Politecnico di Torino

Detection of the type of physical activity based on IMU sensors.

Master's Thesis in Biomedical Engineering - Biomechanics

Promoters:

Prof.ssa Laura Gastaldi

Prof. Olivier Bruls

Prof. Cédric Schwartz

Candidate:

Alessia Paolino

s309406

Academic year 2023-2024

Alla mia famiglia,
a voi dedico tutti i miei traguardi
passati e futuri.

Abstract

Chronic diseases, particularly those affecting mobility such as Parkinson's disease, present significant challenges in diagnosis and management due to their progressive nature. Understanding the correlation between physical activity (PA) and chronic disease progression is crucial for developing effective treatment strategies. Engaging in physical activities provides valuable insights into individuals' daily routines, health conditions, habits, and mental well-being. Continuous monitoring of motor activities is essential for early detection, tracking disease progression, and guiding therapeutic interventions. Traditional Optical Motion Capture (OMC) systems, considered the gold standard for motion analysis, offer high precision but are expensive, require complex setups, and are limited to controlled environments. These limitations constrain their applicability in real-world settings. Given these constraints, this thesis project explores the potential of Inertial Measurement Units (IMUs) as a more accessible, scalable, and practical alternative for Human Activity Recognition (HAR) and chronic disease management. IMUs, which include accelerometers and gyroscopes, are wearable sensors capable of accurately detecting and classifying human motions in natural environments over extended periods. This approach enables continuous, real-world data collection, which is crucial for enhancing the monitoring of chronic conditions and improving patient outcomes through timely interventions. The research develops and validates a HAR system based on IMU data, focusing on identifying different movement patterns using Delsys IMU sensors. The study investigates the relationship between daily physical activity and the progression of chronic diseases, with an emphasis on detecting activities related to these conditions. An experimental protocol was implemented on 7 healthy subjects to establish a baseline for detecting activities relevant to the intended clinical applications. This protocol also facilitated the simultaneous capture of motion data from both IMUs and EMG sensors, allowing for a comprehensive analysis of their performance. IMUs were

strategically placed on various body parts, such as the wrist, thigh, and pocket, to determine optimal sensor placement for activity detection while ensuring patient comfort.

Various machine learning models were evaluated for their ability to classify different physical activities, with the Multilayer Perceptron (MLP) neural network identified as the most effective model, offering a balance between accuracy and computational efficiency. The performance of the IMU-based system yielded satisfactory results, indicating that IMUs can achieve high accuracy levels for almost all placements.

Additionally, this thesis discusses the broader implications of using IMUs for chronic disease management. Continuous data collection enabled by these sensors provides a non-invasive method for tracking motor impairments' progression, allowing for more precise assessment of treatment efficacy. This capability is especially valuable for chronic conditions, where subtle changes in motor function may require adjustments in therapeutic strategies.

In conclusion, this research demonstrates that IMUs, combined with advanced machine learning algorithms, offer a viable, cost-effective alternative to traditional OMC systems for human motion analysis. The portability and scalability of IMUs make them particularly well-suited for continuous monitoring in real-world settings, presenting significant potential for early detection and management of chronic diseases. Future research will focus on refining the system for specific clinical applications and exploring integration with other wearable technologies to further enhance diagnostic capabilities.

The study highlights the importance of daily physical activity in managing chronic diseases and underscores the potential of sensor-based monitoring in improving health outcomes and extending longevity.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Human Activity Recognition

Human Activity Recognition (HAR) plays a critical role in understanding and interpreting human actions within specific time frames. It relies on the analysis of time-based data, which can be captured in various formats such as videos, images, or discrete measurements. This field finds wide-ranging applications across diverse sectors, including healthcare, security, and infrastructure management. Traditionally, sensor data for HAR has been collected using two main motion capture methods: camera-based computer vision systems and inertial sensor-based systems [1].

In computer vision systems, human activities are observed and recorded through cameras, but automating the recognition of these activities from image sequences poses significant challenges. Moreover, continuous camera surveillance raises privacy concerns and may not always be feasible, particularly in scenarios requiring uninterrupted monitoring. In the medical field, Optical Motion Capture (OMC) systems paired with force plates are widely regarded as the gold standard for measuring gait time series in patients. However, OMC systems come with several limitations, including high expenses, extensive data processing durations, and spatial restrictions. These disadvantages have led to a growing interest in wearable sensor technologies as an alternative solution.

In recent decades, wearable sensors have emerged as pivotal tools in response to the constraints of traditional gold standard method and technological advancements.

The compact design of wearable sensors allows for unhindered analysis, a notable difference from optoelectronic motion capture (OMC) systems, which, despite their precision, are hindered by high costs, lengthy data processing times, and spatial limitations. Consequently, wearable sensors are increasingly recognized as a promising alternative solution. Inertial sensors, such as accelerometers and gyroscopes, have gained interest due to advancements in semiconductor and Micro-Electro-Mechanical Systems (MEMS) technologies. When miniaturized, these sensors can be discreetly attached to or worn on the human body, providing a cost-effective and non-intrusive method for data collection.The information gathered from wearable sensors serves as valuable input for HAR systems, which utilize machine learning and pattern recognition techniques to analyze human activity patterns [2]. As Human Activity Recognition (HAR) technologies advance, the integration of wearable devices and non-intrusive sensors offers significant potential for gaining deeper insights into human behavior. Nonetheless, challenges remain, including ensuring user acceptance and determining optimal sensor placement during various daily activities. Furthermore, lightweight wireless sensor devices offer versatility and comfort, making them suitable for continuous monitoring during activities of daily living, including sleep. For elderly individuals, these devices can be instrumental in detecting alarm conditions triggered by unusual behaviors or changes in routine activities associated with psychomotor pathologies. In the realm of sports and fitness, such sensors enable tracking of exercise routines and repetitions, facilitating the monitoring of workout regimens and energy expenditure assessment. A notable advancement in sensor technology involves combining surface electromyography (sEMG) and accelerometer sensors into a single device. This integration enables the collection of comprehensive data on muscle activity, force, fatigue, direction, and acceleration. Such data are essential for evaluating sports performance, preventing injuries, facilitating rehabilitation, and monitoring human activity in various contexts [3].

The primary objective of this study is to collect data from subjects through the strategic placement of practical and non-intrusive sensors during routine daily activities. The overarching aim is to identify optimal positions for IMU sensors that minimize discomfort while addressing medical needs for individuals with chronic conditions and respecting privacy considerations. To assess the effectiveness of wearable sensors in detecting various physical activities and their potential for monitoring different health conditions, this study will develop and implement an algorithm capable of recognizing a range of easily performed and captured daily activities. This is particularly relevant for patients with chronic illnesses such as type 2 diabetes, Parkinson's disease, and Alzheimer's disease. The goal is to use these activities as key indicators for tracking disease progression and detecting disease advancement. Sensors will be strategically placed on different parts of the body to enhance detection accuracy. The primary aim is to create a monitoring system that is practical and comfortable for all patients, including those with chronic conditions, while also demonstrating and addressing the limitations of wearable technology compared to established gold standard techniques. Furthermore, this study aims to build a robust neural network for accurate activity classification, thereby improving patient compliance with sensor usage. By carefully selecting sensor positions that ensure minimal discomfort, the system enables patients to carry out daily activities without inconvenience.

The study will focus on a cohort of healthy young adults, aiming to identify and classify a variety of everyday activities. This approach ensures that the tested algorithm can be effectively applied to a broad spectrum of common daily activities. The study will focus on a cohort of healthy young adults aged 20 to 27 years, aiming to identify and classify a variety of everyday activities. This approach ensures that the tested algorithm can be effectively applied to a broad spectrum of common daily activities. A total of 7 volunteers, comprising both women and men, participated in the study. The participants' ages ranged from 20 to 27 years, with a mean age of 23.0 years and a standard deviation of ±2.31 years. Each participant wore three 9-axis Inertial Measurement Units (IMUs) positioned at different locations on the body.

1.2 Protocol definition

Formulating a protocol was essential to conduct an exhaustive analysis aimed at elucidating the influence and magnitude of daily activities on the advancement of chronical pathologies, which is the central starting point of this thesis. Given the wide spectrum of these pathological conditions, the absence of established protocols in the literature presents a challenge to conducting comprehensive analyses.

To rigorously define this protocol for instrumental measurement employing inertial sensors, literature review on various motion capture techniques was undertaken, aimed at delineating both the primary advantages associated with wearable sensors and recent studies involving the application of inertial sensor units (IMUs). This includes identifying optimal sensor placements, defining acquisition protocols, establishing experimental setups, examining investigated parameters, reporting results, and discussing associated limitations. Firstly, an exhaustive examination was conducted on diverse motion capture techniques, encompassing both traditional methods and the emerging use of wearables for gait assessment. The advantages of wearables, including IMU units, were underscored to elucidate their selection as the preferred mode of acquisition in this study.

Subsequently, the focus of the literature review shifted to identifying the optimal sensor placement, taking into account both common and precise locations. Central to the selection of placement was the consideration of physical activity (PA) in various chronic pathologies, with particular emphasis on type 2 diabetes, Alzheimer's disease (AD), and Parkinson's disease(PD). By analyzing established physical protocols for each condition, specific motor tasks were delineated and subsequently acquired. This process has lead the selection of the best sensor placement to acquire data during each activity.

Furthermore, the entire study followed the sensor placement protocol indicated in the reference text "Atlas of Muscle Innervation Zones: Understanding Surface Electromyography and Its Applications" by M. Barbero, R. Merletti, and A. Rainoldi. [4]. The identification of motor tasks was conducted through analysis of the impact of different physical exercises on a restricted spectrum of widely prevalent chronic pathologies, aimed at determining optimal movements and sensor inertial placements on the selected body segments, accompanied by the identification of a local reference system for each segment.

1.3 Algorithm of Human Activity Recognition

The algorithm of Human Activity Recognition involves four primary steps for comprehensive activity analysis, aiming to achieve satisfying results. Each step requires specific decisions aligned with the work's objectives.

Initially, sensors are carefully selected and placed, which is crucial for accurately detecting and analyzing various activities during data collection. Next, the acquired data undergo pre-processing to ensure they are clean and significant, removing noise. Following this, relevant features are selected from the data, which are essential for effective analysis. Finally, the appropriate Machine Learning algorithm is determined to accurately classify and recognize the activities. This brief overview outlines the main steps of the HAR process.[1].

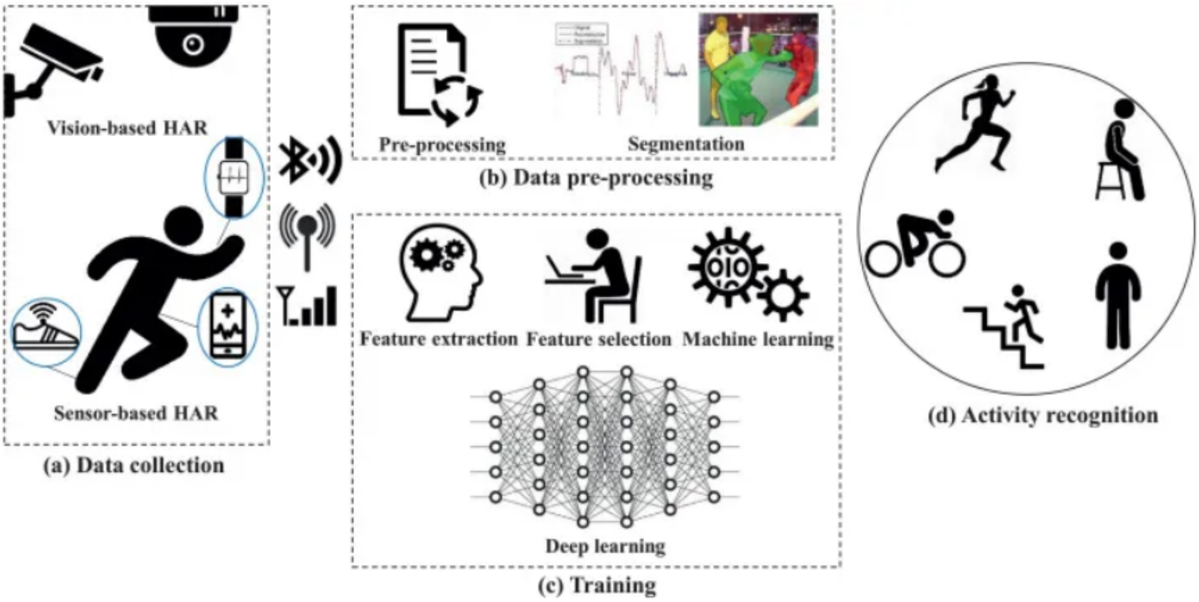


Figure 1: *HAR process* [5]

Based on the study of additional research conducted in the analysis of human movement [1],[2],[5],[6] and expanding on the groundwork laid out in the HAR process described earlier, during this study, in order to adhere to standard machine learning protocols, the following steps have been meticulously executed to develop an effective model for activity detection:

1. **Data Acquisition**: Firstly, the initial step involves selecting sensors that can accurately capture data relevant to the desired activities. Wearable devices offer flexibility in placement ,whether in a pocket, on the wrist, around the waist, and so on. These devices utilize accelerometers, magnetometers, and gyroscopes to gather readings across three axes: x, y, and z. Once the analysis protocol is defined, raw input signals are acquired from these sensors and organized into distinct datasets.

2. **Data Pre-processing**: Before analysis, the sensor output data require cleaning and processing to eliminate noise and artifacts. This includes post-processing accelerometer and gyroscope readings across x, y, and z directions, as well as EMG signals from wearable devices. Once the raw data is collected, pre-processing techniques such as noise removing, sampling, and windowing are applied. Subsequently, the dataset is divided into test, training, and evaluation sets, with careful consideration given to the ratio to effectively train and validate the machine learning model.

3. **Feature Extraction**: Based on the different activities and the different aims of the study,features are extracted manually or automatically, serving as inputs for training machine learning models. Trained models are used to infer human activities in real-world scenarios.

4. **Evalutation**: The evaluation of a machine learning model involves using performance metrics like accuracy, precision, recall, F1-score, and AUC-ROC. Cross-validation ensures the model's consistency and prevents overfitting , meanwhile the analysis of the confusion matrix reveals where the model makes errors. External validation tests the model on new data to ensure it generalizes well beyond the training set. These steps collectively assess the model's effectiveness and suitability for real-world applications.

2 Protocol definition

Formulating a protocol was essential to conduct an exhaustive analysis aimed at elucidating the influence and magnitude of daily activities on the advancement of chronical pathologies, which is the central starting point of this thesis. Given the wide spectrum of these pathological conditions, the absence of established protocols in the literature presents a challenge to conducting comprehensive analyses.

To rigorously define this protocol for instrumental measurement employing inertial sensors, a sequential identification of preliminary phases was undertaken:

1. **Literature review** on various motion capture techniques, aimed at delineating both the primary advantages associated with wearable sensors and recent studies involving the application of inertial sensor units (IMUs). This includes identifying optimal sensor placements, defining acquisition protocols, establishing experimental setups, examining investigated parameters, reporting results, and discussing associated limitations.

2. **Identification of motor tasks** through analysis of the impact of different physical exercises on a restricted spectrum of widely prevalent chronic pathologies, aimed at determining optimal movements and sensor inertial placements.

3. **Placement of sensors** on selected body segments, accompanied by the identification of a local reference system for each segment.

4. **Analysis of the results**, involving the examination of the obtained biomechanical parameters and the extraction the most significant features for

validation of the results.

2.1 Litterature review

The literature review was divided into two main sections. Firstly, an exhaustive examination was conducted on diverse motion capture techniques, encompassing both traditional methods and the emerging use of wearables for gait assessment. The advantages of wearables, including IMU units, were underscored to elucidate their selection as the preferred mode of acquisition in this study.

Subsequently, the focus of the literature review shifted to identifying the optimal sensor placement, taking into account both common and precise locations. Central to the selection of placement was the consideration of physical activity (PA) in various chronic pathologies, with particular emphasis on type 2 diabetes, Alzheimer's disease (AD), and Parkinson's disease(PD). By analyzing established physical protocols for each condition, specific motor tasks were delineated and subsequently acquired. This process has lead the selection of the best sensor placement to acquire data during each activity.

The primary source for the literature review was the *Scopus* database, which provides the ability to search for articles using various keywords. Upon entering the appropriate search string containing keywords related to the topic of interest, Scopus allows users to visualize articles or reviews. The search was confined to the fields of title, abstract, and keywords. These keywords were appropriately separated using boolean operators AND and OR within the search strings.

Through the use of the following keywords: IMUs, Motion Capture, and Gait Analysis, the first part of the literature review was fulfilled. Meanwhile, in order to find the perfect placement of the sensors, the following keywords were used: IMUs, Placement, and HAR, which, as already stated, stands for Human Activity Recognition.

Furthermore, the entire study followed the sensor placement protocol indicated in the reference text "Atlas of Muscle Innervation Zones: Understanding Surface Electromyography and Its Applications" by M. Barbero, R. Merletti, and A. Rainoldi. [4].

3 Motion Capture Systems

Gait analysis, which systematically studies the walking pattern, plays a crucial role in providing insights into an individual's functional level, identifying dysfunctions and injuries, evaluating the effectiveness of rehabilitation, assessing the success of surgeries and also monitoring pathological diseases.

Traditional methods for collecting sensor data for HAR encompass camera-based computer vision systems and inertial sensor-based systems. In the former, human activities are captured via cameras, and the challenge lies in automatically recognizing these activities based on sequences of images. However, limitations arise in scenarios necessitating continuous activity monitoring, where complete camera coverage may not be feasible. Moreover, privacy concerns arise as continuous camera surveillance can be intrusive and discomforting for individuals.

Methods for assessing the gait pattern can be broadly categorized into kinematics and kinetics. Kinematic analysis focuses on movement patterns, specifically spatio-temporal variables, without considering the forces involved in motion production. On the other hand, kinetic analysis is concerned with determining the joint moments and forces, including the ground reaction force (GRF), involved in walking.

As the acquisition of gait kinetics conventionally requires an equipped environment with force plates, there have been recurring efforts to develop alternative strategies in this field. One common approach is the utilization of kinematic motion data to estimate the internal forces experienced by body segments through inverse dynamics. Joint moments are crucial outcomes of motion analysis and are often used to extract relevant information for clinical decision-making. Conventionally, joint moments can be calculated using an inverse-dynamics method, with the measured GRFs serving as input data. However, obtaining proper kinetic data in motion analysis studies can be challenging due to the limitations of force plates, which are typically restricted to laboratory environments. Efforts to address these challenges include exploring methods to estimate kinetic parameters from kinematic data alone, leveraging machine learning algorithms, and incorporating wearable sensors such as IMUs. These approaches aim to enhance the accessibility and feasibility of gait analysis outside of traditional laboratory settings, enabling more comprehensive and continuous monitoring of human movement patterns in real-world environments.

While kinematic and kinetic analyses are frequently performed separately, they can also be seamlessly integrated into a unified evaluation. The gold standard method often employs an optoelectronic system with skin markers for this purpose. Additionally, force plates and Inertial Measurement Units (IMUs) are commonly utilized to gather data from human subjects, offering complementary insights into biomechanical parameters.

3.1 Golden Standard: Optical Motion Capture

In the medical field, Optical Motion Capture (OMC) systems combined with force plates have long been considered the gold standard for accurately quantifying patients' motion capture [2]. Video-based optoelectronic systems rely on devices capable of sourcing, detecting, and controlling light, which fall under the photonics spectrum. The experimental setup typically involves retro-reflective markers visible to multiple video cameras, offering flexibility in capturing images of various body regions.

Existing motion capture systems are expensive, bulky, and require extensive training for proper operation, which is not always feasible for every patient. For three-dimensional (3D) movement capture, these systems necessitate at least three to six precision cameras as shown in Figure 2, typically installed in specialized institutions provided with gait laboratories. This setup poses practical limitations, including complexity and time-consuming operation. Additionally, motion capture systems face constraints due to line-of-sight difficulties, leading to potential data loss [7].

An alternative to traditional motion capture systems would be a marker-less tracking system that is surely less expensive and does not require markers on participants. Therefore, the development of a precise, accurate, cost-effective, and user-friendly marker-less tracking system holds significant potential.

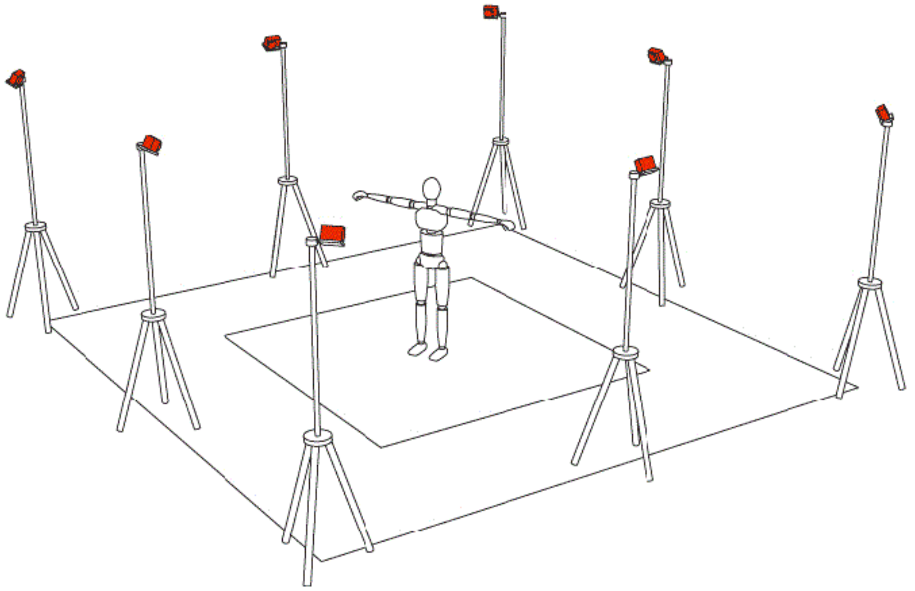


Figure 2: *Graphical representation of the experimental set-up of optoelectronic system and kinetic (force platform) motion analysis system.*[8]

3.2 Wearables for gait assessment

3.2.1 Accelerometers

According to Newton's second law, an object with constant mass [kg] accelerates (m/s^2) in proportion to the sum of the applied net force [N]. Accelerometers are developed from this principle using various approaches (e.g., piezoelectric, thermal, and capacitive). They are highly configurable devices, where their bandwidth or frequency response can be adjusted through coupling filter capacitors. This aspect is crucial for accurate sensing, as the bandwidth must encompass the frequency or vibration of interest. Range (g = 9.81 m/s^2) and sampling frequency (fs, Hertz[Hz]) are additional parameters of interest that need consideration based on the type of activity to be measured.

The dynamic range of an accelerometer is the ± maximum amplitude that can be measured before distorting the output signal during data collection. Low-intensity movements (e.g., postural balance) are assessed more sensitively with lower g values. Conversely, high-insensitivity movements (e.g., gait) are accurately assessed with higher g values to capture high-amplitude (range) movements without distortion or clipping.

Most accelerometer-based wearables offer selectable ranges to work within the optimal range, which depends on both the type of movement and the body part involved. For instance, 3D linear accelerations recorded at joints can range from $3.0g$ to $12.0g$, while lower back vertical acceleration and horizontal acceleration can range from $-0.3g$ to $0.8g$ and from $-0.3g$ to $0.4g$, respectively. Thus, accelerometers must be capable of measuring accelerations up to ±12g regardless of the attachment location but with enough resolution to capture subtle (low g) movements [7].

3.2.2 Gyroscopes

Gyroscopes measure angular velocity [°/s] and are highly utilized as inertial sensors, following closely behind accelerometers in popularity. During deployment, it's crucial to take into account scale factor stability, which indicates the sensitivity of the optical gyroscope. Achieving minimum scale factor stability is essential to minimize sensor errors. This stability can be quantified using the angle random walk (ARW) formula:

$$ARW = \frac{R}{60\sqrt{B}}$$

where R represents resolution and B represents bandwidth.

While combining a tri-axial accelerometer with a tri-axial gyroscope allows for the determination of relative heading/direction, it's important to note that the output may drift over time. This phenomenon highlights the need for ongoing monitoring and calibration to maintain accuracy in long-term applications [7].

3.2.3 Magnetometers

Magnetometers are devices that measure the direction, strength, and changes of a magnetic field (measured in Gauss) at a specific location. They are particularly sensitive to Earth's magnetic field and can be utilized to correct drift or detect rotations in a known direction. In scenarios where magnetometers are absent, using 6 axes (three axes each of accelerometer and gyroscope) can provide relative heading, although it may exhibit some degree of drift.

Integrating magnetometers into the system can mitigate drift by offering an absolute heading reference point based on the Earth's magnetic field. It's important to note that magnetometers may be influenced by localized magnetic fields, which can vary in uncontrolled environments such as free-living conditions. Accelerometers are adept at recognizing numerous daily activities, while gyroscopes notably improve fall detection accuracy. Integrating accelerometer and gyroscope data enhances the reliability of activity recognition processes. Otherwise the magnetometer's reliance on directions often leads to overfitting in training classifiers, rendering its use ineffective for activity recognition, as indicated by studies [7].

3.2.4 Electromyography (EMG)

EMG sensors record myoelectric signals, generated by motor neurons, using different electrode types: needle or surface.

Needle (fine wire) electrodes are directly inserted into the muscle, while surface electrodes are placed on the skin. An explanatory representation in Figure 3

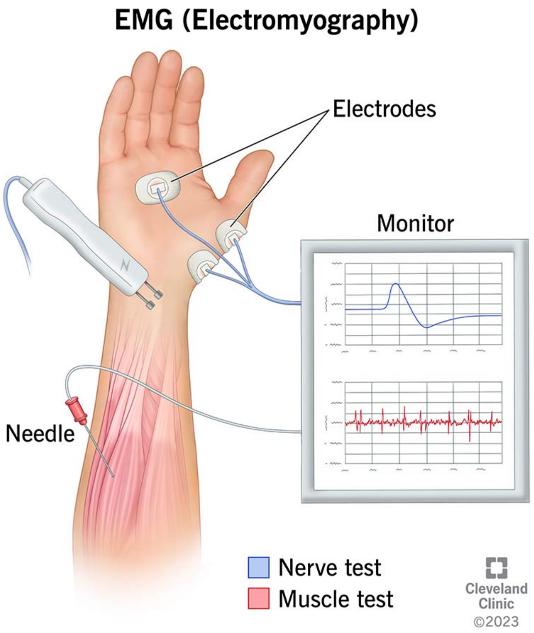


Figure 3: *Different types of EMG's electrodes: on the hand surface electrodes and on the arm needle ones.* [9]

Although the former offer more precise outcomes, their invasive nature limits their practicality. Surface EMG electrodes (sEMG), including wireless option used in this project, provide a more convenient setup for recording muscle activities in both clinical and free-living settings.

Myoelectric signals typically range from 10 to 1000 Hz and are generally measured in millivolts [mV] : muscle contractions produce signals around 10 Hz due to tissue displacement, while ground impact during walking generates signals at 25–30 Hz. However, EMG signals have low signal reception and are susceptible to unwanted noise, mainly from tissue motion and neighboring motors. Nonetheless, these noise sources are identified and mitigated during both signal acquisition and post-processing stages. During signal acquisition, unwanted electronic signals, such as common mode noise, can be minimized using differential amplifiers or instrumentation amplifiers (IA) with a high common mode rejection ratio (CMRR).

Also during acquisition the optimal protocol provides the perfect sensor placement in order to avoid meaningless acquisition. Additionally, post-processing techniques, such as digital low-pass, high-pass, or band-pass filters, are applied to reduce noise while considering the sEMG frequency spectrum [7].

In this project experimental project all the measures adopted to mitigate noise influences are thoroughly explained in the following chapters .

3.2.5 IMUs

The concept of the Inertial Measurement Unit (IMU) emerged in the 1970s. Utilizing independent inertial devices such as gyroscopes and accelerometers, it became possible to separately measure angular velocity, acceleration, and other parameters. By integrating the outputs of these devices, the position of an object could be determined through integration. A typical IMU comprises three orthogonal gyroscopes, three orthogonal accelerometers, and a computer for coordinate conversion, enabling the measurement of attitude, position, and velocity information of a carrier.

Over time, IMUs have continued to shrink in size. Advancements in micro and nanotechnologies have paved the way for the development of micro inertial measurement units (MIMUs). MIMUs integrate multiple micro inertial sensors, micro monitoring and control circuits, and a coordinate conversion circuit to obtain comprehensive inertial measurement parameters of moving objects. The basic structure of an MIMU, shown in Figure 4, typically consists of three micro-gyroscopes and three micro-accelerometers, arranged in orthogonal surfaces of a cube. The sensitive axes x, y, and z are perpendicular to each other, allowing the measurement of ω_x, ω_y, ω_z as the outputs of micro-gyroscopes and a_x, a_y, a_z. as the outputs of micro-accelerometers along each axis [10].

One notable advantage of MIMUs in human movement analysis is their self-containment during operation. Accelerometers and gyroscopes can collect quantitative motion data irrespective of time and environment. Additionally, the omnipresence of a magnetic field on Earth enables the use of magnetometers in most locations. Furthermore, commercially available MIMUs are compact, lightweight, and can be equipped with additional hardware such as Bluetooth, Wi-Fi, or SD card capabilities. This allows for features like wireless data transmission or internal memory recording, facilitating easy data collection without interfering with natural human movement [10].

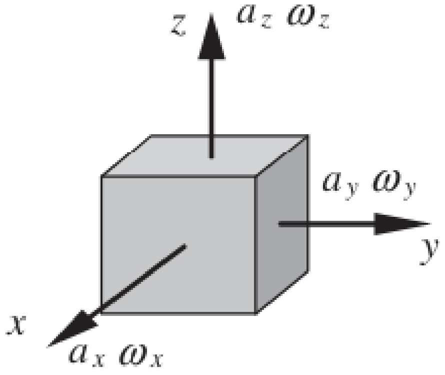


Figure 4: *Basic structure diagram of an MIMU.*[10]

In table 1 I have collected the main features that characterize the sensors used in this project, the IMUs,and the gold standard techniques previously mentioned. This comparison between Optical Motion Capture (OMC) and IMU sensors aims to highlight the primary advantages and disadvantages of each analysis technique.

Optoelettronic systems	**IMUs sensors**
High resolution	Discrete resolution
Accuracy golden standard	Discrete accuracy
High cost	Cost reduction
High accuracy	Discrete accuracy
Power supply	Power efficiency
Motion Lab constrained	Flexibility in deployment
Intrusive monitoring	Non-intrusive monitoring
Bulky	Lightweight
Line-of-sight difficulties	Unlimited field of use

Table 1: Comparison between Optoelectronic Motion Capture systems and IMUs sensors

The main features I want to focus on, as they are crucial for the study's objective, include the lightweight nature of IMUs, which makes them easy to wear even for patients with chronic conditions. Unlike optical motion capture (OMC) systems that require complex setup and a laboratory environment, IMU sensors offer reduced costs and an unlimited field of use, demonstrating their versatility. In this study, this allowed the activity acquisition's process without being limited to a laboratory setting, thus emulating daily activities in a real-world context.

Moreover lightweight and wireless devices can be comfortably worn during daily life activities (ADL), including sleep, for monitoring purposes. For the elderly, these devices could be used to detect alarm conditions triggered by unusual behaviors (such as remaining in bed or lack of activity during a defined time interval) or changes in routine activities associated with psychomotor conditions.In this study, sleep monitoring was excluded because, within the context of chronic diseases, biomechanical parameters related to sleep provide less informative data compared to the acquisition of more dynamic activities of the subject. Nonetheless, it is crucial to emphasize the value of these devices in applications beyond the specific focus of this study, as they hold significant potential for monitoring activities outside the scope of this research.

For healthy individuals, they could be employed during sports activities to track exercise routines and repetitions, facilitating the monitoring of training regimens and determining the energy expenditure for each movement. Mobile fitness coaching has embraced various topics, from assessing the quality of sports actions to detecting specific sports activities. Combining surface electromyography (sEMG) and accelerometric sensors in a single device allows for obtaining all the necessary information to accurately assess muscle activity, strength, fatigue, directionality, and acceleration. These aspects are crucial for evaluating sports performance, preventing injuries, facilitating rehabilitation, and monitoring human activity in general.

4 Data Aquisition

4.1 Experimental Setup

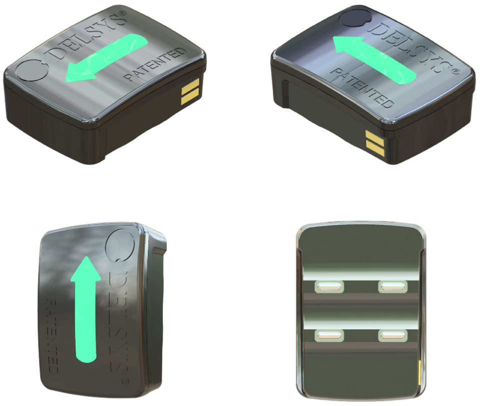
The experimental setup consisted of three Trigno Avanti™ Sensors, 9-axial inertial measurement unit (IMU) sensors with accelerometer, gyroscope and magnetometer in recording mode, paired with EMG electrodes.

The signals can be adjusted across four different ranges for each sensor, covering a spectrum from ±2g to ±16g for accelerometer data and ±250°/s to ±2000°/s for gyroscope data.

With nine channels of data, the sensor can accurately estimate orientation in three-dimensional space. The Trigno wireless communication system ensures reliable data transmission for up to 16 sensors, maintaining connectivity over distances of approximately 20 meters. Sampling frequency for accelerometer and gyroscope was 133.33Hz. To obtain the synchronization between three IMUs all data were timestamped. The sensors feature a rechargeable lithium polymer battery, providing multiple hours of continuous use. The actual duration will vary depending on usage conditions, typically ranging between 4 to 8 hours of performance. In this project the sensors recharging was made with the 4 based station showed in the Figure 7. The main capabilities of the Trigno Avanti™ Sensor are listed in table 2 below:

Capability	**Description**
Onboard configurable precision EMG sensor	EMG electrodes integrated on the back of the sensor unit
Built-in 9-axis inertial measurement unit (IMU)	Each unit includes an accelerometer, gyroscope, and magnetometer in recording mode
Dual-mode ”BLE-Base” communication	Dual-mode communication with the base system
Onboard RMS and Mean calculations	Integrated capability to calculate RMS and Mean values
Onboard orientation calculation	Capability to calculate the 3D orientation of each unit
Onboard median frequency calculation	Capability to output the mean frequency
Software selectable operational modes	Several operational modes can be selected based on the acquired activity
Inter-sensor latency < 1 sample period	Latency period between sensor synchronizations
Wireless transmission range +20 m	Limit of wireless connection range to the station
Self-contained rechargeable battery	Power supply mode
Environmentally sealed enclosure	Environmentally sealed casing
Low power mode	Battery low consumption mode
Internal magnetic switch	Integrated magnetometer unit
LED User Feedback	Status of the units represented by LED indicators

Table 2: Capabilities of Trigno Avanti™ Sensor



Mode Name	Channels	Sample Rate
EMG + IMU	EMG	1000Hz
	ACC_x ACC_y ACC_z	133.3Hz
	$GYRO_x$ $GYRO_y$ $GYRO_z$	133.3Hz
	MAG_x MAG_y MAG_z	66.7Hz

Figure 5: *Sensor view from different point and sensor modes with the specified sampling frequencies.* [11]

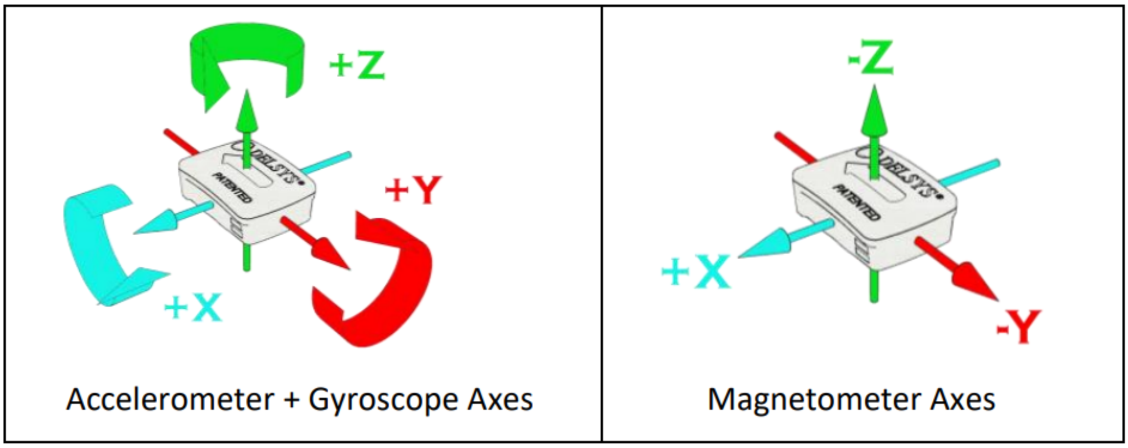


Figure 6: *Orientation axes of the Trigno Avanti™ sensors'unit.* [11]

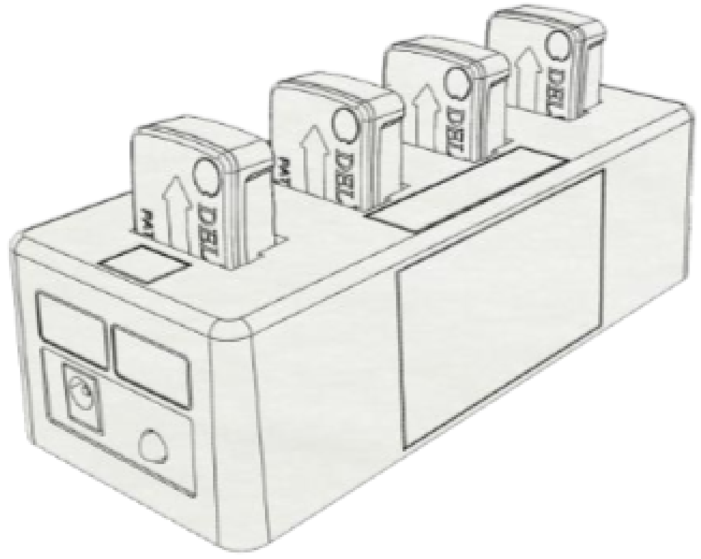


Figure 7: *Trigno System "Charge-4" Station for sensor recharging.Orientation axes of the Trigno Avanti™ sensors'unit.* [11]

As shown in Figure 6, the Trigno Avanti™ sensors have predetermined axis directions that were taken into account during the acquisition process, in order to evaluate the sign of the oscillation of the accelerations acquired. This information could be effectively coupled with the ones from Trigno EMG Sensors to detect synchronized movements based on signals detected by the Electromyography (EMG) system.

Trigno EMG Sensors are equipped with four silver bar contacts designed to detect EMG signals at the skin surface.The sampling frequency of the EMG signal is 1000Hz, and the mobile device allows the choice of two different EMG bandwidths: 10-850Hz or 20-450Hz. To ensure optimal signal amplitude, it's crucial to position these bars perpendicular to the direction of muscle fibers. The top of the sensor features an arrow, aiding in determining this orientation. Aligning the arrow parallel to the muscle fibers, as shown in Figure 8, beneath the sensor ensures proper placement.

Moreover, it's essential to position the sensor at the center of the muscle belly, avoiding placement near tendons or the muscle's edge, in order to acquire the optimal signal amplitude avoiding the muscles' part which not contribute to the contraction of the fibres. This positioning enhances signal accuracy and minimizes interference. To ensure secure and reliable placement during movement analysis, the sensors are attached to the skin using Delsys Adhesive Sensor Interface.

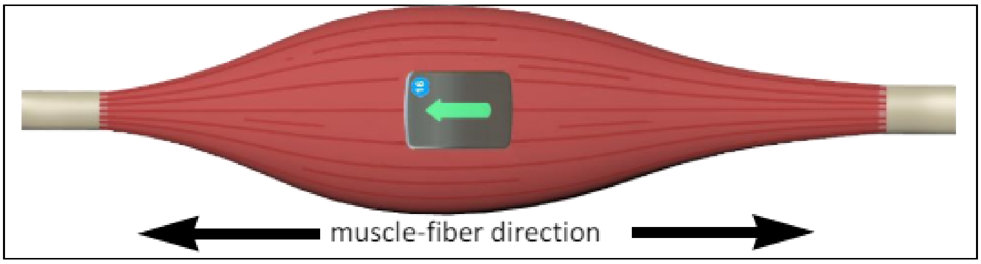
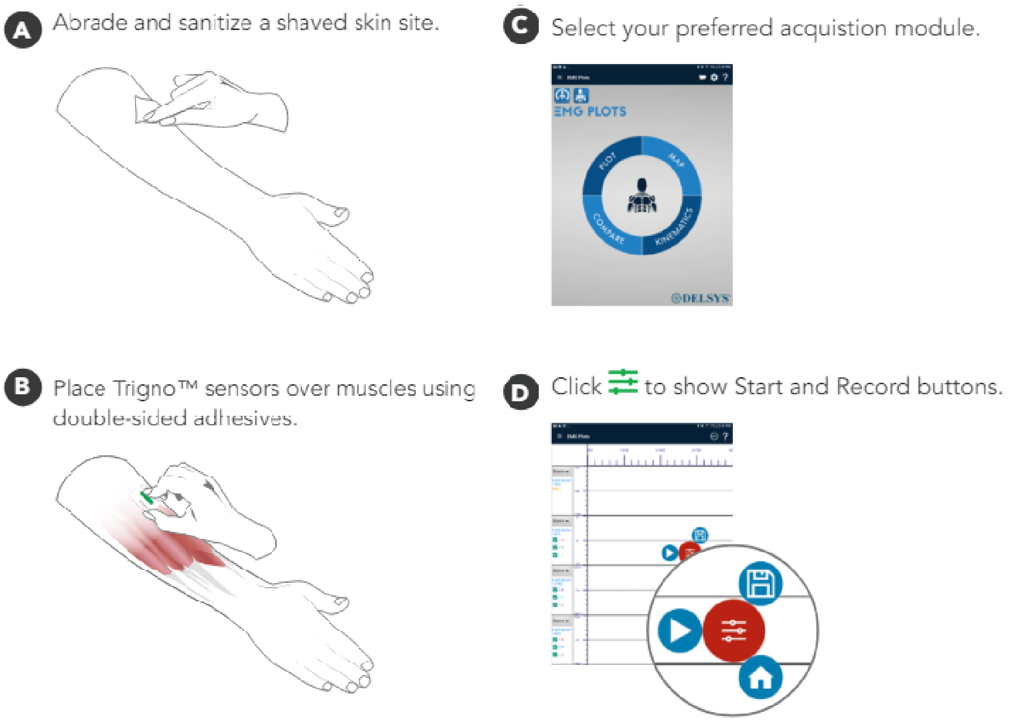


Figure 8: *EMG Sensors must be properly oriented with the muscle fibers. Align the sensor's arrow with the direction of the underlying muscle fibers.* [11]

4.2 Measurement procedure

Data were collected using a mobile tablet to facilitate acquisition both indoors and outdoors during various activities. The Trigno Mobile System App was employed for data collection, which was subsequently exported to a notebook computer for further analysis. As previously mentioned, to maintain signal amplitude and reduce noise, the sensors were secured using an adhesive interface to ensure a stable contact with the skin of the chosen human segments. Additionally, the acquisition procedure required pre-shaving and sanitizing the skin site before placing the Trigno™ sensors over muscles using double-sided adhesives. Through the use of the Trigno Mobile System App interface, it is possible to pursue different acquisition modules and directly export the data in the desired format.


A Abrade and sanitize a shaved skin site.

C Select your preferred acquistion module.

B Place Trigno™ sensors over muscles using double-sided adhesives.

D Click ☰ to show Start and Record buttons.

Figure 9: *Experiment Set-up and record Data with Mobile EMG Suite. The steps from A to D show how to achieve high quality surface EMG signals*

4.2.1 Choice of the motor tasks

The selection of the motor tasks, as already explained, was guided by an in-depth exploration of previous studies [12], [13] regarding the role of physical exercise (PE) in the progression of chronic diseases, such as Parkinson's disease (PD), Alzheimer's disease (AD), and diabetes, which are among the most prevalent .

Recent research reveals that there is currently no definitive consensus on the optimal physical exercise regimen for improving clinical outcomes in Alzheimer's disease patients. Nevertheless, aerobic physical exercise has emerged as a viable and beneficial practice for individuals with AD, consistently associated with improved cognitive function. Simple activities like regular walking have shown promise in mitigating cognitive decline, while also enhancing postural and motor functions in AD patients, who frequently experience diminished muscle mass and strength.

Moreover, balance training has proven effective in enhancing postural abilities, thereby reducing the risk of falls, particularly in moderately-to-severely affected individuals. Multicomponent exercise training, which integrates balance, aerobic, and strength exercises, has demonstrated notable efficacy in enhancing postural and motor functioning while reducing fall risk in AD patients. High-intensity physical activities like cycling, aerobics, or tennis are linked to a 40% lower risk of disease compared to low-intensity activities such as walking or dancing [12], showing substantial enhancements in balance, mobility, and functional capacities in Parkinson's disease patients. Additionally, structured walking programs, involving 30 minutes of walking five times a week for at least six weeks, show significant improvements have shown substantial enhancements in balance, mobility, and functional capacities. These findings emphasize the importance of intensity and specific exercise types in enhancing overall health outcomes [12].

The subjects had to perform different activities, which were mainly divided in 3 different groups:

- **Aerobic training**: continuous, rhythmic movement of large muscle groups, such as in walking, jogging, and cycling;
- **Resistence training**: synonymous with strength training, resistance exercise involves movements utilizing free weights, weight machines, body weight exercises, or elastic resistance bands;
- **Balance training**: exercises designed to improve stability and equilibrium by challenging the body's balance control systems, reducing fall risk, and enhancing overall balance;

In order to combine the aforementioned task with the need to characterize daily life activities, that as previously mentioned is one of the main aims of this project, the following activities were chosen :

- **Mixed walking**: Walking without a constant speed and without a fixed direction. To simulate daily life walking, which lacks constraints, the subjects were able to change direction, turn, or simulate avoiding obstacles during the acquisitions and the aquisition were taken both indoor and outdoor.
- **Organizing things**: The acquisition required subjects to be seated, thus a sitting acquisition was performed while engaging in daily life activities, such as scrolling on mobile screens or organizing things on the desk. The main joint

involved was the wrist, so activities were performed on both the dominant and non-dominant sides to see any correlation.

- **Climbing stairs**: The activity was performed both with and without the handrail to observe any influence on the data. Climbing stairs can be considered an activity that encompasses all three main groups mentioned above.
- **Descending stairs**: This activity was chosen to evaluate how well the machine learning algorithm could distinguish it from the previous one and to assess the patient's balance and stability. Moreover, descending stairs, like climbing, is a useful daily life activity.

As previously mentioned, the acquisitions were carried out both outdoors and indoors, with an average duration of 30 seconds for each acquisition, and each task was performed 3 times, resulting in 12 acquisitions per subject. Before proceeding with a new acquisition, a 60-second break is observed during which the subject is free to move in order to limit fatigue.

4.2.2 Sensors' placement

The methodology for conducting motion analysis relies on protocols, which I designed to ensure test repeatability and operational standardization during data acquisition. Firstly, the anatomical reference system is established for each human segment where sensors are placed, as detailed in the following section 4.3, and sensor axis orientation relative to the corresponding anatomical segments is determined. In order to define the local anatomical system in the right way the sensors were carefully placed on the relevant body segments according to predetermined locations. Subsequently, a static trial was conducted, during which participants were instructed to maintain an upright position for a few seconds. This step is essential for system calibration and ensures accurate data collection.

Finally, participants performed a dynamic trial involving the execution of the specific motor tasks aimed at evaluating various parameters. This comprehensive approach facilitates an in-depth analysis of motion patterns, offering significant insights into the dynamics of human movement. Special emphasis was placed on meticulous sensor placement, aiming to align the sensor axis parallel to the anatomical axis of the segment under examination.

All the volunteers wore three 9-axis Inertial Measurement Units (IMUs) positioned at different locations on the body. These positions were selected to acquire useful information based on the chosen motor tasks. The following sensor placements were established:

1. **Wrist**;

2. **Thigh**;

3. **Pocket**;

In the first configuration, sensors were placed on the wrist using a bracelet-like setup. In the second configuration, sensors were positioned on the thigh to capture both the IMUs parameters and the EMG signal of the quadriceps during activities. The third configuration involved placing the sensors in a pocket without any constraints. The sensor placement is illustrated in Figure 10, which is directly extracted from the Trigno Avanti App, providing a useful tool for monitoring the data recorded by the sensors.

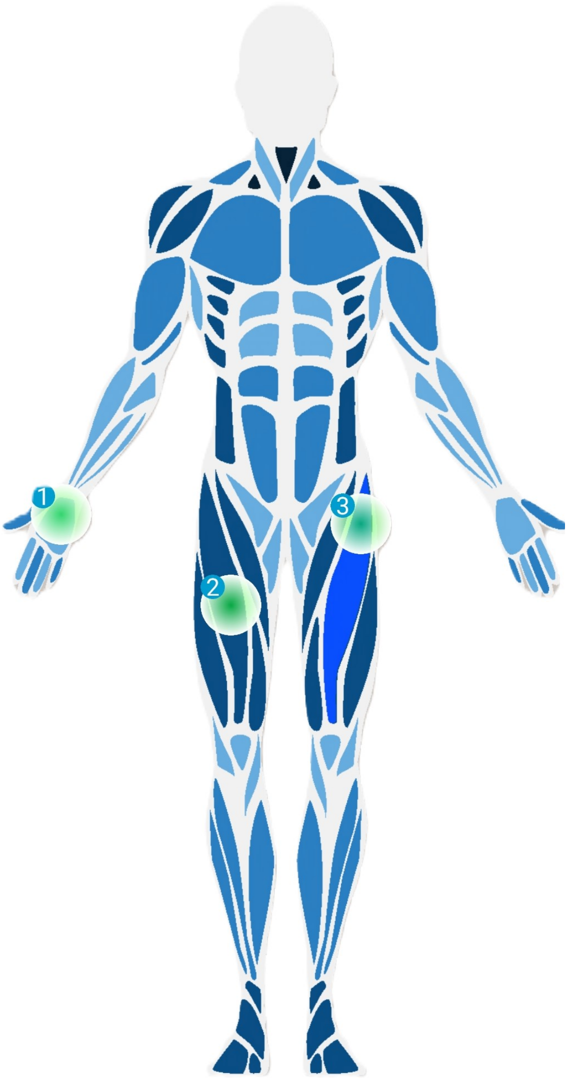


Figure 10: *IMU sensors' placement on the participant's body.*

The EMG signal was acquired to discriminate difference of intensity between the selected activities. The aim was to examine the same tasks and daily activities with increasing precision of acquisition, starting from the simplest setup, the pocket, which does not require any physical constraint, and ending with the most complete and noiseless acquisition,the quadiceps placement, involving both IMUs and EMG sensors.

sEMG attached to lower limb muscles provides reliable muscle activity and force information for gait assessment in neurological conditions. Specifically, activities of 28 major muscles controlling each lower limb can be readily identified. Ideally, lower leg and foot muscles such as the gastrocnemius medialis, gastrocnemius lateralis, soleus, tibialis anterior, and peroneus longus-brevis are suitable for sensor placement, with the reference electrode located at the ankle. Following SENIAM recommendations, the tibialis anterior, lateral gastrocnemius, and rectus femoris muscles were selected to collect EMG parameters for gait assessment in Parkinson's disease (PD) [7].

The decision to focus on the quadriceps muscle for monitoring muscle activation and physical activity intensity was based on previous research showing the significant involvement and activation of both the gastrocnemius and quadriceps muscle groups during typical daily activities such as walking, running, and stair climbing. It is conceivable that the relationship between muscle characteristics and gait performance varies across different lower extremity joints. Specifically, the strength of knee muscles seems to have a more pronounced impact at slower speeds, while the strength of ankle muscles becomes critical at maximal speeds. The quadriceps muscle primarily contributes to the braking phase, whereas the gastrocnemius is predominantly involved in the propulsion phase[14],[15]. Additionally, I chose to focus on the quadriceps muscle due to the more comfortable placement of the sensor. The quadriceps muscles are the strongest in the human body. Literally quadriceps means ”four-headed muscle” from Latin, in fact the quacripes comprises four distinct muscles : *the rectus femoris, vastus medialis, vastus intermedius, and vastus lateralis* as showed in Figure 11.

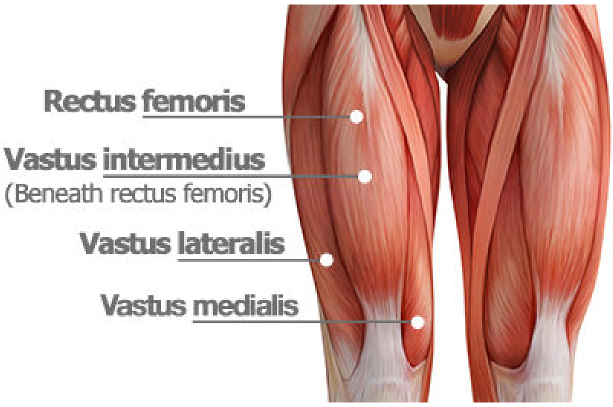


Figure 11: *Anatomy of the quadriceps muscles.* [16]

In this project the focus was centred on the activity of rectus femoris, which stands out for crossing both the hip and knee joints. It is a fusiform muscle, characterized by its spindle-like shape, and is composed of two heads. Originating from two sites on the ilium—the anterior inferior iliac spine (straight head) and the supraacetabular groove (reflected head)—these heads converge into a single muscle belly. This belly runs vertically down the thigh, covering its anterior aspect.

The muscle fibers of the rectus femoris come together to form a thick tendon that inserts into the base of the patella, contributing to the overall function of the quadriceps. Occasionally, the rectus femoris may exhibit a third head, originating from the iliofemoral ligament, adding variability to its structure.

For EMG placement, adherence to guidelines established by SENIAM (Surface EMG for Non-Invasive Assessment of Muscles) was ensured. These guidelines encompass various aspects, including identification of electrode types, skin preparation, patient positioning, determination of electrode locations, and electrode fixation [17]. Furthermore, additional recommendations for sEMG placement provided in a previous review conducted by M. Barbero, R. Merletti, and A. Rainoldi [4] were followed. These guidelines were instrumental in avoiding errors such as interference from soft tissue or inappropriate muscle selection, which could compromise the collection of meaningful data during sEMG measurements. The best placement of the sensor, in order to acquire an optimal EMG signal, was between 0% and 50% or between 83% and 100% of the Anatomical landmark frames (ALF), as showed in Figure 12.

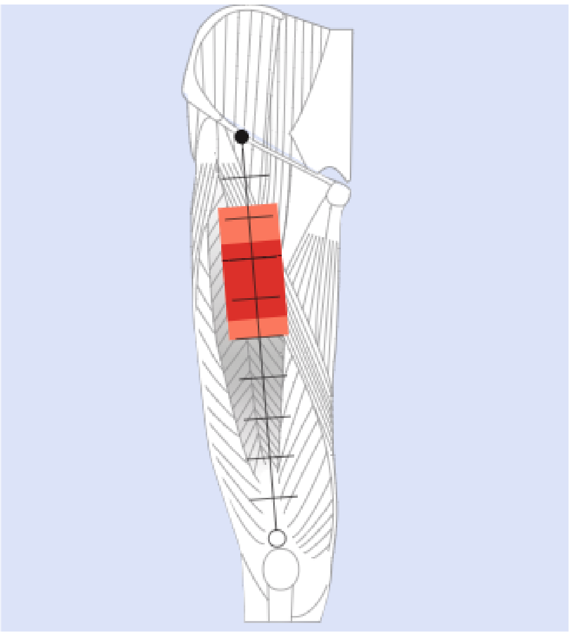


Figure 12: *Optimal EMG sensors' placement between the superior side of the patella and the anterior superior iliac spine.* [4]

The wrist placement involved both the dominant and non-dominant sides to investigate potential correlations between them. These locations were chosen in activity recognition studies for sensor placement due to their representativeness of typical motion patterns, crucial for correlating physical exercise (PE) with chronic pathologies. Additionally, these placements were chosen to mitigate potential acquisition issues stemming from health concerns associated with the analyzed pathologies. Furthermore, as previously mentioned, the IMUs placements were selected with the idea to have positions feasible for daily life acquisition, ensuring comfort for the patient and minimizing privacy or discomfort problems.

4.3 Definition of the anatomical reference system for body segments

To define an anatomical reference system for each sensor fixed to the analyzed segment, particularly those placed on the wrist and thigh, the definitions proposed by the Standardization and Terminology Committee (STC) of the International Society of Biomechanics were followed [18] [19]. The STC proposes a definition of a joint coordinate system (JCS) for each joint, establishing a standard for the local axis system in each articulating segment or bone. These recommendations by the STC aim to facilitate further revisions and obtain more reliable feedback. For the sensor placed in the pocket, defining a fixed reference system is impossible due to the sensor's lack of stability. Consequently, additional measurements were considered during the feature analysis phase, as reported in the section 5.1 later.

4.3.1 Wrist - Carpal bones coordinate system

The global wrist motion typically involves the movement of the second and/or third metacarpal in relation to the radius, with emphasis placed on the third metacarpal. This motion is achieved through the movement of the carpal bones in relation to the radius and the articulations of the eight carpal bones among themselves [Figure 14]. Each bone is assigned a coordinate system, assuming the forearm is initially in the standard anatomical position, with the palm facing forward (anteriorly) and the thumb positioned laterally. The back of the hand and forearm face posteriorly.

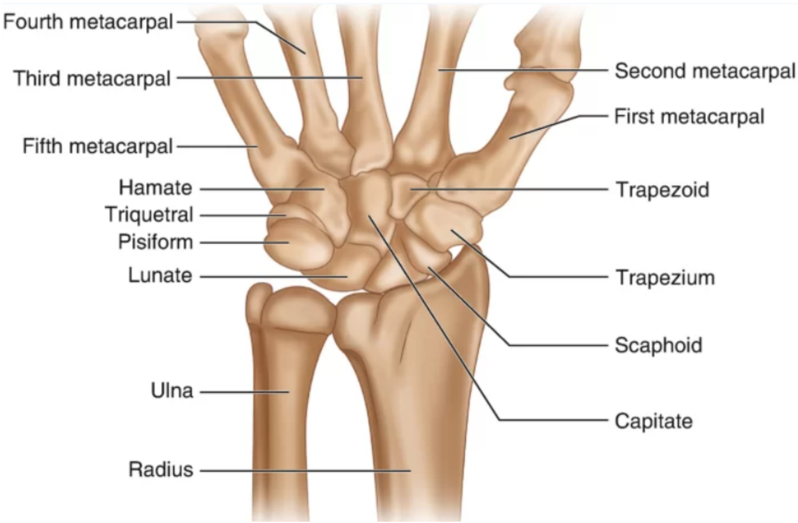


Figure 13: *Bones involved in the wrist joint.*[19]

For a right arm, the positive *Y-axis* is directed proximally, the positive *X-axis* is directed volarly, and the positive *Z-axis* is directed to the right in the anatomical position. To maintain consistency between the clinical motion of left and right arms, for a left arm, the *Y-axis* is directed distally, the *X-axis* dorsally, and the *Z-axis* to the right in the anatomical position.

The eight carpal bones— *caphoid, lunate, triquetrum, pisiform, trapezium, trapezoid, capitate, and hamate*—are collectively considered. The neutral wrist position is used as reference. This position occurs when the wrist is in neutral flexion/extension and neutral radial/ulnar deviation, aligning the long axis of the third metacarpal parallel to the *Y-axis* of the radius.

Typically motion is defined relative to the radius rather than the ulna. Therefore, the orientation of the coordinate systems for each carpal bone should be parallel to the radial coordinate system when the wrist is in the neutral position. Thus, the *Y-axis* of each carpal bone is parallel to the *Y-axis* of the radius, and similarly for the *X and Z axes*, as shown in Figure 14.

Currently,the volumetric centroid of the bone is used to define the origin of a coordinate system within a carpal bone. It is proposed that, when necessary, the origin of a coordinate system within a carpal bone should be located at the volumetric centroid of the bone[19].

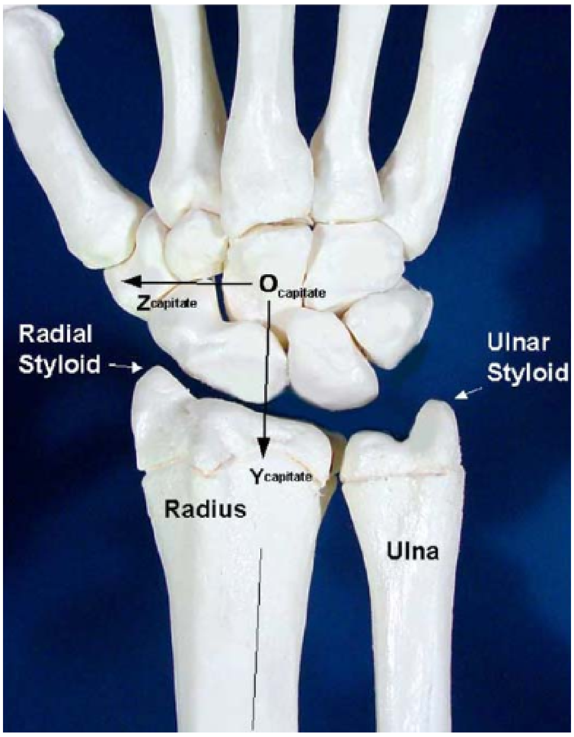


Figure 14: *Dorsal view of a right wrist joint illustrating the capitate coordinate system as an example of the carpal coordinate systems.*[19]

4.3.2 Thigh - Femoral coordinate system

To ensure the proper placement of the Trigno Avanti™ Sensor on the thigh, two essential factors were taken into consideration: the femoral coordinate system relative to the hip joint and the optimal placement relative to the center of rectus femoris' belly [4], as discussed in 4.2.2. The combination of these factors led to the correct positioning of the sensor.

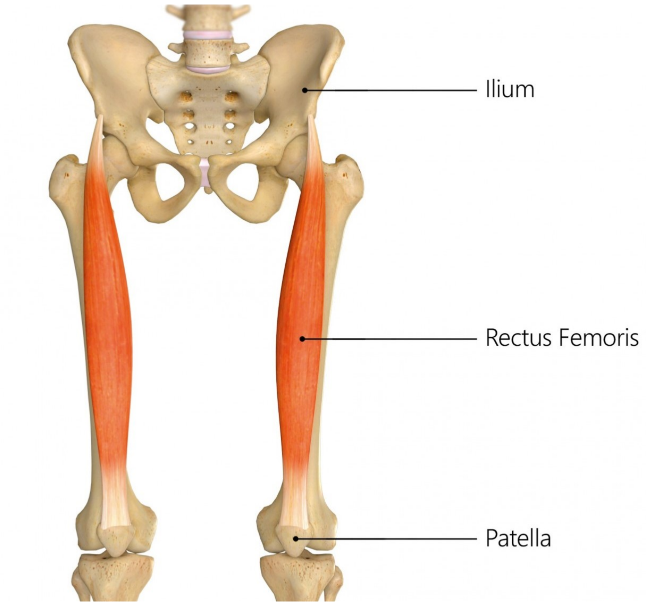


Figure 15: *relative position od rectus femoris muscle and femur bone.*[18]

This section provides a thorough explanation of the local coordinate system of reference, following the directives of the ISB [18]. The femur, located within the thigh, constitutes the upper segment of each lower limb. It is the only bone present in the thigh, articulating with the pelvis at the proximal extremity and with the tibia to form the knee joint at the distal extremity. The femoral coordinate system is derived from the hip joint and is determined by the following axes:

- O: The origin coincident with the hip center of rotation.
- Z: The line parallel to a line connecting the right and left anterior superior iliac spine, and pointing to the right.
- X: The line parallel to a line lying in the plane defined by the two anterior superior iliac spine and the midpoint of the two posterior superior iliac spine, orthogonal to the Z-axis, and pointing anteriorly.
- Y: The line perpendicular to both X and Z, pointing cranially.

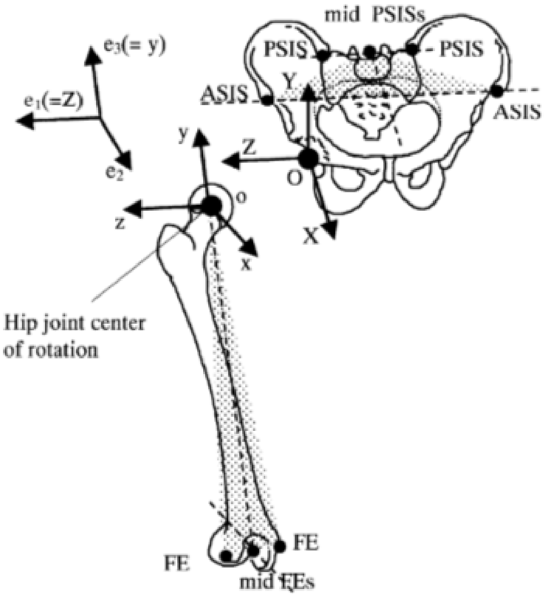


Figure 16: *Pelvic coordinate system (XYZ) and femoral coordinate system (xyz).*[18]

5 Methods of analysis

5.1 Data Pre-processing

Once the data processing is completed, and all markers have been accurately labeled and any gaps resolved, the data is exported into a .mat file format for further analysis in MATLAB (version 9.10.0 R2024a, MathWorks Inc., Natick, MA, USA), where I made data-driven decisions that optimized the results. This .mat file contains both Inertial Measurement Unit (IMU) and Electromyography (EMG) data. The processing pathways diverge based on the data's sampling frequencies (fs).

I applied a dual-pass 4th order Butterworth low-pass filter with a cutoff frequency of 10 Hz to the IMU data. This choice was made after carefully analyzing the signal and evaluating the energy from different frequencies using the Fast Fourier Transform (FFT). Additionally, the mean value was removed, thereby eliminating the DC component and the constant gravity component associated with the sensor's inclination from subsequent parameter calculations.

For post-processing noise reduction in sEMG signals, I followed scientific guidelines from the International Society of Electromyography and Kinesiology (ISEK) and the Surface EMG for Non-Invasive Assessment of Muscles (SENIAM) project [17]. These guidelines recommend using band-pass filters with a low cutoff frequency of 10Hz and a high cutoff frequency of 500Hz, which is particularly relevant for sEMG data sampled at 1kHz, the fs I used. This approach is designed to enhance data quality by removing unwanted noise, which I found crucial for ensuring accurate and reliable results. In applying this strategy, I opted for a Butterworth 4th order filter with the specified cutoff frequencies. I chose this filter because its characteristics are well-suited for smoothing out noise while preserving the integrity of the signal, which was essential for the quality of my data. This step has been invaluable in refining the data, and I felt confident that it would prepare the data effectively for the next stages of analysis.

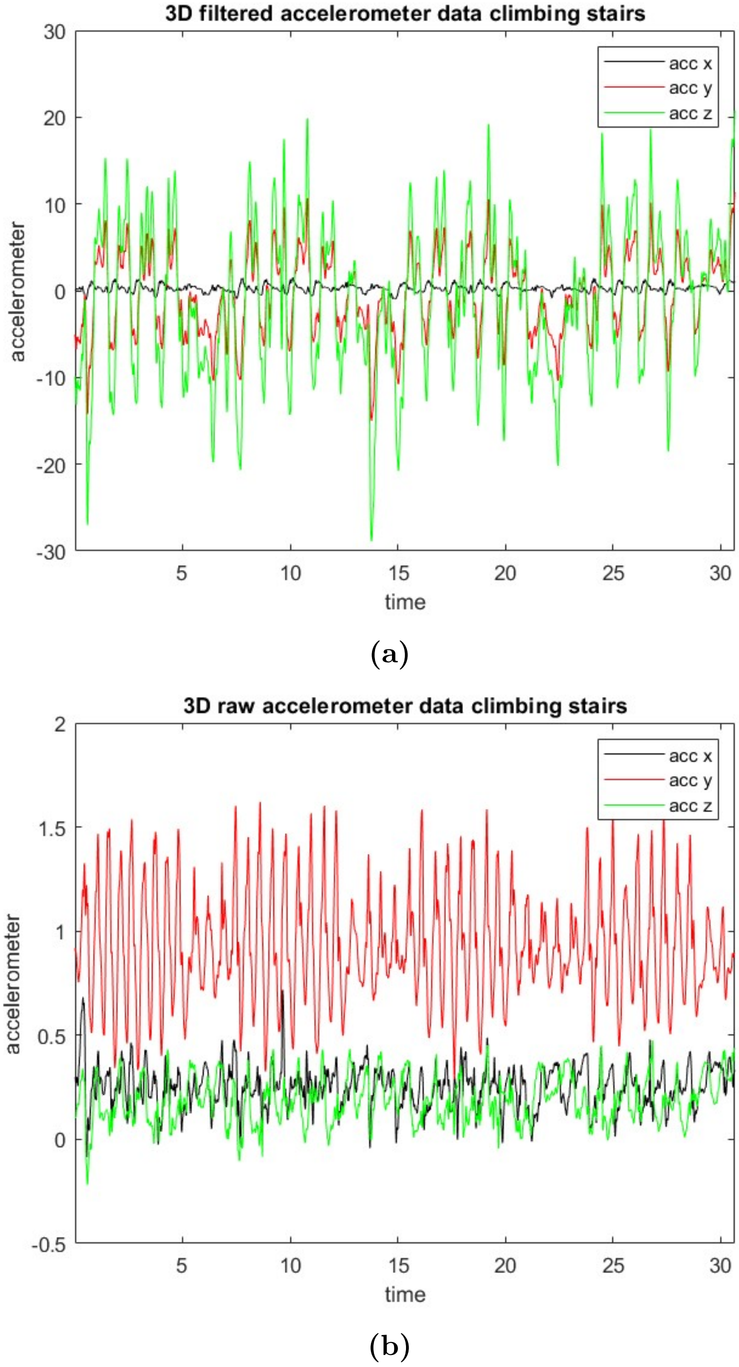
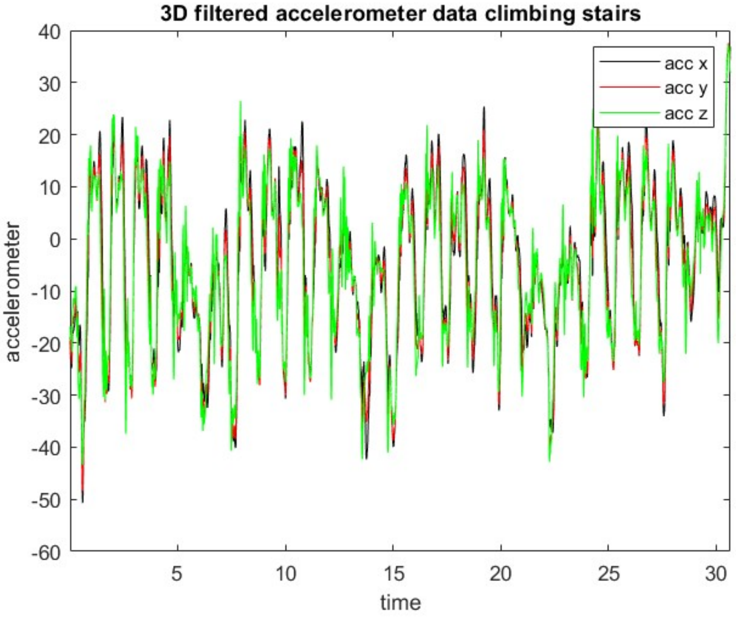
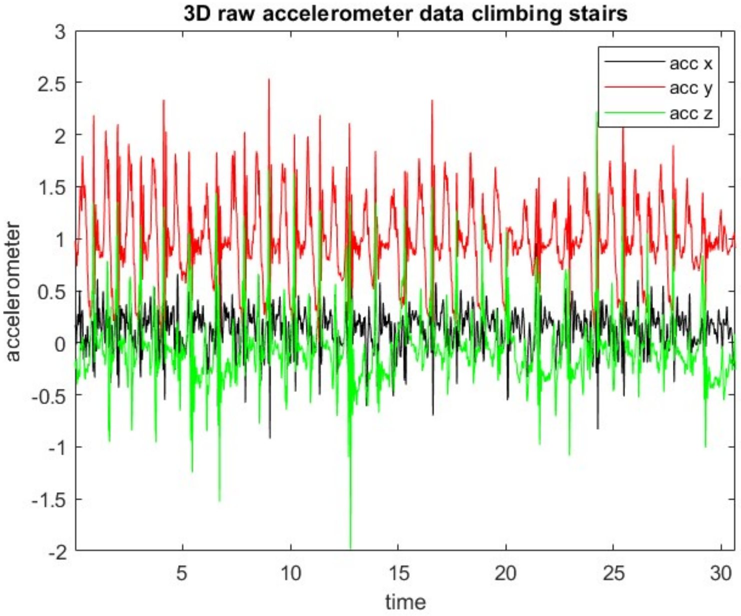


Figure 17: *3D raw **(b)** and filtered data **(a)** for climbing stairs on IMU sensor placed on the wrist*



(a)



(b)

Figure 18: *3D raw **(b)** and filtered data **(a)** for climbing stairs on IMU sensor placed on the thigh*

5.2 Feature Extraction

Features are generally abstractions of data designed to accurately represent the original information. Feature extraction involves transforming large input data into a reduced set of features, known as a feature vector. These feature vectors contain discriminative information that helps differentiate between various activities or data classes and are used as inputs for classification algorithms [20].

I decided to extract features from the data using the `signalTimeFeatureExtractor` and `signalFrequencyFeatureExtractor` objects. These tools are part of the `Signal Processing Toolbox` and the `Statistics and Machine Learning Toolbox` (version 9.10.0 R2024a, MathWorks Inc., Natick, MA, USA). By using these tools, I successfully reduced the data's dimensionality while preserving the key characteristics necessary for accurate classification.

5.2.1 IMU

For the IMU data, I calculated the following features:

- In the time domain: I extracted the mean of the unfiltered signals (`meanFE`), and for the filtered signals, I calculated the root mean square (RMS), shape factor (SF), peak value (PV), crest factor (CF), clearance factor (CLF), and impulse factor (IF).
- In the frequency domain: I determined the mean frequency (MNF) and median frequency (MDF, MNP).

I chose to analyze features in both the time and frequency domains to take advantage of the unique insights each domain provides. Time domain features are straightforward to extract because they involve basic statistical measures of the signal and do not require the computationally intensive process of power spectrum analysis. This simplicity allows for efficient extraction and initial assessment of the signal's characteristics.

In the frequency domain, I focused on the mean frequency (MNF) and median frequency (MDF) because these features offer critical insights into the signal's frequency content. The mean frequency provides an average measure of the signal's frequency distribution, while the median frequency offers a robust central tendency measure, helping to identify dominant frequency components and detect shifts in signal characteristics.

For the frequency domain analysis, I created a feature extractor object (`freqFE`) to systematically compute these features. I selected a Fast Fourier Transform (FFT) window length and spectral estimation window size of 512 samples. This choice was made to strike a balance between frequency resolution and computational efficiency. A window length of 512 samples provides sufficient resolution to accurately capture the frequency components of interest while keeping the computational demands manageable. I also checked the length of the input signal (`Filtered`) and stored it as `signalLength`. If the signal length was shorter than the window length, I adjusted the parameters accordingly to avoid errors. Finally, I configured the frequency domain feature extractor to use Welch's method (`"WelchPSD"`) for spectral estimation, which is well-regarded for providing accurate power spectral density estimates with reduced variance. This approach ensures precise frequency analysis while accommodating the signal's length and avoiding potential errors in the feature extraction process. In table 3 below, I reported all the mathematical formulas used and the namely parameters.

Feature	Mathematical Definition
Mean Value (MV)	$\frac{1}{N}\sum_{i=1}^{N} x(i)$
Root Mean Square (RMS)	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x(i)^2}$
Shape Factor (SF)	$\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} x(i)^2}}{\frac{1}{N}\sum_{i=1}^{N} x(i)}$
Peak Value (PV)	$\max\{x(t)\}$
Crest Factor (CF)	$\frac{\max\{x(t)\}}{\sqrt{2}}$
Clearance Factor (CLF)	$\frac{\max\{x(t)\}}{(\frac{1}{N}\sum_{i=1}^{N} \sqrt{\|x(i)\|})^2}$
Impulse Factor (IF)	$\frac{\max\{x(t)\}}{\frac{1}{N}\sum_{i=1}^{N} \|x(i)\|}$
Median Frequency (MDF)	$\int_0^{\mathrm{MDF}} f \cdot P(f)\, df = \frac{1}{2}\int_0^{\infty} f \cdot P(f)\, df$
Mean Power (MNP)	$\frac{\int_0^{\infty} f \cdot P(f)\, df}{\int_0^{\infty} P(f)\, df}$

Table 3: Features and their Mathematical Definitions

As previously mentioned, the pocket sensor is unique among the sensors used in this study due to its variable positioning during activity recordings, which leads to inconsistent results. To standardize this sensor with the thigh and wrist sensors, which follow the joint coordinate system (JCS) directives provided by the ISB [18], [19], I calculated the Euclidean norm for each acquisition frame from the pocket sensor. This calculation aimed to average the recorded accelerations and velocities, compensating for the absence of a fixed reference axis system.

This method was crucial in addressing the pocket sensor's challenges. Although it excels in comfort and non-invasiveness, which are essential qualities for my study aim, it is also the noisiest sensor due to its lack of a defined axis system. By using the Euclidean norm to average the data, I effectively reduced noise and improved the sensor's reliability. This approach mitigated one of its main limitations, allowing for a standardized record that could be directly compared with the data from the more stable thigh and wrist sensors. Overall, this process was valuable in managing the variability introduced by the pocket sensor's positioning. It facilitated meaningful comparisons between sensors with different orientations, enhanced the accuracy of the analysis, and simplified data integration, ensuring that all measurements were on a comparable scale.

The Euclidean norm $\|\mathbf{x}\|$ of an n-dimensional vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is defined as:

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Where $\mathbf{x} = (x_x, x_y, x_z)$ is a vector where each x_i represents the values of the axes x, y, z at a given time interval. The Euclidean norm $\|\mathbf{x}\|$ is calculated as follows:

$$\|\mathbf{x}\| = \sqrt{x_x^2 + x_y^2 + x_x^2}$$

This norm represents a single value derived from the three directions x, y, z in a defined second of time. In addition to the features extracted from the IMU data recorded, other features were extracted from the norm vectors. These features were then computed to see their correlation with the previously mentioned ones.

5.2.2 EMG

For the EMG signal analysis, I calculated specific features extracted from both the time domain and the frequency domain. In the time domain, I focused on the Mean Absolute Value (MAV) and Root Mean Square (RMS), as these metrics provide straightforward insights into the signal's amplitude characteristics. In the frequency domain, I analyzed the Median Frequency (MDF) and Mean Frequency (MNP), which are useful for understanding the signal's frequency distribution. Additionally, I chose to analyze the Hjorth parameters, which are well-established statistical descriptors used in biomedical signal processing to characterize signals such as EEG and EMG [21]. These parameters, Activity, Mobility, and Complexity, offer valuable insights into the time-domain and frequency-domain characteristics of a signal:

- **Activity (Act)**: This represents the overall energy or power of the signal, calculated as the variance. It provides a measure of the signal's intensity.
- **Mobility (Mob)**: This measures the mean frequency or rate of change of the signal, offering insights into how quickly the signal's characteristics change over time. It is computed as the square root of the ratio of the variance of the first derivative to the variance of the signal itself.
- **Complexity (Com)**: This assesses the irregularity or complexity of the signal waveform. It is determined by the ratio of the mobility of the first derivative of the signal to the mobility of the original signal.

These descriptors are advantageous because they efficiently analyze signals in the time domain without relying on computationally intensive methods like Fourier

Transform. Activity quantifies the signal's power by measuring the squared standard deviation of amplitude. Mobility serves as an estimate of mean frequency, and Complexity provides an estimate of signal bandwidth and its deviation from a pure sine wave. While Hjorth parameters are efficient, they are not without limitations. They are susceptible to noise due to their reliance on signal derivatives. To address this, I also calculated the Spectral Purity Index (SPI) to evaluate the proportion of the signal's power within a specified frequency band. This index helps to assess how much of the signal's energy is concentrated within the desired frequency range, which is particularly useful for EMG signals.

I set the sampling frequency to the instrument's nominal value of $f_{\text{samp}} = 1000\text{Hz}$ and used Welch's method to compute the Power Spectral Density (PSD) of the EMG signal. I defined the frequency range of interest from 10 Hz to 500Hz and calculated the power within this band by summing the PSD values within the identified indices. I then computed the SPI as the ratio of the power in this fundamental band to the total power.

The SPI values for each acquisition were close to unity, validating the filtering process previously conducted. This result indicates that the signal's energy is well-focused within the desired frequency range, which is crucial for analyzing EMG signals accurately. This approach not only helped in assessing signal quality but also ensured that the filtering process was effective in reducing noise and improving data reliability.

This combination of methods allowed me to address the challenges inherent in EMG signal analysis, such as noise and variability, while also leveraging the strengths of each approach to enhance the overall quality and accuracy of the data.

Feature	Mathematical Definition
Mean Absolute Value (MAV)	$\frac{1}{N}\sum_{n=1}^{N} \lvert x[n] \rvert$
Root Mean Square (RMS)	$\sqrt{\frac{1}{T}\int_0^T x(t)^2\,dt}$
Median Frequency (MDF)	$\int_0^{\text{MDF}} f \cdot P(f)\,df = \frac{1}{2}\int_0^{\infty} f \cdot P(f)\,df$
Mean Power (MNP)	$\frac{\int_0^{\infty} f \cdot P(f)\,df}{\int_0^{\infty} P(f)\,df}$
Activity (Act)	$\mathrm{Var}(x(t))$
Mobility (Mob)	$\sqrt{\frac{\mathrm{Var}(\dot{x}(t))}{\mathrm{Var}(x(t))}}$
Complexity (Com)	$\sqrt{\frac{\mathrm{Var}(\ddot{x}(t)) \cdot \mathrm{Var}(x(t))}{\mathrm{Var}(\dot{x}(t))^2}}$
Spectral Purity Index (SPI)	$\frac{\int_{f_1}^{f_2} P(f)\,df}{\int_0^{f_{\max}} P(f)\,df}$

Table 4: Features and their Mathematical Definitions

The MATLAB scripts I used to extract features table 4 and in table 3 are detailed in section 7. For analyzing the frequency domain characteristics, I took a different approach compared to the IMU data, specifically I developed two distinct functions: `fmean` and `fmedian`, whose scripts are also included in section 7.

The `fmedian` function is designed to calculate the median frequency of a given signal epoch. To ensure the accuracy of this calculation, I first preprocess the signal by removing its mean. This step is crucial as it centers the signal around zero, effectively mitigating the influence of any DC components that might otherwise skew the results. These DC components typically arise from noise and the inherent non-zero mean of the EMG signal. After preprocessing, I proceed to define the parameters necessary for calculating the Power Spectral Density (PSD). This includes utilizing a rectangular window with no overlap between segments to ensure a clean and straightforward PSD calculation. Additionally, I set the Nyquist frequency, the frequency resolution, and the number of points for the Fast Fourier Transform (FFT) to optimize the analysis.

Employing Welch's method, I then compute the PSD of the signal, which yields both the PSD values P and their corresponding frequencies f. To determine the median frequency, I sum the PSD values cumulatively until this cumulative sum reaches half of the total power of the spectrum. This cumulative point effectively divides the spectrum into two equal halves. Subsequently, I calculate the median frequency f_{median} as the average of the two frequencies that bracket this cumulative sum point. This approach helps to identify where the spectral power is centered within the signal, providing valuable insights into the distribution of frequency components. I have found this method particularly effective for capturing the median frequency, which is instrumental in understanding the spectral characteristics of the EMG signal. In parallel, for calculating the mean frequency, I have developed the `fmean` function. This function starts by determining the number of points required for the FFT based on the sampling frequency f_{samp} and the length of the signal epoch epoch_len. As with the median frequency calculation, the signal is preprocessed to remove its mean before proceeding. I then compute the Power Spectral Density (PSD) using Welch's method, which provides the PSD values P_{xx} along with their corresponding frequencies f.

To find the mean frequency f_{mean}, I perform a weighted sum of the frequencies f, where each frequency is multiplied by its corresponding PSD value P_{xx}. This weighted sum is then divided by the total power in the PSD to yield the mean frequency. This measure represents the center of mass of the spectral power distribution, offering a comprehensive view of the average frequency of the signal. This method is particularly useful for understanding the overall frequency characteristics of the signal, as it highlights the dominant frequency components and provides a clearer picture of the signal's frequency distribution.

After extracting these features using the MATLAB Toolbox, I compiled them into a table array and imported this data into Excel for further analysis. These data encompassed the three-axis readings from the accelerometers and gyroscopes for all three sensors placed on the patients. I chose this detailed approach to ensure that each aspect of the sensor data was thoroughly analyzed, allowing for a comprehensive evaluation of the sensor performance and accuracy.

By carefully preprocessing the data and employing both median and mean frequency calculations, I was able to gain valuable insights into the signal's frequency content and improve the overall quality of my analysis.

5.3 Features selection

The different features were organized into Excel worksheets to enable meaningful export to the data mining environment Orange (Demsar et al. 2013). Orange is a component-based visual programming software package for data visualization, machine learning, data mining, and data analysis [22]. The worksheets exported were divided and organized by sensor: there were mainly four data worksheets per subject, one for the EMG data and three for the IMU data.

Before diving into all the features, I find it essential to figure out which ones are crucial and which ones aren't necessary. So, I first identify the features that influence activity selection, then narrow down the features based on how they correlate with each other. In the world of machine learning, not every feature within a dataset contributes equally to the model's predictive power, some features may introduce noise or redundancy, potentially obstructing the model's performance. Thus, the process of feature selection is paramount to ensure the inclusion of only the most relevant and informative features.

To optimize the feature selection process and move beyond the traditional models typically used in data analysis, I decided to employ two complementary methods in order to provide a more comprehensive comparison of the relevant features. First, I focused on the correlation between the features themselves by analyzing the correlations' coefficients, which enabled me to identify and eliminate redundant features, thereby reducing complexity and potential multicollinearity.

Second, I constructed and classified a decision tree, a method that helps identify the features most important in distinguishing between different motor tasks. By evaluating the significance of each feature within the decision tree, I was able to prioritize those that contribute most effectively to the classification process.

By combining these two methods—correlation analysis and decision tree classification—I implemented a robust feature selection protocol that ensures the selected features are both non-redundant and highly informative for classifying motor tasks. This dual approach not only streamlines the model by eliminating unnecessary features but also enhances its predictive performance by focusing on the most significant features.

5.3.1 Correlation Analysis

Correlation is a statistical measure indicating the degree to which two variables are related. When variables are correlated, a change in one variable is associated with a change in another, either in the same direction (positive correlation) or in the opposite direction (negative correlation). Often, correlation is discussed in the context of a linear relationship between two continuous variables, showing the strength of the relationship between them. When features exhibit high correlation, they convey similar information about the target variable. This redundancy can be a problemfor constructing the model, as it may lead to overfitting and make the model unnecessarily complex.

Feature selection is a crucial step in the modeling process, aiming to identify the most informative elements within a dataset. By focusing on the most relevant features, It's possible to implement a model that captures the essential patterns in the data while avoiding the pitfalls of irrelevant or redundant information.

To streamline the model and enhance its efficiency, it is essential to identify and eliminate highly correlated features. Features that provide similar information can lead to redundancy, which not only complicates the model but also increases its computational complexity. Furthermore, high correlation among features can result in multicollinearity—a situation where predictors are highly interrelated. This interrelation can inflate the variance of coefficient estimates, making the model unstable and difficult to interpret. By removing one of the correlated features, it is possible to mitigate these issues, ensuring that the model remains both robust and interpretable.

A judicious feature selection process, which takes into account feature correlation, optimizes the model's predictive performance. This process involves retaining a sufficient number of features to capture the complexity of the data while discarding those that do not contribute additional, meaningful information. This approach enhances the model's robustness and facilitates a clearer understanding of the underlying data patterns by focusing on the key drivers of the target variable. Consequently, the model becomes more streamlined, efficient, and easier to interpret, and is more likely to generalize well to new data.

In summary, by concentrating on the most informative features, I managed to develop models that are not only effective in prediction but also provide valuable insights into the structure and relationships within the data. This refined approach ensures that the model remains both powerful and practical, offering accurate predictions and a deeper understanding of the data's underlying dynamics. Before analyzing and evaluating the degree of correlation, it's crucial to normalize or standardize the dataset. This step ensures that all features are on the same scale, which facilitates better comparison and enhances the training performance of the neural network. Specifically, normalization can be achieved using the min-max normalization method [23], which scales the values of a variable to a range between 0 and 1. The min-max normalization is implemented as:

$$\tilde{x}_i = \frac{x_i - \min\{x\}}{\max\{x\} - \min\{x\}}$$

where $\tilde{x}_i$ is the normalized value of the feature x_i, and $\min\{x\}$ and $\max\{x\}$ represent the minimum and maximum values of the feature, respectively. This ensures that:

$$0 \leq \tilde{x}_i \leq 1$$

Normalization significantly improves the model's convergence rate and overall accuracy. By ensuring that features are on a comparable scale, normalization allows for more effective training of the neural network. Consequently, I am confident in building models that are both efficient and effective, with enhanced performance and stability [23].

To identify the degree of correlation between the different features in the working dataset, the `Correlation Widget` was employed. I calculated the degree of correlation between the different features using the Pearson product-moment correlation, which is commonly abbreviated as *"r"*. The values obtained are showed in the below table 5, table 6 and table 7.

Feature 1	**Feature 2**	**Correlation *r***
RMS	STD	+1.000
CLF	IF	+0.959
CF	IF	+0.937
CF	CLF	+0.909
PV	STD	+0.898
PV	RMS	+0.898
MDF	MNF	+0.844
IF	SF	+0.743
CLF	SF	+0.740
PV	SF	+0.593
CLF	MNF	+0.589
CF	SF	+0.576
CF	MNF	+0.575
IF	MNF	+0.565
MDF	PV	+0.560
CLF	MDF	+0.514
CLF	PV	+0.492
SF	STD	+0.481
RMS	SF	+0.480
MDF	SF	+0.472
MDF	STD	+0.471
MDF	RMS	+0.469

Table 5: Pearson Correlation between Variable Pairs Pocket sensor

Feature 1	Feature 2	Correlation r
RMS	STD	+1.000
CLF	IF	+0.986
CF	IF	+0.972
CF	CLF	+0.932
CLF	SF	+0.854
IF	SF	+0.825
PV	STD	+0.795
PV	RMS	+0.795
CF	SF	+0.721
CF	RMS	-0.489
CF	STD	-0.488
IF	RMS	-0.470
IF	STD	-0.469
MDF	MNF	0.454
CLF	RMS	-0.434
CLF	STD	-0.433
MNF	RMS	-0.392
MNF	STD	-0.391
MNF	PV	-0.346
IF	MNF	+0.329
CF	MNF	+0.312
CLF	MNF	+0.306

Table 6: Pearson Correlation between Variable Pairs Thigh Sensor

Feature 1	**Feature 2**	**Correlation *r***
RMS	STD	+1.000
CLF	IF	+0.990
CF	IF	+0.978
CF	CLF	+0.959
CLF	SF	+0.825
PV	STD	+0.809
PV	RMS	+0.807
IF	SF	+0.767
CF	SF	+0.675
MDF	MNF	+0.590
MNF	PV	+0.481
CF	MNF	+0.461
PV	SF	+0.403
IF	MNF	+0.398
CLF	MNF	+0.389
MDF	CF	+0.372
CLF	PV	+0.328
MDF	IF	+0.321
MDF	CLF	+0.316
IF	PV	+0.310
MDF	IF	+0.307

Table 7: Pearson Correlation between Variable Pairs Wrist Sensor

To present a clearer view of the data, I have summarized the correlation levels in the following bar charts:

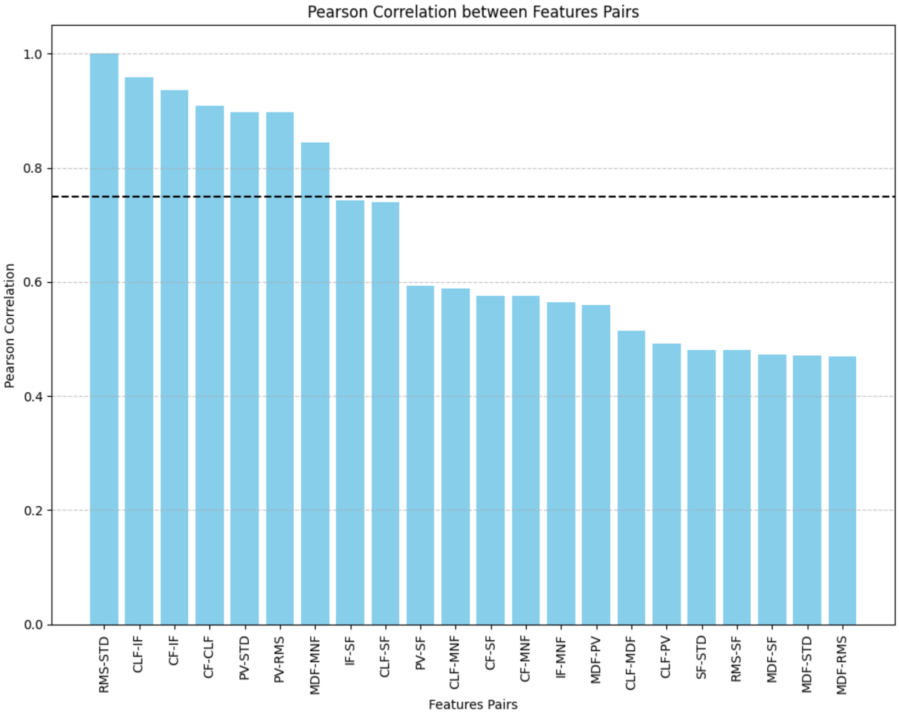


Figure 19: *Bar chart illustrating the Pearson correlation between pairs of features from table 5. The dashed horizontal line marks the threshold value of 0.750, used as a reference to highlight correlations exceeding this value.*

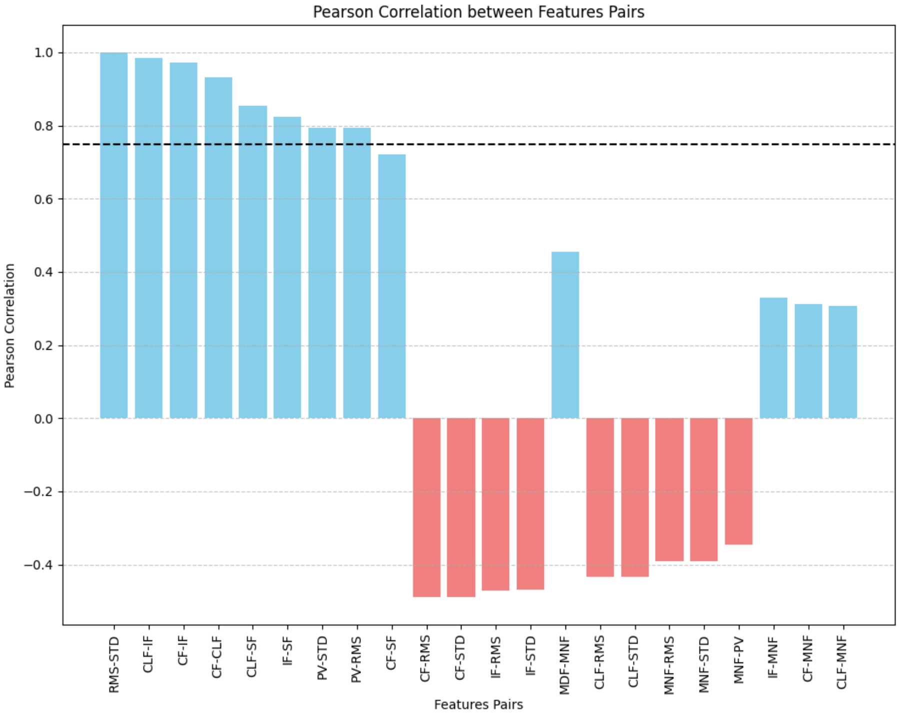


Figure 20: *Bar chart illustrating the Pearson correlation between pairs of features from table 6. The bars represent the correlation value for each feature pair, with different colors indicating the direction of the correlation. The dashed horizontal line marks the threshold value of 0.750, used as a reference to highlight correlations exceeding this value.*

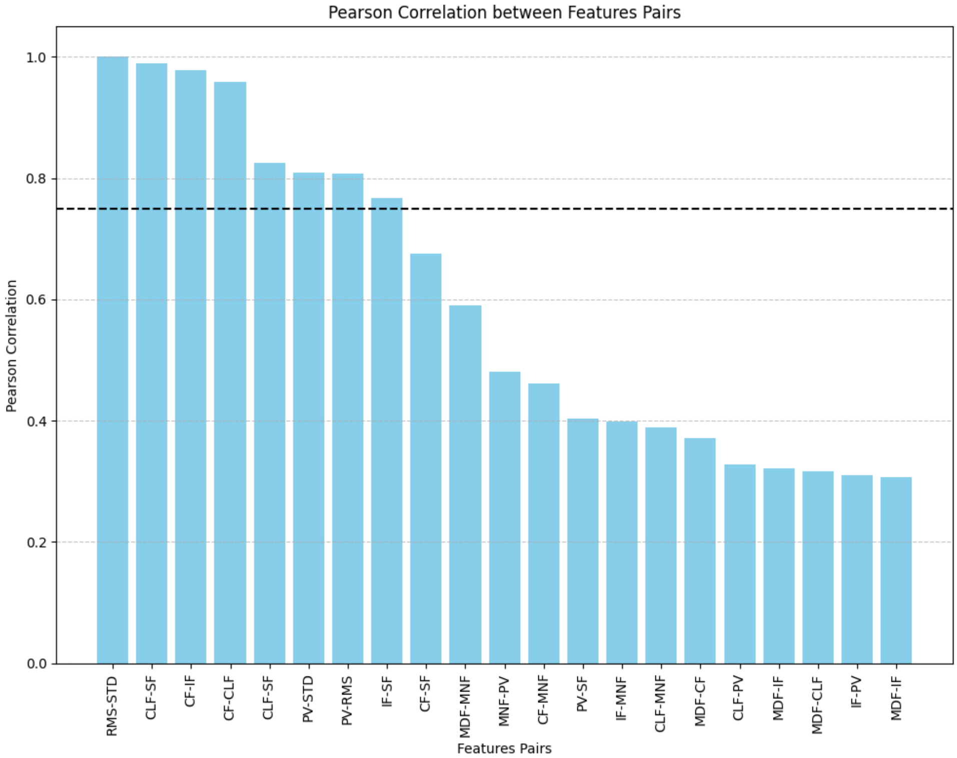


Figure 21: *Bar chart illustrating the Pearson correlation between pairs of features from table 7. The dashed horizontal line marks the threshold value of 0.750, used as a reference to highlight correlations exceeding this value.*

5.3.2 Decision Tree selection

The decision tree model plays a crucial role in refining feature selection and enhancing model performance. After analyzing feature correlations to eliminate redundant variables, I further optimized the feature selection process using a decision tree approach. Specifically, I utilized the Automatically Decision Tree model, as detailed by Parkka et al. (2006) [24]. This model was implemented and visualized through the `Tree` and `Tree Viewer` widgets, providing an in-depth look at the decision-making process. A decision tree is a powerful supervised learning algorithm used for both classification and regression tasks. It operates similarly to a flowchart, starting at a root node that poses a specific question about the data. Based on the answers to this question, the tree branches out into internal decision nodes, each leading to further questions and additional branches. This process continues iteratively until the data reaches terminal nodes, or "leaf" nodes, where the final decision or prediction is made. The strength of decision trees lies in their ability to offer a clear, visual representation of how algorithms process data to reach outcomes. This visual clarity facilitates an intuitive understanding of the decision-making process, from the initial query to the final classification or prediction [25]. A key consideration in developing a decision tree is the choice of the splitting criterion. For the decision tree created using the Orange 3 widget, I chose the default criterion: Information Gain. This criterion evaluates the reduction in entropy or uncertainty within the dataset caused by the splits, thereby prioritizing the most informative features in the decision-making process. By leveraging the decision tree's visual and analytical capabilities, I managed to refine the feature selection process further, enhancing the overall predictive performance and interpretability of the model. In the realm of machine learning, Information Gain serves as a pivotal guidepost in decision tree construction. It's a metric designed to quantify the effectiveness of a potential feature in splitting a dataset. Essentially, it aims to minimize uncertainty or entropy in the dataset, calculated using the formula [26]:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Here, $H(S)$ represents the entropy of the dataset before splitting, p_i is the proportion of examples in class i, and c is the number of classes. When splitting the data based on a feature, the goal is to maximize the reduction in entropy. This reduction, known as information gain, is calculated using the formula:

$$IG(S, A) = H(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} H(S_i)$$

Where $H(S)$ is the entropy of the parent node, $|S_i|$ is the number of examples in the ith child node, $|S|$ is the total number of examples in the parent node, and $H(S_i)$ is the entropy of the ith child node after splitting [26]. By systematically evaluating information gain across all features, the decision tree algorithms were employed to navigate the data landscape. This approach ensured that the features selected provided the most clarity for decision-making, leading to a model that was both accurate in its predictions and insightful in its analysis. To enhance the model's reliability and effectiveness I methodically divided the dataset, selecting features that most effectively classified the data at each stage. Throughout each training and validation session, I built the decision tree using the training data and then refined it to achieve the optimal level, defined by the lowest error rate. I rigorously tested the decision tree model to evaluate its accuracy, ensuring that it was not only sufficient but also reliable for feature selection. The accuracy levels achieved, showed in table 8, demonstrated that the method was efficient and that the model was robust and trustworthy for selecting the most relevant features.

Placement	AUC	CA	F1	Prec	Recall	MCC
WRIST	0.930	0.785	0.788	0.812	0.785	0.719
THIGH	0.907	0.732	0.730	0.732	0.732	0.644
POCKET	0.891	0.688	0.684	0.684	0.688	0.584

Table 8: Performance Evaluation of the Costume Decision Tree for each sensor placement

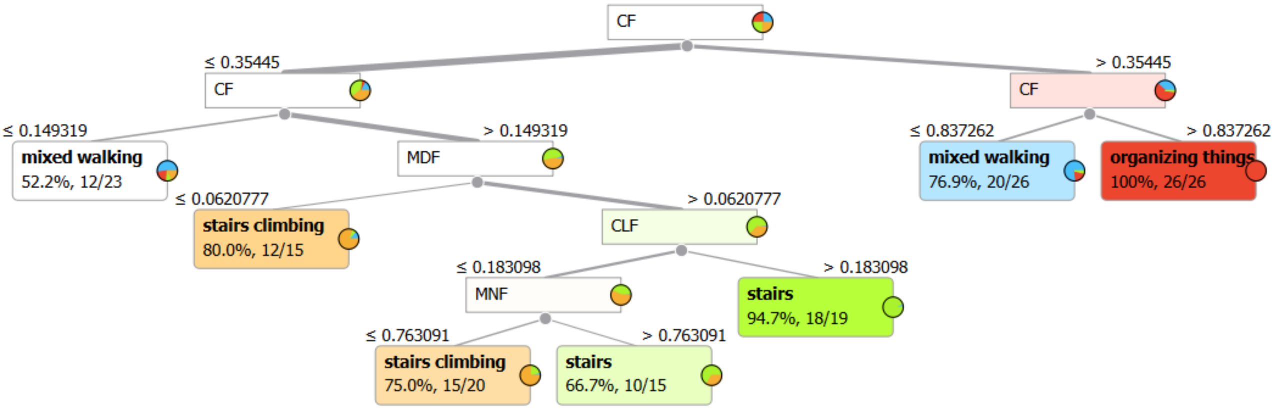


Figure 22: *Decision Tree relative to the wrist's sensor data*

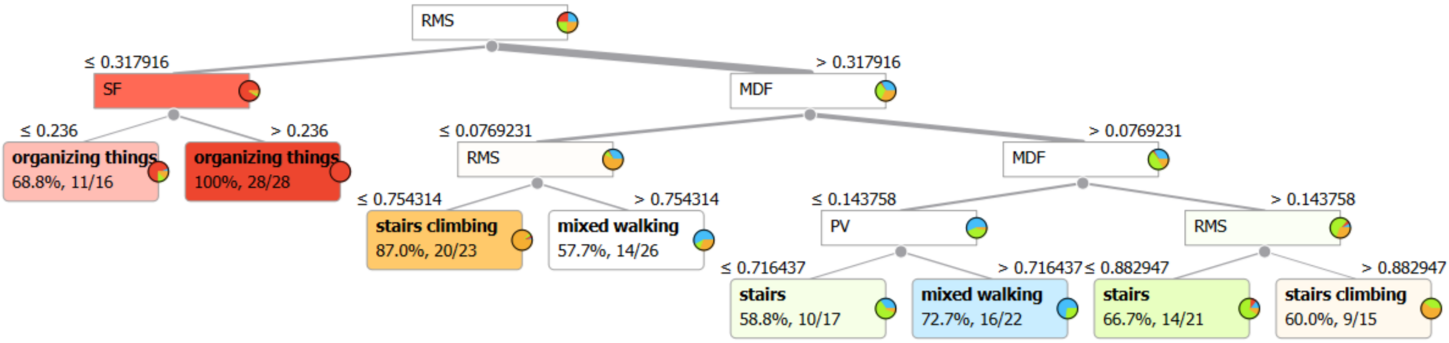


Figure 23: *Decision Tree relative to the thigh's sensor data*

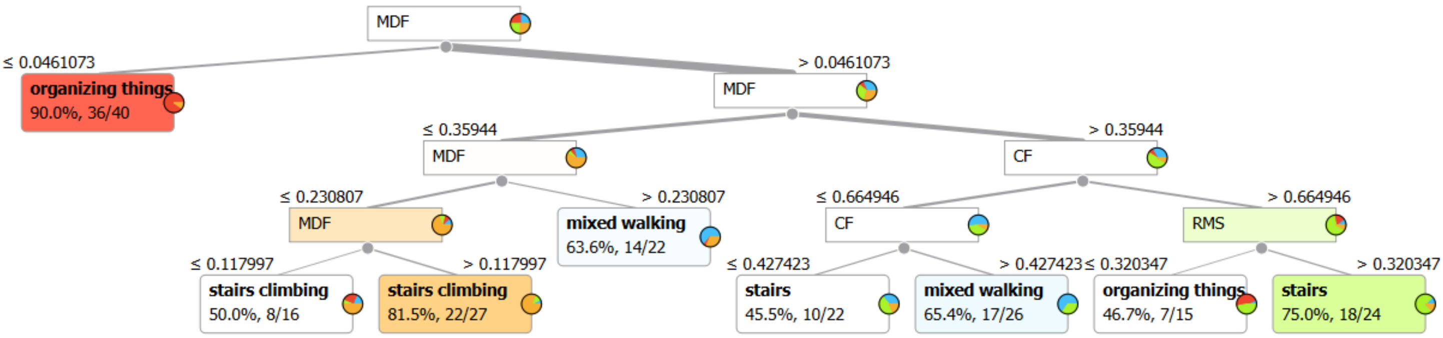


Figure 24: *Decision Tree relative to the pocket's IMU data*

5.3.3 Feature Exclusion Rationale

Upon reviewing the correlation table and analyzing the features used in the decision tree, I decided to exclude specific features from the dataset. These decisions were based on high correlations with other features and overlapping roles in the analysis. The Pearson correlation coefficient r was calculated by evaluating and comparing the same features for each axis and for each magnitude —acceleration and angular velocity— separately. This axis-specific and magnitude-specific correlation analysis aligned with the overall data analysis, confirming that the features identified for removal due to correlation were consistent with those indicated by the general analysis.

The rationale for excluding each feature, considering these correlations and their implications, is detailed below:

1. **Standard Deviation (STD)**: The strong correlation between Standard Deviation (STD) and Root Mean Square (RMS) stems from their intrinsic mathematical relationship and complementary functions in representing dataset characteristics. Both metrics are related to the distribution of data

around the mean: STD measures the dispersion of values relative to the mean, while RMS provides an average magnitude of the values, considering both positive and negative deviations from the mean. Mathematically, RMS can be viewed as an extension of standard deviation when dealing with datasets that include both positive and negative values. RMS is particularly sensitive to extreme values because it squares each value before averaging, which amplifies the impact of deviations from the mean. This inherent connection makes RMS and STD closely correlated, especially in contexts where values are not symmetrically distributed or include outliers.

The high correlation coefficient between STD and RMS reflects not only their theoretical relationship but also how both metrics capture similar information about the variability and intensity of the data. Since they both quantify how data oscillates around a central value, it is natural for them to exhibit significant correlation.

Reflecting on this, I concluded that including both features could introduce redundancy into the model without providing significant additional value. The presence of strong multicollinearity could not only complicate the interpretation of the model but also negatively impact the stability of the coefficient estimates. For these reasons, I decided to exclude STD from the dataset, relying on RMS to represent the variability and magnitude of the data in the dataset aiming to simplify the analysis, while still maintaining the dataset's descriptive capacity. This choice allows me to focus on features that contribute distinct and useful information to the model, thereby improving its efficiency and effectiveness.

2. **Impulse Factor (IF)**: The Impulse Factor (IF) shows a strong correlation with both the Shape Factor (SF) and the Clearance Factor (CLF). This high correlation can be attributed to the interdependent relationships within the system under study, as shown in **??**, and can be explained by several interconnected aspects of the system. The IF, which measures the peak value of a transient event, can be influenced by the Shape Factor(SF), which in turn affects the CLF, a measure that captures the severity of peaks relative to the average vibration. This interdependence results in overlapping information among the factors. By excluding IF from the dataset, the model can be simplified, reducing the risk of overfitting while retaining the most relevant information provided by SF and CLF. This approach ensures a more robust and interpretable model by avoiding noise introduced by redundant variables. Practical considerations, such as system design and measurement limitations, may also contribute to the observed correlations, as these factors are affected by common dynamics and measurements. Thus, while IF captures the intensity of transient changes, SF characterizes geometric properties influencing force distribution, and CLF reflects the severity of vibration peaks. Given their interrelated nature, careful consideration in modeling is required to improve both interpretability and predictive performance.

3. **Clearance Factor (CLF)**:The significant correlation between the Clearance Factor (CLF) and the Crest Factor (CF), as shown in **??**, reflects their shared reliance on specific characteristics of the signal or system behavior. The Crest Factor measures the ratio between a waveform's peak value and its RMS value, commonly used to assess the peakiness or transient nature of a signal. The

high correlation suggests that systems with higher Crest Factors, indicating more pronounced signal peaks, are also likely to exhibit higher Clearance Factors, which quantify the severity of these peaks compared to the average vibration levels. Conversely, systems with lower Crest Factors typically exhibit lower Clearance Factors. My analysis revealed that including both factors could introduce redundancy due to their interconnected nature. Thus, CLF was excluded to improve model clarity and focus on features with distinct contributions.

4. **Mean**: I decided to exclude the mean of the unfiltered signal from the analysis based on a detailed consideration of the nature of IMU sensor data and its characteristics. IMU sensors, which measure acceleration and angular velocity, produce data that are inherently dynamic and vary significantly with changes in motion and orientation. The mean value of such data, especially when unfiltered, tends to be low and relatively constant due to the averaging effect of constant motion or lack of significant deviations.

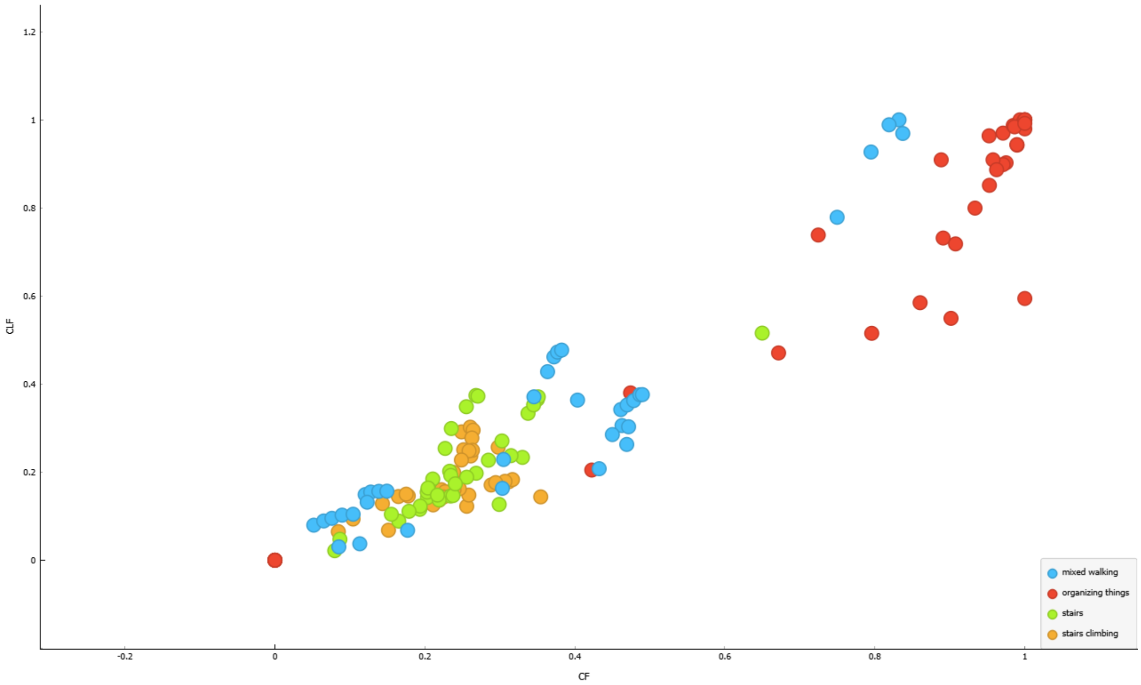
   Given that IMU data are primarily used to capture transient events and dynamic changes in movement, the mean value, which represents a static measure of central tendency, does not reflect the critical variations and patterns needed to differentiate between different activities. For instance, activities involving rapid changes or specific motion patterns are better characterized by features such as peak values, frequency content, and variations, which the mean value fails to capture adequately. Moreover, the presence of high-frequency noise and transient fluctuations in IMU data means that the mean value often remains stable and uninformative, as it is influenced more by the constant or average state rather than the activity-specific dynamics. Including a feature that does not vary significantly with different activities introduces redundancy and can dilute the effectiveness of the feature set, leading to potential multicollinearity issues and complicating the analysis.

By excluding the mean, the dataset is streamlined to focus on features that better capture the dynamic behavior of the system, such as variations in acceleration and angular velocity that are more indicative of specific activities. This decision aligns with best practices in sensor data analysis, where the goal is to extract and emphasize features that provide meaningful insights into the varying dynamics of the system, ensuring a more robust and insightful model for activity detection.

In summary, the exclusion of Standard Deviation (STD), Impulse Factor (IF), Clearance Factor (CLF), and Mean from the dataset was based on their high correlations and overlapping roles with other features. Removing these features reduces redundancy, avoids multicollinearity, and focuses on more informative metrics. This refined selection improves the model's clarity and performance by ensuring that only unique and relevant features are included, thereby enhancing its ability to effectively detect and classify activities. In table 9 below, I reported the features removed for each placement after comparing both the Pearson correlation coefficient r and the classification made by the Decision Tree.

Placement	Features Removed
Wrist	STD, IF, Mean
Thigh	STD, CLF, IF, Mean
Pocket	STD, CLF, IF, PV, Mean

Table 9: Features removed for each placement to increase the goodness of the analysis.



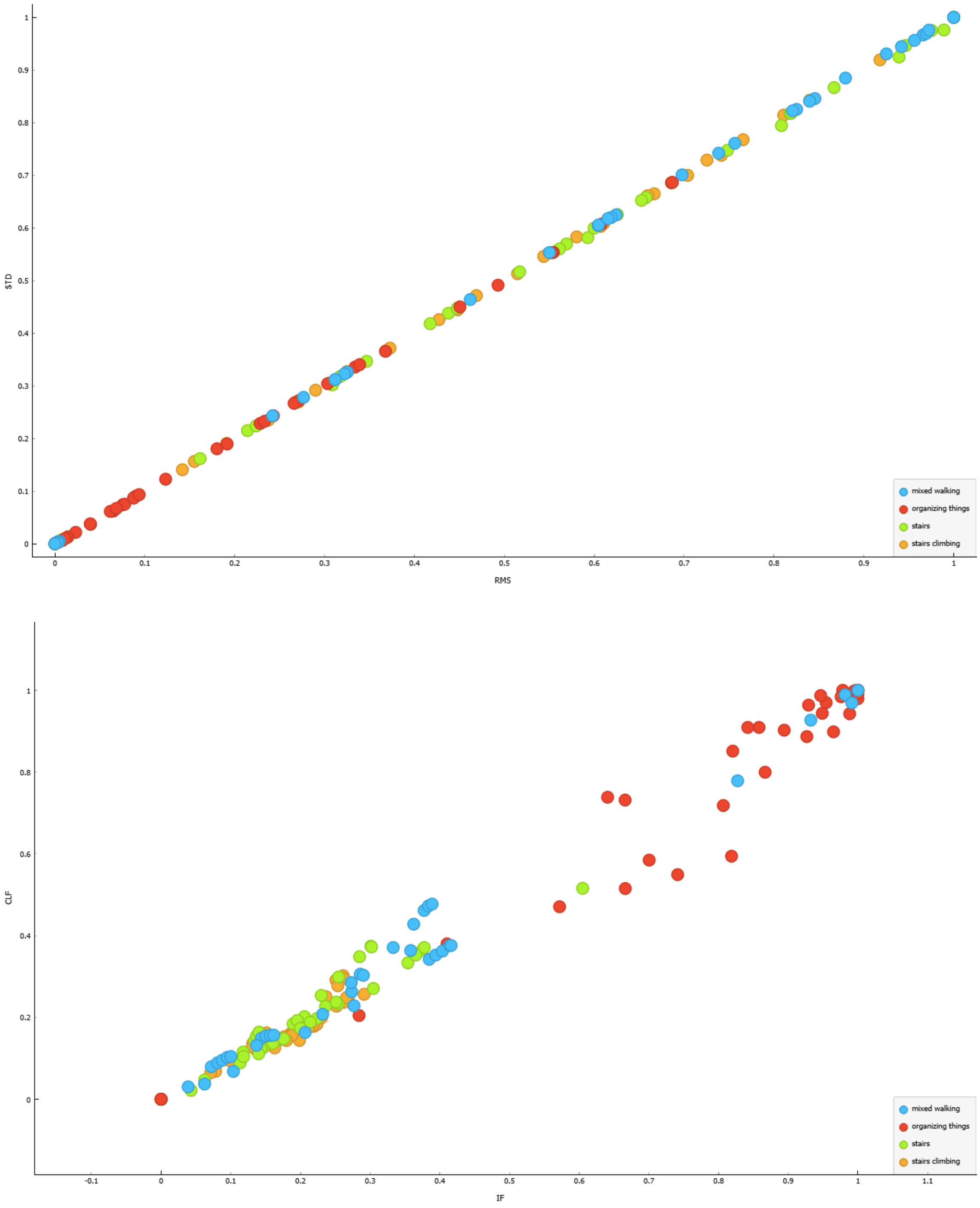


Figure 26: *Statistical relationship between the features based on the data calculated on the wrist sensor:*
the first Figure shows the correlation between between CF and CLF, the second Figure shows the correlation between RMS and STD, and the last Figure shows the correlation between the IF and CLF values.

5.4 Machine Learning

This chapter explores the statistical and mathematical techniques utilized in recognition and evaluation processes within Machine Learning (ML) and Neural Network (NN) applications, with a particular focus on supervised learning, a fundamental approach in Human Activity Recognition (HAR) as demonstrated by various studies [27], [28], [29]. Supervised learning is crucial because it relies on known examples, such as training sets, to estimate a function that maps inputs to outputs. By training on a labeled dataset, where each example is paired with a corresponding output label, the model learns to generalize the relationship between inputs and outputs. This process allows the model to make accurate predictions on new, unseen data, which is essential in the HAR process as it enables the model to generalize across new patient groups. Highlighting supervised learning is essential because it forms the backbone of many practical applications in ML and NN, where precise mapping from inputs to outputs is crucial for tasks such as classification, regression, and pattern recognition.

Li, Xin, et al. [28] aimed to enhance exoskeletons' ability to recognize human motion patterns by analyzing the biomechanics of the lower limbs and developing wearable devices equipped with electromyography (EMG) sensors and inertial measurement units (IMUs). They introduced a Dual Stream Convolutional Neural Network (CNN)-Relief method to handle the data acquired. Convolutional Neural Networks (CNNs) are particularly effective for modeling complex, non-linear relationships between variables, such as those encountered in the HAR process. CNNs are widely used across various domains due to their ability to model intricate patterns and make accurate predictions. Although threshold-based and time-frequency algorithms offer high accuracy in data classification, these approaches often require calibration, and pre-determined threshold rates may not be applicable across different neurological conditions.

Similarly, Y. Celik et al. [29] provide an overview of existing inertial and EMG-based algorithms and summarize previous gait assessment studies in neurological contexts. They discuss the advantages and limitations of using wearable devices for gait assessment and explore future research directions. Specifically, they address gaps in feature selection and classification, proposing the use of ML and NN techniques, particularly deep learning methods, to improve activity recognition with inertial data from wearable devices. Given these considerations, I decided to focus on supervised methods, specifically neural networks (NN) within the realms of machine learning (ML) and artificial intelligence (AI). Neural networks are chosen for their exceptional accuracy and adaptability to diverse conditions and datasets. Their flexibility and robustness make them particularly well-suited for improving activity recognition and addressing some of the limitations found in traditional methods.

5.4.1 Multilayer Perceptron

Among the various neural network architectures, the Multilayer Perceptron (MLP) has been selected for this study due to its proven effectiveness and versatility in addressing a wide range of problems.

The MLP consists of an input layer, one or more hidden layers, and an output layer, with each neuron in one layer fully connected to every neuron in the subsequent layer. The input layer processes the data, with each neuron representing a specific feature or variable from the dataset. The hidden layers are responsible for performing computations and extracting features from the input data, allowing the network to learn and model complex relationships. The number of hidden layers and neurons per layer can be adjusted based on the complexity of the problem at hand. The output layer generates the final prediction or classification.

The Multilayer Perceptron (MLP) is well-suited for this study due to its ability to model complex, non-linear relationships and its well-established training methodologies. Its robust and flexible architecture enables effective learning of hierarchical feature representations, which is crucial for capturing intricate patterns in the data. A key challenge in developing the MLP model was determining the optimal configuration for recognizing various motor tasks. This involved selecting the appropriate number of neurons and hidden layers tailored to the specific dataset. Designing this structure required meticulous consideration to balance model complexity and performance, ensuring that the MLP could learn effectively and generalize from the data. The implementation of the MLP was carried out using the Orange widget `Neural Network`, which provides a range of options for configuring the MLP architecture and learning process.

The learing process involves adjusting the weights of the connections between neurons to minimize the discrepancy between the predicted and actual outputs. This adjustment is accomplished using backpropagation, a technique that computes the gradient of the loss function with respect to the weights and updates the weights accordingly to reduce the error. The training procedure typically includes at least one complete pass through the dataset (epoch). To mitigate overfitting and manage training time, stopping rules are employed. In the table 10 are reported the most common stopping rules used:

Stopping Rule	Description
SR1	Minimum relative change in error achieved
SR2	Error cannot be further decreased
SR2	Maximum training time has been exceeded

Table 10: Stopping rules for the MLP training process.

These rules might include stopping training when the error on the validation set increases (SR1) or when the error reduction falls below a certain threshold over a set number of epochs (SR2) [27]. The SR2 stopping rule is used by default, which limits the algorithm to a maximum number of iterations. In this instance, the maximum number of iterations has been set to 200. Two other important settings that need consideration are the type of activation function for the neurons in the hidden layers and the type of solver used.

I decided to use the hyperbolic tangent activation function, also known as `tanh`, because examining the data distribution indicated that this function would effectively discriminate between the classes [30]. Let z be the weighted sum of inputs and biases:

$$z = \sum_{i=1}^{n} (w_i \cdot x_i) + b$$

where:

- w_i are the weights,

- x_i are the input values,

- b is the bias term.

The output of the hyperbolic tangent function, also known as the `tanh` activation function, is computed as follows:

$$\hat{y} = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

where:

- $\tanh(z)$ is the hyperbolic tangent function,

- e is the base of the natural logarithm.

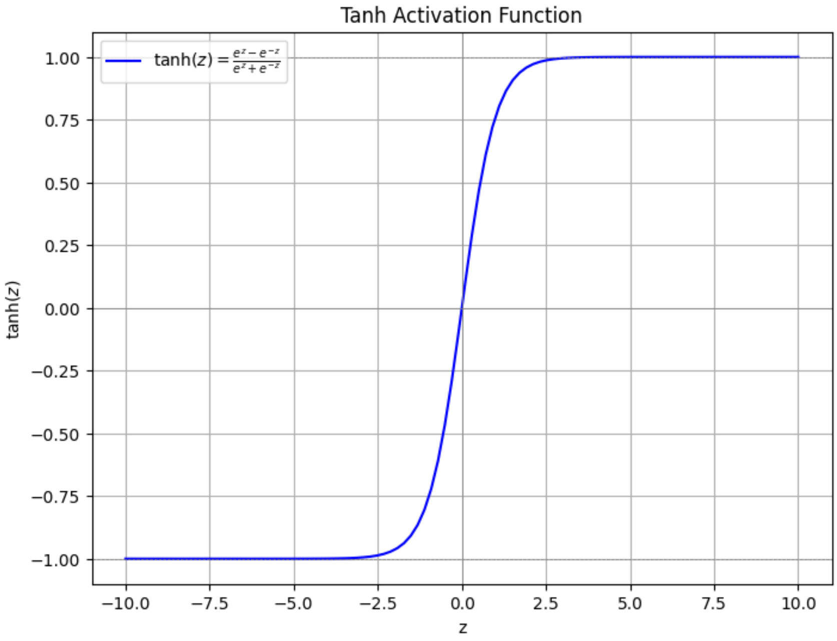


Figure 27: *Representation of the activation function.*

The `tanh` function is particularly powerful in neural networks as it maps outputs to a range between -1 and 1, centering the data around zero, as shown in Figure 27. This helps in achieving better convergence during training. Additionally, `tanh` introduces non-linearity, provides smooth gradients for optimization, and allows for a normalized output that can enhance the network's ability to model complex patterns. During the backpropagation algorithm, the gradients of the loss function with respect to the weights and biases are calculated using the derivative of the `tanh` function:

$$\frac{\partial \tanh(z)}{\partial z} = 1 - \tanh^2(z)$$

This derivative shows how the `tanh` function's output changes with respect to its input, and it is essential for adjusting the weights and biases in the network to minimize the error during training.

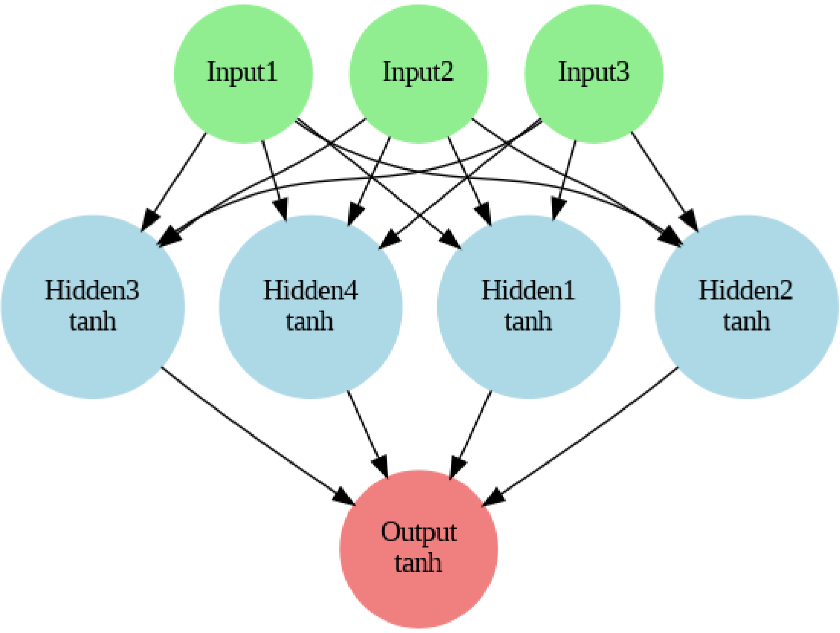


Figure 28: *Example of MLP with* `tanh` *activation function*

These gradients are then used to update the weights and biases to minimize the error between the predicted and actual outputs [30]. Regarding the optimization solver, I opted to use the default option, which is L-BFGS-B. This optimizer is a member of the quasi-Newton methods family, known for its efficiency in handling large-scale problems.

As previously emphasized, determining the number of neurons in different layers of a neural network is a crucial step in designing an effective machine learning model, thus to address this, I followed guidelines from Orr and Muller's "Neural Networks: Tricks of the Trade" [31] to optimize and personalize the number of neurons in the input layer, hidden layers, and output layer. Here the guidelines that I took in count to select the right number of neurons for each layer:

1. **Number of Input Neurons**

   The number of neurons in the input layer is determined by the number of features in the dataset. Each neuron in the input layer corresponds to one feature.

   $$\text{Number of input neurons} = \text{Number of features in the dataset}$$

   The number of input neurons is determined by the features selected in the previous step, to ensure consistency and facilitate comparison across different sensor placements, I set the number of input neurons to 7, which accommodates all feature selections. This standardization allows for a fair evaluation of the neural network's performance across varying feature sets, ensuring that performance differences are attributable to the model itself rather than variations in the input configuration.

2. **Number of Output Neurons**

   The number of neurons in the output layer depends on the type of problem you are trying to solve: for classification problems, the number of output neurons typically corresponds to the number of classes.

$$\text{Number of output neurons} = \text{Number of classes}$$

Training neural networks to estimate posteriors typically requires a 1-out-of-N coding of training targets, where the number of output neurons must match the number of classes. However, training a neural network with thousands of output neurons is impractical. As the number of classes increases, the complexity of the optimal discriminant functions also increases, raising the potential for conflicts between classes. Therefore, while neural network classifiers are effective for tasks with relatively few classes, they become impractical for tasks involving a large number of classes due to these limitations.

In this case, the number of output neurons is predetermined by the different motor tasks to be recognized, which correspond to the number of target classes. Consequently, the number of output neurons is set to 4, matching the 4 target classes.

3. **Number of Neurons in Hidden Layers**

   Determining the number of neurons in the hidden layers is a crucial step in designing a neural network architecture as it significantly influences the model's ability to learn and generalize. Hidden layers, although not directly interacting with the external environment, have a tremendous impact on the final output.

While there is no fixed rule, a common empirical approach involves starting with a moderate number of neurons and then adjusting based on the model's performance, typically selecting a number that falls between the input and output neurons [31]. Common practices and experimental methods can provide valuable guidance in this process [32]. To optimize the number of neurons in the hidden layers, I combined two different approaches, which I will describe here:

- **Trial and Error Method:**

  This method involves repeated attempts until optimal performance is achieved and can be divided into two strategies:

  - **Forward Approach:** This approach starts with a small number of hidden neurons, such as two, and gradually increases the number, training and testing each time until performance improves.
  - **Backward Approach:** This approach begins with a large number of hidden neurons and gradually decreases the number, training and testing each time until performance improves.

- **Rule of Thumb:**

  The rule of thumb suggests that the number of hidden neurons should be between the size of the input and output layers, often using $\frac{2}{3}$ of the input layer size plus the output layer size, and less than twice the input layer size [32].

I selected a combining approach based on both the Trial and Error Method and the Rule of Thumb empirical methods. Initially, a trial configuration of

9 neurons in each of the two hidden layers was chosen, based on the following calculation:

$$\text{Number of neurons} \approx \frac{2}{3} \times 7 + 4 \approx 9$$

Subsequently, following a forward approach, the Trial and Error Method was used to reach the best configuration, resulting in 13 neurons in the first hidden layer and 9 neurons in the second hidden layer.

The design of MLP, as describe above, is represented then in the following Figure 29, where it's possible to see in detail the architecture of the network with all the different layers and links.

Input Neurons	7
Neurons in the 1st Hidden Layer	13
Neurons in the 2nd Hidden Layer	9
Output Neurons	4

Table 11: Neural network architecture

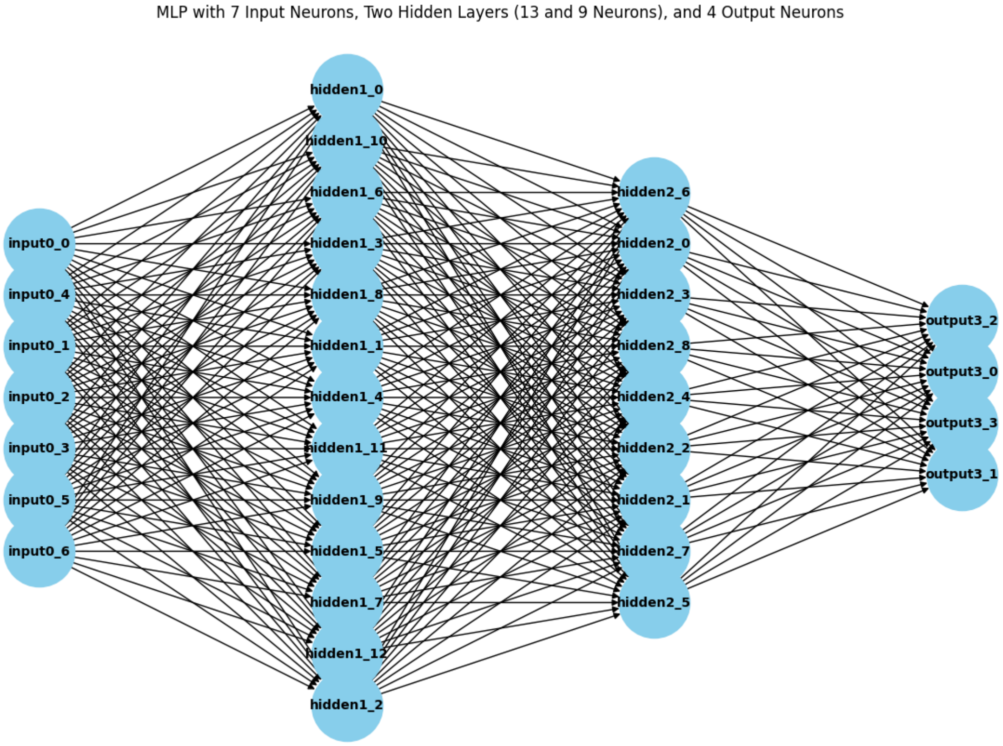


Figure 29: *Visualization of a Multi-Layer Perceptron (MLP) with 7 input neurons, two hidden layers (with 13 and 9 neurons respectively), and 4 output neurons. The diagram illustrates the structure and connectivity between the layers.*

6 Evaluation of the Model

To evaluate the model comprehensively, I utilized another Orange widget, `Prediction`. The performance of the MLP model is evaluated and compared to a Logistic Regression model, with both models trained on the same dataset.

The analysis focuses on demonstrating the MLP's superior performance, primarily attributed to its capability to combine multiple `tanh` activation functions. These functions are more effective for the learning process with the dataset under analysis compared to the logistic function σ employed by the Logistic Regression model.

Training and testing both models on the same data could lead to the problem of overfitting, where the model becomes too attached to the training data and struggles to generalize to new, unseen data. To avoid this issue, I adopted a cautious approach: instead of training and testing on the same dataset, I used manual random sampling. The random sampling of the dataset was performed using the `Random Sampler` widget, where I decided to randomly split my dataset into two parts— a Training Set and a Test Set. I allocate 70% of the data to the Training Set and the remaining 30% to the Test Set. Moreover, to make this process even more robust, each split is non-replicable: every time I perform the split, it is different, introducing variability into my training and testing subsets. By doing this, I'm not only avoiding overfitting but also allowing for a more comprehensive evaluation of my models. This variability in training and testing data enables me to gauge how well my models generalize to new data, ultimately enhancing their classification capabilities and providing more reliable assessments of their performance.

The evaluation of the models' performances was conducted using the `Predictions` widget, which generates a report containing key metrics for assessing the models' effectiveness. Initially, I focused on analyzing the `Classification Accuracy (CA)` of the model. This metric measures the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances. Although it provides a general indication of the classifier's performance, CA can be misleading when applied to imbalanced datasets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

Initially, I evaluated the CA to gauge the accuracy of the neural network model, as recommended in relevant literature [6]. After achieving a satisfactory level of accuracy, I proceeded to evaluate additional metrics to further assess the model's performance and robustness:

- **AUC**: Evaluates binary classifiers by measuring the area under the ROC curve, plotting sensitivity against (1 - specificity). An AUC of 0.5 indicates random performance, while 1.0 indicates perfect discrimination. It is typically computed using numerical integration.

- **F1 Score**: The harmonic mean of precision and recall, useful for imbalanced datasets as it balances both metrics.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{2}$$

- **Precision**: The ratio of true positive instances to the sum of true positive and false positive instances. Important when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

- **Recall(Sensitivity)**: The ratio of true positive instances to the sum of true positive and false negative instances. Important when the cost of false negatives is high.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

- **MCC**: A balanced measure considering true and false positives and negatives, suitable for imbalanced datasets. The MCC ranges from -1 (total disagreement) to 1 (perfect prediction).

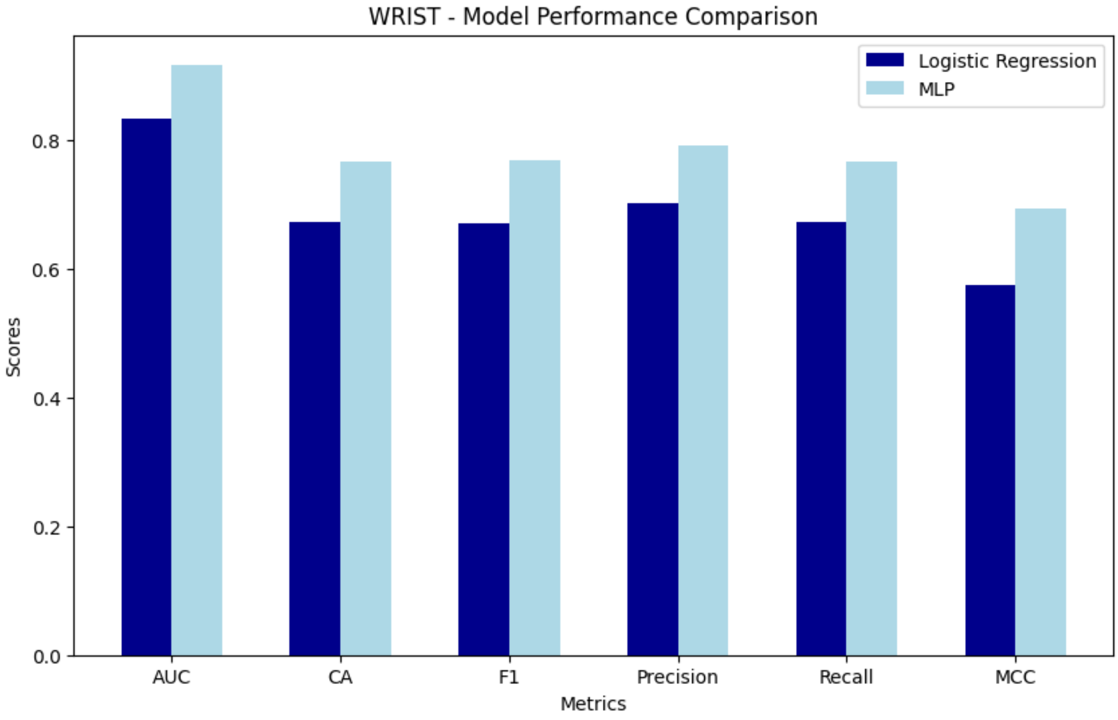
$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

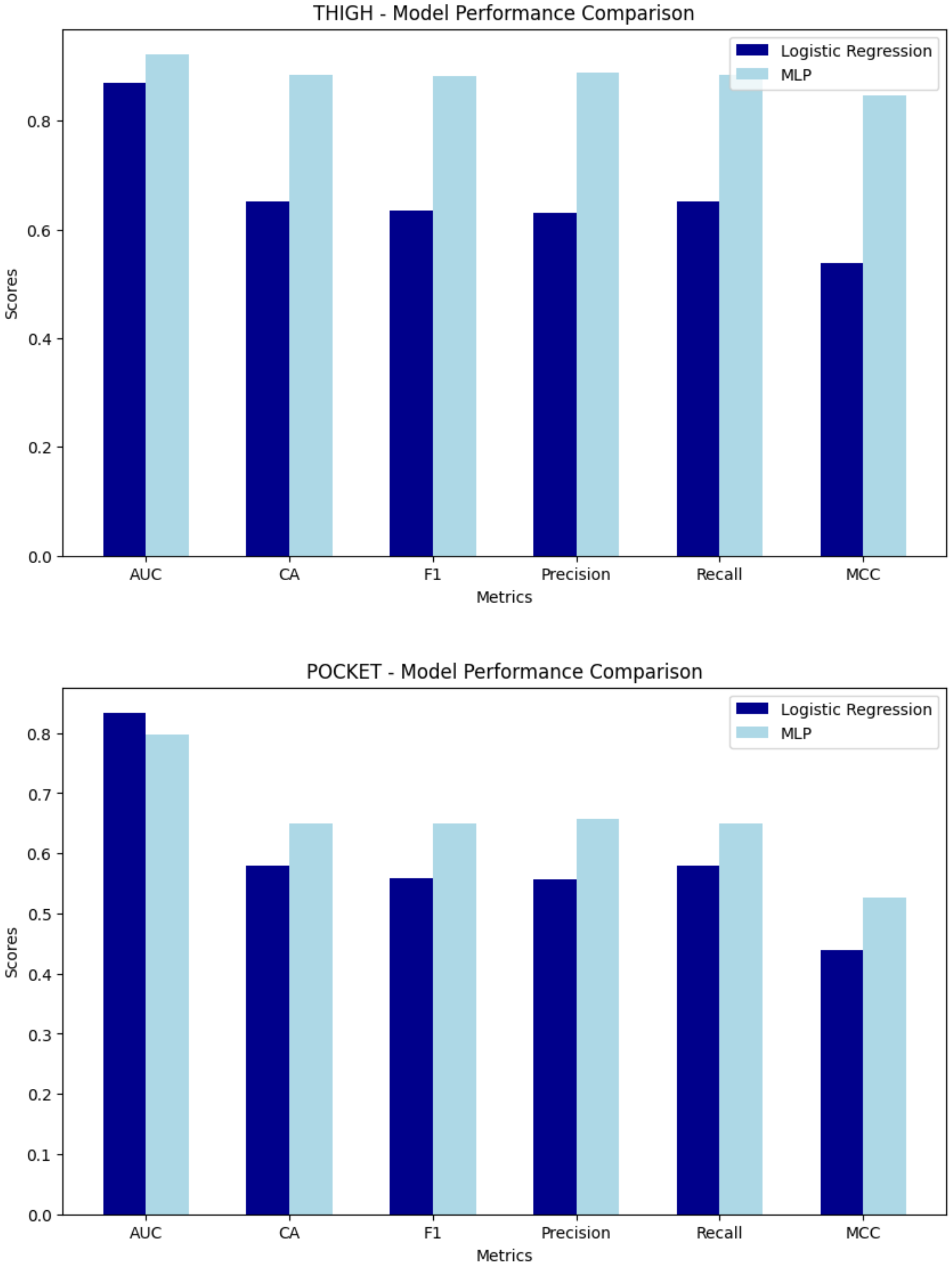
These metrics provide a comprehensive toolkit for evaluating and comparing the performance of classification models in various contexts [22]. The resulting performance of the two models are reported in the following table 12:

Model	Placement	AUC	CA	F1	Prec	Recall	MCC
Logistic Regression	WRIST	0.834	0.674	0.671	0.702	0.674	0.576
	THIGH	0.869	0.651	0.634	0.630	0.651	0.538
	POCKET	0.834	0.579	0.558	0.556	0.579	0.439
MLP	WRIST	0.918	0.767	0.770	0.793	0.767	0.694
	THIGH	0.923	0.884	0.882	0.889	0.884	0.847
	POCKET	0.798	0.649	0.650	0.657	0.649	0.527

Table 12: Prediction of the goodness of the model based on the performance on the data with the random sampling process with a ratio 70:30.

To provide a clearer representation of the performance of the two models, I have depicted in Figure 31 the bar charts corresponding to the values in table 12. Each chart visually compares the performance of the Logistic Regression and MLP models across various metrics and sensor placements.




THIGH - Model Performance Comparison

POCKET - Model Performance Comparison

Figure 31: *Comparison of model performance across different sensor placements. The first figure illustrates the performance metrics (AUC, CA, F1, Precision, Recall, MCC) for the WRIST placement, the second figure for the THIGH placement, and the third figure for the POCKET placement.*

As illustrated in Figure 31, the MLP model consistently surpasses the Logistic Regression model across the dataset, demonstrating higher accuracy and sensitivity. This performance aligns with expectations, given the MLP's ability to model complex patterns effectively. To ensure that these observations are both robust and reliable, it is crucial to perform a comprehensive evaluation of the models.

For a more accurate assessment, it is essential to avoid relying solely on metrics from a single random sampling instance. Instead, repeating the evaluation process and computing average metrics over multiple trials provides a more reliable estimate of model performance. This approach is motivated by the need to capture a true reflection of the model's generalization ability rather than potential anomalies in a single dataset split. Cross-validation is employed as a fundamental technique to achieve this robust evaluation. This method involves partitioning the dataset into several subsets, typically into training, validation, and testing sets. The training set is used to fit the model, the testing set evaluates performance during training, and the validation set provides an unbiased measure of generalizability. In practice, k-fold cross-validation is applied by dividing the dataset into k folds. The model is trained on $k-1$ folds and validated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once, and the results are averaged to provide a more accurate performance estimate. I chose to use 10-fold cross-validation to ensure a reliable assessment, minimizing potential biases and overfitting associated with a single train-test split.

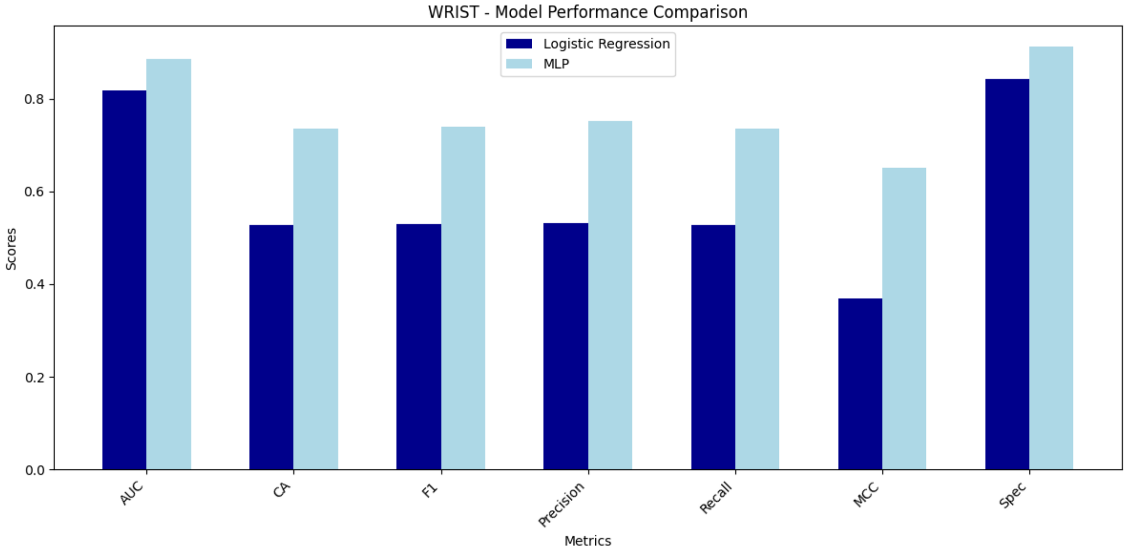
While cross-validation offers insights into model accuracy across a broad dataset, scoring on distinct datasets further verifies the model's generalizability. The evaluation was conducted using the `Test and Score` widget, which facilitated a detailed analysis of the models' performance and helped confirm their robustness. The metrics obtained through these methods are presented in tables 13 and 14. These comprehensive evaluations are essential for developing models that not only perform well on training data but also generalize effectively to new, unseen data. This rigorous approach underscores my commitment to ensuring that the model developed in more both accurate and robust, providing reliable predictions in practical applications.

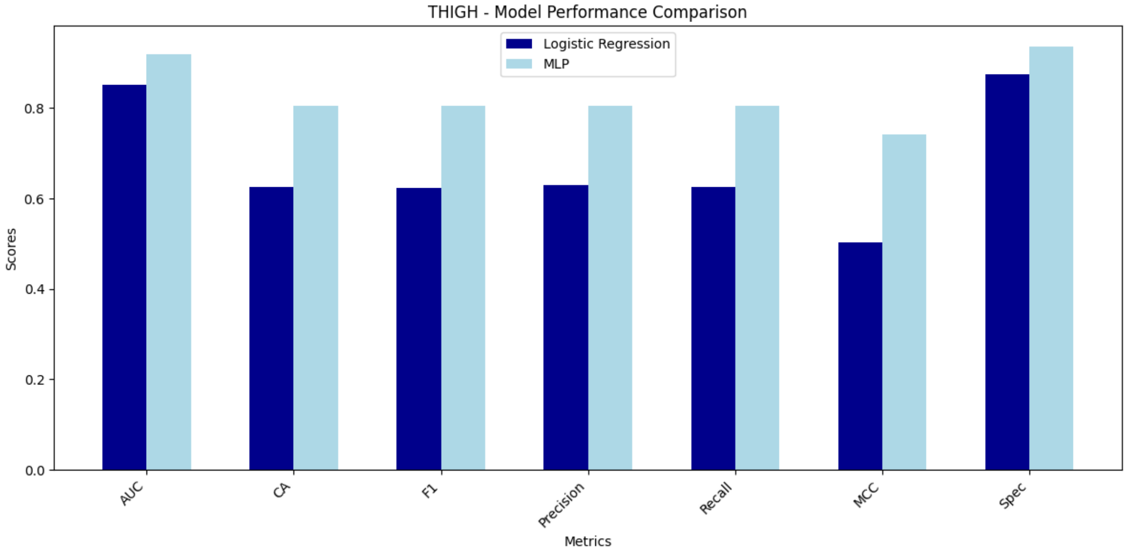
Placement	**AUC**	**CA**	**F1**	**Prec**	**Recall**	**MCC**	**Spec**
WRIST	0.886	0.736	0.739	0.752	0.736	0.651	0.912
THIGH	0.919	0.806	0.805	0.805	0.806	0.741	0.935
POCKET	0.809	0.625	0.621	0.621	0.625	0.501	0.875

Table 13: Performance Evaluation of the Multilayer Perceptron Model Using the cross-validation process

Placement	**AUC**	**CA**	**F1**	**Prec**	**Recall**	**MCC**	**Spec**
WRIST	0.819	0.528	0.530	0.532	0.528	0.370	0.843
THIGH	0.852	0.625	0.624	0.630	0.625	0.502	0.875
POCKET	0.816	0.583	0.559	0.556	0.583	0.451	0.861

Table 14: Performance Evaluation of the Logistic Regression Model using the cross-validtion process


WRIST - Model Performance Comparison
Logistic Regression
MLP
Scores
0.0
0.2
0.4
0.6
0.8
AUC
CA
F1
Precision
Recall
MCC
Spec
Metrics


THIGH - Model Performance Comparison
Logistic Regression
MLP
Scores
0.0
0.2
0.4
0.6
0.8
AUC
CA
F1
Precision
Recall
MCC
Spec
Metrics

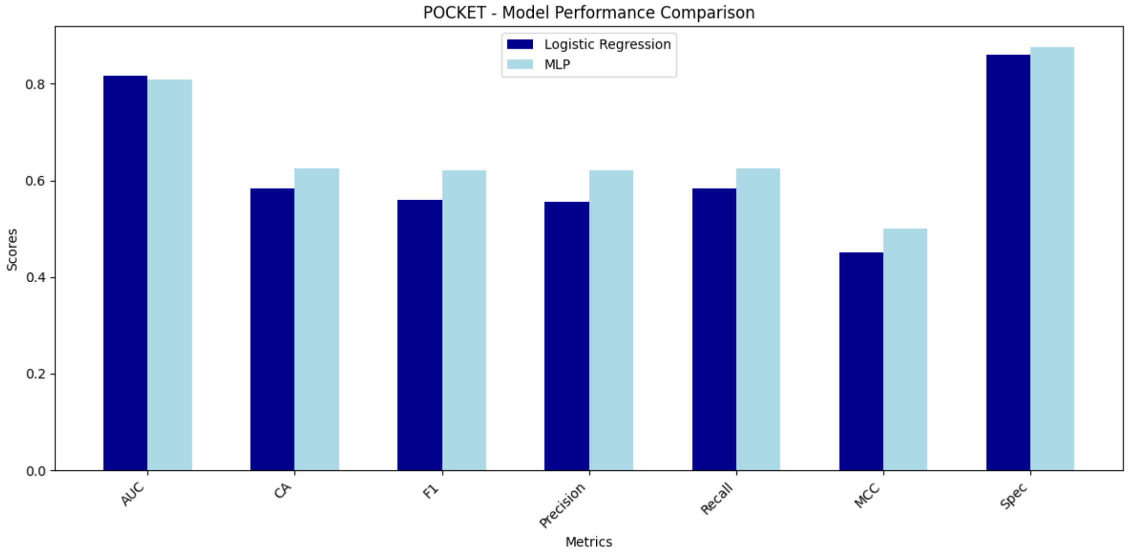


Figure 33: *Comparison of model performance across different sensor placements through the cross-validation process. The first figure illustrates the performance metrics (AUC, CA, F1, Precision, Recall, MCC) for the WRIST placement, the second figure for the THIGH placement, and the third figure for the POCKET placement.*

Based on the provided tables (13 for MLP and 14 for LR), along with the corresponding bar charts in Figure 33, a comparative analysis clearly demonstrates that the Neural Network (MLP) consistently outperforms Logistic Regression (LR) across various performance metrics. Specifically, the MLP model achieves higher scores in AUC, CA, F1 Score, Precision, Recall, and MCC compared to the LR model. This suggests that the MLP is more effective in handling classification tasks, offering greater robustness and accuracy across different sensor placements. In detail, the MLP achieves notably higher AUC (0.919 vs. 0.852 for THIGH, 0.886 vs. 0.819 for WRIST, and 0.809 vs. 0.816 for POCKET), which demonstrates its ability to distinguish between classes more effectively. The classification accuracy (CA) also shows a clear advantage for MLP, with scores of 0.806 vs. 0.625 for THIGH, 0.736 vs. 0.528 for WRIST, and 0.625 vs. 0.583 for POCKET. Similarly, the F1 Score, Precision, Recall, and MCC metrics favor MLP, reinforcing its superior performance across various evaluation criteria. On the other hand, LR, while demonstrating respectable performance, falls short in achieving the high scores seen with MLP, particularly in scenarios involving different sensor placements. This suggests that for tasks requiring high accuracy and reliability in activity classification, the MLP model is more suitable based on the provided performance metrics.

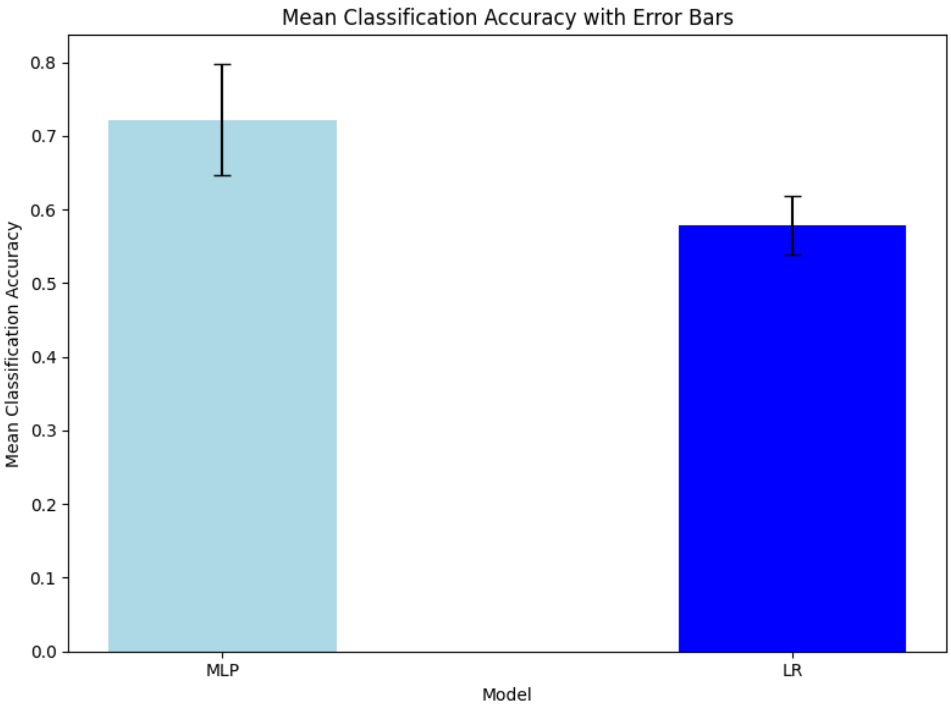


Figure 34: *Comparison of Neural Network (MLP) and Logistic Regression (LR) models by Classification Accuracy (CA) across different sensor placements (Wrist, Thigh, Pocket). The bars represent the average classification accuracy for each model, with error bars indicating the variance. The MLP consistently outperforms LR across all placements, with the wrist sensor showing the highest accuracy for both models.*

The inclusion of error bars in Figure 34 illustrates the variability or uncertainty associated with the classification accuracy measurements for both the Neural

Network (MLP) and Logistic Regression (LR) models across different sensor placements (Wrist, Thigh, Pocket). For each model, the error bars indicate the standard deviation in accuracy values across the different sensor placements, providing insight into the consistency of model performance. For the MLP model, the mean classification accuracy is 0.722, with a standard deviation of approximately 0.075. For the LR model, the mean classification accuracy is 0.579, with a standard deviation of approximately 0.040. The error bars thus represent the dispersion of accuracy values around these mean accuracies.

Understanding this variance is crucial for evaluating the stability and robustness of each model. The error bars highlight the extent of variability in accuracy across sensor placements, reflecting the consistency of model performance. A narrower spread of error bars suggests greater consistency and higher reliability in classification accuracy. For instance, the smaller variance in LR indicates a more stable performance across different sensor placements compared to MLP. By incorporating error bars, the figure offers a more nuanced interpretation of the results. While the mean accuracy values provide a central measure of performance, the error bars reveal how much the accuracy fluctuates. This helps assess how well each model performs under varying conditions and with different sensor data, providing essential insight into the robustness and practical applicability of the models in real-world scenarios.

The `Test and Score` widget allows for a detailed comparison of models using various metrics. In this analysis, I decided to delve whit a more precise pairwise comparison to evaluate each model based on Classification Accuracy (CA). The following table (table 15) presents the results of this pairwise comparison, including the negligible difference values:

Comparing Models by Classification Accuracy			
Placement		**Neural Network (NN)**	**Logistic Regression**
Wrist	**Neural Network (NN)**		0.984 (0.016)
	Logistic Regression	0.000 (0.016)	
Thigh	**Neural Network (NN)**		0.878(0.121)
	Logistic Regression	0.001 (0.121)	
Pocket	**Neural Network (NN)**		0.254(0.685)
	Logistic Regression	0.061(0.685)	

Table 15: Pairwise comparison of models using the selected score (available only for cross-validation). The number in the table gives the probability that the model corresponding to the row has a higher score than the model corresponding to the column. The smaller number below shows the probability that the difference between the pair is negligible.

The probability table provides a comprehensive comparison of Classification Accuracy (CA) between Neural Network (NN) and Logistic Regression (LR) classifiers across different placements: Wrist, Thigh, and Pocket. Each cell contains two values: the upper value represents the probability that the NN model outperforms the LR model regarding CA. the , while the lower value in parentheses indicates the probability that the difference in scores between the two models is negligible, providing insights into whether the performance differences are statistically significant or likely due to random variation.

Analysis of the Pairwise Comparison:

- **Wrist Placement:** The NN model shows a very high probability (0.984) of outperforming the LR model in CA, with a negligible difference probability of 0.016. This suggests that the observed performance difference is statistically significant.

- **Thigh Placement:** The NN model also exhibits a strong probability (0.878) of achieving higher CA compared to the LR model, with a negligible difference probability of 0.121. This indicates that while the NN model generally performs better, the difference might not be substantial in some cases.

- **Pocket Placement:** The probability that the NN model outperforms the LR model is lower (0.254), and the negligible difference probability is quite high (0.685). This suggests that the performance gap between NN and LR is less pronounced at this placement and could be attributed to random variation.

Overall, the analysis underscores that while the Neural Network model generally offers superior performance, the extent of its advantage varies with different sensor placements. The negligible differences in performance probabilities further emphasize that, apart from the Wrist placement, the performance disparities between the models are less likely to be statistically significant. This nuanced understanding highlights the importance of considering specific conditions and placements when evaluating and selecting the most appropriate model for a given application.

In particular, the comparable performance of the Logistic Regression model in the Pocket placement may be attributed to the nature of the data collected. It is plausible that the data from the Pocket placement exhibit greater homogeneity due to increased noise levels, which can affect the discriminative power of the models. This noise may result in more consistent accuracy measurements for the Logistic Regression model, as it might better handle the variability introduced by the data.

In contrast, the Neural Network, which typically excels in distinguishing between features in cleaner datasets, might not leverage its strengths as effectively under these conditions. Thus, the relatively uniform data quality at the Pocket placement could favor the Logistic Regression model, leading to performance levels that are comparable to, or even better than, the Neural Network in this specific context.

6.1 Confusion Matrix

To assess the effectiveness of the two models in classifying the different motor classes across the various subjects analyzed, I computed the corresponding confusion matrices for each sensor placement. This approach allows a direct evaluation of the classifiers' performance, highlighting which classes have the highest rates of misclassification. By identifying these problematic areas, I could investigate potential reasons that lead to these errors and suggest improvements for the neural network model in future studies. The confusion matrices, which were generated using the Orange widget `Confusion Matrix`, are presented in the following tables.

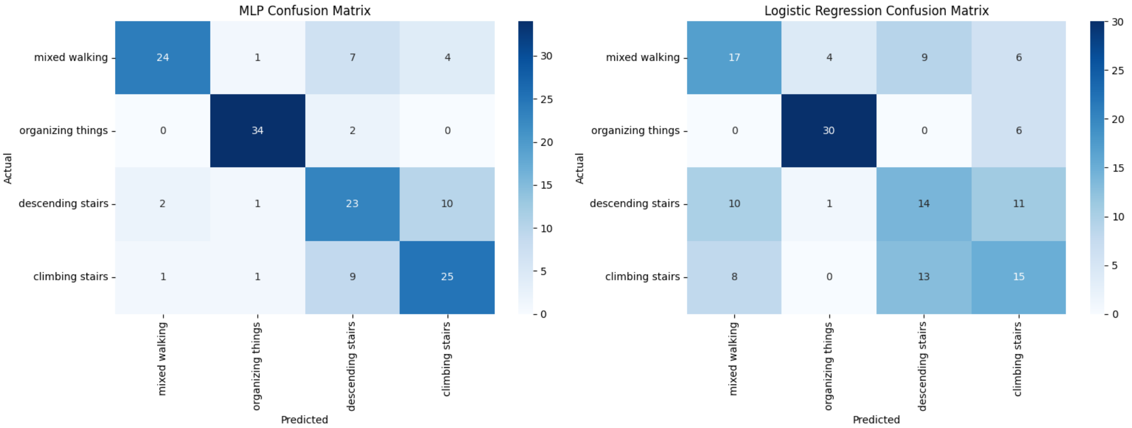


Figure 35: *Comparative analysis of the Neural Network (MLP) and Logistic Regression (LR) models for wrist sensor placement. The confusion matrix displays the classification performance of each model across different motor classes. The table provides insights into which classes are most frequently misclassified, helping to identify areas for potential model improvement.*

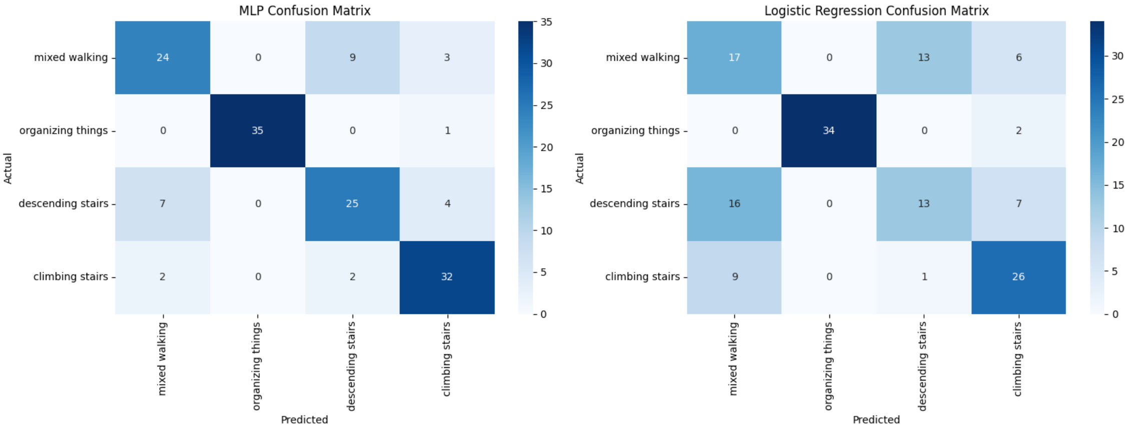


Figure 36: *Confusion matrix for the Neural Network (MLP) and Logistic Regression (LR) models applied to thigh sensor placement. This figure illustrates the accuracy of each model in classifying motor activities, highlighting the rate of misclassification for each motor class.*

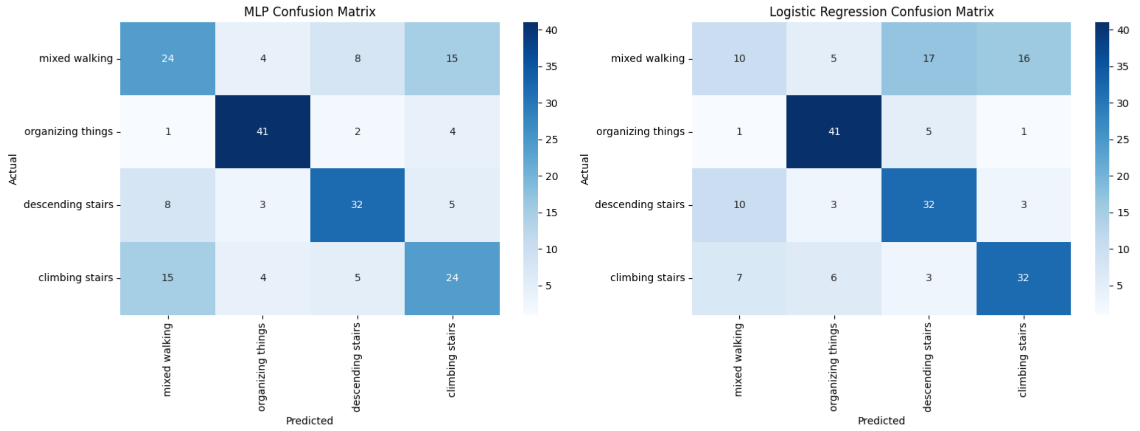


Figure 37: *Confusion matrix for the Neural Network (MLP) and Logistic Regression (LR) models using pocket sensor placement. The matrix visualizes the performance of the classifiers, showing the distribution of correct and incorrect classifications across motor classes.*

6.1.1 Sensor Location:Pocket

MLP Classifier

- **Mixed Walking:** The classifier correctly identified 21 instances of mixed walking, with some confusion with climbing stairs(15 misclassifications) and descending stairs (8 misclassifications).
- **Organizing Things:** This class was well recognized, with 41 correct predictions out of 46 instances, indicating strong performance.
- **Descending Stairs:** 32 instances were correctly classified, but some confusion was present with climbing stairs (5 misclassifications) and mixed walking (8 misclassifications).
- **Climbing Stairs:** Only 24 out of 48 instances were correctly classified, with significant confusion with mixed walking (15 instances).

Logistic Regression

- **Mixed Walking:** Logistic Regression struggled with this class, correctly identifying only 10 instances. Confusion occurred with climbing stairs (16 instances) and descending stairs (17 instances).
- **Organizing Things:** The model performed well, with 41 correct predictions out of 49.
- **Descending Stairs:** 32 correct predictions were made, though confusion occurred with mixed walking (10 instances).
- **Climbing Stairs:** 32 instances were correctly classified, but confusion remained with mixed walking (7 instances) and organizing things (6 instances).

The MLP model demonstrated better performance than Logistic Regression in recognizing most activities using the pocket sensor. It excelled particularly in identifying "Organizing Things" with high accuracy. However, both models struggled with "Mixed Walking" which was frequently confused with "Climbing Stairs " and " Descending Stairs". The MLP model had fewer misclassifications compared to Logistic Regression, especially in "Mixed Walking" and "Climbing Stairs". This suggests that the MLP model is more reliable for tasks involving the pocket sensor, although there is still room for improvement in distinguishing between similar activities.

6.1.2 Sensor Location: Wrist

MLP Classifier

- **Mixed Walking:** The classifier improved with 24 correct identifications, though confusion persisted with descending stairs (7 instances) and climbing stairs(3 instances).
- **Organizing Things:** Almost perfect recognition was achieved, with 34 correct identifications out of 36.
- **Descending Stairs:** 23 instances were correctly classified, with some confusion with climbing stairs (10 instances).
- **Climbing Stairs:** Recognition improved, with 25 correctly classified instances and minimal confusion respect to the pocket placement.

Logistic Regression

- **Mixed Walking:** 17 instances were correctly identified, though confusion remained with descending stairs (9 instances) and climbing stairs(6 instances).
- **Organizing Things:** 30 instances were correctly classified, with 6 misclassification.
- **Descending Stairs:** Only 14 correct classifications were made, with notable confusion between climbing stairs (11 instances) and mixed walking (10 instances).
- **Climbing Stairs:** Only 15 correct classifications were made , with particular confusion with descending stairs (13 instances).

The wrist sensor location revealed that the MLP model consistently outperformed Logistic Regression, particularly in recognizing "Organizing Things" and "Climbing Stairs ." While the MLP model showed improvements in classifying "Mixed

Walking" compared to the pocket sensor, some confusion with "Descending Stairs" and "climbing stairs" remained. Logistic Regression lagged behind significantly in identifying "Descending Stairs" and "Climbing Stairs " showing more frequent misclassifications in these categories. The wrist sensor thus benefits more from the MLP model, which better handles the complexity of the activities, though minor misclassification issues persist

6.1.3 Sensor Location: Thigh

MLP Classifier

- **Mixed Walking:** The model maintained 24 correct classifications, though confusion persisted with descending stairs(9 instances).
- **Organizing Things:** Strong recognition was achieved with 35 correct predictions, though 1 instances were misclassified.
- **Descending Stairs:** 25 instances were correctly classified, with some confusion across climbing stairs (4 instances) and mixed walking (7 instance).
- **Climbing Stairs:** The classifier correctly identified 32 instances, with some confusion with descending stairs(2 instances) and mixed walking(2 instances).

Logistic Regression

- **Mixed Walking:** The model correctly classified 17 instances, but showed confusion across descending stairs (13 instances) and climbing stairs (6 instances).
- **Organizing Things:** 34 correct predictions were made, with 2 misclassifica-

tions.

- **Descending Stairs:** The model struggled, with only 13 correct predictions, and confusion with mixed walking (16 instances).
- **Climbing Stairs:** Logistic regression performed correctly classifying 26 instances, with confusion with the mixe walking task (9 instances).

The thigh sensor provided relatively stable performance for the MLP model, which effectively recognized "Organizing Things" and "Climbing Stairs" similar to the wrist sensor. However, a slight decline in performance was observed in classifying "Descending Stairs" with some confusion with "Mixed Walking" and "Climbing Stairs." Logistic Regression continued to struggle with "Mixed Walking," "Descending Stairs," and "Climbing Stairs" showing higher rates of misclassification compared to the MLP model. Overall, the thigh sensor's performance highlights the MLP model's superior ability to handle complex activities, but it also points to the need for further refinement to improve accuracy, especially for activities that share similar movement patterns.

Overall considerations

In summary, the MLP model consistently outperformed Logistic Regression in recognizing the four activities, with particularly strong performance in identifying "Organizing Things" and "Climbing Stairs." The wrist sensor proved to be the most effective across both models, showing high accuracy in classifying "Organizing Things" and "Climbing Stairs." However, both models struggled with accurately classifying "Mixed Walking," which was frequently confused with "Descending

Stairs” and ”Climbing Stairs ” due to the similarities in movement patterns. Upon reflection, I attribute this difficulty in recognizing ”Mixed Walking” to the fact that participants emulated normal walking patterns during the mixed walking task, with each individual exhibiting a unique gait. This variability likely disrupted the establishment of a clear recognition pattern, making it more challenging to differentiate ”Mixed Walking” from other motor tasks. This insight underscores the importance of considering individual differences in gait when developing and refining activity recognition models, as these variations can significantly impact the accuracy of classification, particularly for activities with overlapping characteristics.

7 Conclusions

The main aim of this study was to advance the field of human activity recognition (HAR) by utilizing a combination of Inertial Measurement Unit (IMU) and Electromyography (EMG) signals. These sensors, although not considered the gold standard for HAR, offer several advantages, such as lower cost and the ability to be used in various settings beyond controlled lab environments. The research focused on developing and evaluating machine learning (ML) models capable of accurately classifying different motor activities based on data collected from these wearable sensors. The study was motivated by the growing demand for reliable activity recognition systems in areas like healthcare monitoring, rehabilitation, and sports performance analysis, with particular emphasis on detecting activities suitable for patients with chronic diseases. Consequently, the target motor tasks selected were designed to be feasible for a wide range of individuals and useful in monitoring the progression of chronic conditions.

The study began with an extensive review of existing methods in activity recognition, underscoring the potential benefits of combining IMU and EMG signals. This was followed by the design and implementation of a comprehensive data processing pipeline. The pipeline encompassed stages such as data acquisition, noise reduction, and feature extraction, with a focus on deriving meaningful features from both the time and frequency domains of the collected signals. These features were then used as inputs for the machine learning models developed in the next phase of the research. At the core of the study was a comparative analysis of two machine learning algorithms: Logistic Regression and Multi-Layer Perceptron

(MLP). Logistic Regression, a linear model, was chosen for its simplicity and interpretability, while the MLP, a type of neural network, was implemented for its ability to model complex, non-linear relationships in the data. The MLP was specifically chosen for its customizability and its ability to use the hyperbolic tangent function as the activation function, which is more suitable than the sigmoid function—commonly used in Logistic Regression—when dealing with datasets with higher variability. The performance of these models was evaluated based on different sensor placements on the body, aiming to determine their effectiveness in recognizing four distinct motor activities. The results demonstrated the superior performance of the MLP model across all sensor locations, particularly in recognizing complex activities such as *stairs-related activities (climbing and descending stairs).* The MLP consistently outperformed Logistic Regression in key metrics, including accuracy, AUC, F1 Score, Precision, Recall, and MCC, showcasing its robustness in capturing the intricate patterns within the data.

Among the various sensor placements, the *wrist sensor* proved to be the most effective for both models, offering the most consistent and accurate recognition of activities like *organizing things* and *climbing stairs.* However, even with this sensor, both models struggled to accurately classify *mixed walking*, with Logistic Regression showing particular difficulty.

The *mixed walking* class was the most challenging to classify correctly across all sensor placements and models. This class was frequently confused with *descending stairs* and *climbing stairs*, likely due to the similar movement patterns involved in these activities. This challenge highlights the need for more sophisticated models or additional data sources to improve classification accuracy for such complex activities.

A specific challenge of this study was evaluating the performance of the sensor placed in the pocket, which emerged as the least accurate positioning compared to the wrist and thigh placements. Although the pocket sensor allowed for natural movement without restrictions, this freedom led to significant variations in sensor orientation, which affected the consistency of data capture. As a result, both the MLP and Logistic Regression models faced difficulties in accurately classifying activities like *mixed walking* and *climbing stairs* when using data from the pocket sensor. The comparable performance of Logistic Regression in the pocket placement is likely due to the increased noise in the data, which may have made the data more homogeneous. This homogeneity could reduce the advantage typically provided by the MLP's ability to capture non-linear dependencies, leading to a better-than-expected performance of Logistic Regression. Despite these challenges, the analysis indicated that the MLP model still managed to achieve a moderate level of accuracy, outperforming Logistic Regression in most cases. However, the overall performance of the pocket sensor was lower than the other placements, suggesting that while the results were acceptable, they were not optimal for precise activity recognition.

Based on the analysis, the MLP model is recommended as the most suitable for the classification tasks addressed in this study. Its ability to capture non-linear dependencies in the data makes it particularly well-suited for recognizing complex activities. Future research should focus on refining this model or exploring more advanced neural network architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), to further enhance recognition rates, especially for the difficult *mixed walking* class.

In conclusion, this study demonstrates that both sensor placement and the choice of machine learning models significantly influence the performance of human activity recognition systems. The MLP model, particularly when combined with wrist-mounted sensors, shows great potential for providing reliable and accurate activity recognition. However, challenges remain, particularly in distinguishing between activities with similar movement patterns. Addressing these challenges will require further research, possibly involving more advanced machine learning techniques or additional sensor modalities, to improve the robustness and accuracy of activity recognition systems in real-world applications.

Appendix A

Matlab R2024a scripts

The analysis employed the following functions for feature extraction.

The function `fmedian` calculates the median frequency of a given signal epoch. It takes three inputs: the signal epoch x, the sampling frequency f_{sam}, and the length of the epoch in seconds epoch_len.

```matlab
function fmedianv = fmedian(x, fsamp, epoch_len)
% Function to calculate the median frequency of a signal epoch.
% Input parameters:    x             signal epoch (vector)
%                      fsamp         sampling frequency
%                      epoch_len     epoch length (in seconds)
% Output parameters:   fmedianv      median frequency

% Calculate P using a simple periodogram
x = x - mean(x);

win = rectwin(length(x));
noverlap = 0;
fNy = fsamp / 2;
```

```
df = 1 / epoch_len;
NFFT = fsamp / epoch_len;

[P, f] = pwelch(x, win, noverlap, NFFT, fsamp);

A = sum(P) / 2;
i = 1;
S = 0;
while S < A
    S = S + P(i);
    i = i + 1;
end

fmedianv = (f(i) + f(i - 1)) / 2;
end
```

Listing 1: MNP function's script

The function **fmean** calculates the mean frequency of a given signal epoch. It takes three inputs: the signal epoch x, the sampling frequency f_{samp}, and the length of the epoch in seconds epoch_len.

```
function fmeanv = fmean(x, fsamp, epoch_len)
% Function to calculate the mean frequency of a signal epoch.
% Input parameters:    x              signal epoch (vector)
%                      fsamp          sampling frequency
%                      epoch_len      epoch length (in seconds)
% Output parameters:   fmeanv         mean frequency
NFFT=fsamp/epoch_len;

% Body of the function
x=x-mean(x);
[Pxx,f]=pwelch(x,[],[],NFFT,fsamp);
fmeanv=sum(f.*Pxx)/sum(Pxx);

end
```

Listing 2: MDP function's script

```
% Hjorth parameters
% Activity
Act = var(EMG_fil);
```

```

% Mobility
Mob = sqrt(var(diff(EMG_fil))./var(EMG_fil));

% Complexity
Com = sqrt(var(diff(diff(EMG_fil)))./var(EMG_fil));

% signal parameters
fs = 1000; % sample frequency

% Evaluation of the Power spectrum
[pxx, f] = pwelch(EMG_fil, [], [], [], fs);

% Defining the cut-off frequencies
f1 = 10; % Low cut-off frequency (Hz)
f2 = 500; % High cut-off frequency (Hz)

% Finding indices for f1 and f2
idx_f1 = find(f >=f1, 1);
idx_f2 = find(f <= f2, 1, ’last’);

% Evaluation of the power spectrum of interest (f1 - f2)
P_f = sum(pxx(idx_f1:idx_f2));

% Evaluation of the total Power content (0 - f_max)
```

```
P_t = sum(pxx);

% Spectral Purity Index
SPI = P_f / P_t;

% MEAN ABSOLUTE VALUE
MAV = sum(abs(x))/(length(x)/fsamp);
% Root Mean Square
RMS=std(x)
```

Listing 3: Hijort parameters' script

```
% Create feature extractor objects
meanFE = signalTimeFeatureExtractor("Mean", true, "SampleRate", fs);

timeFE = signalTimeFeatureExtractor("RMS", true, ...
    "StandardDeviation", true, ...
    "ShapeFactor", true, ...
    "PeakValue", true, ...
    "CrestFactor", true, ...
    "ClearanceFactor", true, ...
    "ImpulseFactor", true, ...
    "SampleRate", fs);

```

```
freqFE = signalFrequencyFeatureExtractor("PeakAmplitude", true, ...
    "PeakLocation", true, ...
    "MeanFrequency", true, ...
    "BandPower", true, ...
    "PowerBandwidth", true, ...
    "SampleRate", fs);

% Create datastores
meanFeatureDs = arrayDatastore(Data’, "IterationDimension", 2);
timeFeatureDs = arrayDatastore(Filtered’, "IterationDimension", 2);

% Extract features using the datastores and feature extractors
meanFeatureDs = transform(meanFeatureDs, @(x) meanFE.extract(x{:}));
timeFeatureDs = transform(timeFeatureDs, @(x) timeFE.extract(x{:}));

% Read the features
meanFeatures = readall(meanFeatureDs, "UseParallel", true);
timeFeatures = readall(timeFeatureDs, "UseParallel", true);

```

```
% Combine the features
features = [meanFeatures timeFeatures];

% Convert channel labels to categorical
Channels = cellstr(Channels);
categories = categorical(Channels);
disp(categories);

% Create the feature table
featureTable = array2table(features);
actioncats = categorical(categories)’;
% featureTable.ActivityID = ’’; % Uncomment and adapt if necessary

% Display the first rows of the table
head(featureTable)
```

Listing 4: Extraction of features through Matlab Signal Processing Toolbox and Statistics and Machine Learning Toolbox

References

[1] Sowmiya Seenath and Menaka Dharmaraj. "Conformer-Based Human Activity Recognition Using Inertial Measurement Units". In: *Sensors (Basel, Switzerland)* 23.17 (Aug. 2023), p. 7357. ISSN: 1424-8220. DOI: 10.3390/s23177357.

[2] Shima Mohammadi Moghadam, Ted Yeung, and Julie Choisne. "The Effect of IMU Sensor Location, Number of Features, and Window Size on a Random Forest Model's Accuracy in Predicting Joint Kinematics and Kinetics During Gait". In: *IEEE Sensors Journal* PP (Nov. 2023), pp. 1–1. DOI: 10.1109/JSEN.2023.3317366.

[3] Giorgio Biagetti et al. "Human Activity Monitoring System Based on Wearable sEMG and Accelerometer Wireless Sensor Nodes". In: *BioMedical Engineering OnLine* 17.Suppl 1 (Nov. 2018), p. 132. ISSN: 1475-925X. DOI: 10.1186/s12938-018-0567-4. (Visited on 05/07/2024).

[4] *Dox @ CHU-ULiège - Document in the boX, Université de Liège*. https://dox.uliege.be/index.php/s/hTj5ibOToKMNxtR. (Visited on 05/05/2024).

[5] L. Minh Dang et al. "Sensor-Based and Vision-Based Human Activity Recognition: A Comprehensive Survey". In: *Pattern Recognition* 108 (Dec. 2020), p. 107561. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2020.107561. (Visited on 08/11/2024).

[6] K. Lee et al. "Deep Learning Model for Classifying Shoulder Pain Rehabilitation Exercises Using IMU Sensor". In: *Journal of NeuroEngineering and*

Rehabilitation 21.1 (2024). Cited By :1. ISSN: 1743-0003. DOI: 10.1186/s12984-024-01343-8.

[7] *Gait Analysis in Neurological Populations: Progression in the Use of Wearables - PubMed.* https://pubmed.ncbi.nlm.nih.gov/33461679/. (Visited on 05/05/2024).

[8] Akihiro Takeuchi, Dai Hasegawa, and Hiroshi Sakuta. “Web-Based Avatar Represented Lecture Viewer toward Interactive e-Lecture Performed by 3D Avatar”. In: Mar. 2015. DOI: 10.1109/EDUCON.2015.7095984.

[9] *Electromyography (EMG).* https://my.clevelandclinic.org/health/diagnostics/4825-emg-electromyography. (Visited on 08/11/2024).

[10] C. V. Bouten et al. “A Triaxial Accelerometer and Portable Data Processing Unit for the Assessment of Daily Physical Activity”. In: *IEEE transactions on bio-medical engineering* 44.3 (Mar. 1997), pp. 136–147. ISSN: 0018-9294. DOI: 10.1109/10.554760.

[11] “Trigno Wireless Biofeedback System User’s Guide”. In: ().

[12] Elizabeth Anderson and J. Larry Durstine. “Physical Activity, Exercise, and Chronic Diseases: A Brief Review”. In: *Sports Medicine and Health Science* 1.1 (Sept. 2019), pp. 3–10. ISSN: 2666-3376. DOI: 10.1016/j.smhs.2019.08.006. (Visited on 05/05/2024).

[13] Hideyuki Tashiro et al. “The Effect of Cognitive Tasks on Reactive Stepping in Young and Older Adults”. In: *Gait & Posture* 107 (Jan. 2024), pp. 287–292. ISSN: 0966-6362. DOI: 10.1016/j.gaitpost.2023.10.018. (Visited on 05/05/2024).

[14] Alessandro Mengarelli et al. "Co-Activation Patterns of Gastrocnemius and Quadriceps Femoris in Controlling the Knee Joint during Walking". In: *Journal of Electromyography and Kinesiology: Official Journal of the International Society of Electrophysiological Kinesiology* 42 (Oct. 2018), pp. 117–122. ISSN: 1873-5711. DOI: 10.1016/j.jelekin.2018.07.003.

[15] Richard G. Ellis, Bonnie J. Sumner, and Rodger Kram. "Muscle Contributions to Propulsion and Braking during Walking and Running: Insight from External Force Perturbations". In: *Gait & Posture* 40.4 (Sept. 2014), pp. 594–599. ISSN: 1879-2219. DOI: 10.1016/j.gaitpost.2014.07.002.

[16] *Quadriceps Muscles - Anatomy, Structure, Function, Exercise.* May 2024. (Visited on 08/12/2024).

[17] Rajat Emanuel Singh et al. "A Review of EMG Techniques for Detection of Gait Disorders". In: *Artificial Intelligence - Applications in Medicine and Biology.* IntechOpen, Mar. 2019. ISBN: 978-1-78984-018-6. DOI: 10.5772/intechopen.84403. (Visited on 05/08/2024).

[18] Ge Wu et al. "ISB Recommendation on Definitions of Joint Coordinate System of Various Joints for the Reporting of Human Joint Motion–Part I: Ankle, Hip, and Spine. International Society of Biomechanics". In: *Journal of Biomechanics* 35.4 (Apr. 2002), pp. 543–548. ISSN: 0021-9290. DOI: 10.1016/s0021-9290(01)00222-6.

[19] Ge Wu et al. "ISB Recommendation on Definitions of Joint Coordinate Systems of Various Joints for the Reporting of Human Joint Motion–Part II: Shoulder, Elbow, Wrist and Hand". In: *Journal of Biomechanics* 38.5 (May

2005), pp. 981–992. ISSN: 0021-9290. DOI: 10.1016/j.jbiomech.2004.05.042.

[20] Xuechuan Wang and Kuldip Paliwal. “Feature Extraction and Dimensionality Reduction Algorithms and Their Applications in Vowel Recognition”. In: *Pattern Recognition* 36 (Oct. 2003), pp. 2429–2439. DOI: 10.1016/S0031-3203(03)00044-X.

[21] MOUZE-AMADY M and HORWAT F. “Evaluation of Hjorth parameters in forearm surface EMG analysis during an occupational repetitive task. = (Evaluation des paramètres de Hjorth appliqués à l’analyse de l’EMG de surface de l’avant-bras durant des mouvements répétitifs au poste de travail).. 101”. In: *Evaluation of Hjorth parameters in forearm surface EMG analysis during an occupational repetitive task. = (Evaluation des paramètres de Hjorth appliqués à l’analyse de l’EMG de surface de l’avant-bras durant des mouvements répétitifs au poste de travail).. 101* (1996). (Visited on 05/08/2024).

[22] Janez Dems˘ar et al. “Orange: Data Mining Toolbox in Python”. In: ().

[23] Luciano da Fontoura Costa. “Data Normalization in Signal and Pattern Analysis and Recognition: A Modeling Approach”. working paper or preprint. June 2022. (Visited on 07/11/2024).

[24] Juha Pärkkä et al. “Activity Classification Using Realistic Data From Wearable Sensors”. In: *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* 10 (Feb. 2006), pp. 119–28. DOI: 10.1109/TITB.2005.856863.

[25] *Decision Tree in Machine Learning.* https://www.geeksforgeeks.org/decision-tree-introduction-example/. Dec. 2017. (Visited on 06/07/2024).

[26] Kanglin Qu et al. "Feature Selection Using Information Gain and Decision Information in Neighborhood Decision System". In: *Applied Soft Computing* 136 (Mar. 2023), p. 110100. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2023.110100. (Visited on 08/18/2024).

[27] *Human Activity Classification Using Multilayer Perceptron - PMC.* https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8473251/. (Visited on 06/07/2024).

[28] Xin Li et al. "Human Motion Pattern Recognition and Feature Extraction: An Approach Using Multi-Information Fusion". In: *Micromachines* 13.8 (July 2022), p. 1205. ISSN: 2072-666X. DOI: 10.3390/mi13081205. (Visited on 05/12/2024).

[29] Y. Celik et al. "Gait Analysis in Neurological Populations: Progression in the Use of Wearables". In: *Medical Engineering & Physics* 87 (Jan. 2021), pp. 9–29. ISSN: 1873-4030. DOI: 10.1016/j.medengphy.2020.11.005.

[30] P. Arena et al. "Multilayer Perceptrons to Approximate Complex Valued Functions". In: *International Journal of Neural Systems* 06.04 (Dec. 1995), pp. 435–446. ISSN: 0129-0657. DOI: 10.1142/S0129065795000299. (Visited on 06/08/2024).

[31] Genevieve B. Orr et al., eds. *Neural Networks: Tricks of the Trade.* Vol. 1524. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1998. ISBN:

978-3-540-65311-0 978-3-540-49430-0. DOI: 10.1007/3-540-49430-8. (Visited on 06/06/2024).

[32] Foram S. Panchal and Mahesh Panchal. “Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network”. In: [TLDR] The proposed method finds the near to optimal number of hidden nodes after training the ANN from real world data and the advantage of proposed method is that it is not approximately calculating number ofhidden nodes but based on similarity between input data, they are calculated. 2014. (Visited on 06/06/2024).