 7

EMG Electromyogram. 6, 7, 13, 16, 18

FFNN Feedforward Neural Network. 13, 15

IE Individual Evaluation. 44, 45, 57, 59, 60, 62, 65, 69

IMU Inertial Measurement Unit. 3, 4, 6, 14–20, 31, 32, 37, 39–41, 43–46, 49, 52, 53, 55, 57, 59, 60, 62, 65–67, 69, 72–74, 76, 78, II, III, IX, X

KF Kalman Filter. 20, 32, 35, 36, II, III

KRLS Kernel Recursive Least Square. 17

LShin Left Shin. 4, 19, 43–45, 49, 52, 54, 55, 57–60, 62, 65–67, 69, 73, 74, II, III, X

LSTM Long-Short Term Memory. 13, 16, 17

LThigh Left Thigh. 19, 43–45, 49, 53, 55, 57–60, 62, 65–67, 69, 73, 74, II, III

MAE Mean Absolute Error. 14, 15, 19

ML Machine Learning. 1–4, 6, 10, 11, 16, 19, 21, 29, 36, 37, 41, 43, 45–47, 49, 59, 73, 74, 76, II, III

MLP Multilayer Perceptron. 4, 16, 17, 19, 22, 29, 42, 44, 50–55, 57–60, 62, 65–67, 69, 73, 74, 76, 78, II

MLR Multiple Linear Regressor. 4, 19, 22, 41, 42, 44, 50, 51, 53, 54, 73, II

NN Neural Network. 13, 16–18, 42, 50, 75, 78

NRMSE Normalized Root Mean Square Error. 14

NV New Volunteer. 65–67, 69

RBF Radial Basis Function. 17

RBFNN Radial Basis Function Neural Network. 18

ReLu Rectified Linear Unit. 27

RF Random Forest. 16, 17, 19

RMSE Root Mean Square Error. 4, 5, 13–16, 19, 29, 35, 42–44, 46, 47, 50, 51, 53–55, 57, 59, 62, 65–67, 69, 74, III

RShin Right Shin. 45, 57, 60, 62, 65, 67, 69

RThigh Right Thigh. 45, 57, 60, 62, 65, 67, 69, 74

SDG Sustainable Development Goal. 73, 76

SVC Support Vector Classifier. 22, 23

SVM Support Vector Machine. 4, 16–18, 22, 23

SVR Support Vector Regressor. 19, 22, 23, 25, 26, 41, 42, 44, 50, 51, 53, 54, 57, 73, II

TB Treebagger. 4, 19, 22, 41, 42, 44, 50–55, 57–60, 62, 65–67, 69, 73, 74, 76, II, III

TW Temporal Window. 4, 5, 15, 19, 20, 36, 41, 42, 44, 45, 49–55, 57–59, 65, 74, II, III

UKF Unscented Kalman Filter. 3, 5, 19, 35, 36, 40, II

UN United Nations. 76

Chapter 1

Introduction

1.1 Context

Gait is a fundamental activity and, at first glance, easy to perform that is part of our daily life. Walking allows us to move efficiently and perform various physical activities [1]. Our locomotor system is set in motion through precise coordination of the nervous system, muscles and bones. However, gait disturbances can have a significant impact on people's quality of life, especially those who suffer from neuromuscular disorders, or injuries, or are elderly.

Gait prediction is a promising field of study that seeks to understand and predict movement patterns associated with human gait [2]. Prediction serves to reduce the limitations associated with sensory solutions, in addition to requiring a smaller number of sensors to record information and measurements [3]. These measurements will be processed to obtain gait parameters using different Machine Learning (ML) algorithms. The applications of gait analysis can be for clinical purposes, such as rehabilitation, or for non-clinical purposes, such as monitoring athletes recovering from injuries [4]. The ability to anticipate and predict gait can provide valuable information for designing more efficient rehabilitation strategies, for evaluating the efficacy of therapeutic interventions, and to generate custom gait patterns to be included in the development of prostheses and exoskeletons.

Prostheses are artificial devices that replace or restore a body part that has been amputated or is missing due to malformation or other causes [5]. This project focuses on the lower extremities, in cases lower limb prostheses are used to replace the legs, feet, or specific parts of them. Figure 1.1 shows an example of a lower limb prosthesis. They are designed to provide support, improve mobility and allow amputees to lead a normal life [6]. Lower prostheses are classified according to the level of amputation: transtibial prostheses (below the knee), transfemoral prostheses (above the knee), foot and ankle prostheses.

The materials used are diverse and are also a great area of research. The materials are intended to be biocompatible, such as scaffolds, metals, plastics or ceramics. It depends on the characteristics of the person [7].

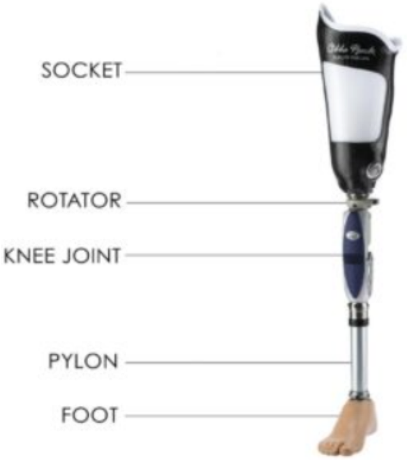


Figure 1.1: Example of leg prosthesis. Its parts are defined [8].

On the other hand, exoskeletons are wearable robotic devices that are attached externally to the human body [9]. They are designed to improve movement and functionality of the extremities. They can be used both by people with physical disabilities and by people without disabilities with the aim of improving their performance in different activities. In the context of the lower extremities, exoskeletons provide support and assistance to the legs and hips [10].

The use of exoskeletons and smart prostheses offers several significant advantages for people with physical disabilities. These technologies have the potential to improve the quality of life, mobility and independence of users. These smart devices are based on gait prediction using algorithms and sensors integrated into these devices. They allow to collect data and use that information to adjust the assistance provided [10].

The use of sensors in healthy people is relevant to be able to simulate movement in people with amputated or damaged body parts. In addition, this would improve rehabilitation and functionality since more personalized solutions are offered [11]. Individualized rehabilitation lies in recognizing the specific needs of the person suffering from a pathology. The benefits are optimal treatment, improved results, patient motivation and prevention of complications [12].

ML can pay a key role in these smart devices detecting gait patterns [13]. By harnessing the power of ML algorithms, these algorithms can intelligently adjust to the individual needs of users, improving gait prediction and providing efficient, personalized assistance. The gait cycle is a repetitive movement, so from the movements of a leg or parts of it, it is possible to estimate the movements that would be made in a healthy person in the part to be predicted. In addition, in the literature there are discrepancies about the number of sensors to be used in gait analysis and their position [14], so this project seeks to optimize the use of sensors.

1.2 Motivation

The predictive analysis of gait is a topic of great relevance, since the ability to walk and move efficiently is essential for people's independence and quality of life [15]. By understanding and predicting gait patterns, this work can help in the improvement of the care and treatment of patients with neuromusculoskeletal disorders, injuries, or disabilities that affect their ability to walk.

This Bachelor's Thesis aims to analyze the use of Inertial Measurement Units (IMUs) for the prediction of angles of the lower extremities so that people with prostheses or exoskeletons have a better individual and personalized adaptation of movement. This prediction is analyze by using data from healthy people pulled from a publicly available database, PHYTMO [16]. PHYTMO is a database with a wide variety of volunteers and collected by four IMUs, one on each thigh and one on each shin. An Unscented Kalman Filter (UKF) commonly used to estimate the angles with respect to the horizontal plane is applied by using the IMU data [17]. The angles are used as a reference to train and test the ML algorithms. These are the angles that are predicted by ML and by varying the number of IMUs used.

Furthermore, this project gives the opportunity to apply and develop advanced technical skills in the field of biomechanics and data analysis in a real context. Besides, cutting-edge technologies are applied, such as motion sensors, motion capture systems, and ML algorithms.

1.3 Objectives

The main objective of this work is to develop an estimator of the angles of a limb from inertial measurements of other IMUs during walking. The estimator is based on ML techniques and sensor location optimization. However, in order to apply ML, prior preprocessing is needed, which are also part of the objectives to be achieved. The objectives of this work are set out below:

1. **Angle estimation.** In the database used there is no reference for the measured angles, so it is necessary to estimate them with respect to the horizontal plane using a Bayesian Filter called UKF.

2. **Evaluation and selection of algorithms and Temporal Window (TW).** The algorithms to be initially tested are: Treebagger (TB), Support Vector Machine (SVM) with Gaussian, Linear and Polynomical kernels, Multiple Linear Regressor (MLR) and Multilayer Perceptron (MLP). The criterion to assess the results is Root Mean Square Error (RMSE). The algorithms and the TW with the best performance according to the criterion are chosen.

3. **Angle prediction.** Six experiments are carried out to predict the IMU angles following two cross-validation methods. In the first, the data of the person to be predicted is known what is called *K-Fold cross-validation*. In the second method, the data of the person to be predicted is not known, so it is validated by *leave-one-out cross-validation*.

4. **IMUs Value.** Data is provided by 4 IMUs located on the thighs and shins. In the experiments, the number of IMUs used varies in order to obtain the number of these and the optimal positions. Left Shin (LShin) is predicted, so it focuses on possible knee exoskeletons, since thigh information is also available.

1.4 Structure

This project is composed of six chapters. However, this study is divided into two main phases. The first phase is based on contextualizing and giving information on the topic to be discussed, Chapters 1, 2 and 3. The second phase has to do with the experiments carried out, Chapters 4, 5 and 6. The chapters are explained below.

- **Chapter 1 -** *Introduction:* This chapter explains the need to use ML algorithms for gait prediction, the use of filters for orientation estimation, and the use of sensors to record information.

- **Chapter 2 -** *State-of-the-Art:* In this chapter a compilation of academic and scientific literature on the use of IMUs for gait prediction is presented, also adding orientation estimation, angle prediction, evaluation criteria and validation methods. The state of the art provides a necessary context to understand the current research landscape.

- **Chapter 3 -** *Theoretical Framework:* This chapter is dedicated to explaining the different ML models used and the IMUs used to capture the data used in the experiments.

- **Chapter 4 -** *Methodology:* This chapter is dedicated to explain the experiments carried out using two cross-validation methods: knowing the data of the person to be predicted and without knowing them. In addition, the data processing from the original database, its labeling, passed through the UKF and TW. Finally the RMSE evaluation criterion is explained.

- **Chapter 5 -** *Results and Discussion:* Along this chapter, the results of previously explained experiments are exposed with a brief description and discussion.
- **Chapter 6 -** *Conclusion and Future Lines:* This final chapter is a summary of the results. The contribution of the project is discussed along with the limitations faced during the study. Besides, future lines and impacts are exposed.

Chapter 2

State of the Art

Gait analysis involves the measurement and estimation of various parameters, depending on the type of analysis. The analysis can include spatial-temporal parameters, Electromyogram (EMG) activity, kinematics, kinetic parameters during walking and parameters obtained by carrying out motion analysis from various activities [18]; such as jumping or doing squats. To carry out this analysis by obtaining objective measurements, technological solutions are needed. These technological solutions commonly are wearable and non-wearable sensors. Wearable sensors include IMUs with accelerometers and gyroscopes. Sensors for EMG, Electroencephalogram (EEG), foot pressure sensors, among others, can also be used. Non-wearable sensors include ground reaction forces, plates, and motion capture systems [18] [19].

The measurements obtained with the wearable sensors are processed to obtain walking parameters. There are many algorithms to process that information. For example, conventional algorithms using angular velocity [20], musculoskeletal models to estimate intention using EMG [21], and ML and Deep Learning (DL) techniques.

The different prediction techniques used have been purchased and analyzed in many works, as well as the advantages and disadvantages of using sensors. In addition, there is controversy about the number of sensors to use, the location, the model, what parameters are the most important to measure, as well as the type of optimal prediction model according to the type of sensor [22]. Some authors focus on studying one type of gait analysis algorithm, such as DL algorithms, while others focus on a specific type of algorithm and sensor, such as Artificial Intelligence (AI) using inertial sensors.

ML algorithms have advantages over conventional gait analysis methods. The gait is temporal and has a nonlinear relationship with its parameters, so ML models can find the relationships between the inputs and outputs of nonlinear systems [23] and are better managing data variability. The use of ML models means greater accuracy in predictions in less time mainly because parameter adjustment is done automatically. This allows devices that interact with humans, specifically those that help in gait, to adapt to the needs of patients in real time [24]. This makes the rehabilitation more effective.

Nowadays, an immense number of portable robots that interact with humans are being developed for various medical and therapeutic purposes [25]. Within these devices, the most used are exoskeletons, orthoses and prostheses.

An exoskeleton is an electromechanical device made up of sensors, controllers and actuators that guarantee torque to the joints [26]. The provision of a support torque allows actions to be performed more easily and with less strain. The exoskeletons are designed to support the lower and upper extremities, the whole body and can be passive, active or semi-passive [27].

In the 1960s, exoskeletons were designed for military use [26]. General Electric Company developed Hardiman I to increase the strength of soldiers, creating what was described as "the union of man and machine" [28]. Half a century later, exoskeletons evolved to be used in different applications [29], such as in industry, rehabilitation and in the restoration of gait in patients with spinal cord injuries [30], Multiple Sclerosis [31], neurological disorders, among others.

One of the exoskeletons that stands out in the industry is the Berkeley Lower Extremity Exoskeleton (BLEEX) which has seven degrees of freedom [32]. It has hydraulic systems, which allow lifting heavy loads. MIT exoskeleton is another semi-passive exoskeleton for weight lifting, made up of springs and dampers instead of actuators. It acts as an intermediary, transferring 80% of the load from the person to the ground. HAL-5, Hybrid Assistive Limb, by the University of Tsukuba, is a full-body exoskeleton designed for all types of people and allows paraplegics to walk by decoding their intentions [33]. MINDWALKER uses the paraplegic user's EEG and EMG in real time to control the exoskeleton [34].

Orthoses is another category of assistive technology, sometimes used interchangeably with exoskeletons [35]. The purpose of orthoses is to help people with pre-existing pathologies. However, exoskeletons can also be worn by healthy people to enhance human capabilities. Orthoses can be passive or active [36].

Prostheses are devices designed to replace or enhance a body part that is missing or not working properly [37]. These can be used to replace amputated limbs, such as arms or legs, or to restore function to organs or body systems. Prostheses can be mechanical, electrical, or a combination of both, and are designed to fit the individual needs of each person. Thanks to advances in technology, modern prostheses offer greater functionality, comfort and naturalness, allowing people to carry out daily activities and participate fully in society.

Lower extremity exoskeletons and prostheses, when used as assistive devices, have two primary applications: (i) rehabilitation and gait training, or (ii) gait assistance to perform daily activities [38]. The way in which these devices move and interact with the user is controlled by a control strategy. *Trajectory tracking* is a type of control strategy that allows the patient to walk with the device following a gait trajectory pattern, often obtained from healthy people. *Assist as needed*, is another control strategy in which the device provides the necessary help to the patient at all times. The level of assistance vary depending on the gait phase, the force exerted by the patient and the patient's recovery period [38]. The most important part of the control strategy is given by the estimation and detection of the gait.

Controlling exoskeletons and intelligent prostheses is one of the most recent applications of gait analysis. In clinical applications, gait analysis is used for rehabilitation and diagnosis of pathologies [39]. Among non-clinical applications, gait analysis has been used in sports to monitor recovery after injury and to assess athlete activation [40]. Also, it has been used in security, for biometric identification and authentication [41]. In the elderly, to detect possible falls and in well-being, for the identification of the emotional state [42].

Regarding the literature of the mentioned devices, as additional information, before 2002 the number of papers published on exoskeletons and intelligent prostheses was less than 30 per year and the cumulative number of articles was less than 500. Between 2016 and 2020, no paper focused on gait analysis for the control of exoskeletons and prostheses in the lower extremities. In mid-2019, the total number of papers on the subject was close to 4000 [43].

2.1 Gait Features and Related Parameters

According to Whittle, the gait is the terminology used to describe "the way or the style with which we walk in" [44]. When studying the gait, several parameters can be observed. For a healthy gait, these parameters have to be within a normal range, with variations caused by the anthropomorphic characteristics of people [45]. When the parameters are outside these normal ranges, it is said that there is a pathological gait. The different types of gait parameters are explained below:

2.1.1 Gait Cycle Phases

The action of walking consists of cycles of events that are repeated periodically. Each cycle is made up of a stance phase, where the lower extremities are in contact with the ground; and the swing phase, where there is no contact, see Figure 2.1.

The stance phase begins with an initial contact called "heel strike". Then, the foot is fully supported on the ground as a result of dorsiflexion. Then, the heel begins to separate

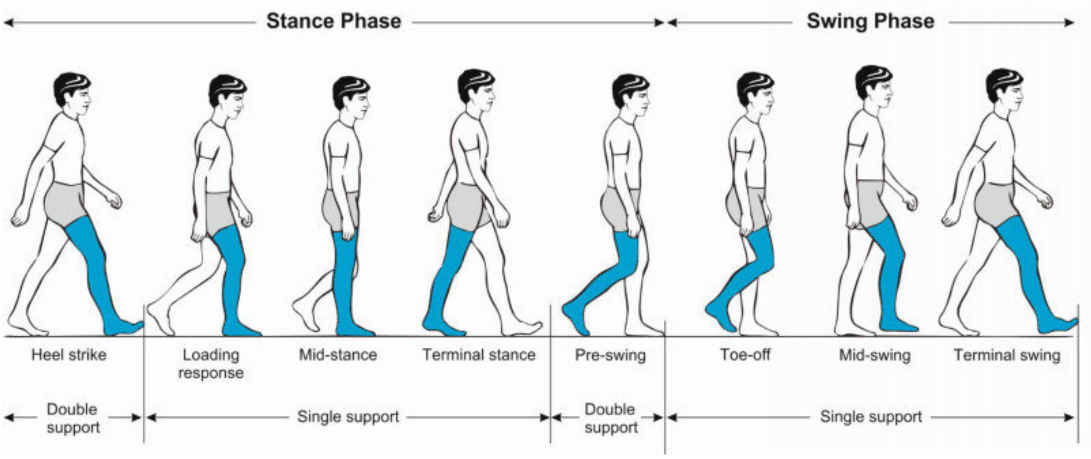


Figure 2.1: Phases of the gait cycle [46].

from the ground, starting the pre-swing phase, which is the last moment of the foot in contact with the ground to start the swing phase. The swing phase is divided into three simple phases: (i) initial swing or toe-off, (ii) mid-swing and (iii) terminal swing. This sequence occurs for the left and right foot alternately, resulting in movement [45]. Gait phases are part of the spatial-temporal parameters [18].

2.1.2 Joint Angle

The angles of the joints change during the gait cycle. The angles of the hip, knee and ankle are usually measured mostly in the sagittal plane since it is where most movement occurs [45]. Joint angles are considered kinematic parameters of gait [47]. The first and second derivatives correspond to angular velocity and angular acceleration, respectively.

2.1.3 Torque / Moment

The moment of force refers to the rotation caused by the application of a force, the magnitude of which depends on the applied force and the shortest distance between the location of the applied force and the pivot. This distance is called the lever arm. There is a certain difference between moment and torque. The moment results in bending and the torque in rotating. Since the mathematical formula is the same, they are often used interchangeably [48].

In biomechanics, the moment of force depends on when muscles contract, causing a pivot to rotate (for example, the knee). Internal moments can be passive due to soft tissue tension and active due to muscle contraction (eccentric, concentric and isometric contraction). On the other hand, external moments are due to external forces such as gravity [45]. The moment belongs to the kinetic parameters of gait [18].

2.1.4 Locomotion Mode

Modes of locomotion refers to different activities such as walking, standing, going up a ramp, or climbing stairs. Various authors use ML classification models according to different modes that the same authors choose, see Rai et al. paper [49], which objective is to predict ankle trajectory angles in unstructured activities and assess whether the rest of the body is involved in the biomechanics of the foot. The disadvantage of the modes is that they cannot be adapted to the specific needs of each patient and that there are terrains and activities that are difficult to categorize.

2.1.5 Intention

Intention is defined as "the need for the robot to have knowledge of some aspect of the human's planned action in order for the robot to appropriately assist toward achieving that action" [50]. The intention can be measured from the central nervous system with the brain activity, from the peripheral nervous system with the electrical activity of the muscles or from the forces of interaction between the human and the robot that can be measured with force sensors [50]. The intention can have discrete states that can be used to trigger the initiation of certain movements or to transition between discrete control modes or transition between continuous states, such as the desired position trajectory.

2.2 Literature Research

A search for scientific articles focused on the prediction of gait has been carried out. PubMed database is the online library chosen for applying the following command filter. We look for papers that focused on the implementation of ML algorithms for gait analysis to be used in exoskeletons and lower limb prosthesis. The keywords selected to extract the desired papers is aimed at being general enough not to leave any valuable papers. The query rule introduced is the following:

(exoskeleton OR orthosis OR orthotic OR prosthesis) AND (gait OR locomotion) AND (recognition OR classification OR prediction OR intention OR selection OR detection OR discrimination OR partitioning OR segmentation OR estimation) AND ("machine learning" OR "deep learning" OR "artificial intelligence" OR "neural" OR "decision trees")

The AND conditional operator is used for the article to bring the words it joins; and the OR conditional operator is used for the article to bring at least one of the words it joins.

2.2.1 Article Selection

With the application of the explained search filter, 585 papers are obtained. As an inclusion method, only papers with free full text were included, leaving us with 253 papers. The relationship of the publications with the year they were published can be seen in Figure 2.2. The increase in studies related to this topic in recent years is notable, with 2020 being the year in which the largest number of works were carried out.

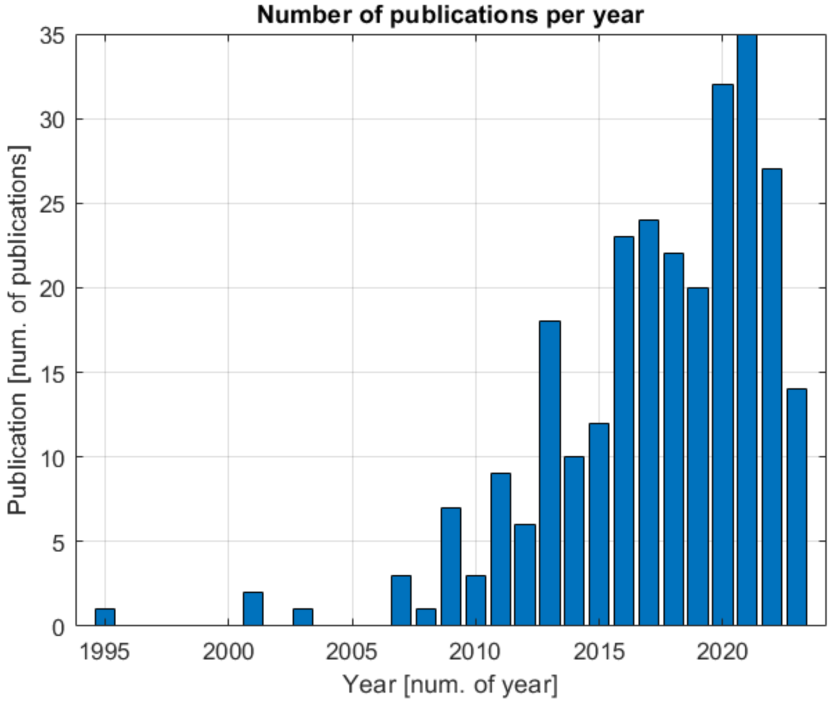


Figure 2.2: Distribution of 253 publications along the time.

These papers have been analysed based on abstract only. Less relevant or irrelevant papers are manually removed; for example, because the scope of these papers has no significance in our review. Figure 2.3 shows the flowchart for article selection. Some reasons for excluding papers include the use of upper limbs instead of lower limbs; or the absence of ML algorithms of interest. Besides, only clinical trials, reviews, systematic reviews, meta-analysis, books and documents were chosen as article type. Finally, there are 30 papers of interest. Furthermore, a complementary search on IEEE, ScienceDirect and Google Scholar to complete this chapter. In this complement search, papers are manually chosen when they are related with this work.

2.2.2 Research Results

In this Section, the results of the performance of the algorithms and the sensors used to obtain the predictions are explained. The discussed parameters are: (1) joint angle and trajectory, (2) gait phase and (3) locomotion mode.

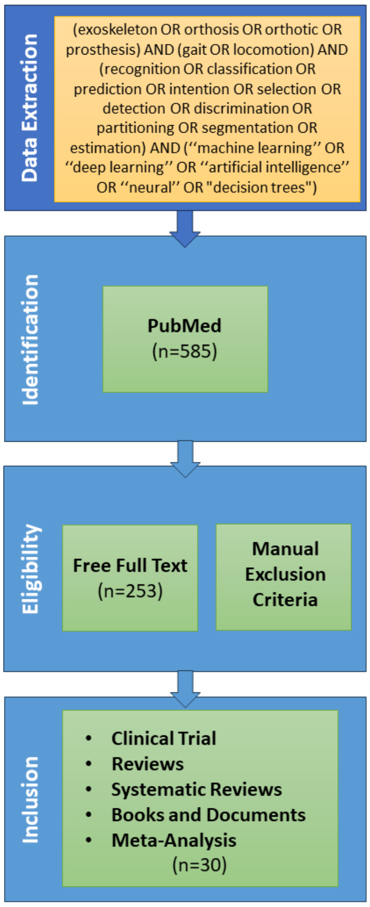

Data Extraction
(exoskeleton OR orthosis OR orthotic OR prosthesis) AND (gait OR locomotion) AND (recognition OR classification OR prediction OR intention OR selection OR detection OR discrimination OR partitioning OR segmentation OR estimation) AND (“machine learning” OR “deep learning” OR “artificial intelligence” OR “neural” OR "decision trees")
Identification
PubMed
(n=585)
Eligibility
Free Full Text
(n=253)
Manual Exclusion Criteria
Inclusion
• Clinical Trial
• Reviews
• Systematic Reviews
• Books and Documents
• Meta-Analysis
(n=30)

Figure 2.3: Flowchart on the methodology of article selection.

2.2.2.1 Joint Angle and Trajectory Estimation

Joint angle and trajectory are considered parameters of motion kinematics. The use of sensors is key to obtain data from the subjects to be predicted. Studies related to these parameters are shown below.

In [51], an Artificial Neural Network (ANN) is implemented by Kutilek and Farkasova, to predict joint angles using cyclograms. The authors used a backpropagation-trained Neural Network (NN) to predict future limb angles. The NN inputs were the actual angles for a single joint, angular acceleration, weight and height of the person. Hip-knee prediction showed more accurate visual results than ankle-knee prediction according to the comparison between the real cyclogram and the predicted one. There are no quantitative results. The cyclograms they are compared to are from past research on the gait cycle.

Mazumder et al. [52] present a method to generate lower limb gait pattern by judging human intention using a wearable sensor system. Data is measured from users with varying anthropomorphic features. Contribution of this paper is in development of a wearable sensor system, multi-channel redundant fusion to calculate stride time and an adaptive gait trajectory generation algorithm. The proposed method of trajectory generation is used to regenerate lower limb joint motion in sagittal plane for wearable robotic devices like prosthesis and active lower limb exoskeleton.

An Elman NN is implemented by Wang et al. [53], to establish a relationship between surface EMG and knee joint angles. The signals were recorded during leg extension exercises at various speeds, with and without load. The results were assessed according to RMSE. The authors report that the higher the leg extension speed, the RMSE is higher. However, when compared to loaded and unloaded exercises at constant speeds, having a load results in a lower RMSE. At low speed with load gets a RMSE(%) of 3.5534.

Another study carried out by Mundt et al. [54] tries to overcome the limitation of measurement discrepancies and the missing information on kinetic motion parameters using a machine learning application based on ANNs. For this purpose, inertial sensor data linear acceleration and angular rate was simulated from a database of optical motion tracking data and used as input for a Feedforward Neural Network (FFNN) and Long-Short Term Memory (LSTM) NN to predict the joint angles and moments of the lower limbs during gait. Both networks achieved mean correlation coefficients higher than 0.80 in the minor motion planes, and correlation coefficients higher than 0.98 in the sagittal plane. These results encourage further applications of AI to support gait analysis.

The position of sensors to obtain data can determine the quality of the results and is a topic of great controversy. In the study carried out by Findlow et al. [55], two inertial sensors are used on the shin and the foot. The objective of this study is to estimate sagittal plane ankle, knee and hip gait kinematics using 3D angular velocity and linear acceleration. Data is collected using reflective markers. They explore the accuracy of intra-subject predictions, i.e., where training and testing uses trials from the same subject, and inter-subject, where testing uses subjects different from the ones used for training. A generalized regression networks algorithm was used to predict obtaining that the best results are from intra-subjects predictions, with correlations 0.93–0.99 and with an mean absolute deviation less than or equal to 2.3° between measured kinematic joint angles and predicted angles. The inter-subject case produced correlations between 0.70 and 0.89; and larger absolute differences between measured and predicted angles, ranging from 4.91° for the left ankle to 9.06° for the right hip. The angular velocity data added little to the accuracy of predictions and there was also minimal benefit to using sensor data from the shin. Thus, a wearable system based only on footwear mounted sensors and a simpler sensor set providing only acceleration data shows potential.

Another use case for a single IMU is the study by Lim et al. [56]. They propose the sensor to be placed at the Center of Mass (CoM) of the subject to predict data from the lower extremities, using an ANN. Data from seven subjects walking on a treadmill at various speeds were collected from a single IMU worn near the sacrum. The data was segmented by step and numerically processed for integration. Six segment angles of the stance and swing leg, three joint torques, and two ground reaction forces were estimated from the single sensor, with fair accuracy. In all cases, the prediction with the lowest error is obtained in the shin at fast speed, with an Normalized Root Mean Square Error (NRMSE) of 3.46±0.66%, 4.99±1.43% and 9.33±3.42% for segment angles of stance leg, segment angles of joint torques of stance leg, respectively. For ground reaction forces a normalized error of 6.16±1.76% is obtained at moderate speed in anteroposterior axis and 6.26±1.24% in the vertical axis. These results indicate the importance of the CoM as a dynamic determinant of multi-segment kinetics during walking. The trade-off between data quantity and wearable convenience can be solved by utilizing a machine learning algorithm based on the dynamic characteristics of human walking.

In [57] an accelerometer placed on the foot is used for sagittal plane lower extremity angle measurement. The study seeks to validate the sensor measurements during running based on a deep learning approach. A Convolutional Neural Network (CNN) is used with data from 10 participants while they run the same course at five different speeds. An optical motion capture system measures the reference angles of the joints. The CNN model predictions deviated from the reference angles with a RMSE of less than 3.5° and 6.5° in intra- and inter-participant scenarios, respectively. Furthermore, they provide an estimation of six important gait events with a Mean Absolute Error (MAE) of less than 2.5° and 6.5° in intra- and inter-participants scenarios, respectively. This study highlights an appealing minimal sensor setup approach for gait analysis purposes.

A concern similar to the one that led to this work is the one proposed by Hernandez et al. [58]. They wanted to find out if DL models accurately predict the angles of the joints of the lower limbs with data from the IMUs during walking. Lower-limb kinematic data were simultaneously measured with a marker-based optical motion capture system and running leggings with 5 integrated IMUs measuring acceleration and angular velocity at the pelvis, thighs and shins. Data was recorded from 27 participants with means of 26.5±3.9 years, 1.75±0.07 m and 68.3±10.0 kg while walking at 4 and 6 km/h and running at 8, 10, 12 and 14 km/h on a treadmill. The model input consists of raw IMU data, while the output estimates the joint angles of the lower body. The model was trained with a nested-fold cross-validation and tested considering a user-independent approach. MAE for the Degrees of Freedom (DoF) ranged from 2.2±0.9° to 5.1±2.7° with an average of 3.6±2.1°. The results of this study show that the proposed model can predict the kinematics of the joints for walking, running and changes in gait without the need to discriminate the subjects wearing the sensors. These results have been validated with treadmill walking and have not yet been confirmed for walking in other environments.

On the other hand, the works face limitations, the small available database being common. So, it is necessary to use methods to extend said data in order to improve predictions. In this work we use a method called TW is used which is the same as one used in [59] but with some modifications. In this paper a linear time-series-based prediction models have been proposed for joint movement for the lower extremity. The joint movement data is collected at RAMAN Lab, MNIT Jaipur. Experimental results indicate that this approach is better than FFNN in the case of linearly correlated data, considering MAE as an evaluation measure. The proposed prediction model could be used for efficient control of lower extremity robot-assisted device for a smooth gait for the patients.

There are other methods for increasing dataset size, such as the one used in [60] where the objective is to analyze the estimation of 3D joint angles and joint moments of the lower limbs based on IMU data using a FFNN. The dataset summarizes optical motion capture data of former studies and additional newly collected IMU data. Based on the optical data, the acceleration and angular rate of inertial sensors was simulated. The data was augmented by simulating different sensor positions and orientations. Gait analysis is performed on 30 participants using an optoelectronic system and force plates in parallel with five IMUs. A mean correlation coefficient of 0.85 for the joint angles and 0.95 for the joint moments was achieved. The RMSE for the joint angle prediction was smaller than 4.8°. The results are good in the sagittal plane and reflect that lengthening the data set improves the prediction of joint angles. This indicates that research on appropriate augmentation techniques for biomechanical data is useful to further improve machine learning applications.

On the other hand, a drawback of using wearable sensors is their dependence on a battery. Trying to minimize this problem, Sung et al. [61] thought about using a single low-frequency IMU (23 Hz) to estimate multi-joint angles based on LSTM NN. The IMU is located on the side of the shin during a walking exercise in which each of the 30 healthy volunteers chooses the speed at which to go. The results show a comparatively good accuracy level, similar to previous studies using high-frequency IMU sensors. Compared to the reference results obtained from the motion capture system, the estimated angle coefficient of determination (R2) is greater than 0.74, and the RMSE is less than 7° and 9.87%, respectively. The knee joint showed the best estimation performance in terms of the RMSE and R2 among the hip, knee, and ankle joints.

Finally, the most important factor when predicting is the choice of ML model. There are many algorithms that can be used to estimate, but finding the one that produces the optimal results can be complex. This is why several models are often used for comparison. An example of this can be seen in the study by Moghadam et al. [62] where four non-linear regression models are used to estimate muscle kinematics, kinetics and muscle forces using IMUs and EMG data. Seventeen healthy volunteers (9F, 28±5 years) had to walk on a flat surface for a minimum of 16 trials. For each trial, marker trajectories and three force-plates data were recorded to calculate pelvis, hip, knee, and ankle kinematics and kinetics, and muscle forces (the targets), as well as 7 IMUs and 16 EMGs. Marker trajectories from a 12-camera optical motion capture system (Vicon Motion Systems Ltd., UK), ground reaction forces from three gound embedded force plates (Bertec, Columbus, Ohio), EMG (Mini-Wave, Italy), and IMUs (Vicon IMeasureU Ltd., NZ). The features from sensors' data were extracted using the Tsfresh python package and fed into 4 ML models; CNN, Random Forest (RF), SVM, and Multivariate Adaptive Regression Spline for targets' prediction. The RF and CNN models outperformed the other ML models by providing lower prediction errors in all intended targets with a lower computational cost. Only kinematics are parameters of interest for this project. The lowest RMSE was obtained with RF model having an error equal to 0.74° related to pelvic obliquity for intra-subject examination and 2.95° for inter-subject examinations. This study suggested that a combination of wearable sensors' data with an RF or a CNN model is a promising tool to overcome the limitations of traditional optical motion capture for 3D gait analysis.

2.2.2.2 Gait Phase Estimation

Another relevant parameter is the classification and identification of gait phases. The magnitude of the torque offered by an active exoskeleton or orthoses varies depending on the phase of gait. A wide variety of sensors and algorithms are used, but only those that are of interest for this project are detailed.

A MLP is implemented by Jung et al. [63] to detect both gait phases: stance and swing. By detecting the phases, they can control the ROBIN-H1 exoskeleton for the rehabilitation of stroke patients. The inputs in the MLP are the pitch angle orientation and angular velocities measured by various sensors. It had one hidden layer with growing nodes that increased from 5 to 50. Ground truth labels were obtained using force plates. To analyze the results, the Classification Success Ratio (CSR) was used for three types of errors: early classification, delayed classification, and miss-classification. Online and offline tests were carried out with an average CSR rate of 90.75% and 97.75%, respectively.

Ma et al. [64] implemented an MLP, as a comparison with the Kernel Recursive Least Square (KRLS) method and SVM to detect 4 gait phases: heel, flat foot, heel out and swing. MLP has an average qualify ratio for 3, 5 and 10-fold cross-validation of 83.17%, 82.42%, 83.23%. MLP has worse results compared to KRLS, with 2.33%, 3.62% and 3.04% classification ratios for 3, 5 and 10-fold cross-validation. The SVM had a Gaussian kernel function and was optimised using particle swarm optimization. The performance was lower than KRLS too, with 83.00%, 82.69% and 83.29%.

Another NN is implemented by Kang et al. [65] to estimate the percentage of the gait cycle, to control a bilateral hip exoskeleton. The phases were treated as continuous. MLP of a hidden layer and 20 neurons. It turned out that using a single thigh IMU better results were achieved than combining the rest of the sensors.

Farah et al. [66] implemented a logistic Decision Tree (DT) model for the detection of 4 swing phases: loading response, swing, terminal swing and push off using knee angles, thigh angular velocity and acceleration. The chosen model was J-48 following the C4.5 node division criteria, with 1643 containing 822 nodes. Training and validation accuracy were 98.76% and 98.61%, respectively.

On the other hand, Pasinetti et al. [67] implements a RF algorithm to detect stance and swing. Cameras embedded in crutches used when walking with the exoskeleton were used. Each of the cameras monitors the contralateral leg. An algorithm detects the floor surface to be able to measure the distances. Two variations of algorithms are used: DT random forest and sigma-z random forest. The RF and sigma-z RF had accuracy values of 81% and 87.3%.

A study using SVM has been previously mentioned, but it is also used by Zhen et al. [68] for the detection of two gait phases. It uses four different kernels: linear, Radial Basis Function (RBF), sigmoid and polynomial, RBF being the one with the best results. Even so, the results are improved by LSTM algorithm. Zhang et al. [69] also implement SVM, but it does not achieve as good results as Back Propagation Neural Network (BPNN).

2.2.2.3 Locomotion Mode Estimation

Finally, knowing the classification of locomotion modes is essential to assist in daily activities. Identifying this parameter allows for a less abrupt transition between activities. The modes of locomotion can be classified as static: sitting and standing; and in dynamics: ascending and descending starts and slopes, and straight level walking. Often, the identification of the mode of locomotion is performed together with the detection of the gait phase.

To carry out the detection of locomotion modes, a BPNN is implemented by Song et al. [70]. They detected 4 static modes and 11 dynamic ones. They used IMUs and foot pressure sensors to acquire the signals and extract features, 141 in all. Three NNs were developed with three layers each. The first classified whether the mode was static or dynamic, with 5 input neurons, 25 neurons in the eye layer, and 1 neuron in the output layer. The neurons in the second model depended on the output. For the static output, 20, 100 and 1 neurons were used, respectively. And in the dynamic output 40, 200 and 1 neurons were used, respectively. The authors found it easier to qualify the dynamic modes. For the single mode classification, the accuracy was 98.28%, higher than the multi-mode classification because involves the transition between modes.

BPNN and Radial Basis Function Neural Network (RBFNN) are implemented by Wang et al. [71], to detect 6 modes of locomotion with IMUs and plantar pressure sensors. The architecture consisted of 3 layers, with 20, 12 and 6 neurons. Results with BPNN were superior with 93.3% accuracy compared to 91.2%. However, these two models were outperformed by SVM, with an accuracy of 96.5%, achieving these results with linear kernel. Polynomial kernel was also used.

An SVM with Gaussian kernel is implemented by Villa-Parra et al. [72], to predict the intention of locomotion based on EMG. The authors used trunk muscle activity to compare it with leg muscle activity. Accuracy range between 76%-83% and 71%-77% for lower limb and trunk muscles respectively. This comparison was made to see the ability of the trunk muscles to predict the intention of locomotion, as an alternative, because the lower limbs are more prone to pathologies and weaknesses.

DT are implemented by Novak et al. [73] to predict the intention to start and finish walking, without using physiological signals. IMUs and pressure insoles were used. Gait initiation occurs when a standing person begins to walk. The termination of the march occurs when a person walks and decides to stand still. IMUs and pressure sensors were analyzed separately. For gait termination, the IMUs predicted better.

2.2.3 Starting point of this project

According to the literature review presented, different studies focus on joint angles and trajectory prediction, gait phases identification and classification of locomotion modes. Knowing these parameters and estimating their characteristics is essential for the future of smart prostheses and exoskeletons. In the reviewed works, the key points are ML algorithms and sensors.

Several ideas have been extracted from the literature as a guide for the completion of this work, especially from Section 2.2.2.1. Regarding the number of IMUs and their position, most of the studies presented propose them to be located in the pelvis, thighs, shins and feet, with the exception of one that proposes the sacral area. Various prediction models are used, both DL and simple regressors. It is of great value to know the results obtained with RF or Support Vector Regressor (SVR), among others, since they are models that are used in the experiments of this study. Regarding the validation methods, there are several that use the two that are also be applied in this study: Train-Test Split and Leave-One-Out. In both studies, better results are obtained when the data of the person being predicted is used, that is, with Train-Test Split, which further accentuates the challenge of extrapolating regressors. Regarding the evaluation methods, RMSE, R2, accuracy, MAE, Pearson, among others, are used, which gives credibility to the use of any of these methods to obtain the results of the experiments. Besides, depending on the study, volunteers have to do tests of walking, running at different speeds, leg extension with or without weight; and with different anthropomorphic characteristics. Finally, in one of the studies a method is used to make the database longer by simulating angles, obtaining that with a quality elongation of data, better results are obtained. This idea is of great value for this project.

This Bachelor's Thesis focuses on the prediction of lower limb angles, especially those formed by the knee joint. Four IMUs placed on the legs are used: on each shin and on each thigh. They are placed on the legs because they monitor leg exercises. Then, a UKF provides the reference data. Besides, different TW sizes are applied on the inertial data to predict the position of the LShin and Left Thigh (LThigh). Knowing the orientation of both sensors, the orientation of LShin with respect to LThigh can be directly obtained, which corresponds to the orientation of the knee. Previously, works have been presented in which different ML methods are used, such as [62]. Based on these studies, the models that are evaluated and compared are MLR, SVR with kernel lineal, polynomial and Gaussian, TB and MLP. However, not only the angles are predicted, but the number of IMUs is also intended to be optimized. All this information is relevant for its application in an intelligent prosthesis or exoskeleton for the knee joint.

Chapter 3

Theoretical Framework

This chapter explains the concepts and tools used to carry out the experiments of this work. Firstly, Section 3.1 briefly explains the concepts related to the biomechanics of human gait. Then, Section 3.2 provides an overview of AI, including the existing types of machine learning and the different algorithms used in the experiments. The two validation methods of the experiments are explained in Section 3.3. Section 3.4 summarizes the uses and composition of IMUs. Then, Section 3.5 explains Bayesian filters and specifically the Kalman Filter (KF). Finally, Section 3.6 a signal segmentation method based on TWs.

3.1 Human gait biomechanics

Human biomechanics is a discipline that combines the principles of biomechanics with human anatomy and physiology to understand how body structures work during movement [74]. Some of the structures involved are the following:

- Leg bones: femur, tibia and fibula. On the other hand, the bones of the foot. They act as movement levers, provide support and stability while walking.
- Hip, knee and ankle joints. They are points where two or more bones meet. The shape and type of joint determine the types of movements that are possible.
- Leg muscles. They are the motors of movement, they provide stability and balance.
- Musculature and bones of the upper extremities. Rocking is performed to maintain balance and achieve a symmetrical and stable gait.
- Nervous system. Coordinate movements while walking.

The gait is made up of a cycle that can be seen in Figure 2.1.This gait cycle includes the time the heel makes contact with the ground until the same heel touches the ground again.

3.2 Artificial Intelligence

Artificial Intelligence refers to the ability of machines to simulate human thought processes and perform tasks that would normally require human intelligence [75]. The main goal of AI is to create systems that can perform cognitive tasks such as reasoning, problem solving, decision making, and understanding human language.

ML is a subdiscipline of AI that focuses on the development of algorithms and models that allow computers to learn from data. Rather than being explicitly programmed, machines can improve their performance on a specific task as they are provided with more data and experience. ML is based on the idea that computers can learn patterns and make predictions or decisions based on those patterns.

On the other hand, DL is a more complex subset within ML that uses Deep Neural Network (DNN). These networks are characterized by having multiple hidden layers capable of finding relationships between the input variables and the variables to be predicted, so they are capable of tackling more complex problems.

3.2.1 Machine Learning Paradigms

Machine learning algorithms rely on using input and output data to train a model and make predictions or classifications [76]. The input is a set of features, usually called x, that are given to the algorithm to learn. The output data forms a response variable, often called y, which represents the expected response that the algorithm should produce from the input data. The algorithm adjusts the model parameters using the data. After training, the algorithm can take new input data and make predictions based on the knowledge gained during training [77].

Machine learning is divided into two types depending on whether the data is labelled: supervised learning and unsupervised learning. Supervised learning involves training with labeled data to predict new outcomes. Depending on the relationship of the labels to the data, two subtypes of machine learning can be distinguished: classification and regression. In classification, the set of labels is countable and the goal is to assign categories, while in regression, the set of labels is uncountable and the goal is to predict continuous values. On the other hand, unsupervised learning involves training with unlabeled data, searching for patterns and characteristic structures. One of the most common techniques of this type is clustering, a technique used to group similar data together [78].

There is a third type within machine learning, reinforcement learning. In reinforcement learning, an agent interacts with the environment by making sequential decisions to maximize a reward obtained based on the correct set of actions. The agent learns through trial and error, through continuous interaction with the environment.

In this work, different supervised learning methods are used, since the data in the database used is labelled. All the models used are regression models and one DL model.

3.2.2 Machine learning algorithms

This section explains the models used to carry out the experiments of this work: MLR, SVR, TB and MLP.

3.2.2.1 Multiple Linear Regression

Multiple linear regression is a supervised machine learning technique used to predict a continuous numeric value based on multiple input variables. One seeks to establish a linear relationship between two or more independent input variables and one output variable. The goal is to find a line or hyperplane that best fits the data, minimizing the difference between the model predictions and the actual values [79].

The multiple linear regression model can be represented mathematically as:

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n + \epsilon \tag{3.1}$$

where y is the output variable to be predicted; $x_1, x_2, ..., x_\mathrm{n}$ are the input variables or features; $b_0, b_1, b_2, ..., b_\mathrm{n}$ are the coefficients or weights associated with each input variable and ϵ is the model error (a.k.a. how much variation there is in our estimate of y).

To train the model, training data containing the values of the input variables and the corresponding output variables are used. Through optimization techniques, such as the method of least squares, the coefficients that best fit the data are estimated.

Once the model is trained, predictions can be made on new data. Given a set of values of the input variables, the model calculates the predicted value of the output variable using the coefficients learned during training.

3.2.2.2 Support Vector Machine

Support Vector Machine is used both for classification with the Support Vector Classifier (SVC) model and for regression with the SVR. Intuitively, the SVM is a model that represents the sample points in space, separating the classes into two spaces as wide as possible by means of a separation hyperplane, which is defined as the vector between the two points of the two largest classes are close, and this vector is called the support vector [80]. In this sense, an SVM builds a hyperplane or a set of hyperplanes in a very high or even infinite dimensional space that can be used in classification or regression problems.

The SVC chooses a hyperplane to separate the groups, with the greatest possible distance. A good margin is one in which you have the maximum distance from the support vectors to the two classes. The representation of this algorithm can be seen in Figure 3.1. Through optimization and the use of kernel functions, the algorithm can handle non-linearly separable data.

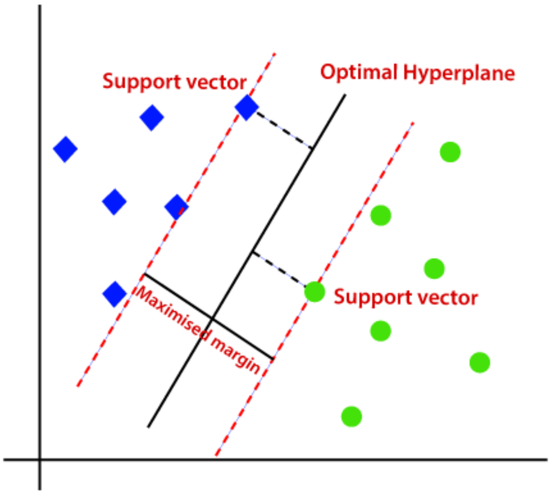


Figure 3.1: Support Vector Classifier graph [81].

On the other hand, SVR uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (ϵ) is set in approximation to the SVM which would have already requested from the problem [80]. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. The representation of an SVR with the margins maximized can be seen in Figure 3.2, whose equation of the hyperplane is:

$$\mathbf{y} = \mathbf{wx} + b \tag{3.2}$$

where w is the slope and b the distance of the hyperplane from the origin.

The optimization problem is established in the Equation (3.3) with its constraints.

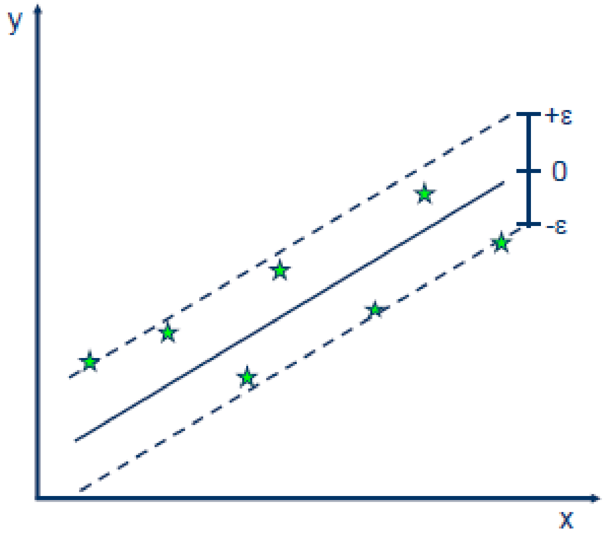


Figure 3.2: Ideal Support Vector Regressor graph [81].

$$
\begin{aligned}
&\min \quad \frac{1}{2} \parallel w \parallel^2 \\
\text{s.t.} \quad & y_i - wx_i - b \leq \epsilon \\
& wx_i + b - y_i \leq \epsilon
\end{aligned}
\tag{3.3}
$$

However, that would be the ideal case, but dealing with data in common would have data that does not fall within the margins, as shown in Figure 3.3. The function of the hyperplane is the same as in Equation (3.2). But, in this case, Equation (3.4) must be minimized.

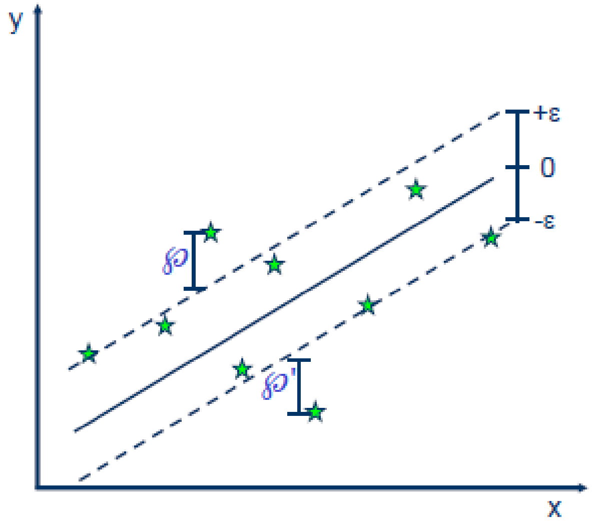


Figure 3.3: Linear Support Vector Regressor graph [81].

$$\begin{aligned} \min \quad & \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{N} (\wp + \wp') \\ \text{s.t.} \quad & y_i - wx_i - b \leq \epsilon + \wp_i \\ & wx_i + b - y_i \leq \epsilon + \wp'_i \\ & \wp_i, \wp'_i \geq 0 \end{aligned} \tag{3.4}$$

In the case of a **linear** kernel, the SVR is given by:

$$y = \sum_{i=1}^{N} (\alpha_i + \alpha'_i) \langle x_i, x \rangle + b \tag{3.5}$$

When the problem is not linear, the kernel function transforms the data into a higher dimensional feature space to make it possible to perform linear separation (Equation 3.6 and Equation (3.7). A representation of this process is shown in Figure 3.4.

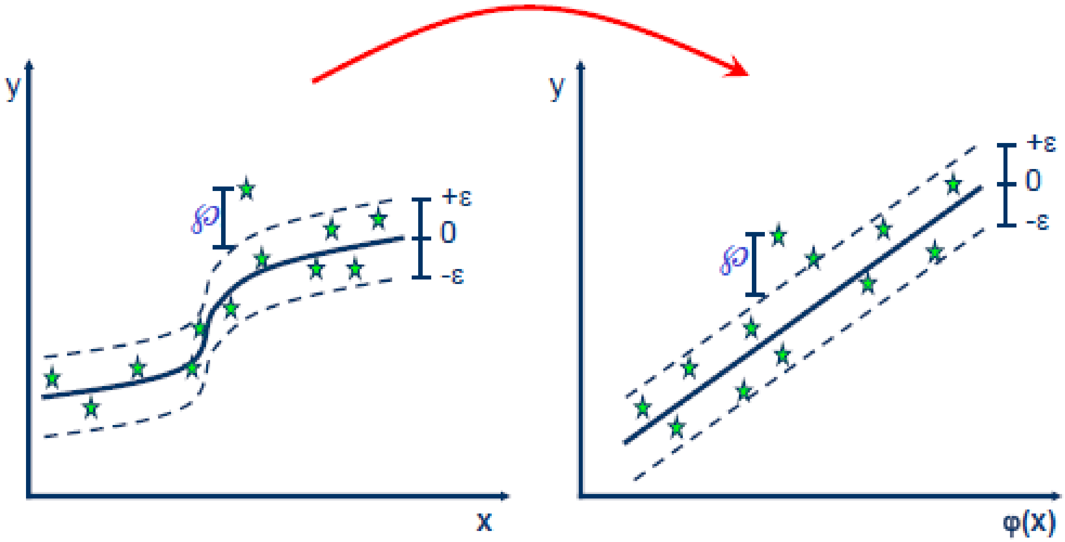


Figure 3.4: Transformation of SVR kernel data [81].

$$y = \sum_{i=1}^{N} (\alpha_i + \alpha'_i) \langle \varphi(x_i), \varphi(x) \rangle + b \tag{3.6}$$

$$y = \sum_{i=1}^{N} (\alpha_i + \alpha'_i) K(x_i, x) + b \tag{3.7}$$

When the problem is nonlinear, the kernel can be **polynomial**:

$$K(x_i, x_j) = (x_i, x_j)^d \tag{3.8}$$

or **Gaussian**:

$$K(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right) \tag{3.9}$$

In this work the SVR used for the experiments are Linear, Polynomial and Gaussian.

3.2.2.3 Regression Trees (Treebagging)

Treebagging, also known as Bagged Decision Trees or Bootstrap Aggregating [82], is an assembly technique used in machine learning to improve the accuracy, performance, and robustness of DT models. This technique is especially useful when working with models that tend to overfit the training data. The central idea behind this technique is the combination of multiple DTs, each one training on a slightly different version of the original training set.

For classification, the button would be obtained by the majority of each algorithm (the mode) and for regression, the mean of the predictions. An example of this is represented in the Figure 3.5. The Treebagging process involves the following steps:

1. **Bootstrap Sampling:** A series of training sets are generated through a process known as bootstrap sampling. In this process, samples are randomly selected from the original training set with replacement. Due to this sampling with replacement, each bootstrap training set is slightly different, which introduces variability in the training data.

2. **Tree training:** A DT is trained on each bootstrap training set. Each tree is built using a different subset of training data and may have different structures due to the variability introduced by bootstrap sampling.

3. **Combined prediction:** Once all the trees have been trained, a combined prediction is performed. In the case of classification problems, the final prediction could be the result of a majority vote among the individual trees. In regression problems, the final prediction might be the average of the predictions of all the trees.

The main advantage of Treebagging lies in its ability to reduce overfitting. By introducing variability into the training data through bootstrap sampling and by combining

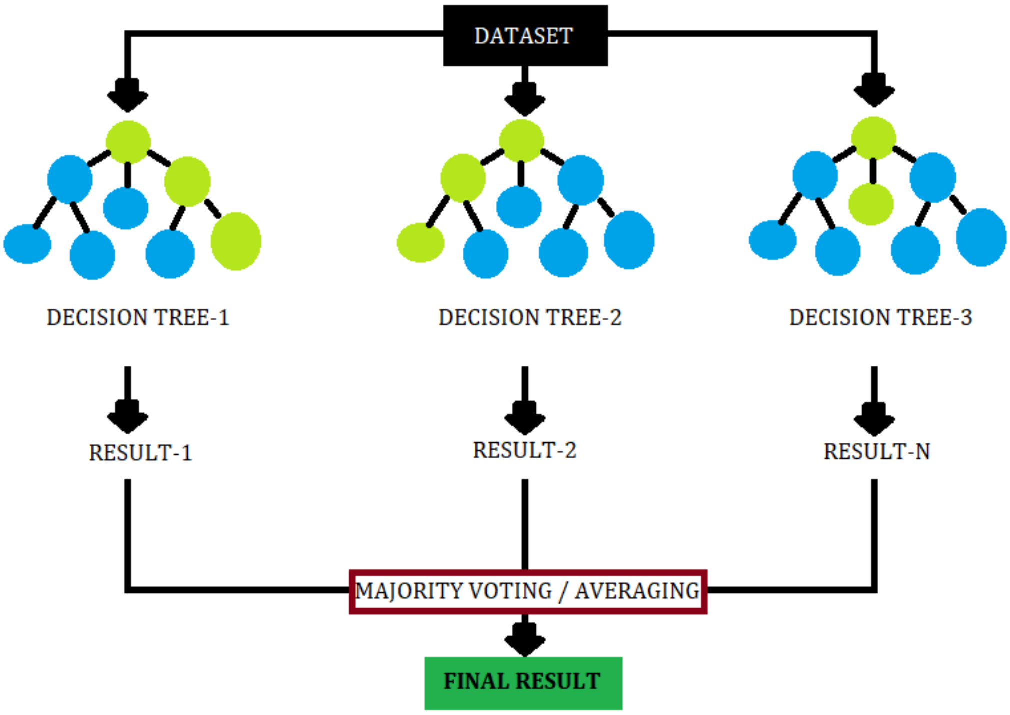


Figure 3.5: How Bagging Tree works.

multiple models, a more robust prediction is obtained that generalizes better to unseen data. Also, because each tree is trained on a different subset of data, all trees are less likely to make the same errors, further improving the overall accuracy of the model.

3.2.2.4 Multilayer Perceptron

Multilayer Perceptron is a basic artificial neural network architecture consisting of multiple layers of neurons [83]. The three main types of layers are: an input layer, one or more hidden layers, and an output layer [84]. Each layer contains a series of interconnected neurons as shown in Figure 3.6. Besides, the structure of a neuron is detailed in Figure 3.7.

The inputs are the features of the model, let say $x_1, x_2, x_3..., x_\mathrm{n}$. Then, each weight is associated with an input and determines its importance in relation to the output. They are denoted as $(\omega_1, \omega_2, ..., \omega_n)$ composing the vector $\boldsymbol{\omega}$. The transfer function is the mathematical operation performed to combine the inputs of the model with the weights associated with each input. It is calculated by multiplying each input by its corresponding weight and adding the results. It is a weighted sum. Then, another mathematical function, the activation function, transforms the weighted input of the perceptron into a nonlinear output. Some of the most common activation functions are the sigmoid, hyperbolic or *Rectified Linear Unit (ReLu)*. Finally, the output is the results of the estimation.

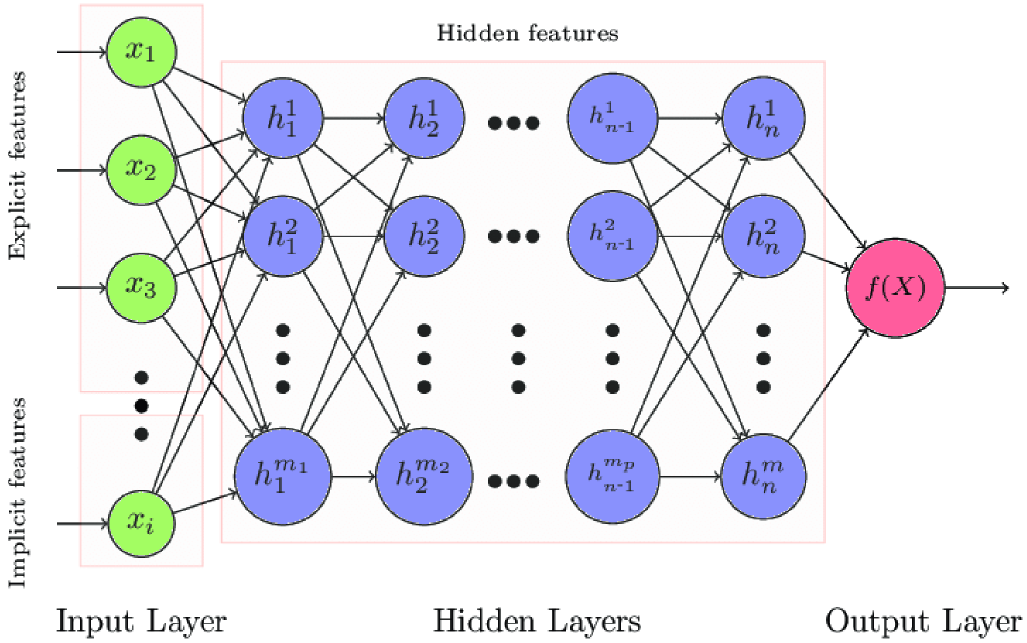


Figure 3.6: Representation of the structure of the multilayer perceptron [83].

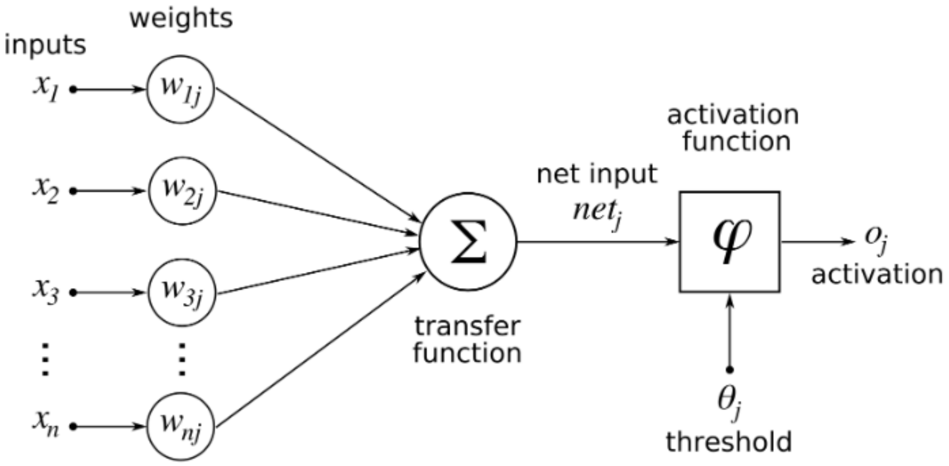


Figure 3.7: Representation of the structure of a neuron [85].

During the MLP training process, the weights are fit iteratively using optimization algorithms, such as error backpropagation, to minimize the difference between the predicted and actual outputs. The MLP is capable of learning and representing complex nonlinear functions due to the presence of multiple hidden layers and nonlinear activation functions. This allows you to model more sophisticated relationships between the input variables and the output variable.

3.3 ML model validation

Validating ML models is a critical step in evaluating the performance and generalization ability of a model before putting it into production or using it in real-world situations. Validation helps ensure that the model is not only memorizing the training data, but is also capable of making accurate predictions on new, previously unseen data. Based on the performance of the model with unknown data, you can determine if it still needs to be adjusted, has been overfitted, or is well generalized.

One of the most used techniques to test the effectiveness of an ML model is "cross-validation". This method is also a re-sampling procedure that allows a model to be evaluated even with limited data. To carry out cross-validation, it is necessary to previously separate a part of the data from the training data set. That data is not used to train the model, but later to test and validate it.

Cross-validation is often used in ML to compare different models and select the most suitable one for the specific problem. Next, the two types of validation that have been used in this project are explained depending on whether or not the data of the person to be predicted is known.

3.3.1 K-Fold cross-validation

The K-Fold technique involves dividing the data set into k approximately equal segments or folds and then performing multiple iterations of the model training and evaluation process using different combinations of these folds. For example, if there are 1000 samples and k equals 5, each fold would contain about 200 samples.

The process is repeated k times. In each iteration, a different fold is used as the test set, while the other $k-1$ folds are used as the training set. This means that in each iteration, the model is trained $k-1$ times on different training sets and evaluated on a different test set.

For each iteration, evaluation metrics, such as precision, RMSE, or any other relevant metrics, are calculated using the corresponding test set. These metrics are stored or averaged to obtain an overall measure of model performance.

After completing the k iterations, the evaluation metrics from all iterations are averaged to obtain a single measure of model performance. This provides a more robust and generalizable evaluation of the model as it has been tested on different subsets of data. An example of this technique is shown in Figure 3.8 where k equals 5, dividing the data into five sets. In this work, in experiments in which subjects with known data are predicted, k equals 5 is also used.

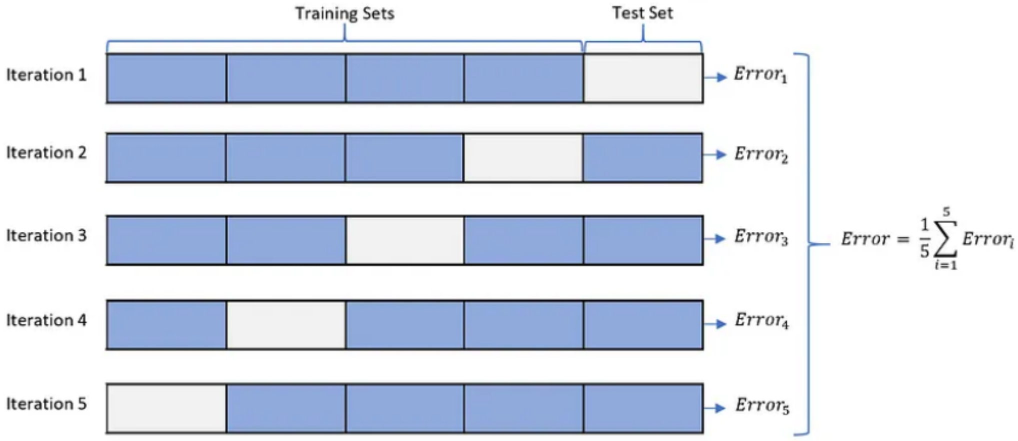


Figure 3.8: Example of 5-Fold cross-validation [86].

The error estimate is highly variable depending on which observations are included as the training set and which as the validation set. This type of cross-validation is useful to obtain a more reliable evaluation of the model and reduce the risk of overfitting.

3.3.2 Leave-One-Out cross-validation

The leave-one-out method is an iterative method that begins using all available observations as a training set except one, which is excluded for use as validation. If a single observation is used to calculate the error, it varies greatly depending on which observation has been selected. To avoid this, the process is repeated as many times as observation available are, excluding a different observation in each iteration, adjusting the model with the rest and calculating the error with said observation. Finally, the error estimated is the average of all the errors calculated. An example of this technique is shown in Figure 3.9 dividing the data into N sets.

This method allows us to reduce the variability that arises if the observations are randomly divided into only two groups. This is because at the end of the process all the available data is used for both training and validation. As there is no random separation of the data, the results are fully reproducible. The main disadvantage of this method is its computational cost.

In this project, this method is applied to predict subjects with unknown data. As five volunteers are used, there are five iterations to obtain the absolute error. This method allows to test the extrapolability of the regressors.

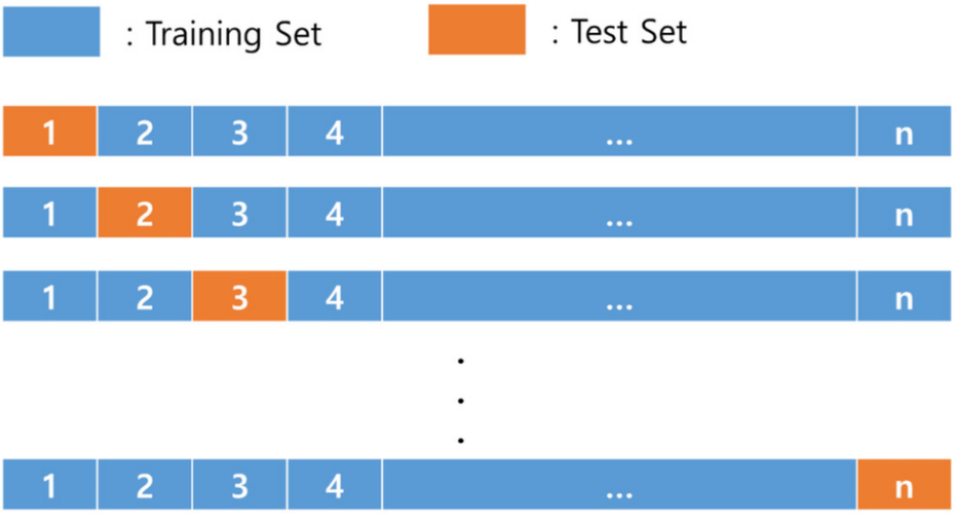


Figure 3.9: Schematic representation of the leave-one-out cross-validation method. The indices refer to individual subjects [87].

3.4 Inertial measurement units: IMUs

An IMU is an electronic device that allows to obtain measurements of angular and linear velocity, orientation and gravitational forces. This is achieved by combining accelerometers, gyroscopes and sometimes magnetometers. These devices are used in mobile phones, navigation systems or study movement [88]. These types of units internally implement three orthogonal axes on which the uniaxial sensors are mounted, so that each axis is assigned an accelerometer, a magnetometer and a gyroscope, as shown in Figure 3.10.

These IMUs work according to Newton's first two laws. The motion of a body remains constant until an external force acts on it, according to Newton's first law. On the contrary, Newton's second law states that the applied force generates an acceleration equivalent to it with Equation (3.10):

$$\mathbf{F} = m \cdot \mathbf{a} \tag{3.10}$$

where $\mathbf{F}$ is the force, m is the mass and $\mathbf{a}$ is the acceleration.

By measuring the instantaneous linear acceleration ($\mathbf{a}$) of the body at a specific moment and gravity ($\boldsymbol{g}$), the accelerometer provides the linear acceleration that corresponds

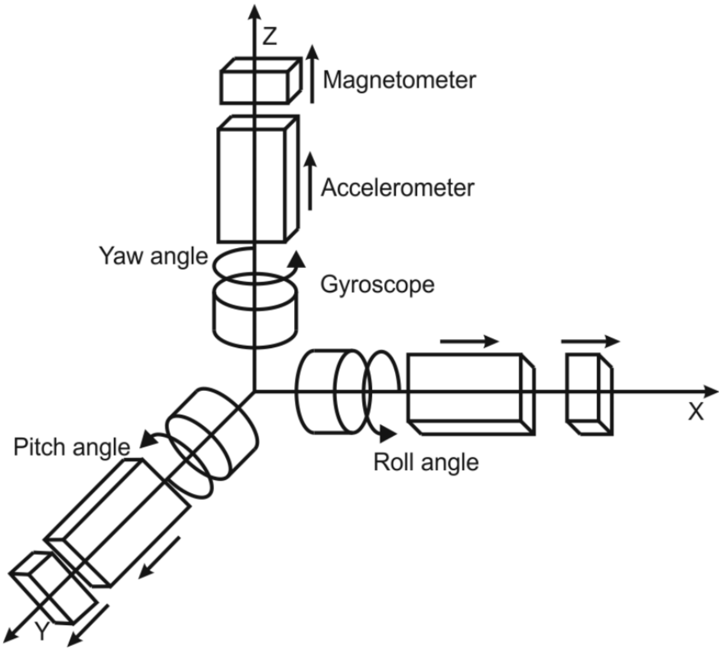


Figure 3.10: The inertial measurement unit body coordinate system. Each axis has an accelerometer and a gyroscope. Rotation of angles in their corresponding axes [89].

to the second derivative of its position. On the other hand, the gyroscope allows to measure the angular velocity ($\boldsymbol{\omega}$) of the body with respect to the reference system, with which it is possible to know the rotation of the body in 3D space. Finally, the magnetometer provides information on the orientation of the body with respect to the terrestrial magnetic north. It acts like a compass, so it is sensitive to the magnetic field. This sensitivity to magnetic fields is the reason why the magnetometer data is not used in this work.

Currently there are many IMUs of different sizes and with different combination of sensors depending on the function to be developed. This allows its use in different parts of the body for the monitoring and estimation of data of interest [90].

3.4.1 Orientation Estimation

Sensor orientation estimation using a KF is a common approach in navigation and signal processing to determine the orientation of an object or system, such as a vehicle, drone, or mobile device [91]. The KF is a best estimation algorithm that combines noisy measurements and predictions of the system state to provide an accurate estimate of the actual state.

In the context of orientation estimation, a quaternion representation is typically considered to describe three-dimensional orientation. Quaternions are an extension of complex numbers and are used to represent rotations in 3D space efficiently and without singularity issues found in other representations, such as Euler angles.

3.4.2 Quaternions

Quaternions are a mathematical extension of complex numbers. Unlike complex numbers, which have a real part and an imaginary part, quaternions have four components: a scalar part and three vector parts [92]. They are usually represented as:

$$\mathbf{a} + \mathbf{b}i + \mathbf{c}j + \mathbf{d}k \tag{3.11}$$

where **a** is the scalar part and **b**, **c** and **d** are the vector parts in the directions i, j and k, respectively.

Quaternions are especially useful for representing rotations in 3D space. They can compactly and unambiguously describe any rotation without suffering from singularity problems, such as those found in Euler angle representations. Also, calculations of rotations and combinations of rotations are done efficiently with quaternions.

In the context of orientation estimation and navigation, quaternions are used to represent an object's orientation in 3D space. Orientation changes are expressed as quaternion multiplications, making it easy to combine and merge rotations.

3.4.3 Euler Angles

Orientation estimation with Euler angles is a common approach to describe the three-dimensional orientation of an object or system. In order to represent the angles of rotation on the three-dimensional axes, it is necessary to know how Euler angles work [93]. Said angles form a set of three angular coordinates that are used to determine the orientation of a reference system, normally mobile, with respect to another fixed reference system.

An angle is represented as the sequence of three angles, each of them rotated about a certain axis, always following a previous established order: the angle of rotation about the Z axis is defined as ψ or α (Yaw), about the Y axis it is θ or β (Pitch) and on the X axis is ϕ or γ (Roll). Figure 3.10 shows the rotation of these angles on their corresponding axes. To better understand the name of each turn, you can see a rule of thumb in Figure 3.11.

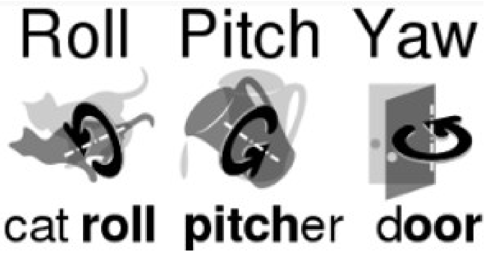


Figure 3.11: Mnemonic rule for Euler angles [94].

However, working directly with Euler angles can lead to certain problems, such as gimbal-lock singularities, where a rotation can lose a degree of freedom and cause unexpected results. This is due to the way the Euler angles interact with each other as is shown in Figure 3.12.

For orientation estimation with Euler angles, the process typically involves the following steps:

1. **Representation of Euler angles:** Euler angles can be represented in different ways, such as Tait-Bryan or XYZ, ZXZ, among others. Each representation defines a specific set of rotations around the axes.

2. **Measurements and Sensors:** Measurements from sensors such as gyroscopes, accelerometers, and magnetometers provide information about the object's orientation in space.

3. **Orientation update:** Using the current Euler angles and sensor measurements, a new orientation estimate is calculated. This is typically done using sensor fusion algorithms that combine information from multiple sources to improve accuracy.

4. **Singularities and Gimbal-Lock:** One of the disadvantages of working with Euler angles is that singularities, such as gimbal-lock, can occur in certain configurations. This limits the range of rotations that can be described without problems.

5. **Conversion to other representations:** Sometimes, to avoid singularity problems and improve stability, it is useful to convert Euler angles to other representations, such as quaternions or rotation matrices, for calculations and estimations.

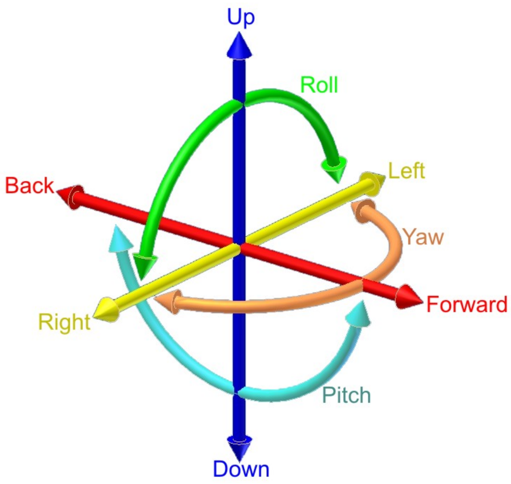


Figure 3.12: The six degrees of freedom: forward/back, up/down, left/right, pitch, yaw, roll [95]

3.5 Bayesian Filters: The filtering problem

Bayesian filters play a critical role in sensor fusion, which is the process of combining information from multiple sensors to obtain a more accurate and reliable estimate of a variable or state of interest [96]. A classic example of an application is the fusion of data from gyroscopes, accelerometers and magnetometers to estimate the orientation of an object in space.

Sensors provide measurements of the same or different aspects of the state of the system. These measurements may be subject to noise and errors. A model is established that relates the measurements of the sensors with the true state of the system. This includes the characteristics of the sensors and the uncertainty associated with their measurements.

A dynamic model is defined that describes how the state of the system evolves over time. This model can be a mathematical representation of physical interactions and relationships between variables. A Bayesian filter, such as the KF or particle filter, is used to combine the sensor measurements with the state model to produce a more accurate and up-to-date estimate of the actual state of the system.

In the prediction stage, the filter uses the state model to predict how the state of the system changes in the next time step. In the update stage, the filter adjusts the state estimate using current sensor measurements and compares them with the state model predictions. The measurement and model uncertainty are used to calculate a weighted estimate that optimizes the data fusion.

As new measurements are received, the filter adjusts the estimate and the associated uncertainty. This allows the system to take into account the quality and reliability of the measurements over time.

In sensor fusion, Bayesian filters make it possible to take advantage of complementary information from multiple sensors and improve the robustness and precision of the estimates, even in situations where some sensors may be subject to noise or errors. In this work, the Bayesian filter to be used is the UKF, which is a derivation of the KF.

3.5.1 Kalman Filter

The Kalman filter is an estimation algorithm that combines observed measurements and dynamic models to provide optimal estimates of the state of a system. It works in two stages: prediction and update [97]. In the prediction stage, the state model is used to predict how the state of the system changes in the next time step. In the update stage, the estimate is adjusted using current measurements and compare them with the predictions. The KF minimizes the RMSE and is especially useful for linear and Gaussian systems. However, it may face challenges in non-linear systems or with non-Gaussian noise.

In this work, a variation of the KF, UKF, are used. The UKF is an extension of the KF that addresses the estimation problem in nonlinear systems. In nonlinear systems, the propagation of uncertainty through nonlinear transformations can lead to suboptimal or inaccurate solutions using the standard KF. The UKF uses a technique called "sigma point transformation" to approximate the statistics of the state distribution through non-linear transformations. This involves taking a set of points from the distribution, passing them through the nonlinear function, and then reconstructing the approximate distribution using these transformed points.

The UKF maintains many of the advantages of the KF in terms of computational efficiency and precision, but is better suited to nonlinear systems. However, the UKF may still have limitations in cases of highly non-linear systems or with non-Gaussian noise.

3.6 Signal Segmentation: Temporal Window

A TW is a concept used in the field of ML and time series analysis that refers to a specific and continuous interval of time within a sequence of data ordered in time. This window scrolls through the time series, allowing you to capture a set of data points within that interval for analysis.

Signal segmentation based on TWs in ML is essential to model patterns and relationships in time series. By dividing the series into overlapping or non-overlapping windows, ML algorithms can process and analyze specific segments of data instead of treating the entire series at once. This helps identify local patterns, trends and changes in data over time. In addition to analyzing future events or behaviors based on the past evolution of the data.

Chapter 4

Methodology

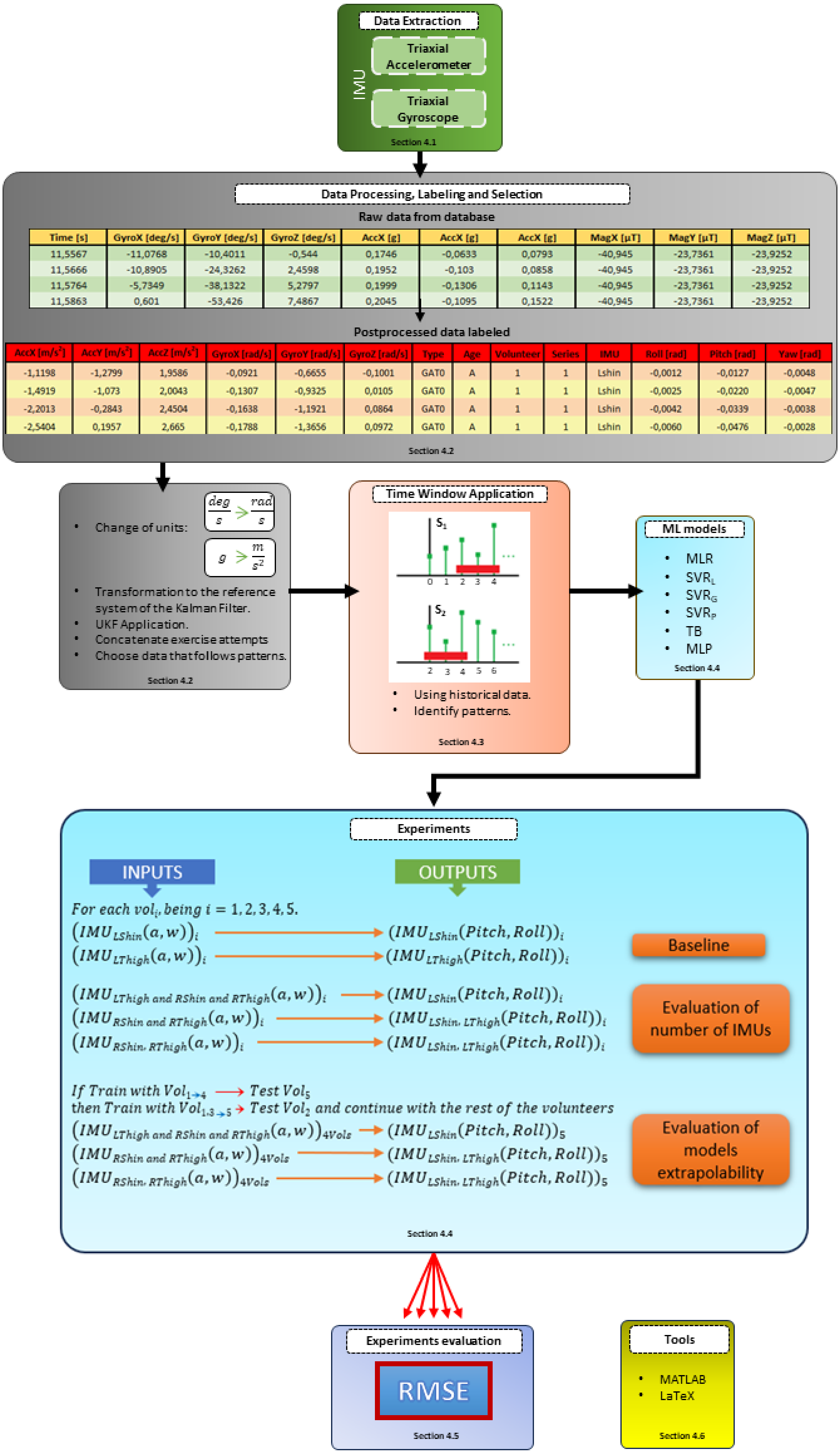
This chapter describes the methodology followed in the project. Figure 4.1 is the workflow that will be explained in detail in the following sections. Section 4.1 describes the database used, as well as how to obtain it. Section 4.2 shows the signals belonging to the database and their processing and labelling. On the other hand, Section 4.3 explains how signals are segmented, by using temporary windows, a method widely used in ML. Section 4.5 explains the algorithms used and the experiments carried out. Finally, Section 4.6 explains the evaluation method and Section 4.7 the tools used in this project.

4.1 Database

This project uses the PHYTMO database [16]. PHYTMO contains information on physical therapy exercises and gait variations of 30 healthy volunteers, aged between 20 and 70 years old.

This database includes data recorded with four IMUs (*NGIMU* IMUs, *X-io Technology*, Bristol, UK) and an optical system (*OptiTrack* system, *NaturalPoint Inc*). In this project only the data obtained by the IMUs are used since *OptiTrack* gait data is not available. Wearable sensors include three-axis gyroscope, accelerometer and magnetometer, with a range of 2000°/s, 16 g and 1,300μT, respectively. The sample rate was set to 100 Hz for the gyroscopes and accelerometers and to 20 Hz for the magnetometers. Regarding the characteristics of the IMUs, they have a size of 56 x 39 x 18 mm with a weight of 46 g. This small size allows it to be worn while doing exercises. Additionally, each IMU is mounted on an *ad-hoc* structure (mounting board) for its placing at the limbs.

The four IMUs are placed on the lower extremities with the $\mathrm{X_I}$-axis pointing to the ceiling, like the reference system that can be seen in Figure 4.2a. The IMUs are placed on the anterior surface so when volunteers are standing, the $\mathrm{Z_I}$-axis is perpendicular to

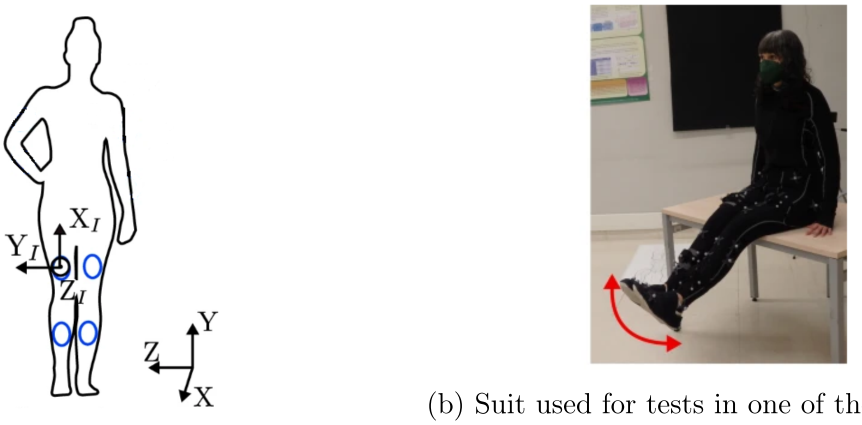


Data Extraction
IMU
Triaxial Accelerometer
Triaxial Gyroscope
Section 4.1
Data Processing, Labeling and Selection
Raw data from database
Postprocessed data labeled
Section 4.2
Change of units:
$\frac{deg}{s} > \frac{rad}{s}$
$g > \frac{m}{s^2}$
Transformation to the reference system of the Kalman Filter.
UKF Application.
Concatenate exercise attempts
Choose data that follows patterns.
Section 4.2
Time Window Application
S_1
0 1 2 3 4
S_2
2 3 4 5 6
Using historical data.
Identify patterns.
Section 4.3
ML models
MLR
SVR_L
SVR_G
SVR_P
TB
MLP
Section 4.4
Experiments
INPUTS
OUTPUTS
For each vol_i, being $i = 1, 2, 3, 4, 5.$
$(IMU_{LShin}(a,w))_i \rightarrow (IMU_{LShin}(Pitch, Roll))_i$
$(IMU_{LThigh}(a,w))_i \rightarrow (IMU_{LThigh}(Pitch, Roll))_i$
Baseline
$(IMU_{LThigh\ and\ RShin\ and\ RThigh}(a,w))_i \rightarrow (IMU_{LShin}(Pitch, Roll))_i$
$(IMU_{RShin\ and\ RThigh}(a,w))_i \rightarrow (IMU_{LShin,\ LThigh}(Pitch, Roll))_i$
$(IMU_{RShin,\ RThigh}(a,w))_i \rightarrow (IMU_{LShin,\ LThigh}(Pitch, Roll))_i$
Evaluation of number of IMUs
If Train with $Vol_{1\rightarrow 4} \rightarrow$ Test Vol_5
then Train with $Vol_{1,3\rightarrow 5} \rightarrow$ Test Vol_2 and continue with the rest of the volunteers
$(IMU_{LThigh\ and\ RShin\ and\ RThigh}(a,w))_{4Vols} \rightarrow (IMU_{LShin}(Pitch, Roll))_5$
$(IMU_{RShin\ and\ RThigh}(a,w))_{4Vols} \rightarrow (IMU_{LShin,\ LThigh}(Pitch, Roll))_5$
$(IMU_{RShin,\ RThigh}(a,w))_{4Vols} \rightarrow (IMU_{LShin,\ LThigh}(Pitch, Roll))_5$
Evaluation of models extrapolability
Section 4.4
Experiments evaluation
RMSE
Section 4.5
Tools
MATLAB
LaTeX
Section 4.6

Time [s]	GyroX [deg/s]	GyroY [deg/s]	GyroZ [deg/s]	AccX [g]	AccX [g]	AccX [g]	MagX [µT]	MagY [µT]	MagZ [µT]
11,5567	-11,0768	-10,4011	-0,544	0,1746	-0,0633	0,0793	-40,945	-23,7361	-23,9252
11,5666	-10,8905	-24,3262	2,4598	0,1952	-0,103	0,0858	-40,945	-23,7361	-23,9252
11,5764	-5,7349	-38,1322	5,2797	0,1999	-0,1306	0,1143	-40,945	-23,7361	-23,9252
11,5863	0,601	-53,426	7,4867	0,2045	-0,1095	0,1522	-40,945	-23,7361	-23,9252

AccX [m/s^2]	AccY [m/s^2]	AccZ [m/s^2]	GyroX [rad/s]	GyroY [rad/s]	GyroZ [rad/s]	Type	Age	Volunteer	Series	IMU	Roll [rad]	Pitch [rad]	Yaw [rad]
-1,1198	-1,2799	1,9586	-0,0921	-0,6655	-0,1001	GAT0	A	1	1	Lshin	-0,0012	-0,0127	-0,0048
-1,4919	-1,073	2,0043	-0,1307	-0,9325	0,0105	GAT0	A	1	1	Lshin	-0,0025	-0,0220	-0,0047
-2,2013	-0,2843	2,4504	-0,1638	-1,1921	0,0864	GAT0	A	1	1	Lshin	-0,0042	-0,0339	-0,0038
-2,5404	0,1957	2,665	-0,1788	-1,3656	0,0972	GAT0	A	1	1	Lshin	-0,0060	-0,0476	-0,0028

Figure 4.1: Representation of workflow.
)



(a) Sensor Location

(b) Suit used for tests in one of the calibration exercises.

Figure 4.2: Location of the IMUs on the lower limbs. Volunteer wearing the suit used for performing the exercises [16].

the coronal plane and the Y$_I$-axis is perpendicular to the sagittal plane. These positions are chosen for the comfort to place the sensors. Each IMU has an identifier with format *Xsegment*, where *X* refers to its position, left ("L") or right ("R"), being segment "thigh" or "shin", referring to the segment of the lower-limb the IMU was placed on. The IMUs are synchronized through exercises at the beginning of each recording. The suit used can be seen in Figure 4.2b.

The four IMUs placed in the lower extremities record the information in CSV files. These files have the name *GNNEEELP_S*, where *G* refers to the letter of the range of age ("A", "B", "C", "D" or "E"); *NN* is number of identification of the volunteer, from "01" to "10"; *EEE* indicates the type of gait variation ("GAT", "GIS" or "GHT"); *L* is the leg with the exercise is performed ("L" or "R"); *P* is a label that indicates the evaluation of the exercise performance, "0" value when the file contains the correctly performed exercise and "1" when exercises are wrongly performed; and finally, *S* indicates the index of the series, being "1" and "2" for the first and second recorded series.

For this project, 5 randomly volunteers are used, whose anthropomorphic data are detailed in Table 4.1. The chosen gait exercise is "GAT", which corresponds with natural gait.

Table 4.1: Anthropomorphic information of the chosen volunteers.

Volunteer	Range	Id	Age (years)	Height (cm)	Weight (Kg)	Sex (M/F)
1	A	A01	22	165	58	F
2	A	A07	25	175	72	M
3	C	C02	46	178	69	M
4	A	A04	23	180	72	M
5	B	B01	30	179	76	M
Summary	-	-	29.2(±8.84)	175.4(±5.46)	69.4(±6.11)	4m,1f

4.2 Data Processing, Labeling and Selection

In each “GAT” file there is information about the time, the accelerometer, the gyroscope and the magnetometer in the three axes. This can be seen in Table 4.2.

Table 4.2: Data belonging to file *A01GAT0_1* of the database before processing it.

Time [s]	**GyroX [deg/s]**	**GyroY [deg/s]**	**GyroZ [deg/s]**	**AccX [g]**	**AccY [g]**	**AccZ [g]**	**MagX [T]**	**MagY [T]**	**MagZ [T]**
11.5567	-11.0768	-10.4011	-0.5444	0.1746	-0.0633	0.0793	-40.9450	-23.7341	-23.9252
11.5666	-10.8905	-24.3262	2.4598	0.1954	-0.1030	0.0858	-40.9450	-23.7341	-23.9252
11.5764	-5.7349	-38.2322	5.2797	0.1999	-0.1306	0.1143	-40.9450	-23.7341	-23.9252
11.5865	0.6010	-53.4260	7.4667	0.2045	-0.1095	0.1522	-40.9450	-23.7341	-23.9252
...	...	...	...	...	...	...	...	...	...

In this project, neither the time variable nor the data from the magnetometer are used. The remaining data that is useful is the three axis gyroscope and accelerometer. The accelerometer data is in g (gravity acceleration) and is converted to m/s^2, as represented in Equation(4.1). On the other hand, the gyroscope data is in ˚ (degrees) and is converted to rad using the Equation (4.2). This change of units is necessary since the UKF is programmed in gs and rads.

$$1g = 9.8\frac{m}{s^2} \tag{4.1}$$

$$1rad = \frac{180^\circ}{\pi} \tag{4.2}$$

The next step is to transform the axes to the reference system in which the UKF is programmed, with Z up and Y to the side. Once the angles have been rotated, the filter is applied to obtain the Euler angles: Roll and Pitch.

All samples are going to be labelled according to the type of movement, the IMU and the series to which the angles belong; the age group to which the volunteer belongs and their identification. An example of the data after joining the inertial data, the motion angles and labeling the samples can be seen in Table 4.3.

Table 4.3: Data belonging to file *A01GAT0_1* of the database after processing it.

AccX [m/s2]	...	**GyroZ [rad/s]**	**Type**	**Age**	**Volunteer**	**Series**	**IMU**	**Roll [rad]**	**Pitch [rad]**
-1.1198	...	-0.1001	GAT0	A	1	1	Lshin	-0.0012	-0.0127
-1.4919	...	0.0105	GAT0	A	1	1	Lshin	-0.0025	-0.0220
-2.2013	...	0.0864	GAT0	A	1	1	Lshin	-0.0042	-0.0339
-2.5404	...	0.0972	GAT0	A	1	1	Lshin	-0.0060	-0.047
...	...	...	...	...	...	...	...	...	...

Each volunteer has two attempts of the gait recorded, so the two attempts of natural gait exercise are concatenated. Figure 4.3 shows the two concatenated attempts of Volunteer 2. It is observed, in red, that at the beginning and at the end of the recordings there are movements that do not correspond to gait, so these data are eliminated. The chosen data that is used in the experiments is the green one. The left part in green corresponds to the first attempt and the one on the right to the second. Both green segments have the same number of samples.

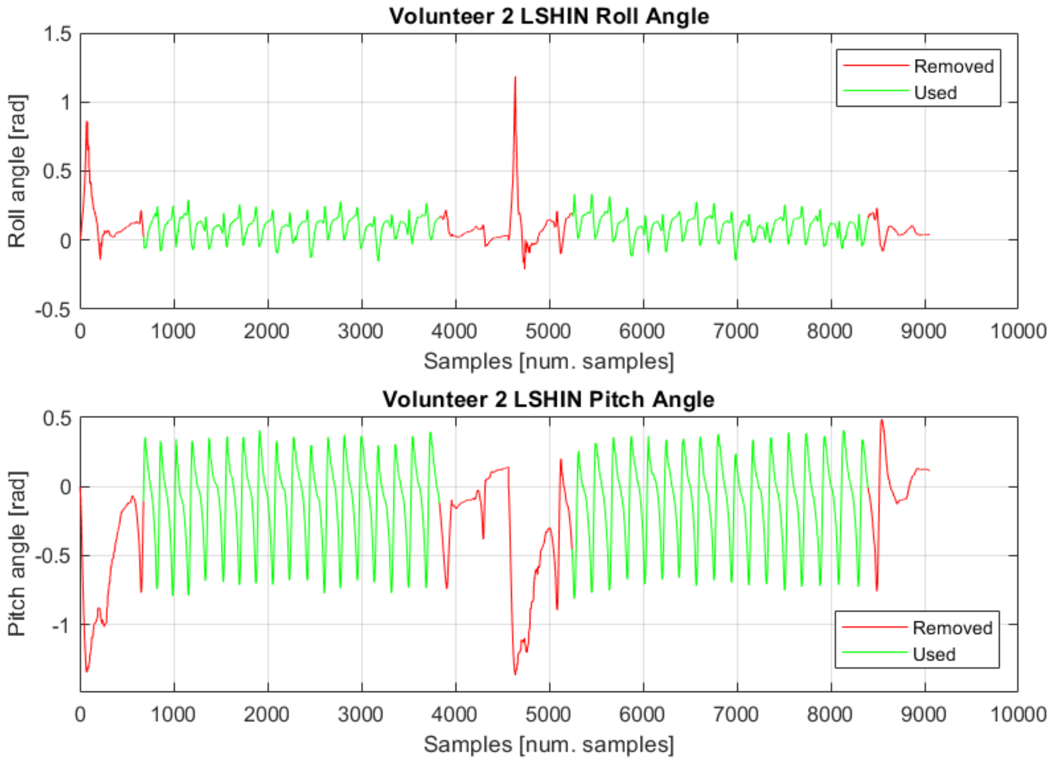


Figure 4.3: Volunteer 2 LShin angles. The measurement frequency is 100 Hz.

4.3 Signal Segmentation

To assess the results of the different methods used in the experiments, the original recorded inertial measurements (gyroscope and accelerometer) are used as inputs. Different MLs models are applied to see how they affect the prediction.

In this work, from the current values, previous values are also taken into account. For each TW sample, there is a vector with W measurements of each of the sensor axes, which are the inputs; and a single sample of the angle at which the IMU is oriented at the final time instant of that interval, which is the output. An example with Pitch angle can be seen in Table 4.4. When creating a TW, the number of input columns is multiplied W-times, where W is the number of TW applied. On the other hand, the number of rows is reduced by W-times plus one.

In the first three experiments, TWs with n = 5, 10, 20, 40 and 60 samples are used to obtain the results of the MLR, SVRs and TB. A maximum of 60 samples are used due to

Table 4.4: Signal resulting from applying a temporal window of 3 samples.

$\mathbf{Acc_X}$ **(t-2)**	$\mathbf{Acc_X}$ **(t-1)**	$\mathbf{Acc_X}$	$\mathbf{Acc_Y}$ **(t-2)**	**...**	$\mathbf{Gyro_Z}$	**PITCH**
Acc_X (0)	Acc_X (1)	Acc_X (2)	Acc_Y (0)	...	$Gyro_Z$ (2)	Pitch (2)
Acc_X (1)	Acc_X (2)	Acc_X (3)	Acc_Y (1)	...	$Gyro_Z$ (3)	Pitch (3)
Acc_X (2)	Acc_X (3)	Acc_X (4)	Acc_Y (2)	...	$Gyro_Z$ (4)	Pitch (4)
Acc_X (3)	Acc_X (4)	Acc_X (5)	Acc_Y (3)	...	$Gyro_Z$ (5)	Pitch (5)

the high training time involved. Furthermore, as time increases, greater movement of the person is considered. However, only 5-sample and 10-sample TW are used to predict with MLP, since a greater number of samples with this model represents a high computational cost that cannot be addressed.

4.4 Implementation of algorithms

The prediction of gait angles are evaluated with four regressors that are MLR,SVR, TB and MLP. Their implementation using the MATLAB programming language is explained below.

In the case of MLR, the function *fitlm* is used in which the training data is introduced. This function creates a model, which is entered into the *predict* function along with the inputs of the test set, so that it provides the prediction of the outputs of the test set. All that remains is to use the RMSE evaluation method to calculate the error committed.

For SVR, the function *fitrsvm* is used in which several arguments must be defined. First of all, it is used *standardize* and add *true* so that the software centers and scales each column of the predictor data by the weighted column mean and standard deviation, respectively. Afterwards, a kernel function is selected by adding *KernelFunction* and depending on what we want, *linear*, *gaussian* or *polynomial* are used. Finally, the *predict* function is used to obtain the estimated outputs by using the trained model provided by *fitrsvm*.

To use TB, the *TreeBagger* function is applied in which the training data is introduced and 100 decision trees are established as a parameter. Additionally, it is indicated that the type of decision is a regression. The *predict* function is used again with the trained model provided by *TreeBagger*.

Finally, for the MLP NN, besides applying a single layer and two layers, they require the selection of the number of neurons per layer. The number of neurons in a single layer are applied using a positive integer number. In the case of two layers, a matrix of positive integer numbers is written, the first number being the number of neurons in the first layer; and the second number, the neurons of the second layer. This is added to the function *feedforwardnet* with the argument *"traingdx"* which applies that the gradient descent learning rate is variable. The learning rate is set to 0.01 and the number of epochs to 100. This is entered into the *train* function along with the training data providing the model as a result. This model is used together with the test set inputs in the *sim* function to obtain the prediction.

4.5 Description of the experiments

This section is divided into two parts according to the evaluation method used and depending on whether the data of the subjects to be predicted are known or not. The first part is in Section 4.5.1 where the data of the volunteers to be predicted is known and is made up of five experiments. The last three experiments, explained in Section 4.5.2, focus on evaluating the extrapolation of the models.

4.5.1 Angle prediction in five independent volunteers

With the purpose of evaluating the performance of the models in the use of the IMUs to predict angles, five experiments are carried out. The body segments of which we want to know the orientation are always LShin and LThigh.

Firstly, the intra-subject predictions are evaluated, where the volunteers' gait data are known. As explained previously, there are five independent volunteers for the experiments (see Table 4.1). The training and testing sets are from the same volunteer. The evaluation method is K-fold cross-validation with K equal to 5, so at all times 80% of the data is used for training and the remaining 20% for testing. The exact function used is *cvpartition(group,'KFold',k)* in MatLab, where "group" is all the dataset and K the number of folds. This function gives us as a result the inputs and outputs of the training set and the inputs and outputs of the testing set. Training and test set are iterated 5 times with the same ML model for the same volunteer, being different each time the test set. The average and standard deviation of the RMSE of the 5 iterations on the 5 volunteers is evaluated.

This workflow is followed in experiments evaluating data from individual subjects. However, the input data used change according to the number of IMUs and their location to be evaluated. The first two experiments, in Section 4.5.1.1 serve as a baseline for the rest of the experiments and in the remaining three, in Section 4.5.1.2, the prediction of angles is analyzed by optimizing the number and position of IMUs.

4.5.1.1 Baseline experiments

The first two experiments consists of predicting the position of one IMU using its inertial data recorded. The first experiment using only the IMU LShin of each volunteer to predict the position of LShin, and the second one is the same for LThigh. We are performing the 5-fold cross-validation method. Figure 4.4 shows the workflow followed in this experiment. In both experiments, the data to be predicted are the Pitch and Roll angles. The RMSE is expected to be the lowest of all the experiments that try to predict these IMUs, which is why they are used as baseline.

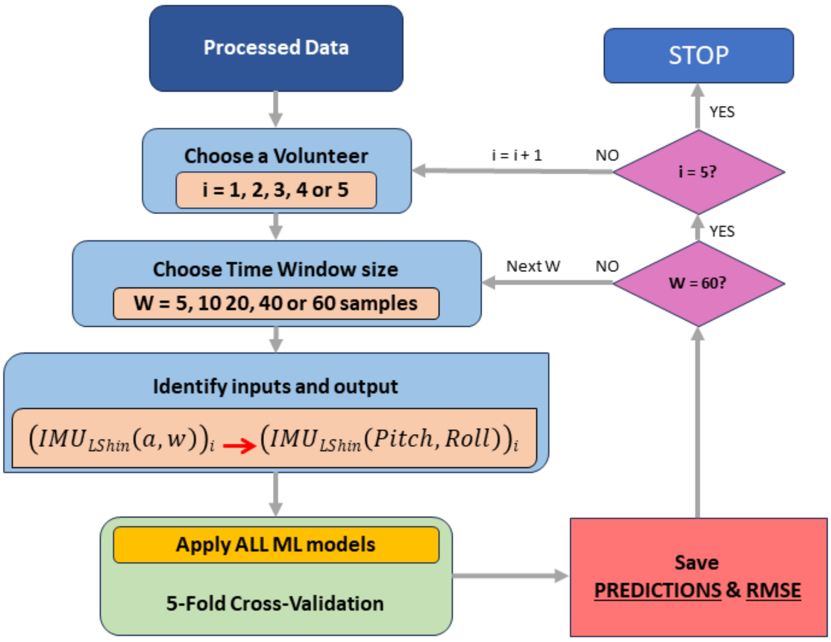


Figure 4.4: Workflow used in the first baseline experiment. The IMU LShin is predicted.

The key values are the number of samples and the models. The first step is to define the TW samples. In these experiments 5, 10, 20, 40 and 60 samples are used for MLR, SVR Linear, Gaussian and Polynomial and TB; and 5 and 10 samples for one-layer and two-layer MLP. The number of samples in the window is applied in an increasing way, to maintain a logical order. The data matrix with the TW size applied is the matrix with which to start each experiment.

4.5.1.2 Optimization of the number and position of IMUs

We perform three experiments to assess the reduction in the number of IMUs needed and their optimal placement.

The first experiment of this section is called 3 Devices for LShin (3DLS) Individual Evaluation (IE). It predicts the angles of the IMU LShin with the inertial measurements of the other three IMUs. LThigh angles are not calculated because it is used in prediction.

All the proposed ML models and all the TW sizes are also applied, as in the two baseline explained above. The workflow followed in 3DLS IE can be seen in Figure 4.5.

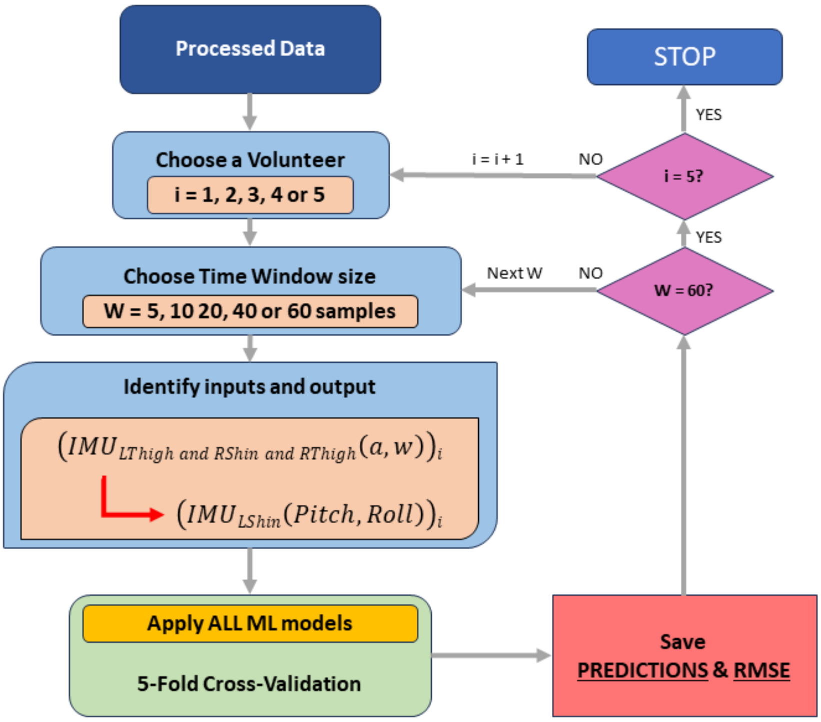


Figure 4.5: Workflow followed to perform the 3DLS IE experiment of evaluation of IMUs. One IMU is predicted from three IMUs data.

In the following experiments, the reduction in the number of sensors and their most appropriate location is evaluated. The experiment named 2 Devices for a IMU (2DLX) IE predicts the angles of the LShin and LThigh IMUs. To do so, we use the inertial measurements of the Right Shin (RShin) and Right Thigh (RThigh) IMUs together. The optimal TW sizes and models are used. So, an IMU is being predicted with data from two IMUs.

The last experiment of this section is named 1 Device for a IMU (1DLX) IE. 1DLX IE is about predicting the angles of the LShin and LThigh IMUs from the inertial measurements of the RShin and RThigh IMUs separately. In this way, the angles of one IMU are being predicted with the data of another IMU, located on the other leg.

4.5.2 Predictions of new volunteers.

In order to see how the already trained and selected models can be extrapolated to the gait of a new volunteer, a *leave-one-out* cross-validation is performed using the five volunteers. To check the extrapolability of the models for new volunteers, three experiments are carried out that are fundamentally the same as the previous three, where the number of

IMUs necessary and their location are analyzed for a good prediction. In this case, the processed data of the five selected volunteers 1, 2, 3, 4 and 5 are used. Five iterations are carried out in which the two proposed models are trained with data from four volunteers, leaving aside the remaining volunteer. In each iteration, a different volunteer is selected to be excluded for testing.

For the training of the algorithms in these three experiments, the number of samples from each volunteer is different since the range of samples from each volunteer that offers the highest quality is taken, see Figure 4.3. The mean of the five volunteers is 5200±963 samples. Samples from the four volunteers used to train the algorithm are randomized to avoid patterns from any specific volunteer. Finally, the performance of the models is evaluated using the data of the independent volunteer who was not used in the training process. The results obtained in each of the three experiments are shown as an average of the five iterations performed. Figure 4.6 shows the workflow of the first experiment of this section.

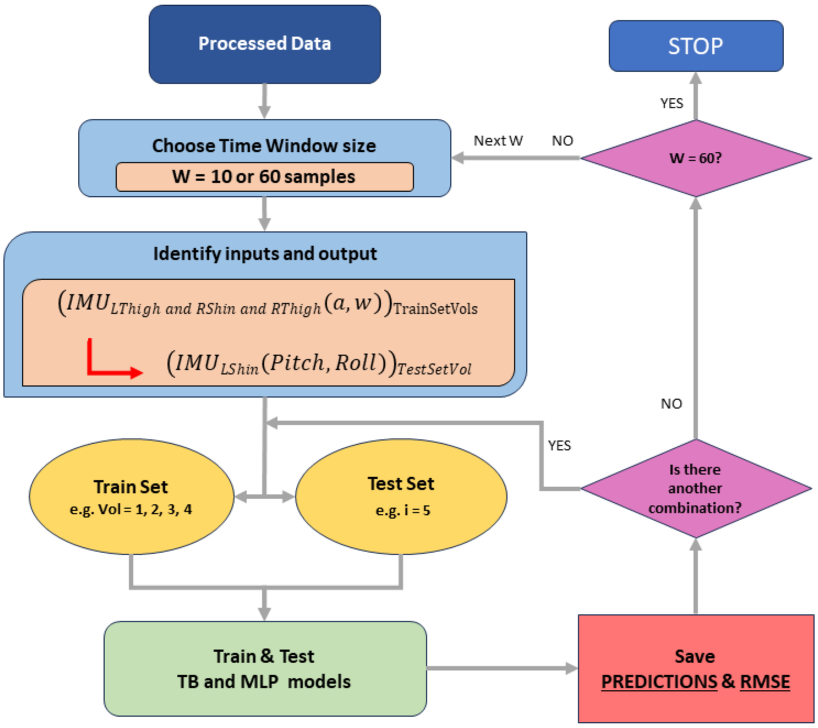


Figure 4.6: Workflow followed to perform the first extrapolation experiment. One volunteer IMU is predicted from three IMUs data of the rest of volunteers.

4.6 Evaluation metrics

The RMSE is a metric commonly used in the field of ML to evaluate the performance of a prediction model compared to actual values. It is especially popular in regression problems, where you want to predict a continuous numeric value. It is a measure of the average difference between the model predictions and the observed values, expressed in the same unit as the original data. In this project, the RMSE is expressed in radians. The objective is to minimize the RMSE, which means that a model is sought that generates predictions close to the true values. The RMSE formula is shown below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.3}$$

where N the number of observations in the data set, y_{i} is the actual or observed value for the i-th observation and $\hat{y}_{\mathrm{i}}$ is the value predicted by the model for the i-th observation.

Some interesting properties of using RMSE are that it penalizes large errors more than small errors due to the square in the formula; it is in the same units as the original data, which makes it easier to interpret and it is sensitive to outliers, since the errors are squared.

In addition, the results are represented in Boxplots. A boxplot provides a visual representation of how the data is distributed, making it easier to understand the spread, concentration of data, and the presence of outliers. Outliers are observations that deviate significantly from the majority of the data. They are represented with dots outside the box, which allows them to be easily identified. Additionally, the boxplot shows key statistics, such as the median, which is the center line of the box; and the range between the first and third quartiles that make up the box. They are commonly used to compare the distribution of data between different groups or categories. In this case, to compare the results of Roll and Pitch.

4.7 Tools

4.7.1 MATLAB

The MATLAB programming language is used in this work. MATLAB is a programming platform and development environment widely used in a variety of fields, including ML. In addition, it allows the visualization of graphs; as well as its manipulation [98]. The software is used for signal processing data manipulation and the training and test of ML models.

4.7.2 LaTeX

LaTeX is a document composition system widely used for creating scientific and technical documents. It is especially useful for mathematical equations, formulas, and structured documents because it automates the appearance and formatting of the document. This work has been written with LaTeX using its *online Overleaf* text editor [99].

Chapter 5

Results and Discussion

In this chapter, the results obtained in the experiments are presented and discussed. Section 5.1 shows the results of predicting the orientation angles of the LShin and LThigh IMUs using their own inertial measurements. These first two experiments serve as the baseline for the rest of the experiments, as well as to define the optimal algorithms and TW that are used in the following the experiments.

Next, Section 5.2 shows three experiments carried out with the data from the IMUs of the same volunteer. The aim of these experiments is the optimization of the number and position of IMUs.

Finally, the aim of experiments in Section 5.3 is to assess model extrapolation to volunteers. The theory is similar to that of the previous section, but to predict the angles of an IMU of a volunteer, the data of the rest of the volunteers is used.

5.1 Baseline experiments

This section is made up of two experiments in which the orientation angles of a single IMU,LShin and LThigh, are predicted. These experiments are called “baseline”, since the prediction error is expected to be the lowest when inertial data of an IMU is used to predict its angles. In the rest of the experiments, the angles of these IMUs are predicted using data from other combinations of IMUs. All the ML algorithms and all the TWs sizes proposed for each algorithm are evaluated. Therefore, the optimal algorithms and TWs sizes are selected to be used in the rest of experiments.

5.1.1 Angle prediction in IMU LShin

This first experiment focuses on the LShin IMU, which is used for training and test. Firstly, all the results obtained for Pitch angle are shown. Table 5.1 shows the mean and standard deviation of the RMSE obtained using MLR, SVR Linear, Gaussian, Polynomial, TB and MLP models; with all the proposed TWs sizes. The errors of the test set and the train set are shown. The MLP NN is only used with 5 and 10 sample TW. The table only includes the lowest RMSE that has been obtained, which is with two-layer MLP. Detailed results of the MLP hyperparameters optimization are shown in Table A.3 and Table A.4 of Appendix A.1.1.

Table 5.1: RMSE mean and deviation obtained in LShin baseline experiment to predict the Pitch angle.

		TW Size [samples]				
		5	**10**	**20**	**40**	**60**
TEST RMSE [rad]	**MLR**	0.152±0.027	0.121±0.025	0.079±0.018	0.052±0.008	0.041±0.006
	$\mathrm{SVR_L}$	0.153±0.023	0.123±0.026	0.080±0.015	0.053±0.008	0.047±0.005
	$\mathrm{SVR_G}$	0.239±0.043	0.312±0.042	0.329±0.036	0.346±0.040	0.347±0.037
	$\mathrm{SVR_P}$	0.662±0.301	0.328±0.172	0.238±0.068	0.747±0.323	1.731±0.801
	TB	0.056±0.010	0.042±0.007	0.036±0.007	0.032±0.009	**0.030±0.007**
	MLP	0.046±0.010	0.042±0.008	-	-	-
TRAIN RMSE [rad]	**MLR**	0.154±0.029	0.122±0.027	0.077±0.017	0.050±0.007	0.043±0.004
	$\mathrm{SVR_L}$	0.151±0.022	0.121±0.015	0.075±0.012	0.046±0.006	0.041±0.004
	$\mathrm{SVR_G}$	0.065±0.007	0.098±0.012	0.103±0.011	0.105±0.009	0.101±0.009
	$\mathrm{SVR_P}$	0.156±0.056	0.101±0.077	0.223±0.069	0.753±0.318	1.712±0.793
	TB	0.032±0.005	0.023±0.004	0.019±0.004	0.016±0.004	**0.015±0.004**
	MLP	0.040±0.008	0.037±0.009	-	-	-

The MLR, $\mathrm{SVR_L}$ and TB models follow the same trend of reducing the RMSE as the TW samples increase. These are the models with the lowest errors. Focusing on the method with the lowest error, TB, its errors decrease from 0.056±0.010 rad to 0.030±0.007 rad when increasing the window from 5 to 60 samples. In the case of MLP with 5 samples obtains 0.046±0.010 rad and with 10 samples, it is reduced to 0.042±0.008 rad for test set. The errors obtained with the train test with these models are similar to those obtained by the test set, so overfitting is not contemplated. These errors are acceptable, so they are the reference metrics in the following experiments. However, from the Table 5.1, it is observed that the $\mathrm{SVR_G}$ model obtains an RMSE of 0.239±0.043 rad with TW of 5 samples and increases to 0.347±0.037 rad with 60 samples. On the other hand, the results of the train set are in a range between 0.065±0.007 rad and 0.105±0.009 rad, which means that overfitting occurs in this model. The next model that stands out is the $\mathrm{SVR_P}$. This model does not follow a visible pattern in any of the sets. In the test set the error with TW of 5 samples is 0.662±0.301 rad, decreasing to 0.238±0.068 rad with 20 samples and increasing again obtaining 1.731±0.801 rad with 60 samples. The RMSE of the train set are in a range between 0.101±0.077 rad and 1.712±0.793 rad.

Table 5.2 shows the mean and absolute deviation of the RMSE for the Roll angle. As for the Pitch angle, all the proposed models and TW sizes are evaluated. Detailed results of the MLP hyperparameters' optimization are shown in Table A.7 and Table A.8 of Appendix A.1.1.

Table 5.2: RMSE mean and deviation obtained in LShin baseline experiment to predict the Roll angle.

		Window Size [samples]				
		5	**10**	**20**	**40**	**60**
TEST RMSE [rad]	**MLR**	0.152±0.015	0.135±0.016	0.080±0.012	0.054±0.014	0.039±0.014
	SVR$_{\mathbf{L}}$	0.165±0.005	0.138±0.009	0.096±0.010	0.070±0.012	0.048±0.011
	SVR$_{\mathbf{G}}$	0.157±0.057	0.168±0.062	0.134±0.039	0.127±0.044	0.144±0.054
	SVR$_{\mathbf{P}}$	1.034±0.711	1.140±0.968	1.246±1.705	1.105±0.802	4.102±7.038
	TB	0.068±0.009	0.056±0.007	0.042±0.009	0.036±0.008	**0.029±0.007**
	MLP	0.046±0.003	0.046±0.001	-	-	-
TRAIN RMSE [rad]	**MLR**	0.147±0.014	0.125±0.011	0.075±0.013	0.050±0.010	0.040±0.009
	SVR$_{\mathbf{L}}$	0.157±0.005	0.136±0.006	0.094±0.009	0.069±0.008	0.051±0.008
	SVR$_{\mathbf{G}}$	0.084±0.068	0.102±0.054	0.096±0.035	0.070±0.013	0.048±0.026
	SVR$_{\mathbf{P}}$	0.817±0.891	1.061±0.993	1.237±1.740	0.943±0.866	3.862±6.747
	TB	0.055±0.011	0.045±0.009	0.030±0.006	0.023±0.006	**0.013±0.004**
	MLP	0.044±0.003	0.046±0.002	-	-	-

As can be seen, the tendency of the models to predict Roll is similar to that of Pitch. The MLR, SVR$_\mathrm{L}$ and TB models follow the same trend of reducing the RMSE as the TW samples increase. In the case of MLR with 5 samples, 0.152±0.015 rad is obtained, which decreases to 0.039±0.014 rad with 60 samples. In the case of SVR$_\mathrm{L}$ it ranges from 0.165±0.005 rad to 0.048±0.011 rad. And in the case of TB from 0.068±0.009 rad to 0.029±0.007 rad, obtaining the lowest RMSE between the regressors, as in the case of the Pitch angle. Two-layer MLP obtains 0.046±0.003 rad and 0.046±0.001 rad with 5 and 10 samples in the test set; and 0.044±0.003 rad and 0.046±0.002 rad, respectively, in train set. The model SVR$_\mathrm{P}$ obtains the highest errors, the lowest being 1.105±0.802 rad in the test set and 0.817±0.891 rad in the train set. Regarding SVR$_\mathrm{G}$, a smaller difference is observed between the errors obtained by the two sets than in the Pitch angles. In the test set, 0.157±0.057 rad are obtained with 5 TW samples, which increases in 10 samples to 0.168±0.062 rad. Afterwards it decreases to 0.127±0.044 rad with 40 samples, to finally increase with 60 samples obtaining 0.144±0.054 rad. On the other hand, in the train set the RMSE varies between 0.048±0.026 rad and 0.102±0.054 rad. This difference between the two sets reflects that overfitting is occurring, to a lesser extent than with the Pitch angle. The lower amplitude that the Roll angle has in gait motion also explains lower absolute errors compared to Pitch, since the angles to predict are in a smaller range.

According to the results in Table 5.1 and Table 5.2, the most appropriate model to estimate orientation angles of a segment from inertial measurements of this segment is TB. Specifically, with the TW of 60 samples since it is the model that provides the lowest RMSE. TB gets 0.030±0.007 rad for Pitch and 0.029±0.007 rad for Roll, what is 1.71°and

1.66°, respectively. The accuracy of the results indicates that the model correctly infers the input-output relationship of the signals and angles. However, two-layer MLP and 10 TW samples is also notable, since with lower computational cost, results close to TB are obtained. For the explained MLP gets 0.042±0.008 rad for Pitch and 0.046±0.001 rad for Roll. Accuracy is lost compared to TB, but the computational cost is lower.

To better understand the error in the predictions with TB, Figure 5.1 displays the final 20% of the data of Volunteer 2. The original and predicted Pitch and Roll angle of the IMU LShin is shown, using the TB model with TW of 60 samples. The predicted Pitch angle is represented in red and the original Pitch angle is overlapped in green. The predicted Roll is in dark blue and the original Roll is overlapped in light blue. It is observed that the predictions follow the trend of the originals, except for isolated cases. Also, as discussed before, Figure 5.1 shows that the amplitude of the Pitch angle, which is the motion in the sagittal plane, is much greater than that of the Roll, which is the motion in the frontal plane.

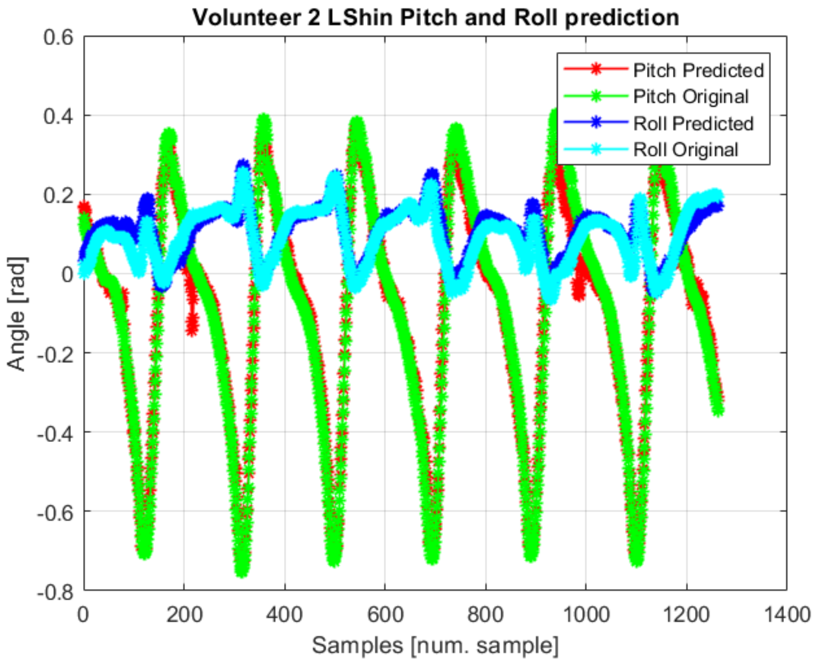
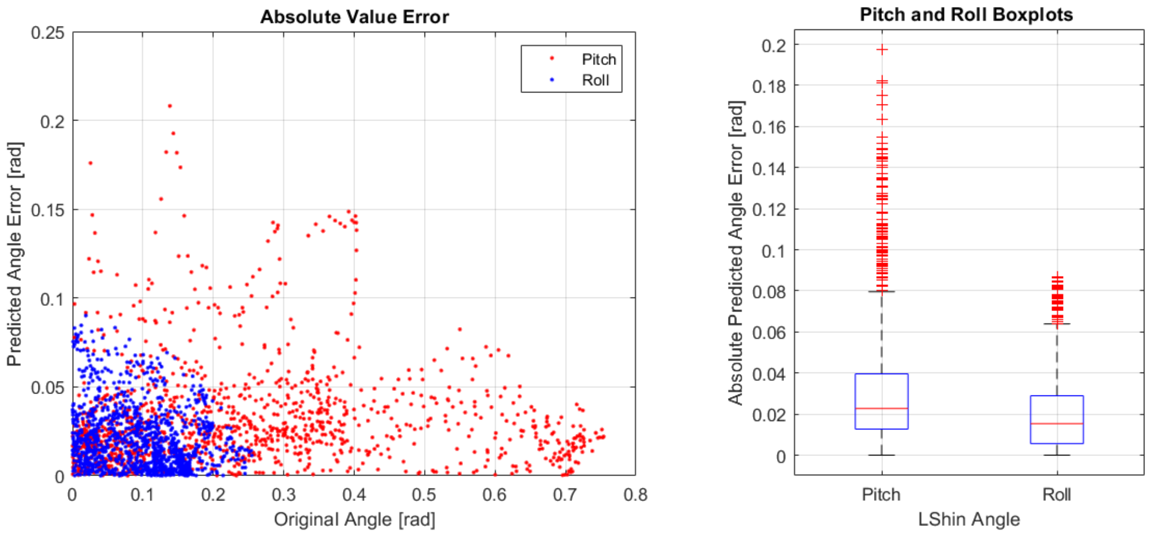


Figure 5.1: Volunteer 2 Pitch and Roll angles of LShin IMU Baseline Experiment using TB and W=60.

Figure 5.2 includes two more graphs to complement the interpretation of the error obtained in Figure 5.1. Figure 5.2a shows the absolute error of the predicted angle depending on the original angle, to analyze possible trends between them. The red color is the Pitch angle and the blue color is the Roll. It can be seen that most points are below 0.05-0.075 rad for both angles. Up to 0.4 rad from the Pitch angle, it has several errors that are out of the average but they are a minority. This shows that the error is not directly related to the angle to predict. However, the range of values of the angle to predict affects the errors by making estimation of higher amplitude angles in low amplitude samples.

Finally, Figure 5.2b present two boxplots corresponding to the absolute error angles. In both cases, there are outliers, which are punctual errors over three times the standard deviation of the absolute errors. In the case of Pitch, the median is 0.024 rad and 50% of the data is between 0.015 and 0.04 rad. In the case of the Roll, the median is 0.018 rad and 50% of the data between 0.01 and 0.03 rad. Most errors in Pitch are below 0.08 rad and those in Roll are below 0.064 rad. They are small errors and are within an acceptable range for motion monitoring.



(a) Comparison of LShin angle error.

(b) Boxplot of LShin angles.

Figure 5.2: Comparison of the absolute error of the predicted angles with respect to the original angles and absolute value error boxplots corresponding to LShin Baseline experiment.

5.1.2 Angle prediction in IMU LThigh.

The second baseline experiment predicts the orientation angles of LThigh. As in the previous experiment, we use the measurements of one IMU to estimate its orientation, being LThigh the chosen IMU. The results provided are of predicting the Pitch and Roll angles.

Table 5.3 shows the absolute mean and deviation of the RMSE obtained using MLR, SVR Linear, Gaussian, Polynomial, TB and MLP models. Besides, all the TWs sizes are used to predict the Pitch angle. The errors of the test set and train set are shown. Detailed results of the MLP hyperparameters' optimization are shown in Table A.11 and Table A.12 of Appendix A.1.2.

MLR, SVR$_{\text{L}}$ and TB, reduce their error as the number of TW samples increases. MLR with 5 samples obtains 0.082±0.005 rad and with 60 samples obtains 0.035±0.017

Table 5.3: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Pitch angle.

		Window Size [samples]				
		5	**10**	**20**	**40**	**60**
TEST RMSE [rad]	**MLR**	0.082±0.005	0.070±0.007	0.054±0.013	0.044±0.016	0.035±0.017
	SVR$_\mathrm{L}$	0.071±0.025	0.062±0.025	0.053±0.017	0.047±0.017	0.041±0.018
	SVR$_\mathrm{G}$	0.261±0.324	0.146±0.036	0.161±0.027	0.192±0.028	0.212±0.027
	SVR$_\mathrm{P}$	0.650±0.847	0.190±0.091	1.261±0.679	0.518±0.205	2.517±1.880
	TB	0.046±0.012	0.043±0.012	0.039±0.010	0.036±0.009	**0.033±0.011**
	MLP	0.045±0.011	0.038±0.002	-	-	-
TRAIN RMSE [rad]	**MLR**	0.082±0.008	0.069±0.009	0.055±0.015	0.047±0.016	0.043±0.019
	SVR$_\mathrm{L}$	0.066±0.025	0.059±0.027	0.051±0.018	0.046±0.017	0.040±0.018
	SVR$_\mathrm{G}$	0.039±0.026	0.051±0.029	0.035±0.008	0.037±0.007	0.035±0.007
	SVR$_\mathrm{P}$	0.638±0.844	0.063±0.060	1.244±0.676	0.515±0.204	2.511±1.860
	TB	0.024±0.008	0.021±0.008	0.018±0.006	0.015±0.006	**0.014±0.006**
	MLP	0.052±0.009	0.038±0.011	-	-	-

rad. SVR$_\mathrm{L}$ with 5 samples obtains 0.071±0.025 rad and with 60 samples 0.041±0.018 rad. Finally, TB obtains 0.046±0.012 rad with 5 samples and 0.033±0.011 rad with 60 samples, being the lowest RMSE in this table. In the case of MLP with 5 samples obtains 0.045±0.011 rad and with 10 samples, it is reduced to 0.038±0.002 rad for test set. In train set, the RMSE is 0.052±0.009 rad and 0.038±0.011 rad, respectively. The trend of the errors is similar to that of the previous experiment. The SVR$_\mathrm{P}$ obtains high errors in both the test set and the train set. The error fluctuates in a range of 0.190±0.091 rad and 2.517±1.880 rad. The SVR$_\mathrm{G}$ model obtains an error of 0.261±0.324 rad with 5 TW samples, which decreases to 0.146±0.036 rad with 10 samples and increases with the rest of the samples until obtaining 0.212±0.027 rad with 60 samples. On the other hand, the train set of this model is substantially smaller, with errors between 0.035±0.007 rad and 0.051±0.029 rad, which indicates that overfitting occurs.

Regarding the prediction of the Roll angle, Table 5.4 follows the same format as for the Pitch angle. Table 5.4 shows the absolute mean and deviation with the different regressors. The trend repeats itself. With 60 samples, MLR obtains 0.040±0.012 rad, SVR$_\mathrm{L}$ 0.049±0.021 rad and TB 0.038±0.011 rad. The latter being the one with the least error. In this three models, the RMSE decreases as the TW samples increase. MLP again gets the lowest RMSE at its TW sizes. It obtains 0.048±0.002 rad with 5 samples and 0.046±0.001 rad with 10 samples in test set. In train, it obviously obtains a lower RMSE, with 0.036±0.009 rad and 0.037±0.007 rad, respectively. SVR$_\mathrm{P}$ offers errors higher than 0.272±0.219 rad, which increases by lengthening the TW. SVR$_\mathrm{G}$ presents overfitting since the lowest error in test is 0.165±0.016 rad and in train, errors are around the average.

According to these results, it can be concluded that the most appropriate model for this application is TB. This model is also best suited for LShin, with the 60 sample con-

Table 5.4: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Roll angle.

		Window Size [samples]				
		5	**10**	**20**	**40**	**60**
TEST RMSE [rad]	**MLR**	0.057±0.010	0.051±0.010	0.045±0.011	0.041±0.012	0.040±0.012
	SVR$_L$	0.064±0.023	0.060±0.019	0.056±0.020	0.053±0.021	0.049±0.021
	SVR$_G$	0.202±0.014	0.165±0.016	0.198±0.017	0.213±0.018	0.237±0.019
	SVR$_P$	0.754±0.596	0.446±0.394	0.862±0.700	1.032±1.694	0.272±0.219
	TB	0.052±0.011	0.044±0.010	0.046±0.012	0.049±0.012	**0.038±0.011**
	MLP	0.048±0.002	0.046±0.001	-	-	-
TRAIN RMSE [rad]	**MLR**	0.056±0.012	0.052±0.013	0.046±0.015	0.043±0.017	0.040±0.019
	SVR$_L$	0.059±0.022	0.055±0.022	0.049±0.018	0.048±0.018	0.046±0.017
	SVR$_G$	0.016±0.005	0.018±0.005	0.020±0.006	0.022±0.008	0.022±0.008
	SVR$_P$	0.565±0.700	0.292±0.413	0.831±0.678	0.940±1.716	0.288±0.222
	TB	0.023±0.009	0.019±0.008	0.018±0.007	0.016±0.007	**0.013±0.006**
	MLP	0.036±0.009	0.037±0.007	-	-	-

figuration. TB obtains 0.033±0.011 rad for Pitch and 0.038±0.011 rad for Roll, which is 1.71°and 2.17°, respectively. This low errors allow the use of this model for gait monitoring. However, two-layer MLP and 10 TW samples offer close results with substantially lower computational cost. With the proposed MLP, 0.038±0.002 rad for Pitch and 0.046±0.001 rad for Roll are obtained. In LShin Baseline experiment, the RMSE is lower than with LThigh, obtaining 0.030±0.007 rad for Pitch and 0.029±0.007 rad for Roll.

To better understand the magnitude of the errors in the prediction of LThigh, Figure 5.3 shows the final 20% of the data from Volunteer 2. The original and predicted Pitch and Roll angle of the IMU LThigh is shown, using TB model with TW of 60 samples. The predicted Pitch is represented in red and the original overlay in green. The predicted Roll is depicted in dark blue and the original in light blue. It is observed that the amplitude of the angles in the thigh are smaller than in the shin. The amplitude of Roll is smaller than that of Pitch. Errors occur when predicting the peaks, since these, at the original Roll angle, are irregular.

Figure 5.4 shows the absolute errors of the predicted angle according to the original angle. It is observed that the Roll angle, even having a smaller amplitude, has greater errors as the original angle increases. In the case of Pitch, the errors are maintained, except in specific cases in which the error exceeds 0.06 rad. This is proof that the error is independent of the amplitude.

The majority of Pitch errors are below 0.07 rad and Roll errors are below 0.079 rad. Errors are acceptable for gait monitoring.

Regressors show that they can predict different movements. The movement of LThigh and LShin are different and find the input-output relationship in both cases. Thus, it is

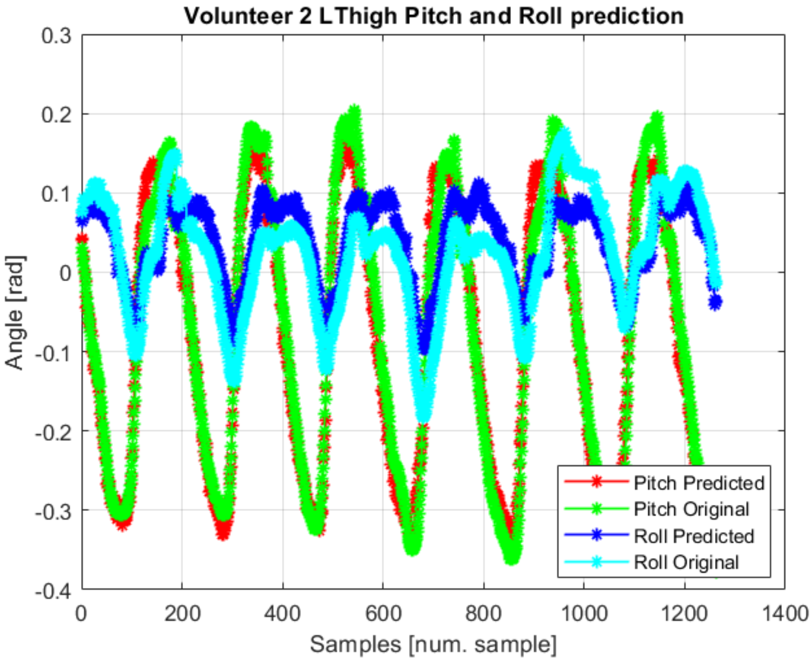
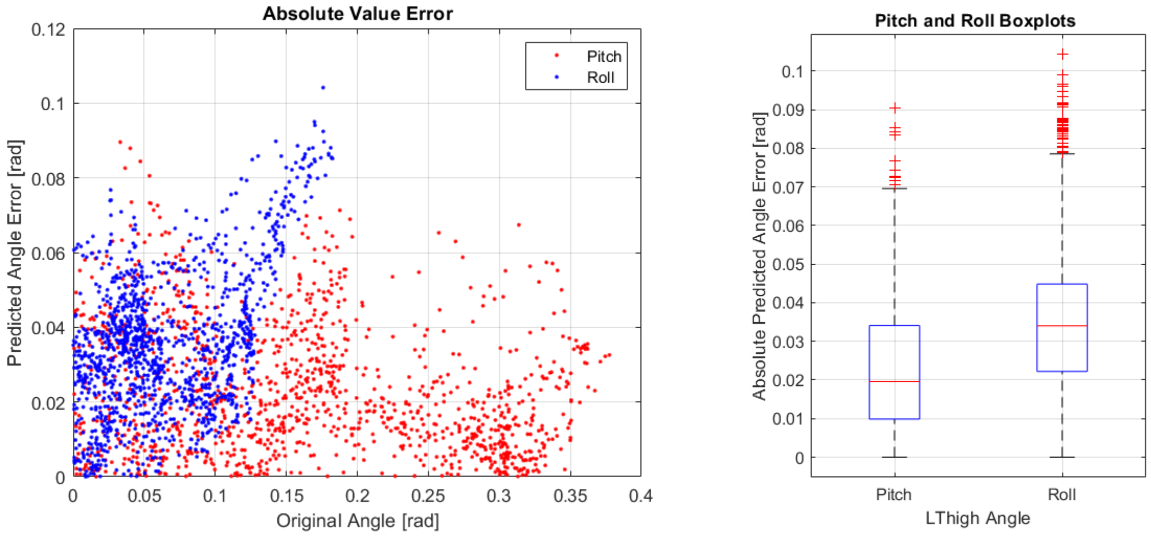


Figure 5.3: Volunteer 2 Pitch and Roll angles of LThigh IMU Baseline Experiment using TB with W=60.



(a) Comparison of LThigh angle error.

(b) Boxplot of LThigh angles.

Figure 5.4: Comparison of the absolute error of the predicted angles with respect to the original angles and absolute value error boxplots corresponding to LThigh Baseline experiment.

shown that both movements can be predicted exactly with inertial measurements of that segment. In subsequent experiments, we analyze whether this input-output relationship can also be inferred with movements of other segments.

5.2 Optimization of the number and position of IMUs

In this section, three experiments are performed using different numbers of IMUs for the prediction of the orientation angles of a single IMU. We use the inertial measurements from one IMU to three IMUs, to predict the angles of the fourth one. The first experiment in this section is the following one to select the prediction models and the optimal TW size that is used in the following experiments. It should be noted that the IMUs used to predict and the IMU angles being predicted all belong to the same person.

5.2.1 3DLS IE experiment

In 3DLS IE experiment, the data from the RShin, RThigh and LThigh IMUs is used to predict the angles formed by the LShin IMU. In this experiment, all the models and all the TWs sizes proposed are used. The LThigh IMU is not predicted because it would be a scenario similar to the baseline experiments.

To begin with, Table 5.5 shows Pitch and Roll prediction results in test. Detailed results of the MLP hyperparameters' optimization are shown in Table A.19, Table A.20 and Table A.24 in Appendix A.2.1. SVR_P and SVR_G provide the highest errors, reaching 11.42±25.39 rad with 60 TW samples and 0.351±0.040 rad, respectively. Furthermore, errors are not acceptable if you want to predict gait, since high errors cause problems that other parts of the body have to compensate. The remaining three models obtain lower RMSE as the number of TW samples increases. TB being the one that obtains the smallest error with 0.036±0.008 rad for Pitch and 0.032±0.017 rad for Roll. This is equivalent to 2.06°and 1.83°, respectively. MLP continues to be the model that best predicts its corresponding TW sizes. With 10 samples, Pitch prediction obtains 0.040±0.010 rad and Roll 0.033±0.006 rad.

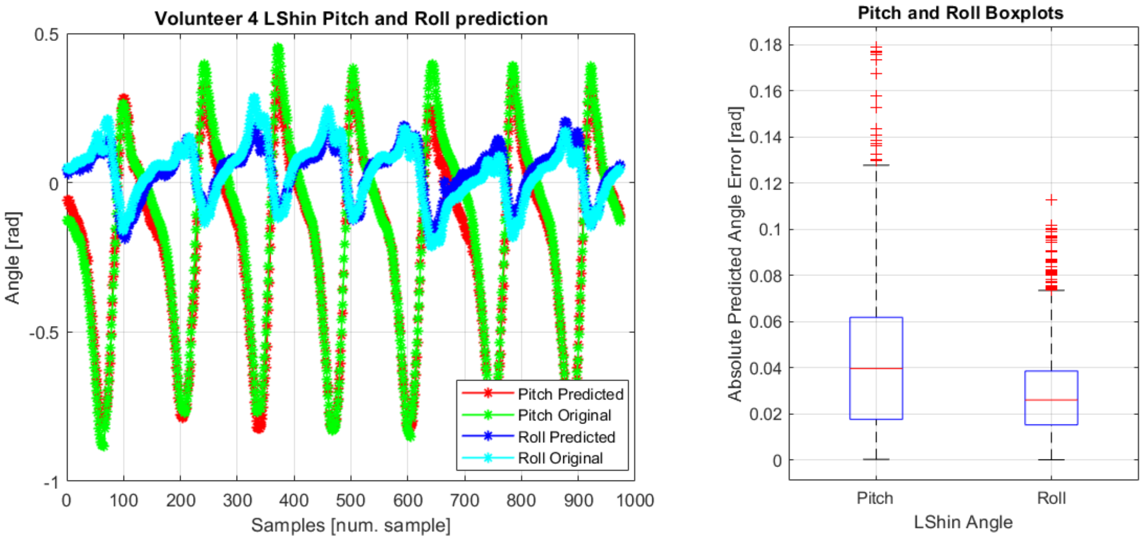
According to these results, the most suitable model for this application is TB with TW of 60 samples to predict Pitch and Roll. However, the performance of two-layer MLP and 10-sample TW obtain close results. Comparing the results of this experiment with those obtained in the LShin Baseline experiment, the RMSE has increased 0.006±0.001 rad to predict Pitch and 0.003±0.010 rad for Roll. Although the error is slightly increased, the models can assumed it to monitor the gait.

To better understand the error, Figure 5.5 shows the final 20% of the data from the IMU LShin of Volunteer 4. It shows the original Pitch and Roll angles and predicted using

Table 5.5: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Pitch and Roll angles.

		Window Size [samples]				
		5	**10**	**20**	**40**	**60**
PITCH RMSE [rad]	**MLR**	0.094±0.016	0.074±0.012	0.062±0.009	0.058±0.010	0.059±0.011
	SVR$_L$	0.096±0.017	0.076±0.013	0.066±0.009	0.064±0.010	0.068±0.010
	SVR$_G$	0.325±0.034	0.342±0.029	0.349±0.039	0.351±0.040	0.350±0.031
	SVR$_P$	0.194±0.153	0.357±0.273	2.560±3.515	1.722±3.257	11.42±25.39
	TB	0.065±0.007	0.059±0.007	0.054±0.006	0.047±0.007	**0.036±0.008**
	MLP	0.048±0.009	0.040±0.010	-	-	-
ROLL RMSE [rad]	**MLR**	0.057±0.016	0.054±0.016	0.051±0.014	0.050±0.015	0.048±0.014
	SVR$_L$	0.064±0.018	0.061±0.020	0.060±0.018	0.060±0.014	0.058±0.014
	SVR$_G$	0.109±0.035	0.107±0.028	0.117±0.038	0.104±0.060	0.122±0.042
	SVR$_P$	0.871±0.936	0.654±0.600	0.979±0.863	1.141±1.293	1.080±2.330
	TB	0.038±0.016	0.037±0.017	0.034±0.016	0.033±0.017	**0.032±0.017**
	MLP	0.047±0.004	0.033±0.006	-	-	-

the TB model with TW of 60 samples. The error image translates into more accurate predictions. The boxplots show that the majority of Pitch errors are below 0.125 rad and Roll errors are below 0.075 rad. The predictions report data within these range of values without outliers. In Pitch, the range of values has increased by 0.04 rad and that of Roll by 0.01 rad.



(a) Original and predicted angles.

(b) Boxplot of LShin angles.

Figure 5.5: Volunteer 4 Pitch and Roll angles of 3DLS IE Experiment using TB with W=60. Absolute value error boxplots.

By carrying out this experiment and LShin and LThigh baseline experiments, it is concluded that the most appropriate model is TB with TW of 60 samples. However, two-layer MLP and 10 TW samples performs correctly with a much lower computational cost. Both methods provide acceptable errors for gait prediction. Also, they are so small that the following experiments are carried out only with those two ML models and TW sizes.

5.2.2 2DLX IE experiment

In 2DLX IE experiment, the angles of a single IMU, LShin or LThigh, are predicted by using the IMUs of the right leg. That is, one IMU orientation is predicted using the inertial measurements of two. As explained above, only the TB models with 60 TW samples and two-layer MLP with 10 samples are used. Detailed results of the MLP hyperparameters' optimization are shown in Table A.25 and Table A.26 in Appendix A.2.2.

Table 5.6 shows the results obtained with TB and MLP for the Pitch and Roll angles, both for the LShin IMU and the LThigh IMU. In LShin, the RMSE of Pitch is 0.039±0.010 rad using TB and 0.068±0.018 rad using MLP. For Roll, it is 0.032±0.006 rad and 0.056±0.003 rad, respectively.

Table 5.6: RMSE mean and deviation obtained in 2DLX IE experiment to predict the Pitch and Roll angles.

	TW size & Model	**Pitch RMSE [rad]**	**Roll RMSE [rad]**
LShin	$\mathbf{TB_{60}}$	**0.039±0.010**	**0.032±0.006**
	$\mathbf{MLP_{10}}$	0.068±0.018	0.056±0.003
LThigh	$\mathbf{TB_{60}}$	**0.034±0.009**	**0.031±0.015**
	$\mathbf{MLP_{10}}$	0.049±0.011	0.043±0.002

On the other hand, for predicting LThigh Pitch angle, 0.034±0.009 rad is obtained with TB and 0.049±0.011 rad with MLP. For Roll, TB obtains 0.031±0.015 rad and MLP 0.043±0.002 rad.

In the case of LShin, the results can be compared with those of the LShin Baseline and 3DLS IE experiments. In the Baseline one, the error increases 0.009±0.003 rad for Pitch and 0.003±0.001 rad for Roll. Regarding the results of 3DLS IE experiment, the RMSE increases 0.003±0.002 rad for Pitch and 0.000±0.011 rad for Roll. Therefore, in 2DLX IE for LShin the error of Roll's prediction is similar to that obtained with 3 IMUs. When predicting LThigh a lower RMSE for Roll is obtained than using only LThigh for predicting in LThigh Baseline experiment. In that experiment, we get 0.033±0.011 rad for Pitch and 0.038±0.011.

To visualize these errors, Figure 5.6c shows the final 20% of the LShin (top) and LThigh (bottom) IMUs with the original and predicted angles, obtained using TB. In the thigh, both angles have a smaller amplitude, which causes the error in this part to be less than in the shin. To better analyze these results, the following graphs offer relevant information. This is best seen in Figure 5.6b where the median of the Pitch angle is 0.045 rad and that the outliers exceed the error of 0.125 rad. On the other hand, the median error in Roll is 0.02 rad and concentrates 50% of the data in a range of ±0.012 rad. Finally, in Figure 5.6c it is observed that 50% of the Roll data produce a greater error than 50% of the Pitch data. In all cases, the errors obtained with the models are within the first and third quartiles of their corresponding angles.

The results obtained reflect that the TB model correctly infers the input-output relationship between the angles and the signal. Comparing the errors produced with those obtained in the 3DLS IE experiment, the conclusion is reached that it is optimal to use 2 IMUs for gait monitoring.

5.2.3 1DLX IE experiment

In this last experiment of individual evaluation, named 1DLX IE, the prediction of the LShin and LThigh IMUs is tested with inertial measurements of a single IMU, either RShin or RThigh. First, all the results obtained with the IMU RShin are presented, and then with RThigh. All detailed MLP tables with the results can be seen in Appendix A.2.3.

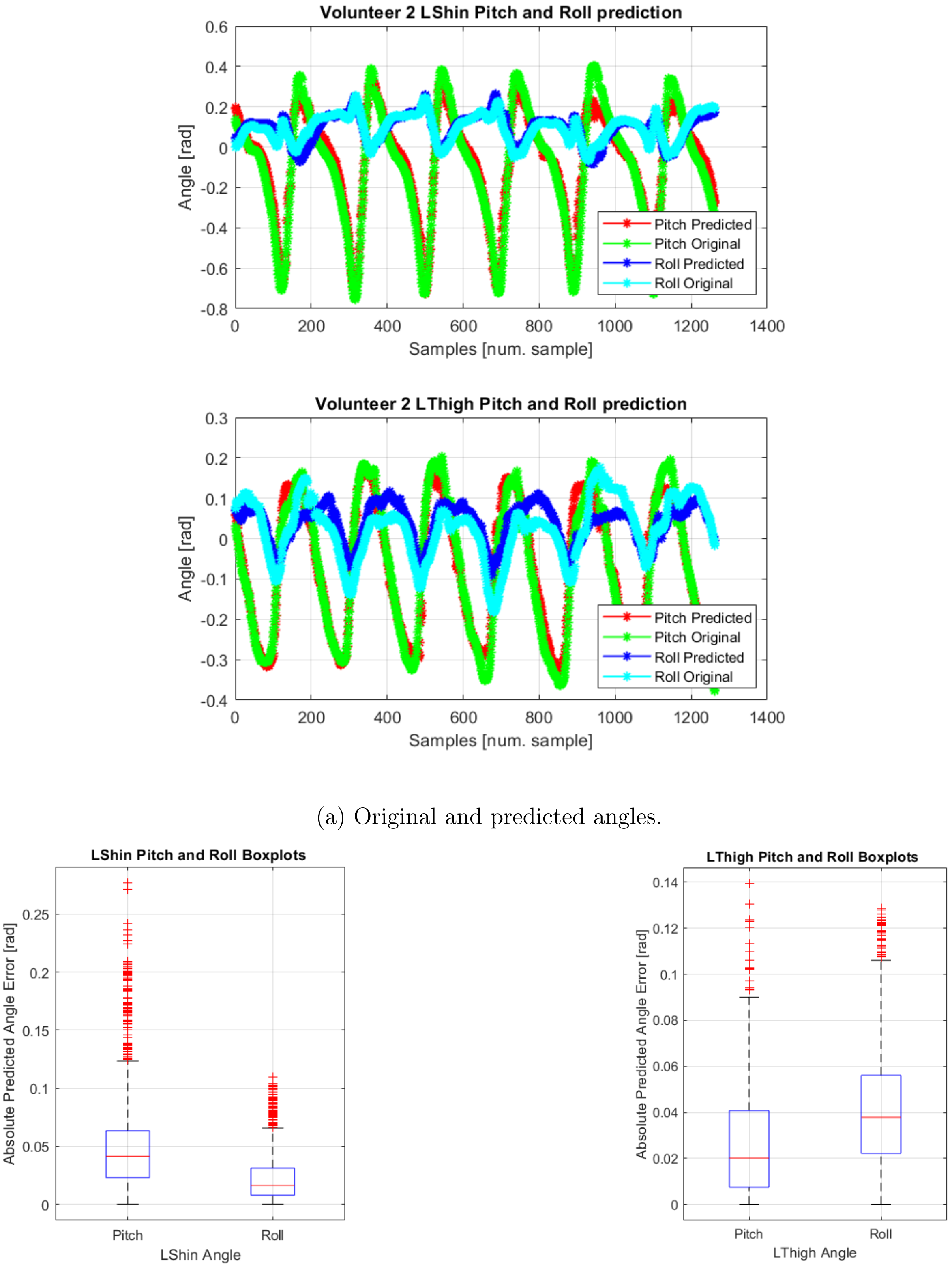
Table 5.7 shows the results obtained by using RShin inertial data. In the prediction of LShin Pitch angle, TB obtains 0.051±0.001 rad and MLP, 0.086±0.005 rad. Predicting Roll angle, we get 0.045±0.010 rad and 0.073±0.004 rad, respectively. When predicting LThigh Pitch angle with TB we get 0.042±0.003 rad and with MLP, 0.056±0.003 rad. For Roll, we get 0.036±0.006 rad and 0.049±0.002 rad, respectively.

Table 5.7: RMSE mean and deviation obtained in 1DLX IE experiment to predict the Pitch and Roll angles knowing RShin data.

	TW size & Model	**Pitch RMSE [rad]**	**Roll RMSE [rad]**
LShin	**TB_{60}**	**0.051±0.001**	**0.045±0.010**
	MLP_{10}	0.086±0.005	0.073±0.004
LThigh	**TB_{60}**	**0.042±0.003**	**0.036±0.006**
	MLP_{10}	0.056±0.003	0.049±0.002

In Figure 5.7a we can see the final 20% of the respective IMUs of Volunteer 3 obtained with TB. The predicted angles adjust to the originals, except in the LThigh Roll due to the irregularity of its peaks.

In the LShin IMU, specific errors occur in both angles, since the majority remains below 0.1 radians of error. Figure 5.7b shows that outliers in Pitch occur when the error



(a) Original and predicted angles.

(b) Boxplot of LShin angles.

(c) Boxplot of LThigh angles.

Figure 5.6: Volunteer 2 Pitch and Roll angles prediction of LShin and LThigh IMUs using right leg inertial measurements in 2DLX IE experiment using TB with W=60. Absolute value error boxplots.

exceeds 0.08 rad and in Roll when it exceeds 0.05 rad. In LThigh, most Pitch prediction errors are below 0.1 rad, except when the original angle is 0.3 rad when isolated errors are made. On the other hand, the Roll error remains below 0.05 rad without any outlier as Figure 5.7c shows.

In the case of using RThigh inertial data for prediction, Table 5.8 shows the results. For predicting LShin Pitch angle we get 0.052±0.002 rad with TB and 0.079±0.008 rad with MLP. For Roll, we get 0.034±0.007 rad and 0.039±0.009 rad, respectively. For predicting LThigh Pitch angle with TB we get 0.036±0.005 rad and with MLP, 0.044±0.009 rad. For Roll, we get 0.032±0.002 rad and 0.026±0.003 rad, respectively.

Table 5.8: RMSE mean and deviation obtained in 1DLX IE experiment to predict the Pitch and Roll angles knowing RThigh data.

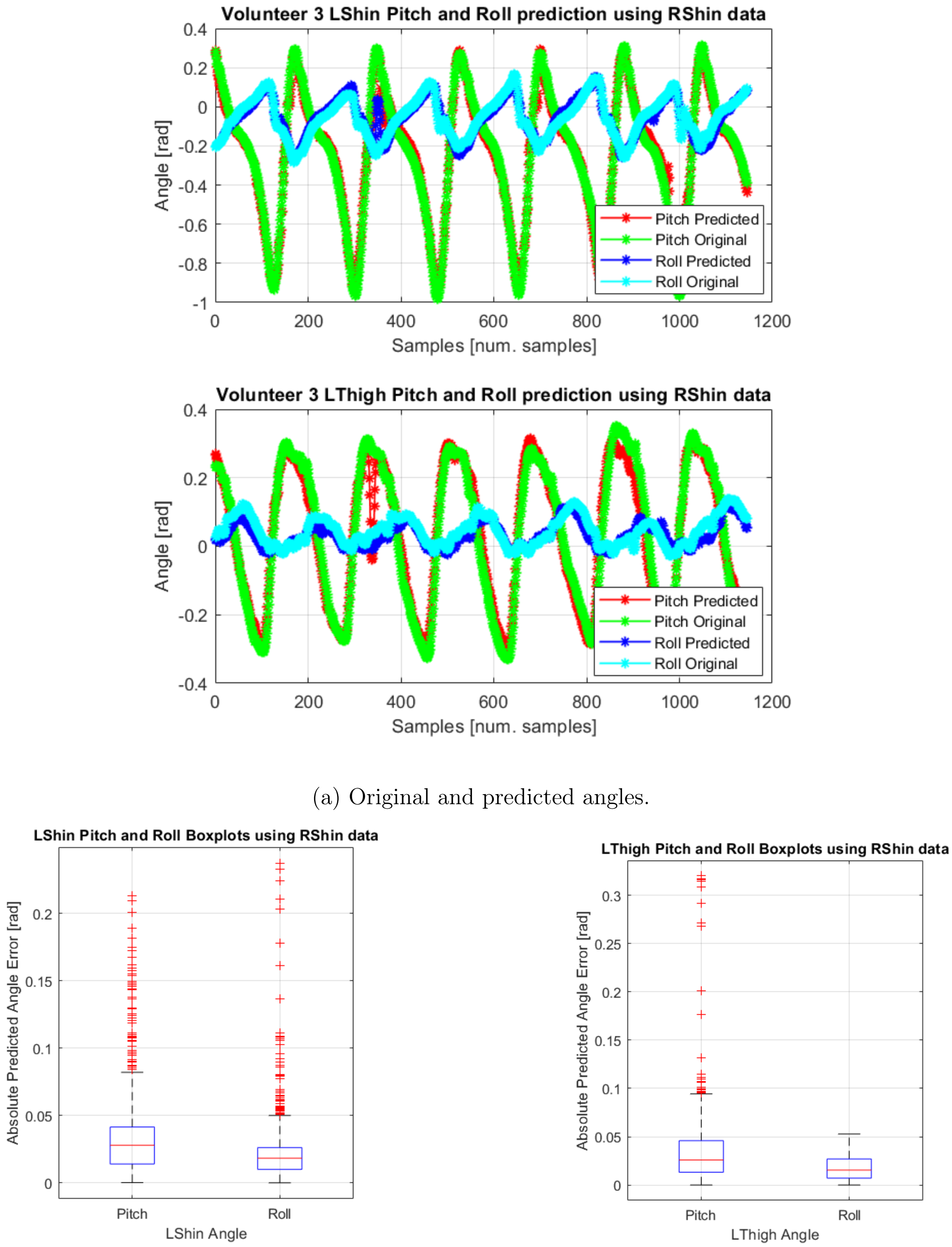
	TW size & Model	**Pitch RMSE [rad]**	**Roll RMSE [rad]**
LShin	**TB$_{60}$**	**0.052±0.002**	**0.034±0.007**
	MLP$_{10}$	0.079±0.008	0.039±0.009
LThigh	**TB$_{60}$**	**0.036±0.005**	**0.032±0.002**
	MLP$_{10}$	0.044±0.009	0.026±0.003

Figure 5.8 shows 20% of the IMUs of Volunteer 3 using TB and the error boxplots obtained. As in the previous case, the original movement is followed with the predictions. In Figure 5.8b and Figure 5.8c it is observed that errors are higher than in 2DLX IE experiment.

The results obtained in this experiment have a higher RMSE than in the rest of the experiments, being lower when using RThigh than RShin. This is because the model better infers the input-output relationships of the thigh. Using RShin to predict all errors increase between 0.005 and 0.011 rad. However, using RThigh, all errors are minor compared to RShin, except the Pitch of LShin. All this is compared with 2DLX IE results. The errors are reasonable, but there is great variability depending on the IMU used to predict, so the results of the 2DLX IE experiment are chosen as optimal; as well as the use of two IMUs placed in the shin and thigh of the same leg.

5.3 Performance of new volunteers

In this last section, three experiments are carried out that are homologous to the Section 5.2 experiments. However, in this section we evaluate how extrapolable the prediction models are to new volunteers motions. This is achieved by predicting the orientation IMU

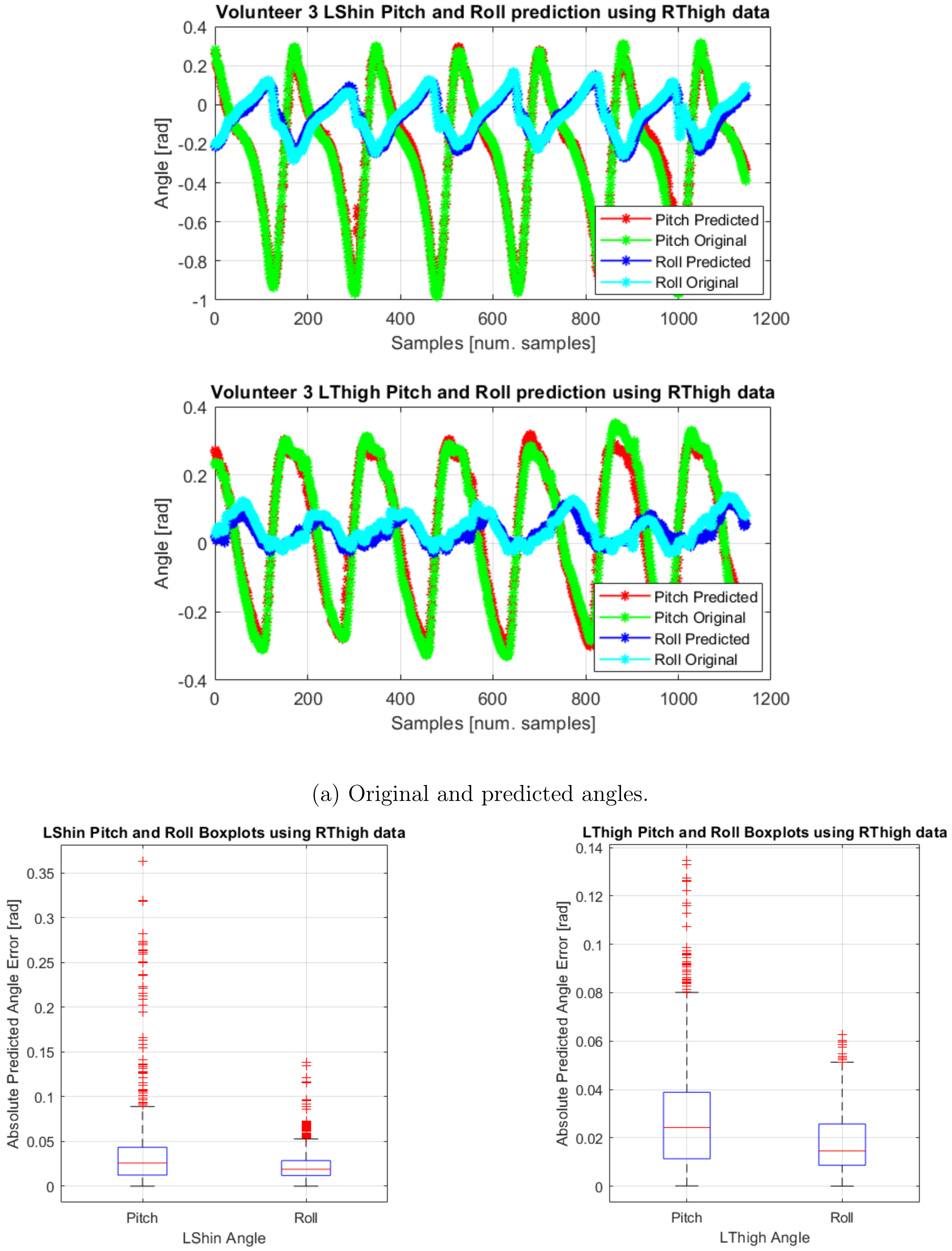


(a) Original and predicted angles.

(b) Boxplot of LShin angles.

(c) Boxplot of LThigh angles.

Figure 5.7: Volunteer 3 Pitch and Roll angles prediction of LShin and LThigh IMUs using RShin inertial measurements in 1DLX IE. Model TB with W=60. Absolute value error boxplots.



(a) Original and predicted angles.

(b) Boxplot of LShin angles.

(c) Boxplot of LThigh angles.

Figure 5.8: Volunteer 3 Pitch and Roll angles prediction of LShin and LThigh IMUs using RThigh inertial measurements in 1DLX IE experiment. Model TB with TW=60. Absolute value error boxplots.

or IMUs of a volunteer using models trained with the IMU data of the rest of the volunteers. This evaluation method is called leave-one-out. For these experiments, the TB models with 60 TW samples and two-layer MLP with 10 TW samples are used.

5.3.1 3DLS NV experiment

In 3DLS New Volunteer (NV) experiment, the Pitch and Roll angles of an IMU of a volunteer is predicted from the inertial measurements of the rest of the IMUs of the rest of the volunteers. In this case, the LShin IMU is predicted using LThigh, RShin and RThigh.

Table 5.9 shows the RMSE obtained by the test set for predicting Pitch and Roll angles of LShin IMU. The error of predicting Pitch is 0.141±0.059 rad with TB; and 0.123±0.011 rad with MLP which corresponds to 7.04°. For Roll, TB obtains 0.140±0.030 rad which corresponds to 8.02°; and 0.173±0.018 rad with MLP. If we compare TB results with its equivalent, 3DLS IE experiment, in which 0.036±0.008 rad for Pitch and 0.032±0.017 rad for Roll are obtained, the error has increased considerably. It makes sense that the error increases with this method, since the model does not have information about the IMU to predict. Detailed MLP results can be seen in Appendix A.3.1

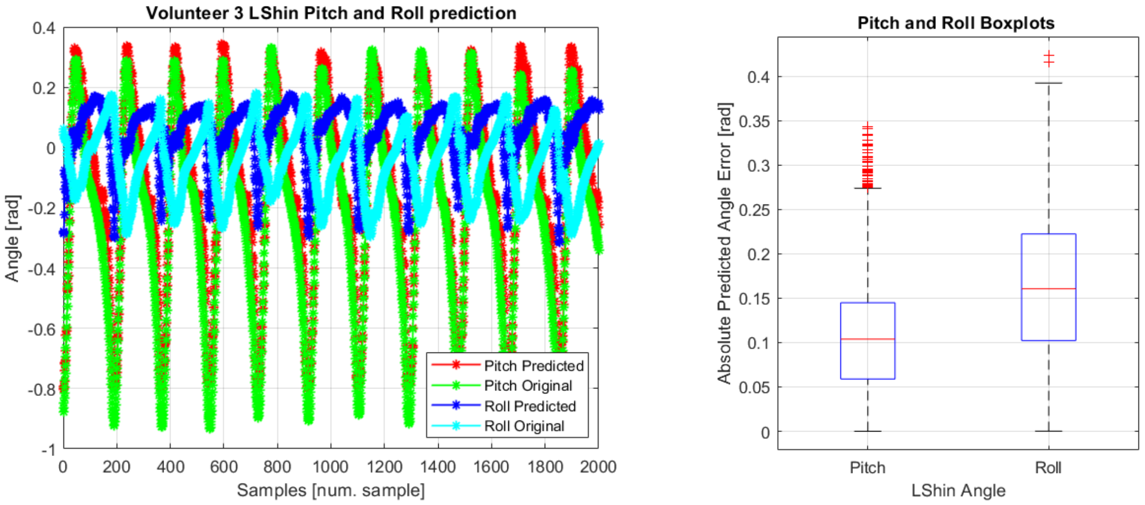
Table 5.9: RMSE mean and deviation obtained in 3DLS NV experiment to predict Pitch and Roll angles.

	TW size & Model	**Pitch RMSE [rad]**	**Roll RMSE [rad]**
LShin	**TB$_{60}$**	0.141±0.059	**0.140±0.030**
	MLP$_{10}$	**0.123±0.011**	0.173±0.018

Figure 5.9a shows the original and predicted Pitch and Roll angles of Volunteer 3. The graph belongs to the IMU LShin. It is observed that there are delayed predictions mainly in the Roll. Furthermore, the peaks of the amplitudes do not follow each other correctly. In Figure 5.9b it is observed that the majority of Pitch errors are below 0.27 rad and Roll errors below 0.37 rad. These errors are acceptable for gait prediction. Therefore, the models can be extrapolated to new volunteers' inertial data.

5.3.2 2DLX NV experiment

In 2DLX NV experiment, the Pitch and Roll angles of each one of the IMUs of the left leg are predicted from the inertial measurements of the two IMUs of the right leg together. That is, with the IMUs RShin and RThigh, the IMU LThigh and LShin are



(a) Original and predicted angles.

(b) Boxplot of LShin angles.

Figure 5.9: Volunteer 3 LShin Pitch and Roll angles prediction with TB and W=60 in 3DLS NV experiments. Absolute value error boxplots.

predicted separately. The objective of this experiment is to check if with two IMUs a good prediction of one IMU can be obtained.

Table 5.10 shows the RMSE of the test set obtained by predicting the Pitch and Roll angle of the IMU LShin and LThigh using TB and MLP. As occurred with the homologous experiment, the prediction in the shin obtains errors greater than those in the thigh, mainly due to the amplitude of the angles that these areas cause. The Pitch error in LShin is 0.236±0.073 rad with TB and 0.166±0.011 rad with MLP. LThigh obtains 0.169±0.035 rad and 0.144±0.012 rad, respectively. The Roll error is slightly lower in both IMUs. LShin obtains 0.224±0.046 rad with TB and 0.183±0.016 rad with MLP. LThigh gets 0.064±0.048 rad and 0.110±0.008 rad. The MLP, for the first time in all this work, is the most appropriate model for an experiment, except for predicting the Roll of LThigh. Detailed information about MLP results can be seen in Appendix A.3.2

Table 5.10: RMSE mean and deviation obtained in 2DLX NV experiment to predict Pitch and Roll angles.

	TW size & Model	Pitch RMSE [rad]	Roll RMSE [rad]
LShin	**TB$_{60}$**	0.236±0.073	0.224±0.046
	MLP$_{10}$	**0.166±0.011**	**0.183±0.016**
LThigh	**TB$_{60}$**	0.169±0.035	**0.064±0.048**
	MLP$_{10}$	**0.144±0.012**	0.110±0.008

The errors of predicting the angles of LShin are higher than those obtained in 3DLS NV. Pitch is 0.043 rad higher and Roll is 0.043 rad higher. If we compare it with its counterpart, the errors are much higher.

The original and predicted Pitch and Roll angles of Volunteer 2 are shown in Figure 5.10a. The graph above belongs to the IMU LShin and the one below to LThigh. Again, delayed predictions are observed in the LShin Roll. Maximum amplitudes are not calculated correctly. Figure 5.10b and Figure 5.10c show that there is variability regarding where the majority of errors are found. The models continue to be extrapolable to new inertial data, but due to the imbalance it provides depending on the segment to be predicted, the results of the 3DLS NV experiment indicate that is preferable to use 3 IMUs.

5.3.3 1DLX NV experiment

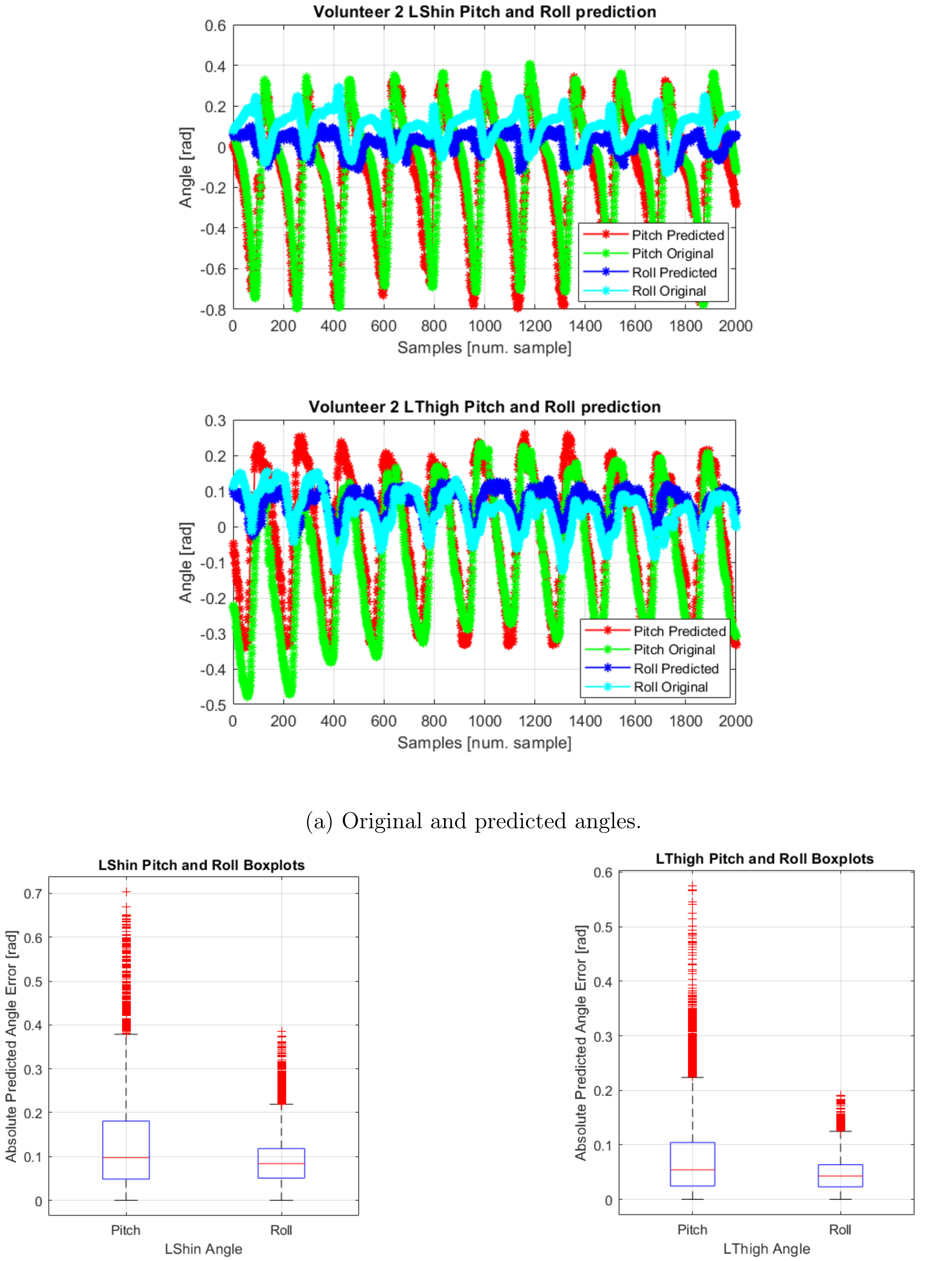
In this last experiment named 1DLX NV, the Pitch and Roll angles of each of the IMUs of the left leg are predicted from the inertial measurements of the IMUs of the right leg. That is, with the IMUs RShin and RThigh separately, the IMU LThigh and LShin are predicted separately. The objective of this experiment is to check if with one IMU a good prediction of one IMU can be obtained.

This section is made up of two parts, the first shows the results obtained from the inertial measurements of the IMU RShin and the second part shows the results obtained from the inertial measurements of the IMU RThigh.

Starting with the first part, Table 5.11 shows the RMSE obtained by the test set to predict the Pitch and Roll angle of the IMUs LShin and LThigh. For LShin, the Pitch errors are 0.163±0.019 rad with TB and 0.173±0.023 rad with MLP. For LThigh is a bit lower obtaining 0.123±0.020 rad with TB and 0.122±0.023 rad with MLP. All the Roll errors are lower than Pitch one when using TB. For LShin, 0.152±0.010 rad and 0.260±0.037 rad are obtained. For LThigh, TB gets 0.102±0.004 rad and MLP 0.150±0.041 rad, respectively. Detailed information about MLP results can be seen in Table A.34 and Table A.35 in Appendix A.3.3.

Table 5.11: RMSE mean and deviation obtained in 1DLX NV experiment to predict Pitch and Roll angles of LShin and LThigh from RShin data.

	TW size & Model	**Pitch RMSE [rad]**	**Roll RMSE [rad]**
LShin	**TB$_{60}$**	**0.163±0.019**	**0.152±0.010**
	MLP$_{10}$	0.173±0.023	0.260±0.037
LThigh	**TB$_{60}$**	0.123±0.020	**0.102±0.004**
	MLP$_{10}$	**0.122±0.023**	0.150±0.041



(a) Original and predicted angles.

(b) Boxplot of LShin angles.

(c) Boxplot of LThigh angles.

Figure 5.10: Volunteer 2 Pitch and Roll angles prediction of LShin and LThigh IMUs using RShin and RThigh inertial measurements in 2DLX NV experiment. Model TB with W=60. Absolute value error boxplots.

Figure 5.11a shows two graphs with the original and predicted Pitch and Roll angles of Volunteer 3. The graph above belongs to the IMU LShin and the one below to LThigh. As can be seen, the amplitude of the Roll angle is much smaller than that of the Pitch angle, being the amplitude of both angles smaller in the IMU LThigh. In this case, delayed predictions are not observed, but errors continue when predicting amplitude maxima. Figure 5.11b shows mainly that Roll has no outliers. Both angles have similar range of the majority of values around 0.3 rad. On the other hand, Figure 5.11c shows that the majority of Roll values are below 0.16 rad and the Pitch ones below 0.225 rad. It makes sense since the amplitude of Roll in LThigh is much smaller than in LShin.

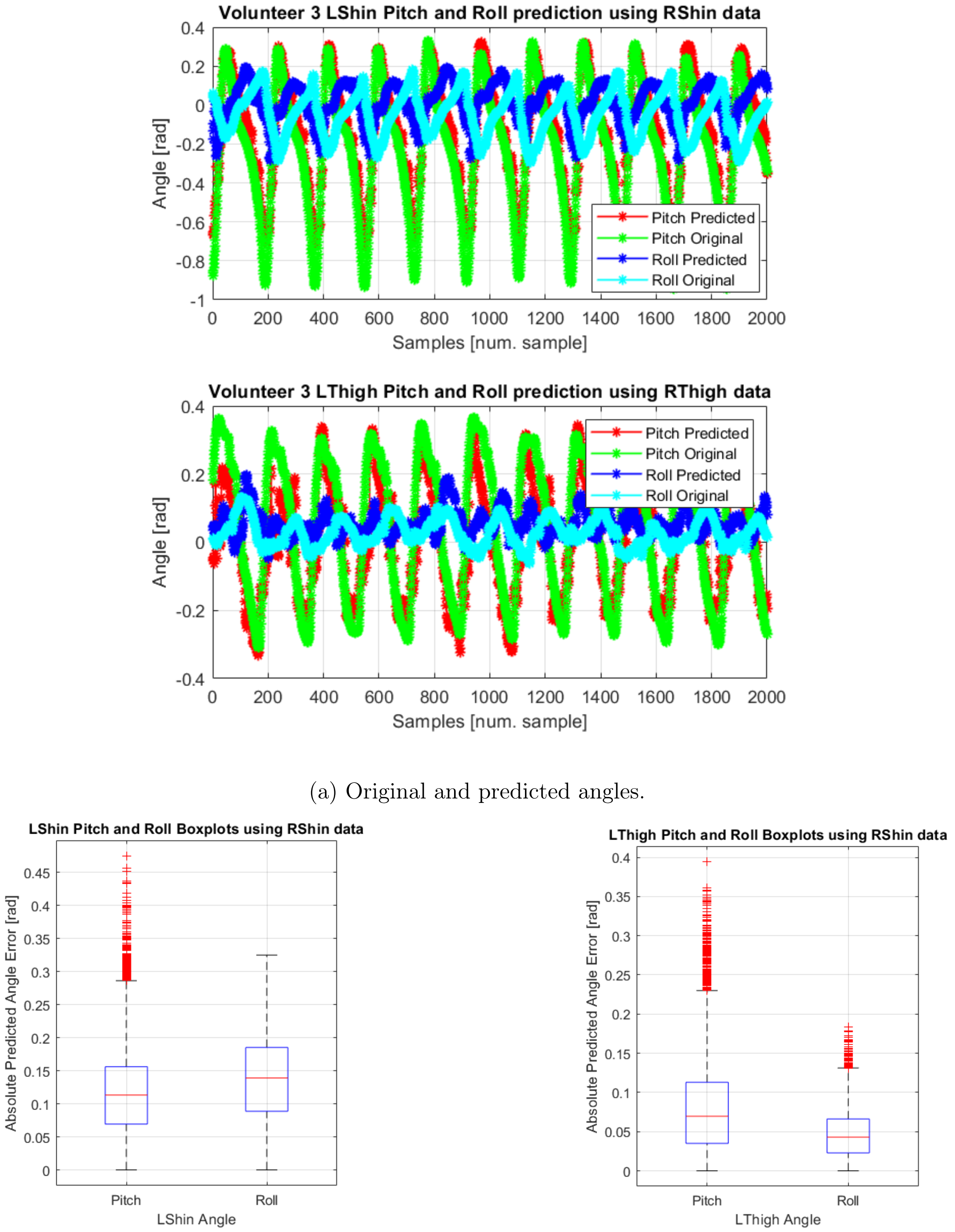
In this second part of the experiment, the results obtained from predicting with the IMU RThigh are presented. Table 5.12 shows prediction results. It obtains lower errors than previous part when predicting LThigh but higher when predicting LShin. Regarding Pitch errors, LShin obtains 0.219±0.045 rad with TB and 0.224±0.019 rad with MLP. LThigh gets 0.107±0.037 rad and 0.151±0.018 rad, respectively. For Roll, LShin gets 0.183±0.033 rad with TB and 0.173±0.019 rad with MLP. LThigh gets 0.046±0.009 rad and 0.052±0.011 rad, respectively.

Table 5.12: RMSE mean and deviation obtained in 1DLX NV experiment to predict Pitch and Roll angles of LShin and LThigh from RThigh data.

	TW size & Model	**Pitch RMSE [rad]**	**Roll RMSE [rad]**
LShin	**$\mathbf{TB_{60}}$**	**0.219±0.045**	0.183±0.033
	$\mathbf{MLP_{10}}$	0.224±0.019	**0.173±0.019**
LThigh	**$\mathbf{TB_{60}}$**	**0.107±0.037**	**0.046±0.009**
	$\mathbf{MLP_{10}}$	0.151±0.018	0.052±0.011

Figure 5.12a shows two graphs with the original and predicted Pitch and Roll angles of Volunteer 3 using TB. As the previous one, the graph above belongs to the IMU LShin and the one below to LThigh. Again, the amplitude of the Roll angle is much smaller than that of the Pitch angle, being the amplitude of both angles smaller in the IMU LThigh. In Figure 5.12b we can see that there are outliers that reach 0.9 rad in amplitude. Both angles have the majority of values around 0.4 rad. Figure 5.12c shows that most values of Roll have half the amplitude of those of Pitch.

Predicting with RShin, a lower RMSE is obtained than in 2DLX NV in all cases. However, using RThigh, the errors of predicting LThigh are the smallest in this entire section; but those of predicting LShin are the greatest. It should be noted that the Roll of LThigh is 0.046±0.009 rad, very similar to that obtained in its IE counterpart which is 0.032±0.002 rad. The model's inference in the input-output relationships is correct, but it infers better in some segments than in others. The models can be extrapolated to new

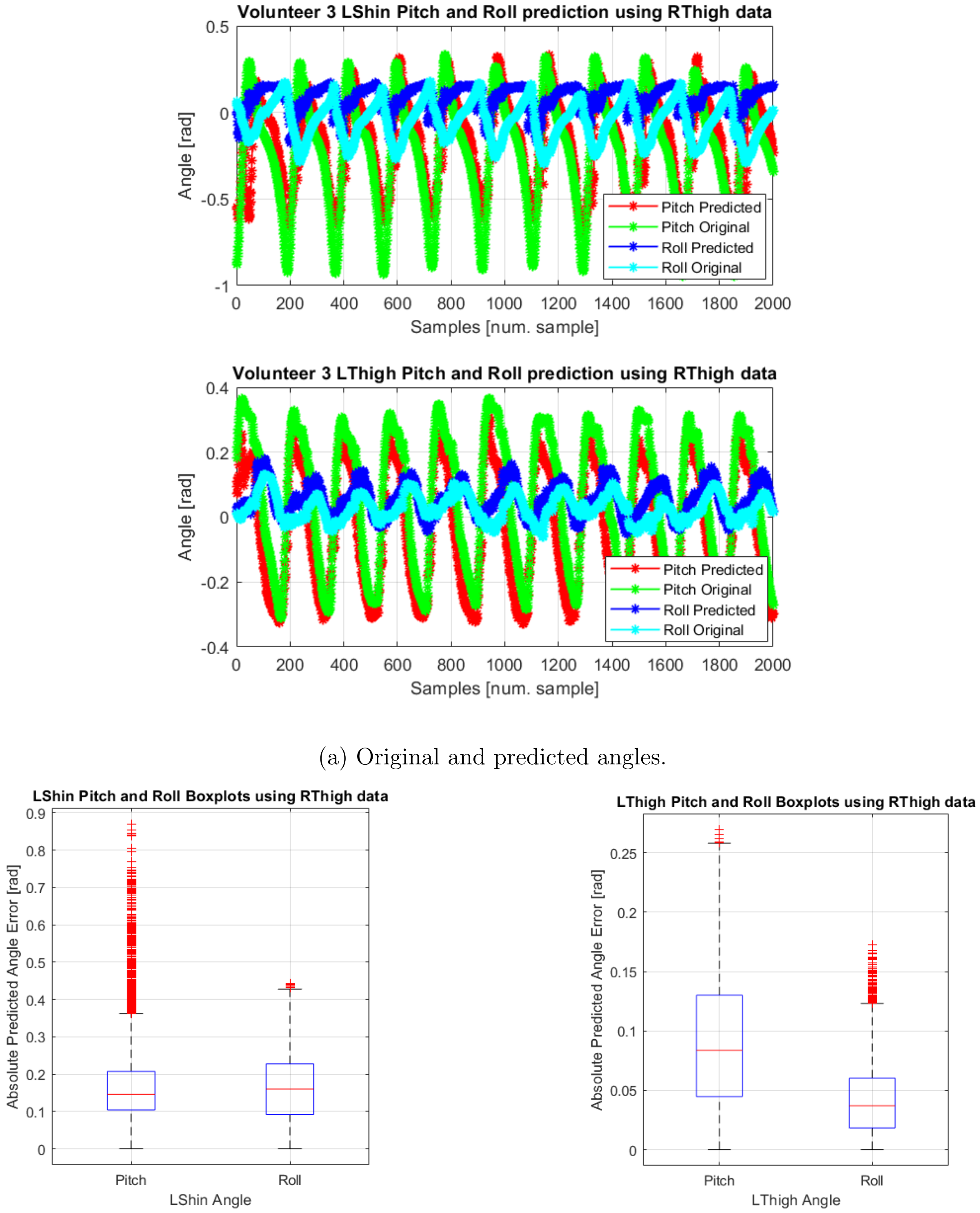


(a) Original and predicted angles.

(b) Boxplot of LShin angles.

(c) Boxplot of LThigh angles.

Figure 5.11: Volunteer 3 Pitch and Roll angles prediction of LShin and LThigh IMUs using RShin inertial measurements in 1DLX NV experiment. Model TB with W=60. Absolute value error boxplot.



(a) Original and predicted angles.

(b) Boxplot of LShin angles.

(c) Boxplot of LThigh angles.

Figure 5.12: Volunteer 3 Pitch and Roll angles prediction of LShin and LThigh IMUs using RThigh inertial measurements in 1DLX NV experiment. Model TB with W=60. Absolute value error boxplots.

volunteers, but the use of 3 IMUs is still optimal when it comes to predicting new inertial data.

Chapter 6

Conclusion and Future Lines

This chapter presents the conclusions obtained after finishing this project; as well as the future lines and the different consequences that this could have. Section 6.1 details the main findings of the research. In Section 6.2, the limitations and future lines of research are exposed. Section 6.3 explains the social, environmental and economic impacts of this work. Besides, Section 6.4 describes the Sustainable Development Goal (SDG) framework in which this work is located. Finally, Section 6.5 explains the lessons learned during the development of this Bachelor's Thesis.

6.1 Conclusions

The main objective of this Bachelor's Thesis was to develop ML models to estimate angles of the lower extremities using inertial measurements of IMUs during walking. This has applications in various areas such as rehabilitation, the design of knee prostheses and exoskeletons, helping to improve the quality of life of people who suffer from pathologies that affect gait.

Eight experiments were carried out that are divided into three groups, depending on the objective of each group. The same five preselected volunteers were used. In the first group, two baseline experiments for predicting LShin and LThigh angles with their own data were performed. The objective of the next group was the optimization of the number and position of IMUs and includes three experiments. The angles of an IMU were predicted with the inertial measurements of three, two and one IMU. The last group also comprises three experiments that were similar to those of the previous group. However, the data was not known, since it was trained with inertial measurements of the rest of the volunteers. The extrapolation of the models in new volunteers was checked.

The ML models applied were MLR, TB, MLP, SVR Linear, Gaussian and Polynomial. In addition, time window series were applied, which allowed the used data to be enlarged, by using past data for present angles. As a conclusion of these experiments, the more samples there were, the greater the computational cost was required. With 60 samples, the most suitable model for the applications of the first three experiments was TB. In the LShin baseline, the error to predict Pitch was 0.030±0.007 rads and to predict Roll was 0.029±0.007 rads. In the LThigh baseline, the Pitch error was 0.033±0.011 rads, and the Roll error was 0.038±0.011 rads.

The following experiment was the first in which IMUs were evaluated. In this case, LShin was predicted with the inertial measurements of the other three IMUs. LThigh was not predicted since it was used for training. The error of predicting Pitch was 0.036±0.008 rads and that of Roll was 0.032±0.017 rads. However, with two-layer MLP and 10 TW samples, the obtained results were valid, although superior to those obtained with TB. By using 10 samples, the computational cost was lower than using 60 samples, so in the rest of the experiments this method and TB of 60 samples were used.

After choosing the optimal TW sizes for the best ML methods, was time to reduce the number of IMUs for training. When using two IMUs to predict, the error of Pitch and Roll were 0.039±0.010 rads and 0.032±0.006 rads for LShin; and 0.034±0.009 rads and 0.031±0.015 rads, respectively, for LThigh. In the last experiment of predicting with a single IMU, the errors are higher in all cases.

Finally, we evaluated the extrapolability of the chosen models to new volunteers data. The predictions produced considerably larger errors. However, using only RThigh to predict, the Roll of LThigh obtained an RMSE of 0.046±0.009 rads with TB. To predict the Pitch of LShin, an RMSE of 0.123±0.011 rads was obtained by using three IMUs and the MLP model. It was the only time that MLP made a more precise estimation than TB.

In summary, the results obtained in this work showed that it is possible to estimate sensor angles with ML techniques. Furthermore, these techniques showed to be effective and obtained accurate results. Having carried out different experiments to assess optimal number of IMUs and their position, it allows us to know the importance of each sensor, also knowing the cost that can be saved.

6.2 Limitations and future approaches

During the development of this work, some limitations have been experienced that affect the results obtained. These limitations serve as a starting point for future prediction studies of human gait:

1. The TW method has been applied to increase the number of data since only 5 volunteers have been used to carry out the experiments, but it would be advisable

to initially use a large database. Additionally, it would be valuable to add volunteers with anthropomorphic differences and gait-related pathologies.

2. The signals used are recorded while walking on a flat surface. It would be interesting to use signs of volunteers doing different activities such as climbing stairs or running; in addition to different speeds, to be able to make predictions of day-to-day activities.

3. It would be advisable to develop regressors and NN in more depth for the estimation of data from unknown volunteers. In this project, acceptable results have been obtained by extrapolating the models, but it would be advisable to optimize hyperparameters.

6.3 Impact

6.3.1 Social Impact

The analysis of human gait or gait biomechanics has a significant impact on various aspects of society, especially in the field of health and rehabilitation. It allows health professionals to evaluate and diagnose disorders. This allows us to design personalized treatments and therapies. It can reveal hidden or underlying health problems, such as neuromuscular diseases or postural imbalances. This can lead to earlier diagnoses and, therefore, more effective treatment.

The biomechanics of gait is essential in the development and improvement of medical devices such as prostheses, orthoses and orthopedic footwear. These devices improve the quality of life of people with disabilities and allow them to move more easily. Even, in jobs that require standing or walking for long periods, gait analysis can be used to improve ergonomics and workplace safety. This can help prevent musculoskeletal injuries and increase productivity.

Besides, understanding how people walk and get around influences the design of public spaces, such as sidewalks and parks, as well as the design of public transportation systems. The aim is to make these spaces and systems more accessible and comfortable for all age groups and abilities.

In general, gait analysis contributes to improving people's quality of life by allowing a better understanding and addressing of mobility and health problems. This is especially important in the aging population, where mobility becomes a key factor in maintaining independence and quality of life.

6.3.2 Environmental Impact

To carry out this work, signals recorded with IMUs are used. The manufacturing of these sensors can generate electronic microwaste that can be harmful to the environment. However, this work aims to reduce the number of these sensors, which contributes to mitigating the waste generated.

Due to the use of MLs algorithms, computing resources are required that consume a large amount of electrical energy during the model training and evaluation phase. Electricity generation can be through fossil fuels or renewable energy. In the first case, this implies the burning of these resources, releasing a large amount of greenhouse gases into the atmosphere, which contributes to global warming. In the second case, it involves the construction of structures such as hydroelectric dams, electric generating mills or solar plants, which affect the landscape and the terrain on which they are located. For example, in this work, the MLP model obtains similar results to those of TB, with much lower computational cost.

By improving gait quality and reducing the risk of injury, sustainable mobility is promoted. People who can walk or run more efficiently may opt for more environmentally friendly means of transportation, such as walking or cycling, rather than relying too heavily on motorized vehicles.

6.3.3 Economic Impact

Finally, the economic impact of this work is found in the inertial sensors. Its low cost, small size and easy placement favor the carrying out of studies with these devices by populations or teams with low resources. There are clothes, watches and accessories on the market that have these sensors built in, which allow movement analysis to be carried out more easily in the future.

6.4 Sustainable Development Goals framework

This Bachelor's Thesis is framed within the SDGs taking into account the impacts in Section 6.3. The SDGs are 17 global goals established in 2015 by the United Nations (UN) as part of the 20230 agenda for sustainable development. The objectives and goals to which this project benefits are shown in Table 6.4.

Table 6.1: Specific SGD goals within the framework of this work.

Goal	**Target**	**Description**
SDG 1: No Poverty	SDG 1.4	By 2030, ensure that all men and women, in particular the poor and the vulnerable, have equal rights to economic resources, as well as access to basic services, ownership and control over land and other forms of property, inheritance, natural resources, appropriate new technology and financial services, including microfinance.
SDG 3: Good Health and Well-Being	SDG 3.8	Achieve universal health coverage, including financial risk protection, access to quality essential health-care services and access to safe, effective, quality and affordable essential medicines and vaccines for all.
	SDG 3.d	Strengthen the capacity of all countries, in particular developing countries, for early warning, risk reduction and management of national and global health risks.
SDG 7: Affordable and Clean Energy	SDG 7.a	By 2030, enhance international cooperation to facilitate access to clean energy research and technology, including renewable energy, energy efficiency and advanced and cleaner fossil-fuel technology, and promote investment in energy infrastructure and clean energy technology.
SDG 9: Industry, Innovation and Infrastructure	SDG 9.5	Enhance scientific research, upgrade the technological capabilities of industrial sectors in all countries, in particular developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers per 1 million people and public and private research and development spending.

6.5 Bachelor's Thesis Based Learning

During the development of this work, knowledge acquired during the Biomedical Engineering Bachelor's Degree has been put into practice. Starting from studying the different parameters of the biomechanics of human gait, such as the phases of gait and modes of locomotion. Although this work has focused particularly on the angles of the knee joints. For that, I needed to understand IMUs operation.

The quality of training data is essential for the effectiveness of machine learning models. I learned that accurate data collection and labeling takes time and care. Besides, it was the first time that I have implemented a NN as MLP. Finally, I have improved my programming skills in MATLAB for signal processing; as well as in writing a scientific document in LaTeX.

Bibliography

[1] Claudiane Arakaki Fukuchi, Reginaldo Kisho FUKUCHI, and Marcos. DUARTE. Effects of walking speed on gait biomechanics in healthy participants: a systematic review and meta-analysis. *Systematic reviews*, 8(1):1–11, 2019.

[2] Seung Eel Oh, Ahnryul CHOI, and Joung Hwan MUN. Prediction of ground reaction forces during gait based on kinematics and a neural network model. *Journal of biomechanics*, 46(14):2372–2380, 2013.

[3] T. Maruyama, H. Toda, S. Kanoga, M. Tada, and Y. Endo. Accuracy evaluation of human gait estimation by a sparse set of inertial measurement units. *Activity and Behavior Computing*, pages 51–61, 2021.

[4] W. Tao, T. Liu, R. Zheng, and H. Feng. Gait analysis using wearable sensors. *Sensors*, 12(2):2255–2283, 2012.

[5] Katherine A Raichle, Marisol A Hanley, Ivan Molton, Nancy J Kadel, Kellye Campbell, Emily Phelps, Dawn Ehde, and Douglas G Smith. Prosthesis use in persons with lower-and upper-limb amputation. *Journal of rehabilitation research and development*, 45(7):961, 2008.

[6] Justin Z. Laferrier and Robert. GAILEY. Advances in lower-limb prosthetic technology. *Physical Medicine and Rehabilitation Clinics*, 21(1):87–110, 2010.

[7] Jan Andrysek. Lower-limb prosthetic technologies in the developing world: A review of literature from 1994–2010. *Prosthetics and orthotics international*, 34(4):378–398, 2010.

[8] Jaime C Paz. Lower-extremity amputation. *Acute Care Handbook for Physical Therapists-E-Book*, page 361, 2013.

[9] Samer Mohammed, Yacine Amirat, and Hala Rifai. Lower-limb movement assistance through wearable robots: State of the art and challenges. *Advanced Robotics*, 26(1-2):1–22, 2012.

[10] Daniel P. Young, Aaron J.and Ferris. State of the art and future directions for lower limb robotic exoskeletons. *Transactions on Neural Systems and Rehabilitation Engineering*, 25(2):171–182, 2016.

[11] E. Rocon and et al. Application of inertial sensors in rehabilitation robotics. *10th International Conference on Rehabilitation Robotics*, pages 145–150, 2007.

[12] Zikang Zhou and et al. Individualized gait generation for rehabilitation robots based on recurrent neural networks. *Transactions on Neural Systems and Rehabilitation Engineering*, 29:273–281, 2020.

[13] Preeti Khera and Neelesh. Kumar. Role of machine learning in gait analysis: a review. *Journal of Medical Engineering & Technology*, 44(8):441–467, 2020.

[14] Arash Salarian, Pierre R Burkhard, François JG Vingerhoets, Brigitte M Jolles, and Kamiar Aminian. A novel approach to reducing number of sensing units for wearable gait analysis systems. *IEEE Transactions on Biomedical Engineering*, 60(1):72–77, 2012.

[15] Marcos Edgar Fernández Cuadros and et al. Análisis de la calidad de vida en pacientes con prótesis de rodilla. *Tesis Doctoral*, 2013.

[16] Sara García-de Villa, Ana Jiménez-Martín, and Juan Jesús García-Domínguez. A database of physical therapy exercises with variability of execution collected by wearable sensors. *Scientific Data*, 9(1):1–13, 2022.

[17] Estefania Munoz Diaz. A review of indoor localization methods based on inertial sensors. *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, 16:311–333, 2019.

[18] C. Prakash, R. Kumar, and N. Mittal. Recent developments in human gait research: Parameters, approaches, applications, machine learning techniques, datasets and challenges. *Artif. Intell. Rev.*, 49(1):1–40, 2018.

[19] A. S. Alharthi, S. U. Yunas, and K. B. Ozanyan. Deep learning for monitoring of human gait: A review. *IEEE Sensors J.*, 19(21):9575–9591, 2019.

[20] Behboodi, A. Zahradka, N. H. Wright, J. Alesi, and S. C. K. Lee. Real time detection of seven phases of gait in children with cerebral palsy using two gyroscopes. *Sensors*, 19(11):2517, 2019.

[21] W. Hassani, S. Mohammed, H. Rifai, and Y. Amirat. Emg based approach for wearer-centered control of a knee joint actuated orthosis. *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, page 990–995, 2013.

[22] W. Hassani, S. Mohammed, H. Rifai, and Y. Amirat. Gait partitioning methods: A systematic review. *Sensors*, 16(1):40–42, 2016.

[23] E. Alpaydin. Introduction. *Introduction to Machine Learning*, pages 1–20, 2014.

[24] Ł. Kidziński, S. Delp, and M. Schwartz. Automatic real-time gait event detection in children using deep neural networks. *PLoS ONE*, 14(1):235–240, 2019.

[25] Ryan M CHAPMAN. From inertial measurement units (imus) to physical therapy (pt): The use of patient biomechanics to inform pt prescription. *Tesis Doctoral.Dartmouth College.*, 2018.

[26] V. Kumar, Y. V. Hote, and S. Jain. Review of exoskeleton: History, design and control. *Proc. 3rd Int. Conf. Recent Develop. Control, Autom. Power Eng. (RDCAPE)*, page 677–682, 2019.

[27] A. S. Khan, D. C. Livingstone, and C. L. Hurd. Wearable exoskeleton systems: Design, control and applications. *U.K.: The Institution of Engineering and Technology*, page London, 2018.

[28] General Electric Company Corporate Development. Research and development prototype for machine augmentation of human strength and endurance hardiman i project. *Schenectady*, pages NY,USA, 1971.

[29] T. Zhang and H. H. Huang. A lower-back robotic exoskeleton: Industrial handling augmentation used to provide spinal support. *IEEE Robot. Automat. Mag*, 25(2):95–106, 2018.

[30] A. S. Khan, D. C. Livingstone, and C. L. Hurd. Retraining walking over ground in a powered exoskeleton after spinal cord injury: A prospective cohort study to examine functional gains and neuroplasticity. *J. NeuroEng. Rehabil.*, 16(1):145, 2019.

[31] F. Di Russo, M. Berchicci, R. L. Perri, and M. Ripani. 'a passive exoskeleton can push your life up: Application on multiple sclerosis patients. *PLoS ONE*, 8(10), 2013.

[32] C. J. Walsh, K. Endo, and H. Herr. A quasi-passive leg exoskeleton for load-carrying augmentation. *Int. J. Humanoid Robot.*, 4(3):487–506, 2007.

[33] K. Suzuki, G. Mito, H. Kawamoto, Y. Hasegawa, and Y. Sankai. Intention-based walking support for paraplegia patients with robot suit hal. *Adv. Robot*, 21(12):1441–1469, 2007.

[34] S. Wang, L. Wang, C. Maijneke, and et al. Design and control of the mindwalker exoskeleton. *IEEE Trans. Neural Syst. Rehabil*, 23(2):277–286, 2015.

[35] H. Herr. Exoskeletons and orthoses: Classification, design challenges and future directions. *J. NeuroEng. Rehabil.*, 6(1):1–9, 2009.

[36] J. A. Blaya and H. Herr. Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. *IEEE Trans. Neural Syst. Rehabil*, 12(1):24–31, 2004.

[37] C. Millen, U. Brägger, and J. G. Wittneben. Influence of prosthesis type and retention mechanism on complications with fixed implant-supported prostheses: a systematic review applying multivariate analyses. *International journal of oral & maxillofacial implants*, 30, 2015.

[38] B. Chen, H. Ma, L. Y. Qin, and et al. Recent developments and challenges of lower extremity exoskeletons. *J. Orthopaedic Transl.*, 5:26–37, 2016.

[39] Z. Zhao, H .and Wang, S. Qiu, Y. Shen, and J. Wang. Imu-based gait analysis for rehabilitation assessment of patients with gait disorders. *Proc. 4th Int. Conf. Syst. Informat. (ICSAI),*, page 622–626, 2017.

[40] S. Wang, L. Wang, C. Maijneke, and et al. Gait analysis of national athletes after anterior cruciate ligament reconstruction following three stages of rehabilitation program: Symmetrical perspective. *Gait Posture*, 48:152–158, 2016.

[41] O. Dehzangi, M. Taherisadr, and R. ChangalVala. Imu-based gait recognition using convolutional neural networks and multi-sensor fusion. *Sensors*, 17(12):2735, 2017.

[42] I. Birch, T. Birch, and D. Bray. The identification of emotions from gait. *Sci. Jus*, 56(5):351–356, 2016.

[43] G. Bao, L. Pan, H. Fang, X. Wu, S. Yu, H. Cai, B. Yu, and Y. Wan. Academic review and perspectives on robotic exoskeletons. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 27(11):2294–2304, 2019.

[44] M. W. Whittle. Gait analysis?: An introduction. *Amsterdam, The Netherlands: Elsevier*, 1991.

[45] D. Levine, J. Richards, and M. W. Whittle. Whittle's gait analysis. *5th ed. Edinburgh, Scotland: Elsevier*, 2012.

[46] Walter Pirker and Regina Katzenschlager. Gait disorders in adults and the elderly: A clinical guide. *Wiener klinische Wochenschrift*, 129, 10 2016.

[47] S. M. Shafiul Hasan, M. R. Siddiquee, R. Atri, R. Ramon, J. S. Marquez, and O. Bai. Prediction of gait intention from pre-movement eeg signals: A feasibility study. *J. NeuroEng. Rehabil*, 17(1):1–16, 2020.

[48] D. Levine, J. Richards, and M. W. Whittle. Whittle's gait analysis. *5th ed. Edinburgh, Scotland: Elsevier*, 2012.

[49] Vijeth Rai and Eric Rombokas. A framework for mode-free prosthetic control for unstructured terrains. *16th International Conference on Rehabilitation Robotics*, 2019.

[50] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Appl. Mech. Rev.*, 70(1):1–19, 2018.

[51] P. Kutilek and B. Farkasova. 'prediction of lower extremities' movement by angle-angle diagrams and neural networks. *Acta Bioeng. Biomech.*, 13(2):57–65, 2011.

[52] O. Mazumder, A. S. Kundu, P. K. Lenka, and S. Bhaumik. Multi-channel fusion based adaptive gait trajectory generation using wearable sensors. *J. Intell. Robotic Syst*, 86(3):335–351, 2017.

[53] J. Wang, L. Wang, S. M. Miran, X. Xi, and A. Xue. Surface electromyography based estimation of knee joint angle by using correlation dimension of wavelet coefficient. *IEEE Access*, 7:60522–60531, 2019.

[54] Marion Mundt, Wolf Thomsen, Tom Witter, Arnd Koeppe, Sina David, Franz Bamer, Wolfgang Potthast, and Bernd Markert. Prediction of lower limb joint angles and moments during gait using artificial neural networks. *Medical & Biological Engineering & Computing*, 58:211–225, 2020.

[55] A. Findlow, J. Y. Goulermas, C. Nester, D. Howard, and L. P. J. Kenney. Predicting lower limb joint kinematics using wearable motion sensors. *Gait and posture*, 28(1):120–126, 2008.

[56] Hyerim Lim, Bumjoon Kim, and Sukyung Park. Prediction of lower limb kinetics and kinematics during walking by a single imu on the lower back using machine learning. *Sensors*, 20(1):130, 2019.

[57] Mohsen Gholami, Christopher Napier, and Carlo Menon. Estimating lower extremity running gait kinematics with a single accelerometer: A deep learning approach. *Sensors*, 20(10):2939, May 2020.

[58] Vincent Hernandez, Davood Dadkhah, Vahid Babakeshizadeh, and Dana Kulić. Lower body kinematics estimation from wearable sensors for walking and running: A deep learning approach. *Gait and Posture*, 83:185–193, 2021.

[59] Chandra Prakash, A Sujil, Rajesh Kumar, and Namita Mittal. Linear prediction model for joint movement of lower extremity. In *Recent Findings in Intelligent Computing Techniques: Proceedings of the 5th ICACNI 2017, Volume 1*, pages 235–243. Springer, 2019.

[60] Marion Mundt, Arnd Koeppe, Sina David, Tom Witter, Franz Bamer, Wolfgang Potthast, and Bernd Markert. Estimation of gait mechanics based on simulated and measured imu data using an artificial neural network. *Frontiers in bioengineering and biotechnology*, 8:41, 2020.

[61] J. Sung, S. Han, H. Park, H. M. Cho, S. Hwang, J. W. Park, and I. Youn. Prediction of lower extremity multi-joint angles during overground walking by using a single imu with a low frequency based on an lstm recurrent neural network. *Sensors*, 22(1):53, 2021.

[62] S. M. Moghadam, T. Yeung, and J. Choisne. A comparison of machine learning models' accuracy in predicting lower-limb joints' kinematics, kinetics, and muscle forces from wearable sensors. *Scientific Reports*, 13(1):5046, 2023.

[63] J.-Y. Jung, W. Heo, H. Yang, and H. Park. A neural network-based gait phase classification method using sensors equipped on lower limb exoskeleton robots. *Introduction to Machine Learning*, pages 1–20, 2014.

[64] Y. Ma, X. Wu, C. Wang, Z. Yi, and G. Liang. Gait phase classification and assist torque prediction for a lower limb exoskeleton system using kernel recursive least-squares method. *Sensors*, 19(24):5449, 2019.

[65] I. Kang, P. Kunapuli, and A. J. Young. Real-time neural network-based gait phase estimation using a robotic hip exoskeleton. *IEEE Trans. Med. Robot. Bionics*, 2(1):28–37, 2020.

[66] J. D. Farah, N. Baddour, and E. D. Lemaire. Design, development, and evaluation of a local sensor-based gait phase recognition system using a logistic model decision tree for orthosis-control. *J. NeuroEng. Rehabil.*, 16(1):1–11, 2019.

[67] S. Pasinetti, A. Fornaser, M. Lancini, M. De Cecco, and G. Sansoni. Assisted gait phase estimation through an embedded depth camera using modified random forest algorithm classification. *IEEE Sensors J*, 20(6):3343–3355, 2020.

[68] T. Zhen, L. Yan, and P. Yuan. Walking gait phase detection based on acceleration signals using lstm-dnn algorithm. *Algorithms*, 12(12):253, 2019.

[69] X. Zhang, S. Sun, C. Li, and Z. Tang. Impact of load variation on the accuracy of gait recognition from surface emg signals. *Appl. Sci*, 8(9):1462, 2018.

[70] J. Song, A. Zhu, Y. Tu, Y. Wang, M. A. Arif, H. Shen, Z. Shen, X. Zhang, and G. Cao. Human body mixed motion pattern recognition method based on multi-source feature parameter fusion. *Sensors*, 20(2):537, 2020.

[71] F. Wang, L. Yan, and J. Xiao. Human gait recognition system based on support vector machine algorithm and using wearable sensors. *Sensors Mater.*, 31(4):1335–1349, 2019.

[72] A. C. Villa-Parra, D. Delisle-Rodriguez, T. Botelho, J. J. V. Mayor, A. L. Delis, R. Carelli, A. Frizera Neto, and T. F. Bastos. Control of a robotic knee exoskeleton for assistance and rehabilitation based on motion intention from semg. *Res. Biomed. Eng.*, 34(3):198–210, 2018.

[73] A. Parri, K. Yuan, D. Marconi, T. Yan, S. Crea, M. Munih, R. M. Lova, N. Vitiello, and Q. Wang. Real-time hybrid locomotion mode recognition for lower limb wearable robots. *IEEE/ASME Trans. Mechatronics*, 22(6):2480–2491, 2017.

[74] M. Roberts, D. Mongeon, and F. Prince. Biomechanical parameters for gait analysis: a systematic review of healthy human gait. *Phys. Ther. Rehabil*, 4(6), 2017.

[75] A. N. et al. Ramesh. Artificial intelligence in medicine. *Annals of the Royal College of Surgeons of England*, 86(5):334, 2004.

[76] T. O. Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 4:19–48, 2010.

[77] C. Sammut and G. I. Webb. Supervised learning. *Encyclopedia of Machine Learning.*, page 941, 2010.

[78] Yogesh Singh, Bhatia Pradeep Kumar, and Omprakash Sangwan. A review of studies on machine learning techniques. *International Journal of Computer Science and Security*, 1(1):70–84, 2007.

[79] Lynn E. Eberly. Multiple linear regression. *Topics in Biostatistics*, pages 165–187, 2007.

[80] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *Workshop Comput. Learn. Theory (COLT)*, page 144–152, 1992.

[81] Derek A. Pisner and David M. Schnyer. Support vector machine. in machine learning . *Machine learning*, pages 101–121, 2020.

[82] Carolin Strobl, James Malley, and Gerhard Tutz. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4):323, 2009.

[83] Matt W. Gardner and Dorling. S. R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[84] W. Ertel. Introduction to artificial intelligence. *London, U.K.: Springer*, 2011.

[85] Rafael Yuste. From the neuron doctrine to neural networks. *Nature reviews neuroscience*, 16(8):487–497, 2015.

[86] Tzu-Tsung Wong and Po-Yang Yeh. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, 2020.

[87] Gi-Wook Cha, Hyeun Moon, Young-Min Kim, Won-Hwa Hong, Jung-Ha Hwang, Won-Jun Park, and Young-Chan Kim. Development of a prediction model for demolition waste generation using a random forest algorithm based on small datasets. *International journal of environmental research and public health*, 17, 09 2020.

[88] N. Ahmad, R. A. R. Ghazilla, and V. Khairi, N. M.and Kasi. Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, 1(2):256–262, 2013.

[89] Dariusz Tomaszewski, Jacek Rapinski, and Pelc-Mieczkowska Renata. Concept of ahrs algorithm designed for platform independent imu attitude alignment. *Reports on Geodesy and Geoinformatics*, 104, 12 2017.

[90] Christos. MOUSAS. Full-body locomotion reconstruction of virtual characters using a single inertial measurement unit. *Sensors*, 17(11):2589, 2017.

[91] J. Wang and W. J. Wilson. 3d relative position and orientation estimation using kalman filter for robot control. *IEEE Computer Society*, pages 2638–2639, 1992.

[92] F. Zhang. Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57, 1997.

[93] A. Kim and M. F. Golnaraghi. A quaternion-based orientation estimation algorithm using an inertial measurement unit. *Position Location and Navigation Symposium*, pages 268–272., 2004.

[94] Chris-martin and Jeremy Ruston. Mnemonics to help remember the terms for rotations around principal axes. *wikipedia*, 2006.

[95] McNeil Alexander R. Functional design in fishes. *wikipedia*, 2023.

[96] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE pervasive computing*, 2(3):24–33, 2003.

[97] Z. Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[98] The MathWorks Inc. Matlab version: 9.13.0 (r2022b). ., 2022.

[99] Robert Browder and Carrie Cross. Welcome to overleaf: A brief overview of opportunities. 2018.

[100] Joan Solà. Quaternion kinematics for the error-state kalman filter. *arXiv:1711.02508*, 2017.

[101] Julio S. Lora-Millan, Andres Hidalgo, and Eduardo Rocon. An imus-based extended kalman filter to estimate gait lower limb sagittal kinematics for the control of wearable robotic devices. *IEEE Access*, 9:144540–144554, 2021.

[102] Antonio Tricoli and Noushin Nasiri. Wearable and miniaturized sensor technologies for personalized and preventive medicine. *Advanced Functional Materials*, 27(15):1605271, 2017.

[103] Manuela Brito Duarte and et al. Validity and reliability of a smartphone-based assessment for anticipatory and compensatory postural adjustments during predictable perturbations. *Gait & Posture*, 96:9–17, 2022.

[104] Manuela Brito Duarte and et al. Lower extremity amputations. *Essentials of Vascular Surgery for the General Surgeon*, pages 119–132, 2015.

[105] E. González Cañas, A. Giménez Gaibar, and S. et al. Bellmunt Montoya. Estudio de calidad de vida en pacientes afectos de isquemia critica a medio plazo. *Angiologia*, 51(1):19–27, 2007.

[106] J. R. Quinlan. Induction of decision trees. *Mach. Learn*, 1(1):81–106, 1986.

[107] C. Buckner and J. Garson. Connectionism, fall 2019. *Fall 2019. Metaphysics Research Lab.*, 2019.

[108] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133, 1943.

[109] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958.

[110] R. Miikkulainen. Topology of a neural network. *Encyclopedia of Machine Learning, C. Sammut G. I. Webb, Eds. Boston, MA, USA: Springer, 2010,*, page 988–989, 2010.

[111] S. I. Gallant. Neural network learning and expert systems. *Cambridge, MA, USA: MIT Press,*, 1993.

[112] S. L. Salzberg. C4.5: Programs for machine learning. *Mach. Learn.*, 16(3):235–240, 1994.

[113] D. Novak, P. Reberšek, S. M. M. De Rossi, M Donati, T. Podobnik, J. andBeravs, T. Lenzi, N. Vitiello, M. C. Carrozza, and M. Munih. Automated detection of gait initiation and termination using wearable sensors. *Med. Eng. Phys.*, 35(12):1713–1720, 2013.

[114] P. Kutilek and S. Viteckova. Prediction of lower extremity movement by cyclograms. *Acta Bioeng. Biomech*, 52(1):51, 2012.

[115] C. A. Ramezan, T. A. Warner, and A. E. Maxwell. Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. *Remote Sensing*, 11(2):185, 2019.

Appendix A

MLP tables

A.1 Baseline experiments

A.1.1 Angle prediction in IMU LShin

Table A.1: RMSE mean and deviation obtained in LShin baseline experiment to predict the Pitch angle using one-layer MLP and 5-sample TW.

	Num. Neurons					
	5	**10**	**15**	**20**	**25**	**30**
TEST RMSE [rad]	0.205±0.032	0.186±0.025	0.184±0.018	0.175±0.022	0.161±0.019	**0.160±0.011**
TRAIN RMSE [rad]	0.203±0.031	0.186±0.025	0.183±0.019	0.176±0.024	0.162±0.019	**0.160±0.016**

Table A.2: RMSE mean and deviation obtained in LShin baseline experiment to predict the Pitch angle using one-layer MLP and 10-sample TW.

	Num. Neurons					
	10	**20**	**30**	**40**	**50**	**60**
TEST RMSE [rad]	0.157±0.018	0.121±0.017	0.130±0.016	0.129±0.013	**0.120±0.017**	0.137±0.019
TRAIN RMSE [rad]	0.160±0.016	0.123±0.016	0.131±0.015	0.127±0.012	**0.118±0.011**	0.137±0.012

Table A.3: RMSE mean and deviation obtained in LShin baseline experiment to predict the Pitch angle using two-layer MLP and 5-sample TW.

	Num. Neurons	5	10	15
TEST	**5**	0.053±0.009	0.053±0.010	0.051±0.013
RMSE	**10**		**0.046±0.010**	0.047±0.009
[rad]	**15**			0.048±0.011
TRAIN	**5**	0.049±0.006	0.046±0.007	0.044±0.008
RMSE	**10**		**0.040±0.008**	0.040±0.007
[rad]	**15**			0.040±0.009

Table A.4: RMSE mean and deviation obtained in LShin baseline experiment to predict the Pitch angle using two-layer MLP and 10-sample TW.

	Num. Neurons	10	20	30
TEST	**10**	**0.042±0.008**	0.043±0.009	0.045±0.011
RMSE	**20**		0.043±0.010	0.044±0.010
[rad]	**30**			0.043±0.010
TRAIN	**10**	**0.037±0.009**	0.037±0.008	0.037±0.008
RMSE	**20**		0.035±0.007	0.034±0.008
[rad]	**30**			0.036±0.008

Table A.5: RMSE mean and deviation obtained in LShin baseline experiment to predict the Roll angle using one-layer MLP and 5-sample TW.

	Num. Neurons					
	5	**10**	**15**	**20**	**25**	**30**
TEST RMSE [rad]	0.202±0.007	0.193±0.010	0.186±0.006	0.178±0.009	0.166±0.008	**0.166±0.007**
TRAIN RMSE [rad]	0.202±0.006	0.190±0.010	0.185±0.006	0.175±0.008	0.171±0.008	**0.165±0.008**

Table A.6: RMSE mean and deviation obtained in LShin baseline experiment to predict the Roll angle using one-layer MLP and 10-sample TW.

	Num. Neurons					
	10	**20**	**30**	**40**	**50**	**60**
TEST RMSE [rad]	0.159±0.010	0.147±0.022	0.145±0.014	**0.138±0.019**	0.144±0.020	0.145±0.020
TRAIN RMSE [rad]	0.157±0.011	0.145±0.023	0.144±0.013	**0.136±0.018**	0.142±0.022	0.144±0.020

Table A.7: RMSE mean and deviation obtained in LShin baseline experiment to predict the Roll angle using two-layer MLP and 5-sample TW.

	Num. Neurons	**5**	**10**	**15**
TEST	**5**	0.048±0.001	0.050±0.001	0.049±0.001
RMSE	**10**		0.049±0.001	0.048±0.003
[rad]	**15**			**0.046±0.003**
TRAIN	**5**	0.048±0.001	0.048±0.003	0.049±0.001
RMSE	**10**		0.048±0.001	0.047±0.003
[rad]	**15**			**0.044±0.003**

Table A.8: RMSE mean and deviation obtained in LShin baseline experiment to predict the Roll angle using two-layer MLP and 10-sample TW.

	Num. Neurons	**10**	**20**	**30**
TEST	**10**	0.048±0.001	0.048±0.001	0.050±0.002
RMSE	**20**		0.048±0.002	0.048±0.001
[rad]	**30**			**0.046±0.001**
TRAIN	**10**	0.048±0.001	0.048±0.001	0.050±0.002
RMSE	**20**		0.047±0.001	0.048±0.002
[rad]	**30**			**0.046±0.002**

Table A.9: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Pitch angle using one-layer MLP and 5-sample TW.

	Num. Neurons					
	5	**10**	**15**	**20**	**25**	**30**
TEST RMSE [rad]	0.101±0.007	0.108±0.003	**0.097±0.003**	0.115±0.005	0.117±0.010	0.114±0.006
TRAIN RMSE [rad]	0.106±0.008	0.114±0.002	**0.105±0.009**	0.111±0.010	0.110±0.005	0.110±0.002

Table A.10: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Pitch angle using one-layer MLP and 10-sample TW.

	Num. Neurons					
	10	**20**	**30**	**40**	**50**	**60**
TEST RMSE [rad]	**0.092±0.010**	0.095±0.007	0.097±0.006	0.105±0.014	0.102±0.005	0.106±0.006
TRAIN RMSE [rad]	**0.090±0.011**	0.092±0.011	0.092±0.009	0.097±0.017	0.096±0.010	0.101±0.009

Table A.11: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Pitch angle using two-layer MLP and 5-sample TW.

	Num. Neurons	**5**	**10**	**15**
TEST	**5**	0.054±0.005	0.057±0.010	0.055±0.006
RMSE	**10**		0.053±0.006	**0.045±0.011**
[rad]	**15**			0.050±0.005
TRAIN	**5**	0.044±0.011	0.041±0.007	0.046±0.008
RMSE	**10**		0.043±0.005	**0.052±0.009**
[rad]	**15**			0.050±0.004

Table A.12: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Pitch angle using two-layer MLP and 10-sample TW.

	Num. Neurons	**10**	**20**	**30**
TEST	**10**	**0.038±0.002**	0.040±0.004	0.048±0.014
RMSE	**20**		0.046±0.013	0.042±0.009
[rad]	**30**			0.051±0.019
TRAIN	**10**	**0.038±0.011**	0.039±0.011	0.037±0.005
RMSE	**20**		0.035±0.008	0.035±0.011
[rad]	**30**			0.032±0.006

Table A.13: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Roll angle using one-layer MLP and 5-sample TW.

	Num. Neurons					
	5	**10**	**15**	**20**	**25**	**30**
TEST RMSE [rad]	0.093±0.007	0.106±0.005	**0.060±0.002**	0.075±0.006	0.065±0.010	0.092±0.005
TRAIN RMSE [rad]	0.081±0.014	0.092±0.006	**0.061±0.007**	0.061±0.010	0.057±0.006	0.081±0.005

Table A.14: RMSE mean and deviation obtained in LShin baseline experiment to predictthe Roll angle using one-layer MLP and 10-sample TW.

	Num. Neurons					
	10	**20**	**30**	**40**	**50**	**60**
TEST RMSE [rad]	0.069±0.013	0.069±0.001	0.070±0.008	**0.068±0.004**	0.069±0.003	0.069±0.003
TRAIN RMSE [rad]	0.063±0.003	0.068±0.001	0.066±0.010	**0.064±0.006**	0.067±0.004	0.068±0.005

Table A.15: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Roll angle using two-layer MLP and 5-sample TW.

	Num. Neurons	**5**	**10**	**15**
TEST	**5**	**0.048±0.002**	0.048±0.005	0.048±0.009
RMSE	**10**		0.049±0.002	0.049±0.010
[rad]	**15**			0.050±0.008
TRAIN	**5**	**0.036±0.009**	0.040±0.001	0.035±0.006
RMSE	**10**		0.035±0.011	0.034±0.009
[rad]	**15**			0.033±0.010

Table A.16: RMSE mean and deviation obtained in LThigh baseline experiment to predict the Roll angle using two-layer MLP and 10-sample TW.

	Num. Neurons	**10**	**20**	**30**
TEST	**10**	**0.046±0.001**	0.047±0.003	0.046±0.002
RMSE	**20**		0.047±0.005	0.047±0.006
[rad]	**30**			0.046±0.008
TRAIN	**10**	**0.037±0.007**	0.036±0.007	0.035±0.004
RMSE	**20**		0.035±0.008	0.031±0.003
[rad]	**30**			0.032±0.004

A.1.2 Angle prediction in IMU LThigh

A.2 Optimization of the number and position of IMUs

A.2.1 3DLS IE experiment

Table A.17: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Pitch angle using one-layer MLP and 5-sample TW.

	Num. Neurons					
	5	**10**	**15**	**20**	**25**	**30**
TEST RMSE [rad]	0.121±0.014	0.124±0.015	0.120±0.015	0.120±0.021	0.122±0.017	**0.120±0.012**
TRAIN RMSE [rad]	0.119±0.013	0.119±0.014	0.117±0.015	0.116±0.020	0.118±0.014	**0.118±0.012**

Table A.18: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Pitch angle using one-layer MLP and 10-sample TW.

	Num. Neurons					
	10	**20**	**30**	**40**	**50**	**60**
TEST RMSE [rad]	**0.095±0.017**	0.099±0.017	0.105±0.017	0.117±0.016	0.123±0.012	0.131±0.015
TRAIN RMSE [rad]	**0.092±0.015**	0.096±0.015	0.101±0.015	0.115±0.012	0.121±0.014	0.126±0.014

Table A.19: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Pitch angle using two-layer MLP and 5-sample TW.

	Num. Neurons	**5**	**10**	**15**
TEST RMSE [rad]	**5**	0.051±0.008	0.051±0.006	0.054±0.008
	10		0.049±0.008	0.049±0.005
	15			**0.048±0.009**
TRAIN RMSE [rad]	**5**	0.045±0.006	0.045±0.006	0.046±0.007
	10		0.04±0.008	0.038±0.006
	15			**0.036±0.007**

Table A.20: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Pitch angle using two-layer MLP and 10-sample TW.

	Num. Neurons	**10**	**20**	**30**
TEST RMSE [rad]	**10**	0.046±0.005	0.051±0.014	0.048±0.010
	20		0.043±0.007	0.044±0.011
	30			**0.040±0.010**
TRAIN RMSE [rad]	**10**	0.035±0.008	0.038±0.010	0.037±0.006
	20		0.031±0.007	0.03±0.010
	30			**0.028±0.009**

Table A.21: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Roll angle using one-layer MLP and 5-sample TW.

	Num. Neurons					
	5	**10**	**15**	**20**	**25**	**30**
TEST RMSE [rad]	0.079±0.002	0.085±0.005	**0.072±0.008**	0.073±0.007	0.076±0.006	0.085±0.007
TRAIN RMSE [rad]	0.076±0.007	0.084±0.011	**0.070±0.009**	0.070±0.005	0.072±0.006	0.083±0.010

Table A.22: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Roll angle using one-layer MLP and 10-sample TW.

	Num. Neurons					
	10	**20**	**30**	**40**	**50**	**60**
TEST RMSE [rad]	**0.071±0.012**	0.071±0.023	0.144±0.119	0.096±0.042	0.081±0.009	0.083±0.024
TRAIN RMSE [rad]	**0.072±0.010**	0.070±0.021	0.138±0.113	0.037±0.040	0.081±0.012	0.084±0.023

Table A.23: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Roll angle using two-layer MLP and 5-sample TW.

	Num. Neurons	**5**	**10**	**15**
TEST RMSE [rad]	**5**	0.048±0.005	**0.047±0.004**	0.044±0.009
	10		0.049±0.005	0.053±0.011
	15			0.051±0.007
TRAIN RMSE [rad]	**5**	0.042±0.009	**0.041±0.003**	0.042±0.007
	10		0.035±0.010	0.034±0.006
	15			0.036±0.008

Table A.24: RMSE mean and deviation obtained in 3DLS IE experiment to predict the Roll angle using two-layer MLP and 10-sample TW.

	Num. Neurons	**10**	**20**	**30**
TEST RMSE [rad]	**10**	0.037±0.010	0.037±0.011	0.038±0.011
	20		0.033±0.007	**0.033±0.006**
	30			0.077±0.069
TRAIN RMSE [rad]	**10**	0.029±0.007	0.030±0.011	0.032±0.006
	20		0.024±0.004	**0.021±0.004**
	30			0.009±0.011

Table A.25: RMSE mean and deviation obtained in 2DLX IE experiment to predict the Pitch angle using two-layer MLP algorithm with 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Pitch RMSE [rad]	**10**	0.070±0.015	**0.068±0.018**	0.072±0.017
	20		0.075±0.018	0.078±0.013
	30			0.076±0.016
LThigh Pitch RMSE [rad]	**10**	0.051±0.011	0.053±0.012	0.053±0.011
	20		**0.049±0.011**	0.052±0.011
	30			0.050±0.012

Table A.26: RMSE mean and deviation obtained in 2DLX IE experiment to predict the Roll angle using two-layer MLP algorithm with 10 samples of TW.

	Number of neurons	10	20	30
LShin Roll RMSE [rad]	**10**	0.060±0.007	0.059±0.004	0.057±0.004
	20		**0.056±0.003**	0.057±0.005
	30			0.059±0.006
LThigh Roll RMSE [rad]	**10**	0.048±0.002	0.047±0.004	0.045±0.003
	20		0.045±0.004	0.047±0.004
	30			**0.043±0.002**

Table A.27: RMSE mean and deviation obtained in 1DLX IE experiment to predict the Pitch angle using two-layer MLP algorithm with 10 samples of TW. LShin and LThigh IMUs from RShin.

	Num. Neurons	10	20	30
LShin Pitch RMSE [rad]	**10**	0.091±0.002	0.090±0.008	0.088±0.008
	20		0.091±0.005	0.087±0.005
	30			**0.086±0.005**
LThigh Pitch RMSE [rad]	**10**	0.057±0.002	0.057±0.002	**0.056±0.003**
	20		0.059±0.003	0.056±0.003
	30			0.059±0.005

Table A.28: RMSE mean and deviation obtained in 1DLX IE experiment to predict the Roll angle using two-layer MLP algorithm with 10 samples of TW. LShin and LThigh IMUs from RShin.

	Num. Neurons	10	20	30
LShin Roll RMSE [rad]	**10**	0.076±0.006	0.076±0.006	0.075±0.006
	20		0.073±0.006	0.074±0.005
	30			**0.073±0.004**
LThigh Roll RMSE [rad]	**10**	0.051±0.002	0.051±0.003	0.052±0.004
	20		0.049±0.003	0.051±0.002
	30			**0.049±0.002**

Table A.29: RMSE mean and deviation obtained in 1DLX IE experiment to predict the Pitch angle using two-layer MLP algorithm with 10 samples of TW. LShin and LThigh IMUs from RThigh.

	Num. Neurons	10	20	30
LShin Pitch RMSE [rad]	**10**	0.102±0.015	0.087±0.018	0.089±0.011
	20		**0.079±0.008**	0.087±0.012
	30			0.096±0.014
LThigh Pitch RMSE [rad]	**10**	0.049±0.008	**0.044±0.009**	0.045±0.002
	20		0.048±0.007	0.045±0.006
	30			0.048±0.009

Table A.30: RMSE mean and deviation obtained in 1DLX IE experiment to predict the Roll angle using two-layer MLP algorithm with 10 samples of TW. LShin and LThigh IMUs from RThigh.

	Num. Neurons	**10**	**20**	**30**
LShin Roll RMSE [rad]	**10**	0.041±0.006	0.042±0.007	0.040±0.010
	20		0.042±0.008	**0.039±0.009**
	30			0.049±0.006
LThigh Roll RMSE [rad]	**10**	**0.026±0.003**	0.026±0.004	0.027±0.004
	20		0.029±0.003	0.033±0.006
	30			0.030±0.005

A.2.2 2DLX IE experiment

A.2.3 1DLX IE experiment

A.3 Performance of new volunteers

A.3.1 3DLS NV experiment

Table A.31: RMSE mean and deviation obtained in 3DLS NV experiment to predict Pitch and Roll angles of LShin using two-layers MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Pitch	**10**	0.140±0.008	0.124±0.006	0.141±0.010
	20		**0.123±0.011**	0.141±0.014
	30			0.133±0.017
LShin Roll	**10**	0.176±0.023	0.174±0.020	0.178±0.021
	20		0.174±0.018	**0.173±0.018**
	30			0.179±0.024

A.3.2 2DLX NV experiment

A.3.3 1DLX NV experiment

Table A.32: RMSE mean and deviation obtained in 2DLX NV experiment to predict Pitch angle of LShin and LThigh using two-layer MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Pitch RMSE [rad]	**10**	0.172±0.012	0.169±0.014	**0.166±0.011**
	20		0.170±0.016	0.176±0.013
	30			0.166±0.015
LThigh Pitch RMSE [rad]	**10**	0.153±0.011	0.147±0.010	0.160±0.012
	20		**0.144±0.012**	0.152±0.007
	30			0.145±0.016

Table A.33: RMSE mean and deviation obtained in 2DLX NV experiment to predict Roll angle of LShin and LThigh using two-layer MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Roll RMSE [rad]	**10**	0.194±0.014	0.194±0.019	0.196±0.017
	20		0.198±0.021	0.200±0.020
	30			**0.183±0.016**
LThigh Roll RMSE [rad]	**10**	0.112±0.007	0.109±0.003	0.117±0.010
	20		0.113±0.008	**0.110±0.008**
	30			0.115±0.012

Table A.34: RMSE mean and deviation obtained in 1DLX NV experiment to predict Pitch angle of LShin and LThigh from RShin data using MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Pitch RMSE [rad]	**10**	0.180±0.028	0.179±0.019	0.193±0.033
	20		**0.173±0.023**	0.192±0.038
	30			0.197±0.040
LThigh Pitch RMSE [rad]	**10**	0.133±0.021	**0.122±0.023**	0.131±0.022
	20		0.128±0.008	0.131±0.013
	30			0.137±0.006

Table A.35: RMSE mean and deviation obtained in 1DLX NV experiment to predict Roll angle of LShin and LThigh from RShin data using MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Roll RMSE [rad]	**10**	0.373±0.052	0.297±0.045	0.325±0.048
	20		0.279±0.044	**0.260±0.037**
	30			0.327±0.037
LThigh Roll RMSE [rad]	**10**	0.177±0.023	0.165±0.027	0.167±0.017
	20		0.171±0.032	**0.150±0.041**
	30			0.164±0.052

Table A.36: RMSE mean and deviation obtained in 1DLX NV experiment to predict Pitch angle of LShin and LThigh from RThigh data using MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Pitch RMSE [rad]	**10**	0.234±0.027	0.238±0.035	**0.224±0.019**
	20		0.238±0.052	0.248±0.039
	30			0.239±0.041
LThigh Pitch RMSE [rad]	**10**	0.154±0.033	0.155±0.027	0.154±0.036
	20		**0.151±0.018**	0.173±0.027
	30			0.162±0.016

Table A.37: RMSE mean and deviation obtained in 1DLX NV experiment to predict Roll angle of LShin and LThigh from RThigh data using MLP algorithm and 10 samples of TW.

	Num. Neurons	**10**	**20**	**30**
LShin Roll RMSE [rad]	**10**	0.188±0.025	0.180±0.014	**0.173±0.019**
	20		0.175±0.022	0.179±0.021
	30			0.179±0.031
LThigh Roll RMSE [rad]	**10**	**0.052±0.011**	0.057±0.009	0.060±0.008
	20		0.056±0.010	0.066±0.012
	30			0.059±0.006