



**POLITECNICO**  
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

1

# Efficient and reliable algorithms for step detection and counting on wearable devices

TESI DI LAUREA MAGISTRALE IN  
BIOMEDICAL ENGINEERING  
INGEGNERIA INGEGNERIA

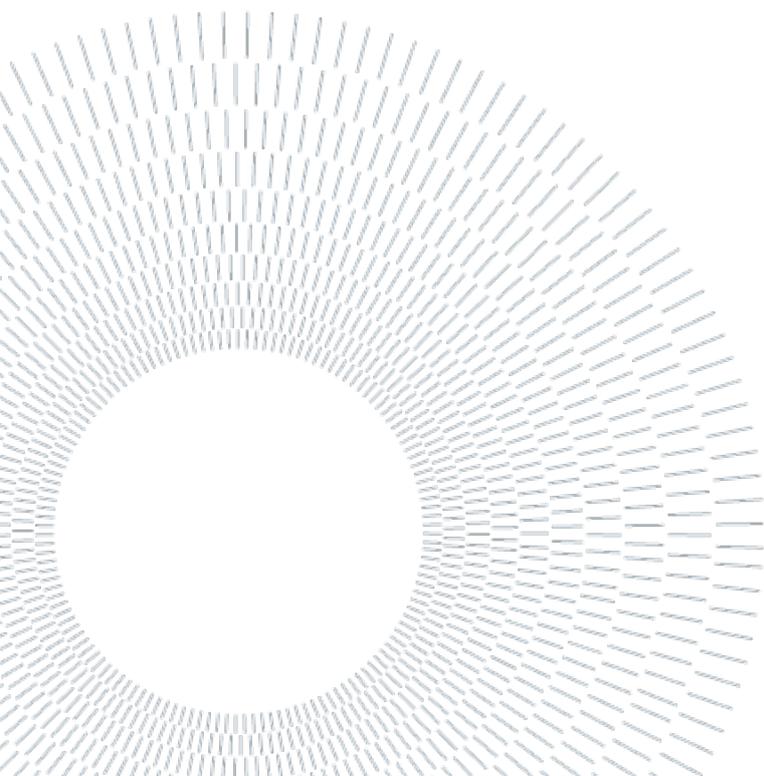
**Authors: Clarysse Allyssa Sarmiento**  
**Virginia Sekules**

Student ID: 10633038, 10629163

Advisor: Andrea Aliverti

Co-advisor: Alessandra Angelucci, Dario Salvi

Academic Year: 2024-25



# Ringraziamenti

# Abstract

Wearable devices have become indispensable tools for monitoring physiological parameters and tracking daily physical activity. Smartwatches and fitness trackers offer users convenient ways to measure key health indicators, such as step count. However, accuracy remains a significant concern, particularly for applications in clinical settings or continuous health monitoring. Many commercial devices face limitations due to interference from body movements and environmental factors, and their proprietary algorithms are often opaque, functioning as “black boxes” that limit user control and adaptability.

This Thesis explores the accuracy of the Bangle.js smartwatch, an open-source wearable device that allows researchers and developers to access and refine its step-counting algorithms. The open-source nature of Bangle.js provides a unique opportunity to enhance algorithm performance, making it a compelling option for both academic research and personal health monitoring.

The objectives of this study are:

- To collect data from the Bangle.js smartwatch alongside a reliable ground-truth, *i.e.*, a device based on an Inertial Measurement Unit (IMU), following a rigorous experimental protocol.
- To develop a step counting algorithm for the data collected with the IMU-based device, which is used as the ground truth.
- To compare and analyze existing open-source algorithms for step counting, identifying those that offer the highest accuracy, with the goal of implementing improvements in Bangle.js. The algorithms tested for this study are: Espruino algorithm, Oxford algorithm, BangleSimple algorithm, Dummy algorithm, Autocorrelation algorithm and Fast Fourier Transform algorithm.

The experimental protocol included a range of activities designed to simulate real-life usage, such as light tasks, resting phases, treadmill walking at varying speeds, stair climbing, and outdoor walking.

The results indicate that, while the Bangle.js’s existing algorithms, such as Espruino and BangleSimple, perform well in many everyday activities, they face challenges in scenarios involving more complex movements, such as treadmill walking. Comparatively, more sophisticated algorithms, like the Fast Fourier Transform-based algorithm, demonstrated strengths in specific contexts, particularly outdoor walking, where step patterns are more natural. However, this algorithm is also computationally demanding, highlighting a trade-off between accuracy and efficiency that is critical for wearable applications. Through this comprehensive analysis, we identified which existing algorithms provide the best balance between accuracy and computational cost for specific activities. These findings lay a foundation for refining the Bangle.js’s step-counting capabilities by integrating algorithmic approaches that optimize both

accuracy and resource efficiency. This research offers insights into potential modifications that could be applied to the Bangle.js's code, enhancing its adaptability and precision in future iterations

**Keywords:** Wearable devices, step-counting algorithm, experimental protocol, algorithm comparison, data analysis.

## Abstract in italiano

I dispositivi indossabili sono diventati strumenti indispensabili per monitorare i parametri fisiologici e tenere traccia dell'attività fisica quotidiana. Gli smartwatch e i fitness tracker offrono agli utenti modi pratici per misurare indicatori di salute chiave, come il conteggio dei passi. Tuttavia, l'accuratezza rimane un problema significativo, in particolare per le applicazioni in ambito clinico o per il monitoraggio continuo della salute. Molti dispositivi commerciali presentano limitazioni dovute all'interferenza dei movimenti del corpo e dei fattori ambientali, i loro algoritmi sono spesso poco trasparenti, funzionando come "scatole nere" che limitano il controllo e l'adattabilità dell'utente.

Questa tesi esplora l'accuratezza dello smartwatch Bangle.js, un dispositivo indossabile open-source che consente a ricercatori e sviluppatori di accedere e perfezionare i suoi algoritmi di conteggio dei passi. La natura open-source di Bangle.js offre un'opportunità unica di migliorare le prestazioni degli algoritmi, rendendolo un'opzione interessante sia per la ricerca accademica che per il monitoraggio della salute personale.

Gli obiettivi di questo studio sono:

- Raccogliere dati tramite lo smartwatch Bangle.js, insieme a una ground-truth affidabile, cioè un dispositivo basato su un'unità di misura inerziale (IMU), seguendo un protocollo sperimentale rigoroso.
- Sviluppare un algoritmo di conteggio dei passi per i dati raccolti con il dispositivo basato su IMU, utilizzato come ground-truth.
- Confrontare e analizzare gli algoritmi open-source esistenti per il conteggio dei passi, identificando quelli che offrono la massima precisione, con l'obiettivo di implementare i miglioramenti in Bangle.js. Gli algoritmi testati per questo studio sono: algoritmo Espruino, algoritmo Oxford, algoritmo BangleSimple, algoritmo Dummy, algoritmo di autocorrelazione e algoritmo di Trasformata Veloce di Fourier.

Il protocollo sperimentale comprendeva una serie di attività progettate per simulare l'uso nella vita reale, come compiti leggeri, fasi di riposo, camminata su tapis roulant a velocità variabile, attività su scale e camminata all'aperto.

I risultati indicano che, mentre gli algoritmi esistenti di Bangle.js, come Espruino e BangleSimple, si comportano bene in molte attività quotidiane, incontrano difficoltà in scenari che comportano movimenti più complessi, come la camminata su tapis roulant. Al contrario, algoritmi più sofisticati, come quello basato sulla Trasformata di Fourier Veloce, si sono dimostrati efficaci in contesti specifici, in particolare nella camminata all'aria aperta, dove i modelli di passo sono più naturali. Tuttavia, questo algoritmo è anche impegnativo dal punto di vista computazionale, evidenziando un compromesso tra accuratezza ed efficienza che è fondamentale per le applicazioni indossabili. Attraverso questa

analisi completa, abbiamo identificato quali algoritmi esistenti offrono il miglior equilibrio tra accuratezza e costo computazionale per attività specifiche. Questi risultati gettano le basi per perfezionare le capacità di conteggio dei passi di Bangle.js, integrando approcci algoritmici che ottimizzino sia l'accuratezza sia l'efficienza delle risorse. Questa ricerca offre spunti per potenziali modifiche che potrebbero essere applicate al codice di Bangle.js, migliorandone l'adattabilità e la precisione nelle iterazioni future.

**Parole chiave:** Dispositivi indossabili, algoritmo di conteggio dei passi, protocollo sperimentale, confronto tra algoritmi, analisi dei dati.

# Contents

.....	2
Ringraziamenti.....	3
Abstract .....	4
Abstract in italiano .....	6
Contents .....	8
1 Introduction.....	10
1.1. An overview of wearable devices .....	10
2 State of the Art.....	12
2.1. Overview of physical activity .....	12
2.2. Gait cycle.....	13
2.3. Step counting.....	14
2.3.1. Techniques for step counting .....	14
2.4. Digital medicine.....	16
2.5. Overview of mobile health applications .....	17
2.5.1. Integration of mHealth app with healthcare systems .....	18
2.5.1.1. Example of mobile health platform: Mobistudy .....	20
2.5.2. Wearable devices .....	22
3 Materials and Methods .....	26
3.1. IMU-based device.....	26
3.1.1. Hardware .....	26
3.1.2. ANT transmission protocol.....	29
3.1.3. Mobile Android application.....	30
3.2. Bangle.js smartwatch.....	30
3.3. Polar H10 Band .....	32
3.4. Experimental Protocol.....	32
3.4.1. Population characteristics.....	32
3.4.2. Set-up.....	33
3.4.3. Data acquisition .....	35
3.5. Pre-processing .....	37
3.6. Step counting from the IMU-based device .....	38
3.7. Step counting from the Bangle.js.....	45

3.7.1.	Benchmarking framework.....	45
3.7.2.	Dummy algorithm.....	46
3.7.3.	Bangle- simple algorithm .....	46
3.7.4.	Espruino algorithm .....	48
3.7.5.	Autocorrelation algorithm .....	50
3.7.6.	Fast Fourier Transform algorithm .....	53
3.7.7.	Oxford Algorithm .....	56
3.8.	Movement Detection Phase.....	60
3.9.	Performance metrics.....	61
3.9.1.	Error Calculation and Metrics .....	62
3.9.2.	Bland-Altman.....	63
3.9.3.	Box plot .....	63
4	Results and Discussion .....	65
4.1.	Bland-Altman analysis.....	65
4.1.1.	Interpretation of Bland-Altman Plot Components .....	65
4.2.	Computational cost of the algorithms .....	71
4.3.	Box plot analysis .....	73
4.4.	Discussion.....	74
4.4.1.	Bland-Altman discussion .....	75
4.4.2.	Computational costs discussion .....	75
4.4.3.	Box plot discussion.....	76
5	Conclusion and Future developments .....	77
5.1.	Conclusion .....	77
5.2.	Future Development .....	79
	Bibliography .....	81
	List of Figures.....	85
	List of Tables.....	87
	List of Abbreviations .....	88
6	Appendix A .....	89
6.1.	A.1 Metrics Calculation.....	89
6.2.	A.2 Mean Percentage Error in 30 seconds time window for each activity	

# 1 Introduction

## 1.1. An overview of wearable devices

In recent years, the proliferation of wearable devices has profoundly transformed the way various physiological parameters and daily physical activity can be monitored. Tools like smartwatches and fitness trackers allow users to keep track of crucial health aspects, such as pulse rate, step count, distance traveled, and calories burned. This technology has quickly gained popularity due to its ease of use and ability to continuously and non-invasively collect data, facilitating the autonomous management of health and well-being by users.

The accuracy of such measurements is of great importance, especially when considering their potential use in clinical settings or for continuous health monitoring. Despite technological improvements, many commercial devices still have limitations, not only in terms of accuracy, particularly due to body movements or environmental interferences, but also the hardware components must be compact and energy efficient. The algorithm that processes the data can significantly affect the reliability of the measurements, and often the technical details on how these algorithms operate remain unknown due to industry proprietary rights. In fact, many products on the market do not provide detailed information about the algorithms employed to monitor parameters such as pulse rate or step count, which limits the ability to assess the accuracy of the measurements or to make improvements. This lack of transparency poses a significant obstacle, especially in the healthcare field, where methods validation and accuracy are essential for safe and reliable use [1].

The aim of this Thesis is to investigate the accuracy of wearable devices for monitoring parameters such step count, using the Bangle.js device. Bangle.js is an open-source smartwatch [2]. Unlike most commercial devices, where the algorithms are proprietary and operate as "black boxes", Bangle.js allows researchers and developers to access, modify, and improve the algorithms that handle pulse rate and step counting. This programmability offers the potential to investigate the data usage and methods as one prefers, making it a powerful tool for both personal health monitoring and academic research [3].

The objectives of the Thesis are the following:

- To collect data from the Bangle.js smartwatch alongside a reliable ground-truth, *i.e.*, a device based on an Inertial Measurement Unit (IMU), following a rigorous experimental protocol.
- To develop a step counting algorithm for the data collected with the IMU-based device, which is used as the ground truth.
- To compare and analyze existing open-source algorithms for step counting, identifying those that offer the highest accuracy, with the goal of implementing improvements in Bangle.js.

By leveraging the open-source nature of Bangle.js, we can modify and potentially enhance these algorithms, ultimately creating a more reliable and accurate monitoring solution for users. Furthermore, open-source development has inherent benefits for security and privacy, which are increasingly important in healthcare. As noted in the computer security community, transparency in design, protocols, and source code generally leads to improved security. Commercial wearables typically rely on companion mobile apps that often come with complex and non-user-friendly privacy policies, raising concerns about the protection of personal data. An open-source approach mitigates these issues by enabling users to audit and improve firmware development for healthcare applications [4] [5].

## 2 State of the Art

### 2.1. Overview of physical activity

Physical activity is a crucial element in assessing a person's health status, as it is closely linked to both physical and mental well-being. It refers to any bodily movement produced by skeletal muscles that results in energy expenditure higher than at rest. The importance of physical activity lies in its numerous health benefits, including improvements in cardiorespiratory and muscular capacity, bone health, balance, and body weight control. Additionally, it plays a fundamental role in reducing the risk of cardiovascular, metabolic, and neurological diseases, making its measurement and monitoring essential in various contexts, including clinical and preventive settings.

According to the guidelines of the World Health Organization (WHO) and other international institutions, adults should engage in at least 150-300 minutes of moderate physical activity or 75-150 minutes of intense activity per week, or an equivalent combination. These levels of physical activity are associated with a reduced risk of all-cause mortality and cardiovascular diseases, with the maximum benefit being achievable with as little as 150 minutes of moderate activity, or approximately 8.25 hours per week. For older adults, a combination of exercises that includes balance training, aerobic activities, and muscle-strengthening exercises is recommended to counteract age-related physical decline and improve physical functionality [6].

Conversely, sedentary behavior is an increasingly widespread problem globally. Physical inactivity is associated with a 20-30% increased risk of mortality and is one of the main risk factors for chronic diseases such as obesity, type 2 diabetes, hypertension, and heart disease. In the United States, the obesity rate has reached alarming levels: 39.8% of adults and 18.5% of youth are considered obese, with annual healthcare costs reaching \$147 billion in 2008, a figure that may be even higher today, exceeding \$320 billion per year. To address this emergency, initiatives like "Healthy People 2020" have sought to increase the percentage of the population with healthy body weight by promoting physical activity and the adoption of healthier lifestyles [7].

The increase in the use of monitoring devices, such as smartwatches and smartphones, has made step count a more accessible metric. Today, more than 80% of Americans own a smartphone, while the wearable device market has seen exponential growth, with sales surpassing 113.1 million units sold worldwide in the first quarter of 2024. The expansion of these devices represents a valuable resource for monitoring and increasing physical activity, although their accuracy in step counting is subject to debate. Some studies suggest that while the devices

tend to be accurate in counting steps in young adults, they may not be as reliable in older populations or individuals with mobility difficulties [8].

The monitoring of physical activity through daily step counts is, therefore, an effective tool, but its accuracy depends on variables such as age, weight, height, and device position. For instance, wrist-worn devices tend to underestimate steps in controlled environments but may overestimate them in real-life conditions. Additionally, older adults, who may experience muscle weakness and altered gait, may record inaccurate step counts. Despite these limitations, achieving an adequate number of daily steps remains an important goal for improving overall health and reducing the risk of disease.

## 2.2. Gait cycle

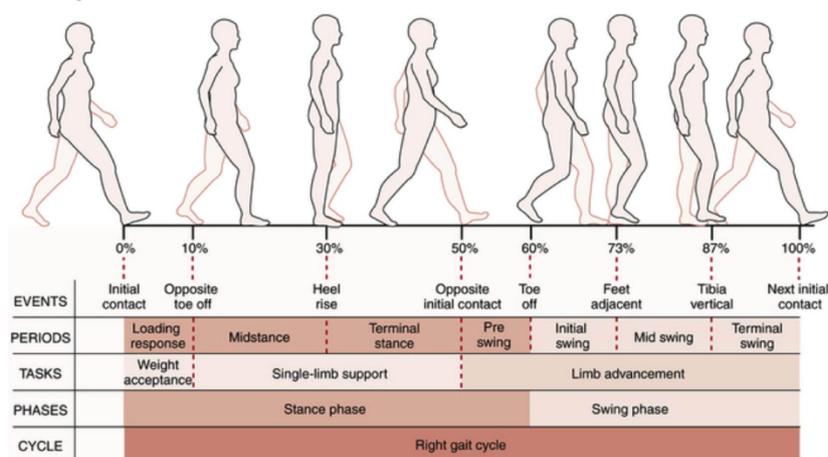


Figure 2.1 - Gait cycle

The human gait cycle, shown in Figure 2.1, represents a rhythmic sequence of movements that repeat during walking. Each phase of this cycle involves a precise and coordinated interaction between muscles, joints, and bones, ensuring balance, stability, and forward propulsion of the body. The gait cycle begins with the initial contact of the foot with the ground (often referred to as "heel strike") and ends with the final lifting of the foot (known as "toe-off"). It is divided into two main phases: the stance phase, where the foot is in contact with the ground, and the swing phase, during which the foot is lifted, and the leg prepares for the next ground contact [9].

The stance phase covers about 60% of the gait cycle and includes a series of sub-sequences that help support the body's weight and absorb the impact with the ground. During this phase, three key movements of the foot can be distinguished: the first occurs at initial contact, when the heel touches the ground, and the foot gradually moves into a flat position; the second movement happens when the tibia leans forward over the planted foot, allowing the body to move ahead; finally, the third movement consists of the heel lifting and pushing forward, facilitated by plantar flexion, which accelerates the body's progression. The

swing phase, on the other hand, is essential for repositioning the foot, allowing the leg to advance freely in preparation for the next ground contact [10].

The study of the gait cycle is of great interest in numerous fields, including biomechanics, sports medicine, and rehabilitation. By analyzing the details of movement, it is possible to identify normal and pathological patterns, providing valuable information for improving physical performance, preventing injuries, and designing rehabilitation programs. For example, in gait pathologies, one or more of the normal phases of the cycle may be altered or absent, leading to compensations in movement and an increased risk of falls or injuries [11].

This description of the gait cycle provides a foundation for understanding how step counting has become a widely used methodology for monitoring daily physical activity. Step counting is based on the measurement and analysis of each gait cycle, providing a simple yet effective indicator of an individual's physical activity level [12].

## 2.3. Step counting

Step counting represents one of the simplest and most accessible methods for monitoring and promoting daily physical activity. Its popularity stems mainly from the ease with which it allows the measurement of daily movement, providing a tangible metric to assess physical effort and encourage a more active lifestyle. The goal of 10,000 steps per day, originally introduced by Japanese pedometer manufacturers in the 1960s, has found widespread support in scientific literature. Numerous studies show that reaching this daily target brings significant health benefits, such as reducing the risk of chronic diseases like type 2 diabetes, cardiovascular diseases, and certain types of cancer.

[9] However, it is important to consider that the optimal number of steps can vary based on the individual and their health conditions. Recent research suggests that for even more effective protection against serious diseases, it may be necessary to increase the daily step count to 12,000 or 15,000 steps. This highlights the importance of personalizing physical activity goals, adapting them to the specific needs and abilities of the individual [13]. In addition to the number of steps, it is also crucial to monitor the intensity of physical activity. To this end, tracking pulse rate is a valuable tool, as it allows the assessment of an individual's aerobic capacity and the adjustment of workouts to achieve the best health results, keeping exercise within a safe and effective effort zone [14].

### 2.3.1. Techniques for step counting

Step counting algorithms have evolved to use various methods, each with distinct advantages depending on the application and context. Some of the most common techniques for counting steps include:

1. **Accelerometer-Based Methods:** These rely on detecting changes in acceleration caused by foot movement. Algorithms analyze the peaks and troughs in the accelerometer data to count steps. This is the most widely used method in consumer wearables due to its simplicity and low power consumption [15].
2. **IMU-Based Methods:** IMUs incorporate accelerometers, gyroscopes, and magnetometers to track motion and orientation. When combined with algorithms that filter and process these signals, IMUs provide accurate step detection, even in complex motion scenarios such as running or navigating uneven surfaces [16].
3. **GPS-Based Methods:** GPS tracks the distance traveled and can infer steps by estimating the user's stride length. While this method is more power-intensive, it is particularly useful for outdoor activities like hiking [15].
4. **Optical Sensors-Based Methods:** While less common for step counting alone, some devices integrate optical sensors to complement the previously mentioned methods, enhancing accuracy by monitoring movement and physiological signals [15].
5. **Pressure Sensors-Based Methods:** Generally placed in footwear, pressure sensors detect footfalls directly, offering an alternative to accelerometer-based wearables for specific use cases like medical monitoring [15].

Step counting plays a critical role in promoting physical activity and monitoring health. It provides an easy-to-understand metric for daily movement, encouraging users to stay active. Research has shown a strong link between daily step counts and health outcomes, including reduced risks of cardiovascular diseases, diabetes, and obesity. Therefore, step-counting algorithms are not just a feature in fitness trackers but have significant implications in healthcare [16].

## 2.4. Digital medicine

Digital medicine marks a significant paradigm shift in healthcare, fundamentally transforming how medical care is delivered. By incorporating advanced software, algorithms, and state-of-the-art technologies, digital medicine enables the continuous collection, analysis, and use of patient data to inform medical decisions. Unlike traditional healthcare systems, which often operate in a reactive manner, responding to symptoms or health issues as they arise, digital medicine takes a proactive approach. Using wearable devices, remote monitoring, and data-driven algorithms, patients' health can be continuously monitored in real-time. This enables healthcare providers to intervene at the earliest signs of a problem, leading to more timely treatments and better outcomes [17].

Central to digital medicine is its closed-loop system, where data is not only collected but also processed in real time to trigger immediate actions or interventions. This feedback loop ensures that patient management becomes dynamic, continuously adjusting to the individual's evolving health status. For example, a patient with a chronic condition like diabetes can benefit from continuous glucose monitoring systems that automatically adjust insulin levels. Similarly, those with cardiovascular issues can use wearable devices that alert healthcare providers in case of abnormal heart rhythms, significantly reducing the risk of severe complications. Studies show that such real-time monitoring and intervention can reduce hospitalizations by up to 38%, emphasizing the immense potential of digital health tools to reshape patient care [18].

Scientific rigor plays a key role in the deployment of these tools. Digital health technologies must go through rigorous processes of verification and validation. Verification ensures that these tools are designed and function as intended, while validation assesses their effectiveness in achieving clinical outcomes. This rigorous approach ensures that patient safety is paramount and that the tools deployed in clinical settings are not only innovative but also reliable.

The scope of digital medicine extends to a wide range of innovative technologies. Wearable devices, such as smartwatches and fitness trackers, have become integral to monitoring vital signs like pulse rate, physical activity, sleep patterns, and even blood oxygen levels. Telemedicine platforms are revolutionizing access to healthcare, allowing patients to consult with their healthcare providers remotely via video calls, reducing the need for in-person visits, and making healthcare more accessible to remote or underserved populations. Digital therapeutics, another growing field, employ software-based solutions to treat and manage various conditions, such as mental health disorders, hypertension, and respiratory diseases, offering non-invasive, scalable, and cost-effective treatment alternatives.

In addition to patient monitoring and management, digital medicine significantly enhances clinical decision-making. Artificial intelligence (AI) and machine learning (ML) technologies are increasingly being adopted in healthcare to analyze vast datasets and detect patterns that may not be apparent to human observers. AI-driven diagnostics, for instance, have shown the potential to match or even surpass the accuracy of human experts in identifying conditions such as diabetic retinopathy and lung cancer in their early stages. These technologies not only improve diagnostic accuracy but also enable predictive analytics, allowing healthcare providers to anticipate complications and make data-informed decisions that improve long-term patient outcomes.

The economic impact of digital medicine is equally substantial. The global market for digital health is expanding rapidly, with projections indicating it could reach \$1.1 trillion by 2031, reflecting a compound annual growth rate of around 13.1% from 2023 to 2031 [19]. The COVID-19 pandemic has further accelerated the adoption of these technologies, as the need for remote healthcare solutions became more pressing. Telemedicine use, for instance, saw a 38-fold increase during the pandemic.

As the role of digital health continues to grow, it is important to acknowledge the broader societal benefits of these advancements. Not only do they provide more personalized and effective care, but they also contribute to reducing healthcare costs by preventing hospital readmissions and optimizing resource allocation. Furthermore, they empower patients to take an active role in managing their health, fostering a more engaged and informed patient population.

## 2.5. Overview of mobile health applications

The proliferation of mobile health (mHealth) applications has transformed the way individuals monitor their health and access healthcare services. Mobile technology, particularly the widespread use of smartphones, has contributed to the rapid expansion of mHealth. This rapid growth has been accompanied by an explosion in the number of available health-related apps, with more than 100,000 such apps available on mobile platforms [20].

mHealth apps are used for a variety of health-related purposes, ranging from fitness tracking and disease prevention to chronic disease management and virtual consultations with healthcare providers. According to the WHO, mHealth supports a broad range of functions, including health monitoring, communication between healthcare providers and patients, and real-time data collection. These apps, supported by mobile devices such as smartphones and tablets, as well as wearable sensors, have changed the landscape of healthcare by allowing users to track steps, monitor heart rates, and even manage chronic conditions.

The utility of mHealth apps extends beyond simple fitness tracking. Patients use these tools to manage long-term conditions like diabetes, asthma, and hypertension. For instance, apps that monitor blood pressure or track glucose levels provide users with actionable insights into their health, helping them make informed decisions about their care. The data collected by these apps is often used to inform treatment plans and can support clinical decision-making, potentially reducing the need for frequent in-person visits.

The exponential growth in mHealth adoption is driven by several factors. The increasing affordability of smartphones, advances in sensor technology, and the desire for personalized healthcare experiences have all contributed to the rise of mHealth. However, while the adoption of these technologies has increased, issues such as data privacy, security, and regulatory oversight remain key challenges [20]

### 2.5.1. Integration of mHealth app with healthcare systems

MHealth applications are increasingly recognized for their ability to assist with various tasks traditionally managed by general practitioners, ranging from routine health monitoring to chronic disease management. A scoping review highlighted that mHealth apps can support tasks such as data gathering, symptom assessment, and even health promotion. These applications, integrated with wearable devices and smartphones, provide users with real-time insights into their health, reducing the need for frequent in-person consultations. For example, apps like Ada and Babylon allow patients to input symptoms and receive potential diagnoses directly on their mobile devices, empowering them to manage their health more independently [20].

Moreover, mHealth apps are particularly effective when combined with wearable technology and other sensors, which enable continuous monitoring of key health metrics. Popular apps like Google Fit and Apple Health act as aggregators, tracking metrics such as step counts, heart rate, and sleep patterns. Wearables can provide a more comprehensive view of users' physical activity and heart health, encouraging preventive care and enabling better management of chronic diseases such as diabetes, asthma, and cardiovascular conditions [21].

While these apps have proven valuable in health self-management, there are still challenges in integrating them with formal healthcare systems. A major limitation is the lack of direct integration with electronic health records and healthcare professionals. Although patients can collect extensive health data, this data is rarely shared with or used by healthcare providers, limiting its potential to inform clinical decisions. Moreover, concerns about data accuracy and the risks of self-diagnosis without professional oversight hinder the full adoption of mHealth apps in clinical settings.

Another notable aspect of the growth in mHealth apps is the broader industry trend, which shows rapid expansion. With the growing number of smartphone users, more than 100,000 health-related apps are now available on major

platforms like the **Google Play Store** and **Apple App Store**, catering to a wide range of health needs, from fitness tracking to chronic disease management. A study highlighted in the *Journal of Medical Internet Research* showed that app users are more likely to meet their health goals, such as increased physical activity and better diet management, compared to non-users. This aligns with findings that mHealth apps are playing a crucial role in public health by promoting healthier behaviors and increasing health awareness among users [21]. As mHealth platforms continue to evolve, their potential to integrate seamlessly into healthcare systems and offer more personalized health monitoring is becoming increasingly evident. Table 2.1, modified from Wattanapisit et al. (2020), illustrates the range of tasks that mHealth apps can perform, such as data gathering, chronic disease management, and health promotion [20].

Task	Category	Example of available Apps
Apply a structured approach to data gathering and investigation	History taking	Ada, Babylon, Medical history builder, Historian Ada, Babylon
Interpret findings accurately to reach a diagnosis	Diagnosis	Ada, Doctor Diagnose Symptoms Check, GBDiagnosis Medical App, My diagnostic, Self Diagnosis, Symptomate, WebMD, Rapid diagnosis - mental health, Your rapid diagnosis – STD, Babylon
Demonstrate a proficient approach to clinical examination	Clinical examination	Runtastic Heart Rate, SkinVision
Demonstrate a proficient approach to the performance of procedures	Medical procedures	None
Adopt appropriate decision-making principles	Medical decision making	Gout Decision Aid

Adopt a structured approach to clinical management	Clinical management	Rapid Diagnosis, Mental Health, Rapid Diagnosis - STD
Make appropriate use of other professionals and services	Health professionals	None
Provide urgent care when needed	Urgent care	Google Assistant, Siri
Enable people with long-term conditions to improve their health	Long-term care	Asthma Manager, Blood Pressure Companion, mySugr, forDiabetes, Pill Reminder and Medication Tracker by Medisafe
Manage concurrent health problems in an individual patient	Health problems	Asthma Manager, Blood Pressure Companion, mySugr, forDiabetes, Pill Reminder and Medication Tracker by Medisafe
Coordinate a team-based approach to patient care	Team-based care	None
Support people through individual experiences of health, illness, and recovery	Health promotion	Appibuddy, Food (lg), HealthHub Track, Healthy 365 HealthWatch, Healthy 365 BECCA - Breast Cancer Support, The Circle of Health

*Table 2.1 – Tasks of a general practitioner that can be potentially performed by mHealth apps*

#### 2.5.1.1. Example of mobile health platform: Mobistudy

A promising example of a mobile health platform that addresses some of these challenges is Mobistudy [22] an open-source, mobile-based platform that supports multi-dimensional data collection for clinical studies. Mobistudy uses smartphones, IoT devices, and health data aggregators such as Google Fit and Apple HealthKit to facilitate real-time data collection. It enables researchers to design studies, recruit participants, and collect data using various sensors, all within a single unified infrastructure. Mobistudy supports multiple studies at

once, allowing participants to contribute to several research projects without needing separate apps for each study.

One of the key features of Mobistudy is its task-based data collection, where participants are prompted to complete scheduled tasks such as questionnaires, activity tracking, and health assessments using connected devices. These tasks are delivered through an easy-to-use mobile app that supports both Android and iOS platforms. In a pilot study involving 18 participants, Mobistudy collected data on physical activity (*e.g.*, step count), heart rate, and heart rate variability from devices like the Xiaomi MiBand 3 and smart peak flow meters. Over 23.8 days (on average) of the 31-day study, participants generated a total of 531 reports, including 265 on step counts, 53 on activities, 130 on heart rate, and 83 on heart rate variability. This highlights the platform's effectiveness in collecting diverse types of health data over extended periods [22].

In terms of use cases, shown in Figure 2.2, Mobistudy is designed to be highly flexible and extensible. Researchers can create studies without needing extensive technical knowledge, leveraging pre-built tasks that include not only step counting but also questionnaires, mood assessments, pulse oximetry, and more. For physical activity tracking, Mobistudy retrieves data on steps, heart rate, and other metrics from health aggregators. The app is also equipped to guide users through fitness tests like the Queen's College step test and the 6-Minute Walk Test, which assess cardio-respiratory health. These use cases enable researchers to gather comprehensive data about participants' physical activity and health, making Mobistudy a powerful tool for studies on conditions like asthma, cardiovascular diseases, and other health concerns linked to mobility and activity.

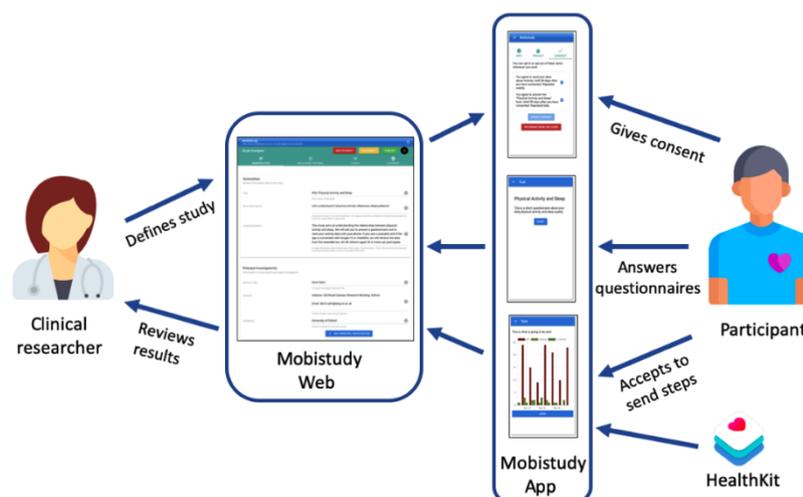


Figure 2.2 - Mobistudy- Use case

While Mobistudy offers substantial advantages, it also faces challenges. One notable limitation is the difficulty in extracting data from certain wearables, such as Fitbit devices, which do not synchronize with **Google Fit** or **Apple HealthKit**.

Additionally, during the pilot study, data loss occurred due to connectivity issues, as Mobistudy does not currently buffer data when participants are offline. Despite these limitations, Mobistudy's open-source nature allows continuous improvement, and its ability to handle multiple studies within one app makes it a powerful tool for researchers.

### 2.5.2. Wearable devices

Wearable devices represent one of the most revolutionary innovations in healthcare technology. Thanks to continuous advances in miniaturization and digital electronics, it is now possible to design tools capable of monitoring a wide range of physiological parameters and daily activities in real-time and remotely. These devices, which include smartwatches, smart bracelets, and other wearable technologies, allow for continuous and non-invasive health monitoring, seamlessly integrating with the daily lives of users, both in work and personal settings [23].

The emergence of advanced technologies in this field has had a significant impact on awareness of personal health conditions and medical care, promoting a greater ability for early diagnosis and chronic disease management [24].

According to market research, the wearable healthcare device sector reached a value of over \$25 billion in 2020, with an expected annual growth rate of 22.9% until 2027. This data highlights not only the growing interest in personal health and fitness but also the central role wearables are assuming as tools for continuous health condition monitoring [25].

#### Technical Components

From a technical standpoint, wearable devices integrate a series of advanced sensors capable of detecting various physiological parameters. The main sensors used include:

- **Photoplethysmographic Sensors.** The most common sensors for measuring pulse rate in smartwatches are based on photoplethysmography (PPG). This method uses an LED light source (typically green, as it is absorbed differently by oxygenated hemoglobin) and a photodiode that measures the light reflected from the underlying tissues. When the heart beats, the volume of blood in the tissues changes, and consequently, the amount of light absorbed and reflected also changes. The data collected by the photodiode is processed to calculate the number of beats per minute (bpm). [26] PPG also allows to estimate peripheral blood oxygen saturation (SpO<sub>2</sub>). In fact, a combination of LEDs with different wavelengths (typically red and infrared) can be used to measure the amount of oxygen bound to hemoglobin in the blood. Infrared light is absorbed more by oxygenated hemoglobin, while red light is absorbed more by deoxygenated hemoglobin. These differences

are detected by photodiodes and used to calculate the percentage of oxygen saturation in the blood [26].

- **Inertial Measurement Units.** These sensors are mainly used to detect movement, step count, and body position. Accelerometers measure changes in speed along one or more axes, while gyroscopes measure changes in orientation or rotation, and magnetometers measure the change in the Earth's magnetic field. Both sensors generate electrical signals in response to body movements, and through advanced algorithms, the raw data is processed to determine the number of steps, overall physical activity, and even sleep cycles [27].
- **Electrocardiogram (ECG) Sensors** Some smartwatches are equipped with sensors to detect electrocardiograms (ECG). This sensor measures the heart's electrical activity through electrodes integrated into the device. The electrodes detect the electrical signals generated by the heart, and through advanced algorithms, the device can identify any irregularities in the heart rhythm, such as atrial fibrillation [28].

These devices not only enable the monitoring of vital parameters such as heart rate and blood pressure but are also capable of detecting changes related to medical conditions like fatigue, diabetes, and inflammatory responses. For instance, recent studies have shown how smartwatches can help predict cardiometabolic conditions and detect early stages of atrial fibrillation [29].

Table 2.2 presents a comparison of some of the most common smartwatches on the market, all using PPG as pulse rate sensor and a 3-axis accelerometer for step counting.

Model	Accuracy of pulse rate measurements	Accuracy of step counting	Battery life	Additional functionalities
<b>Fitbit Sense 2</b>	~4-6% error during physical activity	~3-5% error	Up to 6 days	Electrodermal activity, SpO <sub>2</sub> , stress
<b>Garmin Venu 2</b>	~1-3% error both at rest and during exercise	~1.5-2% error	Up to 11 days	SpO <sub>2</sub> , GPS
<b>E4 Wristband</b>	~1-2% error at rest, 2.7% during physical activity	~1-2% error	Up to 36 hours	Electrodermal activity, skin temperature

<b>Omron HeartGuide</b>	~3-4% error for HR	~3% error	Up to 2-3 days	ECG, SpO <sub>2</sub> , estimation of blood pressure
<b>Samsung Galaxy Watch 6</b>	~2.3% average error	~1-2% error	Up to 40 hours	ECG, SpO <sub>2</sub> , body composition analysis
<b>Apple Watch 9</b>	~1.6% error at rest, ~2.5% during intense exercise	~1-2% error	Up to 18 hours	ECG, SpO <sub>2</sub> , GPS, fall detection

Table 2.2 – Comparison of the major smartwatches

As shown in Table 2.2 devices like the Garmin Venu 2 and Apple Watch 9 feature highly refined step-counting algorithms, offering excellent accuracy both during normal and sport activities. The Fitbit Sense 2 and Omron HeartGuide are less accurate in step counting, especially during high-intensity activities or complex movements [30].

The exploration of wearable technology highlights the transformative potential of smartwatches and fitness trackers in monitoring physical activity and physiological parameters. However, despite their widespread adoption, the accuracy of these devices often falls short, particularly in step counting. Variability in user activity, environmental conditions, and algorithmic limitations are key factors contributing to inconsistent performance.

Proprietary algorithms used in commercial devices, while optimized for general use, lack transparency and adaptability. This creates a critical gap for researchers aiming to improve these technologies. Open-source platforms, such as Bangle.js, address this limitation by offering accessible and modifiable algorithms, fostering collaboration and innovation.

The lack of standardized validation datasets remains a significant barrier to advancing wearable accuracy. Shared datasets capturing diverse activities and user behaviors are essential for benchmarking algorithms, improving their robustness, and ensuring their applicability across a wide range of scenarios.

This work seeks to bridge these gaps by leveraging the open-source capabilities of Bangle.js to evaluate and enhance step-counting algorithms. By generating and analyzing comprehensive datasets and comparing algorithms against gold-

standard references, this study lays the groundwork for improving both the accuracy and adaptability of wearable devices.

## 3 Materials and Methods

### 3.1. IMU-based device

Inertial Measurement Units (IMUs) are electronic devices that combine multiple sensors to measure inertial parameters such as linear accelerations, angular velocities, and, in some cases, the magnetic field along three spatial axes ( $x$ ,  $y$ ,  $z$ ). They are widely used in various applications to detect and track the movement, position, and orientation of an object or person in three-dimensional space. In this project, an IMU-based device, developed at Politecnico di Milano, was placed on the participants' shoe and used as ground truth for step counting.

#### 3.1.1. Hardware

The device's components [31] are shown in Table 3.1:[32]

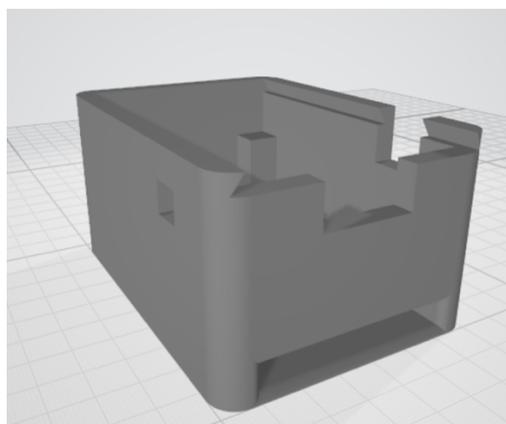
Component	Description
<p><b>nRF52840 microcontroller</b></p>	<p>The MDBT50Q-1MV2 module by Raytac is built on the Nordic nRF52840 SoC. It offers interfaces like GPIO, SPI, UART, I2C, ADC, NFC, and USB for connecting peripherals, and includes a chip antenna and a 32 MHz crystal, simplifying PCB design. The nRF52840 features a 64 MHz Cortex M4F CPU, 1 MB Flash, and 256 kB RAM. It supports voltage inputs of 1.8–5.5 V in high voltage mode and up to 3.6 V in normal mode. The antenna output power, adjustable via firmware, ranges from -20 dBm to +8 dBm.</p>
<p><b>LSM6DSO 6-axis IMU</b></p>	<p>The LSM6DSO is a compact 6-axis IMU by STM Microelectronics, combining a 16-bit accelerometer (<math>\pm 2</math> to <math>\pm 16</math> g) and a 16-bit gyroscope (<math>\pm 125</math> to <math>\pm 2000</math> dps). It features event detection (e.g., free fall, motion, wake-up, step counting), a 9 kB FIFO for power efficiency, and Sensor Hub capabilities for managing external sensors via I2C or SPI. It operates at 1.8–3.6 V, consumes 0.55 mA in high-performance mode, and comes in a 2.5 x 3.0 x 0.83 mm LGA package.</p>
<p><b>LIS2MDL magnetometer</b></p>	<p>The LIS2MDL is a 3-axis 16-bit digital magnetometer with a <math>\pm 50</math> gauss range. It supports 1.8–3.6 V supply, I2C and SPI communication.</p> <p>Combined with a 6-axis IMU like the LSM6DSO, it creates a 9-axis sensor system. It is available in a compact 2.0 x 2.0 x 0.7 mm LGA package.</p>

<p><b>LM3671MF Buck DC-DC converter</b></p>	<p>The LM3671MF-3.3/NOPB by Texas Instruments is a Buck converter ideal for low-voltage circuits. It supports an input range of 2.7 V to 5.5 V and can supply up to 600 mA, which is sufficient for this application. The converter automatically switches between PWM and PFM modes for optimal efficiency. It comes in a compact SOT-23 package (2.9 × 1.6 mm) and requires only three external components: an inductor and two capacitors.</p>
<p><b>MCP73831/2 LiPo charger</b></p>	<p>The MCP73832 by Microchip is a linear charge management controller designed for efficiently charging LiPo batteries via USB. It uses a constant-current/constant-voltage charging algorithm, with adjustable preconditioning and termination via an external resistor. For the project, the 5-lead SOT-23 package was chosen.</p>
<p><b>BJT and MOSFET</b></p>	<p>High-side MOSFET circuits were selected for two purposes: battery voltage sensing and reducing the power consumption of the SD card. A high-side MOSFET circuit was chosen for battery sensing. Similarly, to power off the SD card during idle periods and minimize power dissipation, a high-side MOSFET circuit was employed.</p> <p>The MOSFETs were selected based on experimental testing using electronic circuit simulators like LTSpice and breadboarding, where various low-threshold P-MOSFETs from Microchip were evaluated. The TP2640 was selected for battery sensing, while the LP0701 was chosen for controlling the SD card. Both are available in 8-lead SOIC packages. For both circuits, the NPN BJT used is the MMBT3904-AQ from Diotec Semiconductor (SOT-23 package).</p>
<p><b>Connectors</b></p>	<p>The PCB includes three connectors. The first is a MOLEX micro-USB type B connector, which serves both for charging the LiPo battery and for USB communication with the host PC.</p> <p>The second connector is a JST 2-pin male header with a 1 mm pitch, designed for connecting the LiPo battery. A matching female header is necessary for the connection.</p> <p>For the SD card, the chosen connector is the Adafruit 1660, a push-pull type that provides excellent stability and durability during the development process.</p>

<p><b>Switch and button</b></p>	<p>The KSS231GLFS temporary switch is used to activate the OTA firmware update mode during startup. To power the device on and off, a through-hole sliding switch is employed. The chosen model, EG1213, ensures greater mechanical durability and long-term reliability. The actuator, measuring 5 mm in length, is long enough to extend beyond the case and be operated manually.</p>
<p><b>Red LED/ Yellow LED</b></p>	<p>A RED LED from Lumex Opto Components provides user feedback, matching the type used in Lucchesini's version. The LED has a forward voltage of 1.7 V and a current draw of 3.40 mA when powered on, despite the datasheet suggesting 20 mA, as it was found experimentally that 3.40 mA was sufficient for brightness.</p> <p>Additionally, a yellow LED by Würth indicates the battery charging status. When the battery is charging via USB, the LED stays on until the battery is fully charged.</p>
<p><b>Lipo battery with 3.7 V nominal voltage</b></p>	<p>A 3.7 V rechargeable LiPo battery was chosen to avoid frequent replacements. A 150 mAh battery (25x20x5 mm) was selected for its compact size, compatibility with the PCB case, and ability to power the device for several hours, depending on the acquisition type</p>

*Table 3.1 - IMU-based device's components*

Figure 3.1 and Figure 3.2 illustrate the design of the case (lower part in Figure 3.1, upper part in Figure 3.2). The material selected for the case is DraftGrey, a rigid material designed specifically for the Super High Speed print mode, producing models with medium opacity and smooth finish. The version of the case used in this Thesis features a slot for an elastic strap to secure it to the foot.



*Figure 3.1 - Lower part of the case*

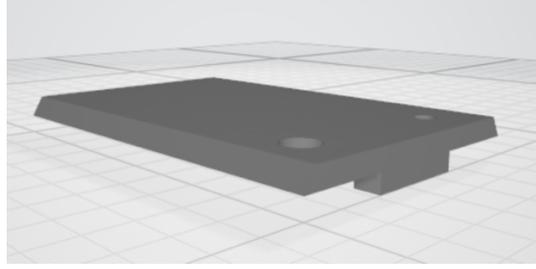


Figure 3.2 - Upper part of the case

### 3.1.2. ANT transmission protocol

Data from the IMU-based device transmitted using the ANT transmission protocol, a low-power and short-range wireless communication method, specifically designed for sensor networks. ANT operates in the 2.4 GHz ISM band, optimizing energy consumption through low data rates (20-60 kbps), short transmission periods, and deep sleep modes, making it ideal for applications requiring long battery life, such as wearable health monitors.

The ANT communication network in this application utilizes a star topology, allowing data acquisition from multiple devices simultaneously, although only one device is used in this specific study. Each device operates as a master node and connects to a smartphone equipped with an ANT USB2 stick, which serves as the slave. A low-frequency shared channel manages the control of all devices. The ANT USB2 stick can handle up to 8 communication channels, with a total combined data rate of 190 Hz for 8-byte data payloads in broadcast mode. After allocating 5 Hz for the control channel, 185 Hz remains available for data transfer. For ease of computation when multiple devices are active, the data rate is capped at 180 Hz.

Given the shared nature of the platform, bandwidth is distributed among the three IMU units used in this study. Each device operates within a calculated maximum data rate based on the total bandwidth divided by the number of active devices. In this case, with 5 Hz reserved for the control channel and three IMUs sharing the remaining bandwidth, the effective maximum data rate per device is 60 Hz.

The data transmission rate is further influenced by the number of sensors active on each device, as sensor outputs occupy space in the ANT packet payload.

Applying the bandwidth and data rate equations:

$$BW_{unit} = \frac{BW_{total}}{n_{units}} = \frac{180 \text{ Hz}}{n_{units}}$$

$$MDR = \frac{BW_{units}}{n_{sensors}}$$

### 3.1.3. Mobile Android application

A smartphone application to collect data from the device was developed in Android Studio during a previous Thesis work and provides a user-friendly interface for patients and caregivers to manage the device and collect data.

Through this application, users can configure the system to meet their specific requirements. After completing authentication and selecting a patient or participant, users can customize various parameters, including:

- **Number of IMU units:** The application allows users to specify the number of IMU-based devices for data acquisition. While three units are currently available, the system is designed to support additional units in the future.
- **Sensor selection:** Users can choose which sensor data to collect, with any combination of accelerometer, gyroscope, and magnetometer outputs supported.
- **Sensor sensitivity:** Sensitivities for the accelerometer and gyroscope can be adjusted based on the acquisition needs. Available accelerometer sensitivity settings are  $\pm 2$  g,  $\pm 4$  g,  $\pm 8$  g, and  $\pm 16$  g, while gyroscope sensitivities include  $\pm 125$  dps,  $\pm 250$  dps,  $\pm 500$  dps, and  $\pm 1000$  dps.
- **Sampling frequency:** Users can set the desired data rate (in Hz) via a text box. The maximum allowable frequency depends on the number of IMU units and sensors selected, as discussed in Section 2.2. The application provides on-screen guidance to ensure valid frequency input.
- **SD card backup:** Users have the option to enable backup data storage on the SD card. If selected, they can also assign a custom file name for the backup. If this option is not selected, the data will be stored in the cloud using Google Firebase.

## 3.2. Bangle.js smartwatch

The Bangle Js.2 [2] is an open-source smartwatch. Designed and produced by Espruino, it is fully programmable using JavaScript. Espruino is known for being energy-efficient and easy to use, allowing developers to write code quickly and with fewer resources compared to more complex programming languages.

The Bangle Js.2 has numerous technical features that enhance its functionality, making it extremely versatile for multiple uses, such as monitoring certain health parameters.

The device, shown in Figure 3.3, is equipped with a 1.3-inch color transfective LCD touchscreen with a resolution of 176x176 pixels. This type of screen is designed to be clearly visible even in strong sunlight, which is a useful feature for outdoor use.

The smartwatch features a 200mAh battery, which guarantees an average autonomy of about 2 weeks with moderate use. The energy efficiency is

improved by low-power technologies, making this smartwatch ideal for daily use and prolonged activities without the need for frequent recharging.

The Bangle.js 2 includes a series of sensors useful for monitoring health and physical activities. Among them:

- 3-axis accelerometer: useful for tracking movements, steps, position detection, and gesture control.
- PPG pulse rate sensor: useful for real time pulse rate monitoring.
- Ambient light sensor: which automatically adjusts the brightness of the display based on the surrounding environment, improving user experience and optimizing energy consumption.



*Figure 3.3 - Bangle.js smartwatch*

The Bangle.js 2 is equipped with Bluetooth Low Energy (BLE) connectivity, which enables interaction with smartphones, tablets, and computers for data synchronization, notification reception, and firmware updates. The BLE connection is also essential for developing applications that require interaction with other devices. BLE is widely used for short-range wireless communication and is well-suited for medical applications where low power consumption is critical. It supports small bursts of data transfer over a short range (up to 100 meters), which is often sufficient for connecting wearable devices like smartwatches or medical sensors to smartphones or computers. Unlike ANT, Bluetooth operates with higher data rates but has greater power consumption. Compared to other smartwatches on the market, the Bangle.js 2 allows for advanced customizations and the development of personalized applications, thanks to its open-source nature, giving users full control over the device's functionalities.

### 3.3. Polar H10 Band

The Polar H10 is a high-precision chest-worn heart rate sensor designed for sports, fitness, and research applications. It features advanced electrodes embedded in a soft, adjustable strap that adhere securely to the skin, ensuring optimal contact for accurate and reliable heart rate monitoring. The sensor is compatible with Bluetooth, ANT+, and 5 kHz transmission, allowing seamless connectivity to a wide range of devices, including smartphones, smartwatches, fitness equipment, and GPS devices. With a water resistance of up to 30 meters, it is suitable for swimming and other water-based activities. The H10 also includes onboard memory for standalone data recording, making it convenient for activities without a paired device. Powered by a CR2025 coin cell battery, it offers an impressive battery life of up to 400 hours. Its high accuracy and versatility make it a trusted choice for both athletes and researchers.

### 3.4. Experimental Protocol

#### 3.4.1. Population characteristics

For the data acquisition 20 healthy participants were recruited at the Politecnico di Milano (Table 3.2, 11 men and 9 women, mean age  $27.9 \pm 8.7$  years, mean weight  $70.8 \pm 10.7$  kg, mean height  $174.5 \pm 8.2$  cm).

Participant number	Age [years]	Sex	Weight [kg]	Height [cm]
1	24	F	73	168
2	24	F	60	168
3	25	M	69	170
4	24	M	69	170
5	24	M	70	186
6	26	M	76	185
7	28	M	76	180
8	24	F	54	163
9	29	F	65	165
10	27	M	77	180
11	24	M	100	193
12	27	M	68	184
13	27	F	68	168
14	24	F	63	174
15	24	M	80	180
16	25	F	60	170
17	22	F	58	175

18	57	M	80	175
19	25	M	64	170
20	48	F	85	165

Table 3.2 - Population characteristics

### 3.4.2. Set-up

The data acquisition protocol involved using the three previously mentioned devices on each participant: a Bangle.js smartwatch worn on their preferred wrist (introducing interpersonal variability), an IMU-based unit housed in a custom case attached to the right foot with a strap (Figure 3.4) and a Polar chest strap. The IMU-based unit was connected to its associated smartphone application via the ANT protocol, with data stored locally on an SD card for reliability and simultaneously transmitted via Google Firebase.



Figure 3.4 - IMU-based device inside the case

The IMU-based was configured with a sampling frequency of 100 Hz and a dynamic range of  $\pm 4g$ . Figure 3.5 shows the setting for the application.

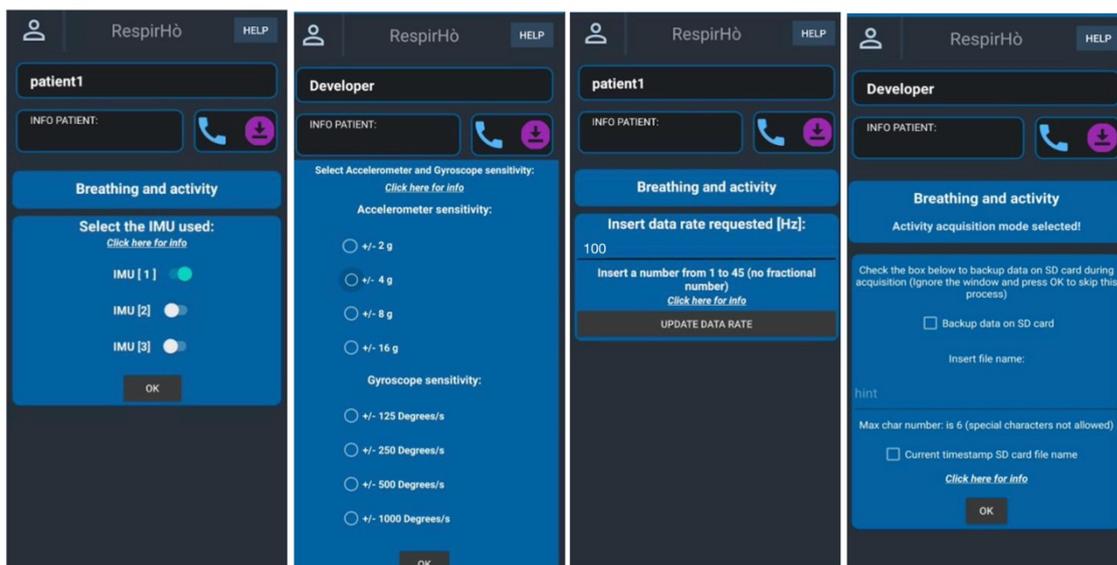


Figure 3.5 - Smartphone application associated with the IMU-based unit.

In the case of the Polar H10 Band, to improve electrode contact with the skin, subjects were instructed to moisten the Polar chest strap's electrodes. Both the Polar strap and the smartwatch connected via BLE to separate web applications, which collected the output data files. The smartwatch web application, provided by Dario Salvi of University of Malmö, was modified to transmit also raw PPG values and confidence levels, while the Polar web application, developed for this thesis, generated a file containing the detected BPM values. Figure 3.6 and Figure 3.7 show the interface of web applications.

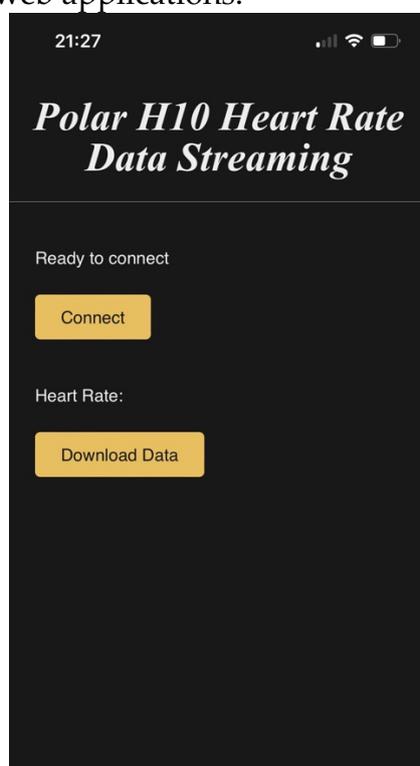


Figure 3.6 - Polar streaming web application

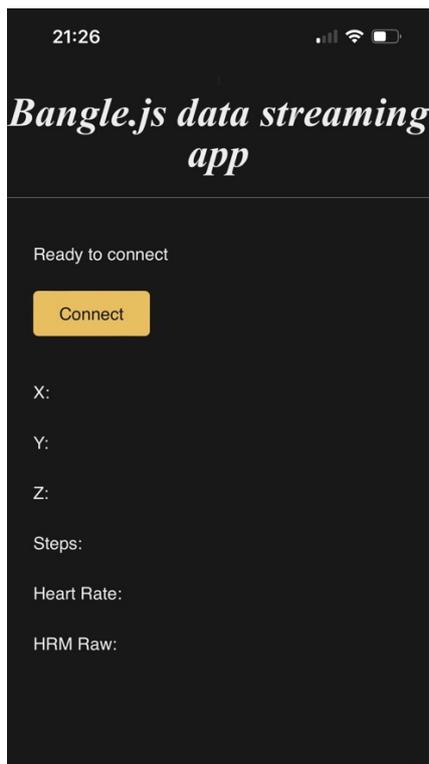


Figure 3.7 -Bangle.js streaming web application

### 3.4.3. Data acquisition

Once the devices were properly positioned on the participant, a sequence of activities was started, each designed to collect specific data under different conditions. The activities were performed in a pre-established order, ensuring a well-structured progression.

There were 5 different activities: 'light task' (5 minutes), 'complete rest' (3 minutes), 'indoor activity' (14 minutes), 'stair climbing' (2 minutes), 'outdoor walk' (~15 minutes). The protocol chosen for the activities is a well-established and tested approach, as referenced in [33]. Figure 3.8 shows an overview of the activities performed with their respective durations.

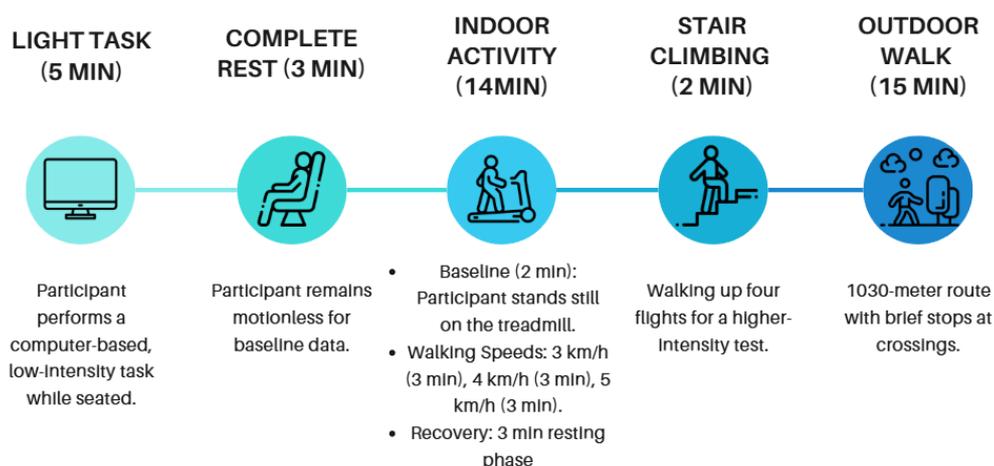


Figure 3.8 - Acquisition protocol

During the first activity ('light task'), the participant performs a computer-based task. This type of activity is low intensity and requires the volunteer to remain seated, performing simple operations that do not involve significant movements, such as using a mobile phone or computer. The duration of this phase is 5 minutes.

After the computer task, the participant enters the 'complete rest' phase, during which they must not perform any movement. This phase is important for collecting baseline data under conditions of total absence of movement and lasts for 3 minutes.

Next, the participant moves on to the treadmill ('indoor activity'). This exercise is structured in several phases, each with a specific speed to test different intensities of movement. The first phase involves 2 minutes of baseline, during which the participant remains stationary on the treadmill without walking. Walking begins at a speed of 3 km/h for 3 minutes, a slow and relaxed pace. Then, the speed increases to 4 km/h for 3 minutes, a moderate yet comfortable pace. Finally, the participant walks at 5 km/h for another 3 minutes, representing a brisk pace. The treadmill activity concludes with a 3-minute recovery phase, during which the participant returns to a resting condition.

The fourth activity is 'stair climbing', a more intense exercise that involves walking up four flights of stairs, with a total duration of approximately 2 minutes.

Finally, the participant performs an outdoor activity, walking a 1030-meter route ('outdoor walk'). During this route, continuous walking is required, with the only exception being brief 15-second stops at pedestrian crossings. The route is completed in about 15 minutes. Figure 3.9 shows the route of outdoor activity.



from the various devices, which provided information in different formats and temporal resolutions. Such pre-processing proved crucial for correctly comparing the measurements and preparing the data for further analysis.

In particular, the focus was on the comparison between the steps recorded by the Bangle.js and those from the IMU-based device, used as the gold standard.

The first phase involved managing the data from the IMU-based device, which was positioned on the participant's shoe and recorded movements through acceleration across three dimensions ( $x$ ,  $y$ ,  $z$ ). After extracting the raw data, we cleaned it by eliminating any anomalies, then converted the measurements into real acceleration values, measured in mg, and synchronized the samples with the temporal data from the other devices. Subsequently, we plotted the data on graphs to visualize changes in movement during different activities and compared the steps recorded by the IMU with those from the Bangle.js.

### 3.6. Step counting from the IMU-based device

From the output files of the IMU-based device, timestamped data on accelerations along the three axes ( $x$ ,  $y$ ,  $z$ ) were obtained. These acceleration data were used to develop the step counter algorithm used as gold standard for our Thesis. The whole process is shown in Figure 3.10:

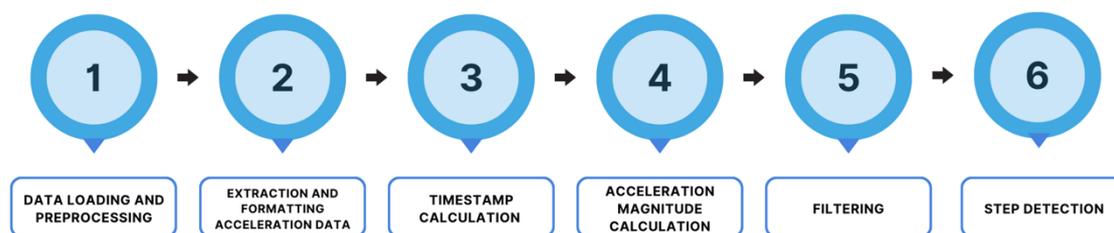


Figure 3.10 - Step counting from the IMU-based device

The algorithm processes the IMU data as follows:

1. **Data loading and preprocessing:** The algorithm begins by reading data from a specified file. Initial details such as sensors used, data rate, and recording time are extracted from the file header. It then reads the main dataset, creating a data frame with data fields for each packet recorded by the IMU-based device.
2. **Extraction and formatting of acceleration data:** Only acceleration data is selected for further processing, filtering out irrelevant fields. The raw hexadecimal data is converted to decimal values, adjusted based on IMU sensitivity settings, and scaled to milligravity units [mg].
3. **Timestamp calculation:** The algorithm calculates timestamps for each data point based on the start time and sampling rate (100 Hz). These timestamps are converted to Unix time format (in milliseconds), allowing for a consistent time scale for further analysis.

4. **Acceleration magnitude calculation:** The algorithm calculates the total acceleration magnitude for each data point using the three-axis data (X, Y, Z) with the formula:

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$

This measure helps to reduce noise from individual axes and provides a single scalar value for each time point.

5. **Filtering with low-pass filter:** A low-pass FIR (Finite Impulse Response) filter is applied to the acceleration magnitude to smooth out high-frequency noise. The cutoff frequency is set to 3 Hz, which allows a variety of walking speeds, and a Hamming window with 300 taps is used. The filter minimizes noise due to rapid, insignificant movements while preserving relevant movement patterns.
6. **Step detection with threshold-based peak identification:** To detect steps, a threshold is calculated based on the mean and standard deviation of the filtered acceleration magnitude as follow:

$$threshold = mean_{magnitude} + n * std_{magnitude}$$

The value of  $n$  was chosen by comparing the numbers of steps obtained with the step counter and the numbers of steps obtained with a manual count. In particular:

- $n = 1$  for light activity, rest activity, outdoor activity
- $n = 0.19$  for stairs activity
- For the treadmill activity
  1.  $n = 0.1$  for the speed 3 km/h
  2.  $n = 0.7$  for the speed 4 km/h
  3.  $n = 1$  for the speed 5 km/h

Peaks that exceed the threshold are identified as steps. The algorithm checks for crossings of the threshold in the upward direction to prevent multiple detections of a single step within proximity.

7. **Visualization and export:** The filtered acceleration magnitude and detected steps are plotted over time for visualization, with the threshold shown as a reference. Finally, the timestamp and step data are saved to an Excel file for further analysis, allowing for an exportable record of detected steps.

Figure 3.11, Figure 3.12, Figure 3.13, Figure 3.14, Figure 3.15 show the accelerometer signals for different activities are processed during different stages of the process.

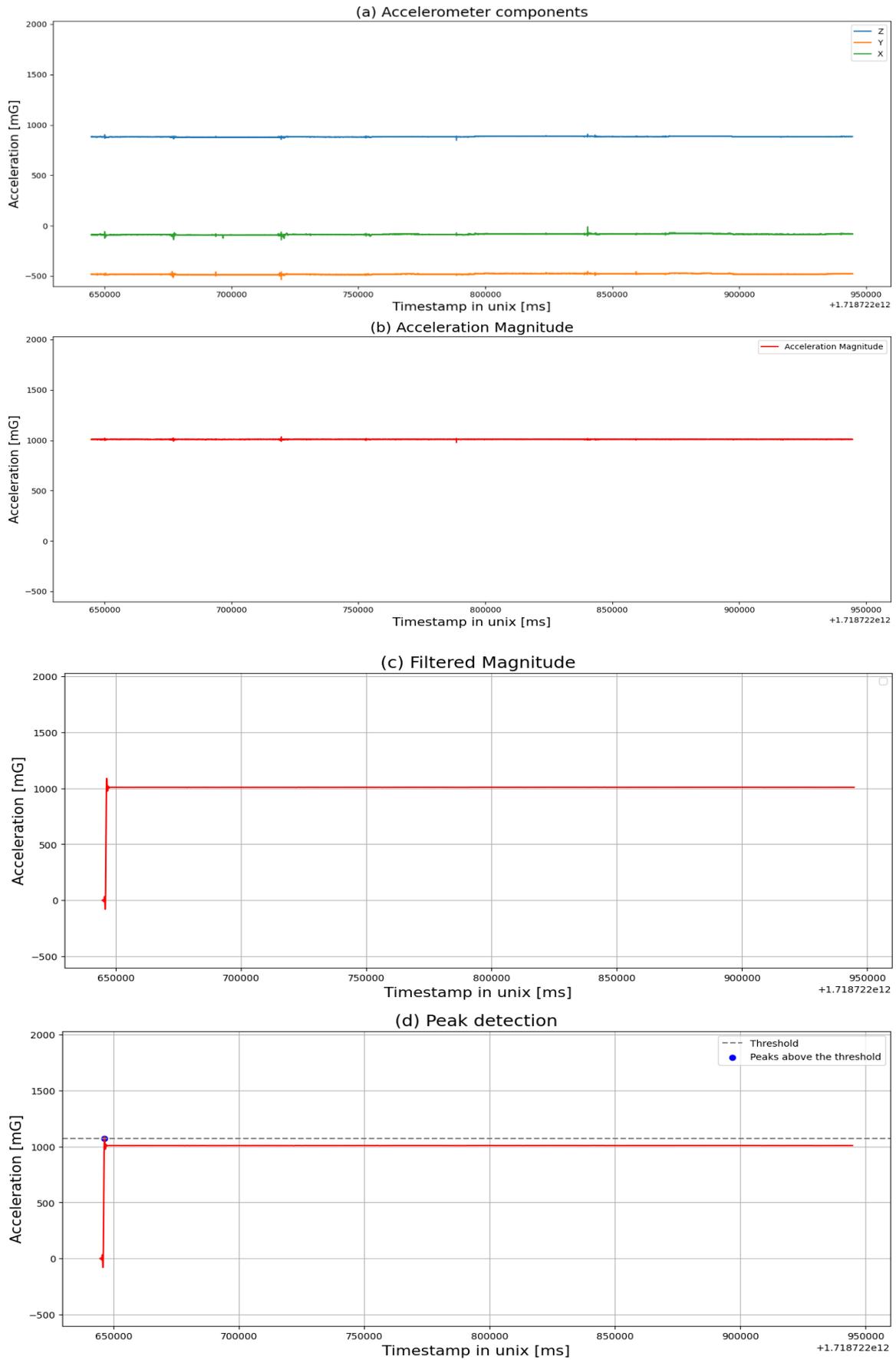


Figure 3.11- Light activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d)

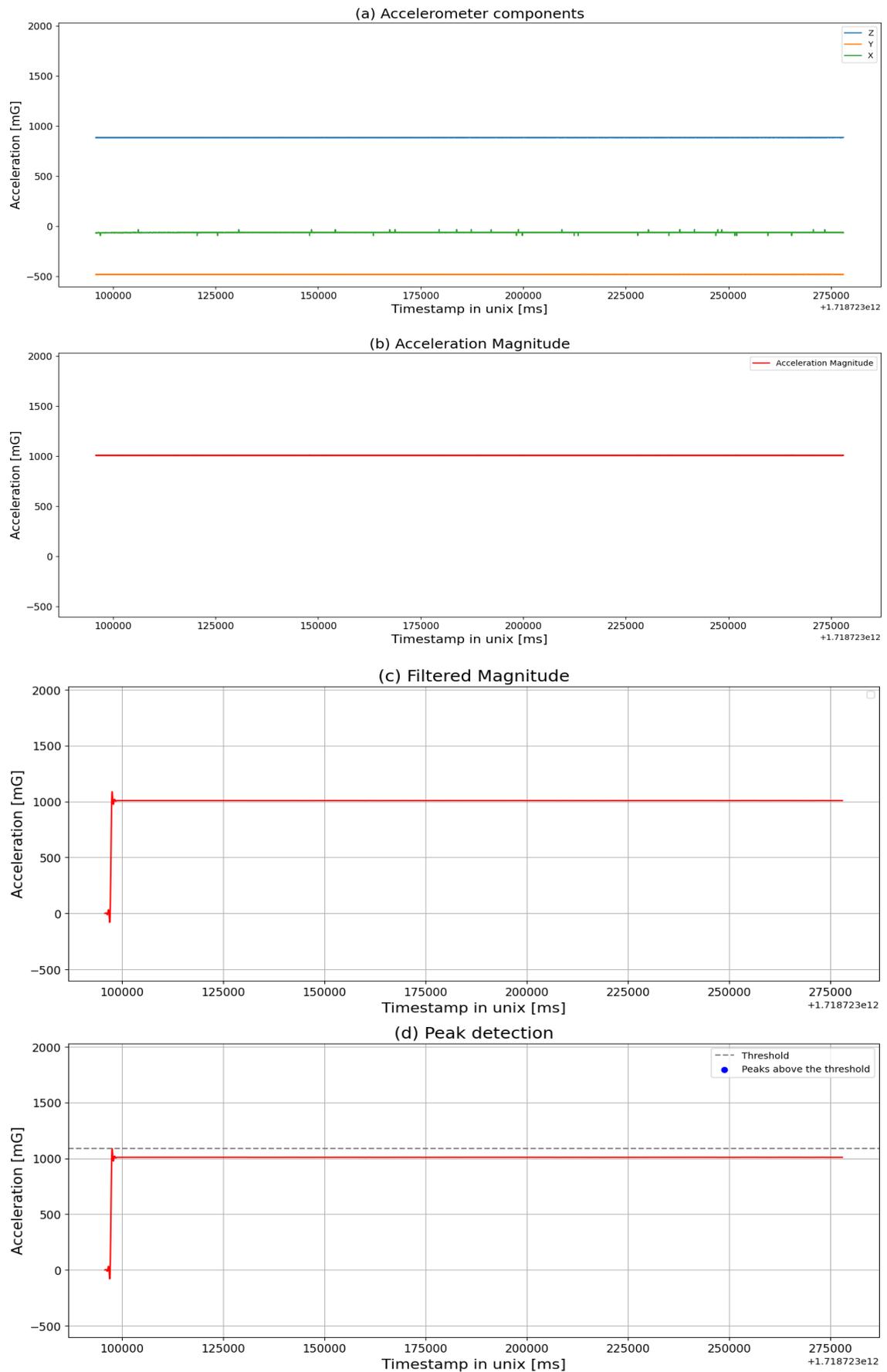


Figure 3.12- Complete Rest: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d)

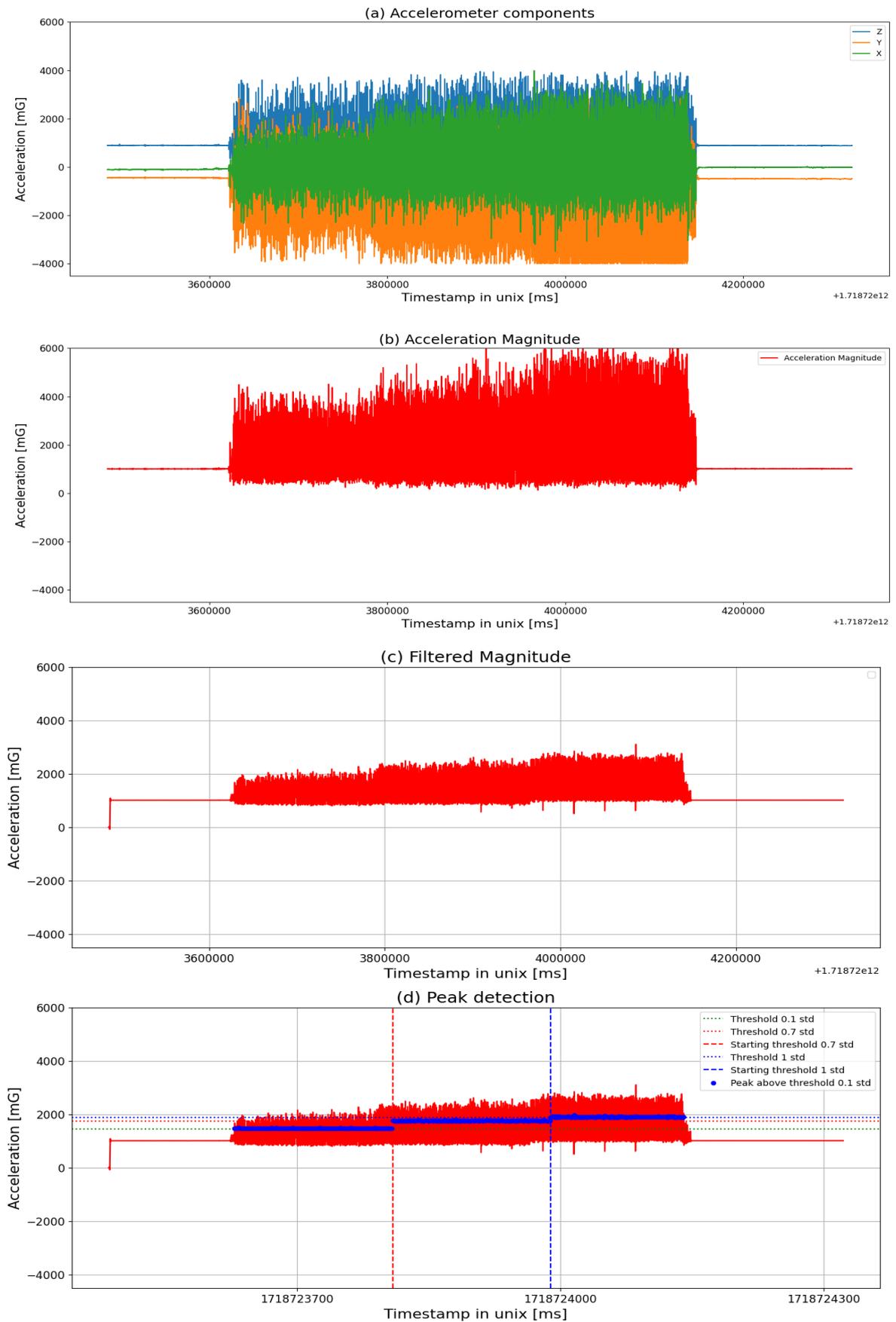


Figure 3.13- Indoor activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d)

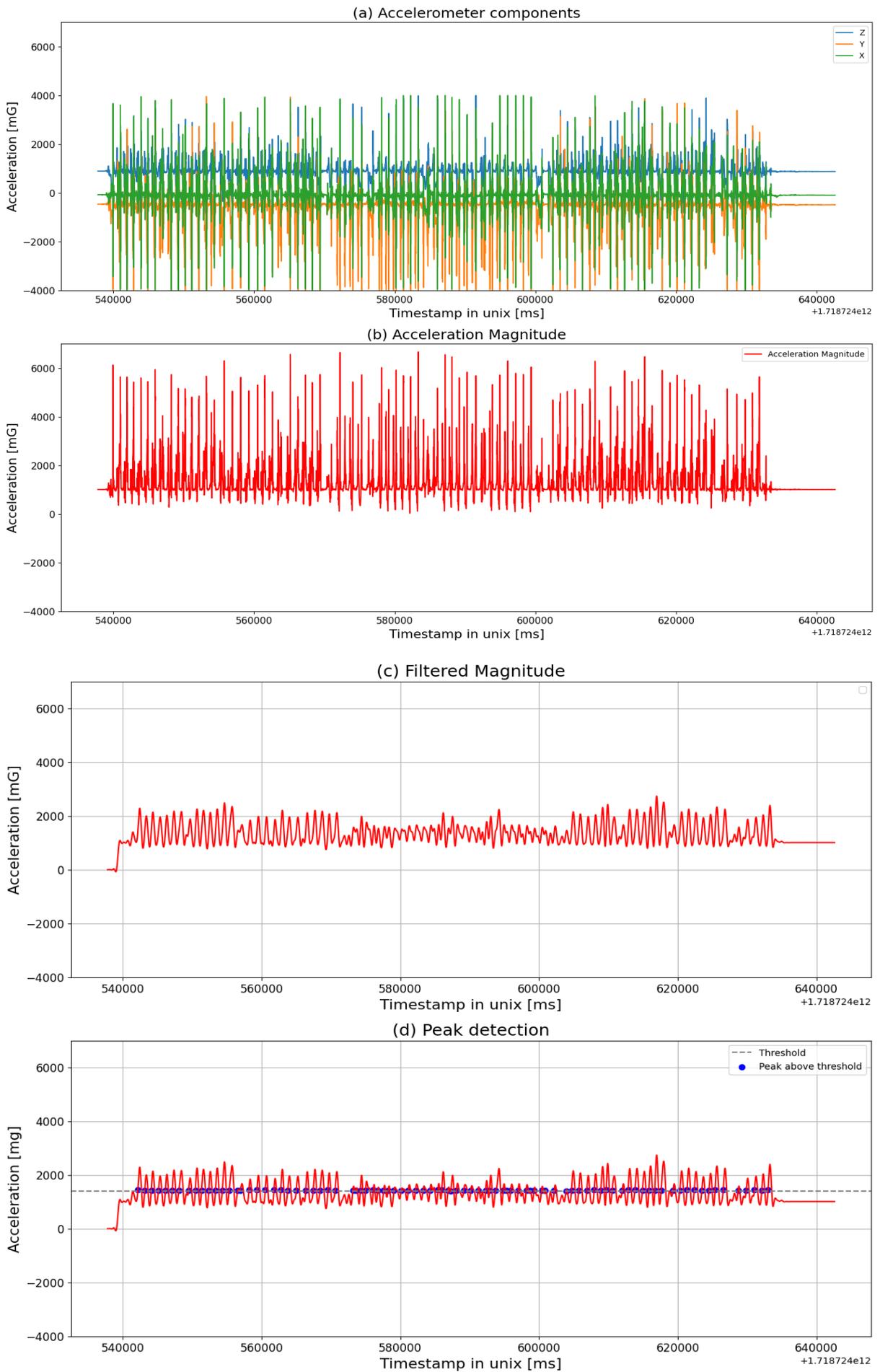


Figure 3.14- Stairs Activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d)

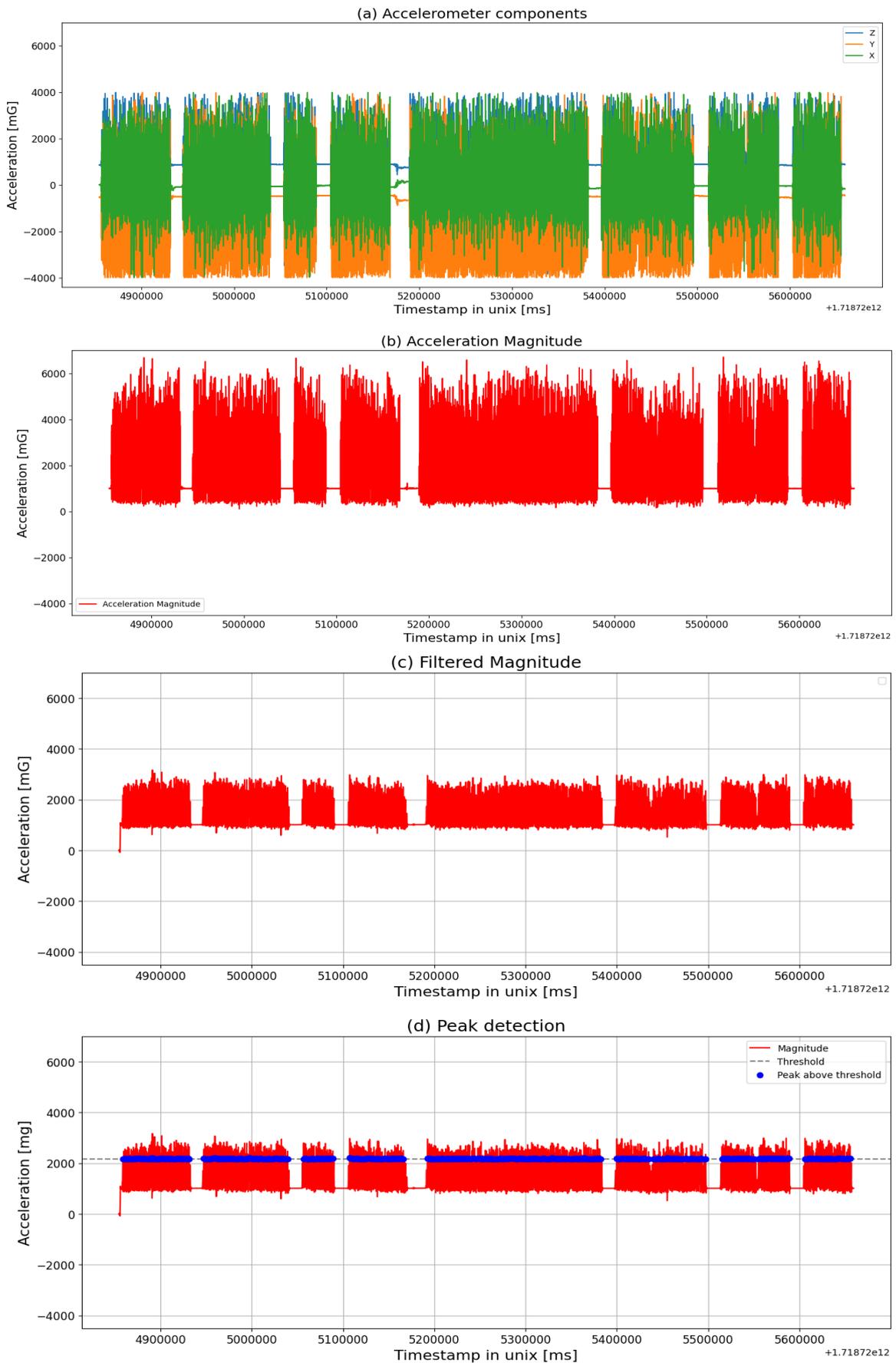


Figure 3.15- Outdoor Activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d)

## 3.7. Step counting from the Bangle.js

After the data collection phase, this Thesis focused specifically on developing step-counting algorithms to be applied on the data from the Bangle.js. Future work will extend this approach to adapt these algorithms for pulse rate monitoring.

The algorithms were implemented in C, leveraging the Visual Studio Code environment for streamlined development and debugging. The selection of step-counting algorithms was informed by an extensive preliminary literature review, prioritizing methods recognized for their accuracy and resource efficiency, critical for wearables where processing power and battery life are constrained. As part of the analysis, six step-counting algorithms were selected and implemented:

- i. The **Dummy algorithm**, a baseline algorithm that performs basic step detection. Its purpose is to serve as a point of comparison with more advanced algorithms.
- ii. The **Bangle Simple** algorithm, an earlier open-source version designed for the smartwatch
- iii. The **Autocorrelation-based algorithm**
- iv. The **Espruino algorithm**, pre-installed on the Bangle.js smartwatch
- v. The **Fast Fourier Transform-based algorithm**.
- vi. The **Oxford algorithm**, a project developed by Jamieson Brynes at the University of Oxford

The implementation details, performance evaluations, and comparisons of these algorithms are discussed in the following sections.

### 3.7.1. Benchmarking framework

The first step was the creation of a benchmarking framework to systematically test and compare the performance of multiple step-counting algorithms. The framework allows for the simultaneous execution of several algorithms on the same dataset, consisting of accelerometer data collected during the data acquisition of the previous experimental campaign.

The structure of the benchmark is flexible and modular, enabling the inclusion of different algorithms with specific initialization processes and step-counting functions. Once the algorithms are loaded, the accelerometer data collected from the Bangle.js smartwatch, stored in .csv format, is processed by each algorithm. The total step counts calculated by each algorithm are then compared to the step counts obtained from the algorithm using accelerometer data derived from the IMU-based device. The benchmark framework manages variations in time intervals and data formats to ensure consistent evaluation across all algorithms.

### 3.7.2. Dummy algorithm

The Dummy step counter algorithm is a simple implementation designed to serve as a baseline for comparison with more complex step-counting algorithms. Its logic is intentionally straightforward, providing a basic reference point. This algorithm was also used to eliminate other algorithms performing below this baseline, ensuring that only methods demonstrating greater accuracy and reliability were retained for further analysis. The whole process is shown in Figure 3.16:

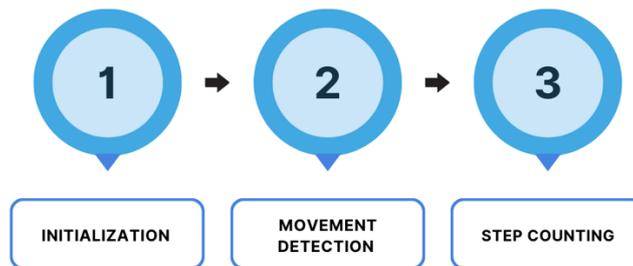


Figure 3.16 - Dummy algorithm

The algorithm works as follows:

- **Initialization.** At the beginning, the algorithm initializes a variable (*dummy\_time\_passed*) that tracks the total elapsed time. A movement detection buffer is also set up using a circular buffer structure. This buffer is used to store accelerometer data for detecting movement.
- **Movement detection.** For each new accelerometer sample, the algorithm calculates the magnitude of the acceleration vector. This is done using the Euclidean norm of the accelerometer data across the x, y, and z axes. The calculated magnitude is then passed to a movement detection function (discussed in the following paragraphs), which determines whether the device is currently moving.
- **Step counting.** If movement is detected and there has been a time increment, the algorithm accumulates the total time passed since the start of the activity. Instead of detecting individual steps, the Dummy algorithm assumes a constant rate of steps per second (in this case, 2 steps per second). It simply multiplies the total time passed by this step rate to estimate the total number of steps, as shown in the following equation:

$$total\_steps = total\_time\_passed \times step\_rate$$

### 3.7.3. Bangle- simple algorithm

The Bangle simple step counter is a straightforward algorithm designed to detect steps based on changes in acceleration. It is one of the original step-counting algorithms developed for the Bangle.js smartwatch, using minimal logic to identify steps by analyzing the magnitude of the acceleration vector. The entire process is illustrated in Figure 3.17:

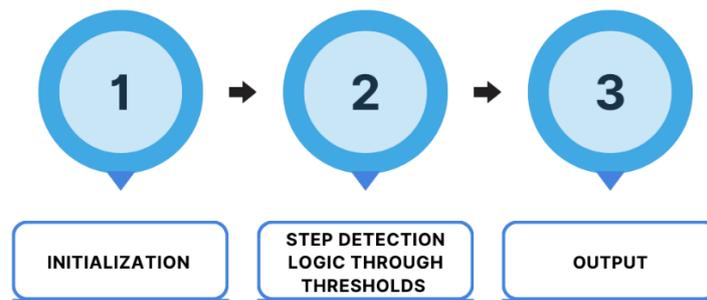


Figure 3.17 - Bangle algorithm

The algorithm works as follows:

1. **Initialization.** The algorithm starts by initializing two key variables:
  - *bangle\_simple\_StepWasLow*: a boolean flag that tracks whether the acceleration magnitude has fallen below a predefined threshold.
  - *bangle\_simple\_StepCount*: a counter that tracks the total number of steps detected.

During initialization, the step counter is reset to zero, and the flag is set to false.

2. **Thresholds.** The algorithm uses two predefined thresholds to detect step patterns:
  - *bangle\_simple\_ThresholdLow*: this defines how low the squared acceleration magnitude must fall before the algorithm considers the next rise a potential step.
  - *bangle\_simple\_ThresholdHigh*: this defines how high the squared acceleration magnitude must rise for it to be recognized as a step.

These thresholds are designed to capture the characteristic up-and-down motion of a step.

3. **Step detection logic.** For each accelerometer sample (with acceleration data in the *x*, *y*, and *z* axes), the algorithm calculates the squared magnitude of the acceleration vector. The logic is based on comparing the acceleration magnitude against predefined thresholds, specifically:
  - *Low Threshold Check*: if the squared acceleration magnitude falls below the *bangle\_simple\_ThresholdLow*, the flag *bangle\_simple\_StepWasLow* is set to true, indicating that the acceleration has dropped low enough for a possible step.
  - *High Threshold Check*: if the squared magnitude exceeds *bangle\_simple\_ThresholdHigh* and the *bangle\_simple\_StepWasLow* flag is true, a step is detected. The flag is reset to false, and the step counter is incremented.
4. **Output.** The algorithm returns the updated step count after each accelerometer sample is processed. This step count is based on the detection of the low-to-high transitions in the acceleration magnitude, which correspond to the characteristic motion of a step.

### 3.7.4. Espruino algorithm

The Espruino step counting algorithm [33] is a more advanced approach compared to simpler threshold-based methods. It involves filtering accelerometer data, applying a state machine to detect valid steps, and maintaining a history of detected steps to improve accuracy. The full procedure is depicted in Figure 3.18:

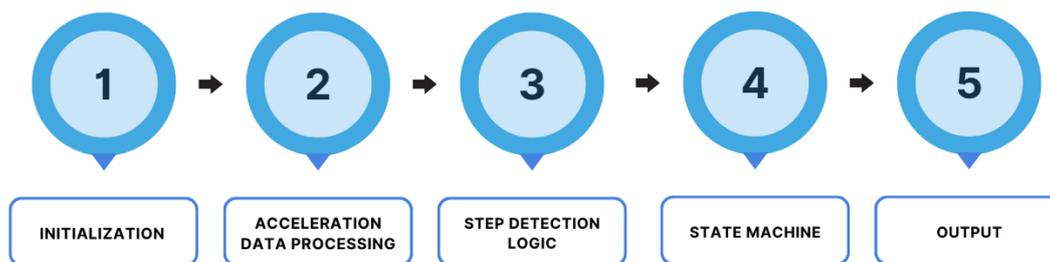


Figure 3.18 - Espruino algorithm

The algorithm works as follows:

1. **Initialization.** The algorithm begins by initializing necessary components:
  - a. *Step count state machine:* the step counter uses a state machine to track the progression of detected steps.
  - b. *Acceleration filters:* the algorithm initializes filters to smooth out noise in accelerometer data, ensuring that only valid step-like movements are considered.
  - c. *DC filter:* This filter removes the direct current (DC) component from the signal to ensure that variations around a mean value are detected, which is essential for detecting steps.

The `stepcount_init()` function sets the initial values for the various filters and state machine, preparing the algorithm to begin processing accelerometer data.

2. **Accelerometer data processing.** The core of the algorithm processes accelerometer data at a frequency of 12.5 Hz. The squared magnitude of the accelerometer data ( $x$ ,  $y$ ,  $z$  axes) is calculated. This value is passed through multiple filters:
  - a. *DC Filter:* removes the DC component from the signal, allowing the algorithm to focus on relative changes in movement.
  - b. *Moving Average Filter:* a 7-tap FIR filter smooths the data to reduce noise and to eliminate small fluctuations in the acceleration signal. This filter uses a predefined set of weights (taps) to apply to the accelerometer data points, improving the detection of significant movement patterns.
3. **Step detection logic.** After filtering, the algorithm evaluates whether the magnitude of acceleration crosses certain thresholds to indicate a step:

- a. *Step History*: the algorithm maintains a history of detected steps to avoid counting noise or small motions as steps. A step is only counted if it occurs in a sequence of valid steps, ensuring that step detection is consistent and avoids false positives.
  - b. *Thresholds*: the algorithm uses a state machine to track whether the acceleration exceeds certain thresholds within a defined time window. If the signal passes a low threshold, the algorithm expects a rise above a higher threshold to confirm a step.
4. **State machine for step counting.** The state machine ensures that only valid steps are counted. It tracks the progression of step-like movements and ensures they occur within the expected time intervals. Table 3.3 shows the states used:

<i>S_STILL</i>	The initial state, where no steps are detected
<i>S_STEP_1</i>	The algorithm detects the first step and begins tracking the time until the next step.
<i>S_STEP_22N</i>	This state is used when multiple steps are detected in sequence. The algorithm expects a certain number of steps ( <i>X_STEPS</i> ) within a specified time range to confirm consistent walking.
<i>S_STEPPING</i>	Once a valid sequence of steps is detected, the algorithm enters this state and continues counting each step detected within the expected time frame.

Table 3.3 - State machine

- If the steps occur too quickly or too slowly, the state machine resets, avoiding false positives.
5. **Espruino Step Counter Wrapper.** The Espruino wrapper is a simple interface that uses the step counting logic defined above. It takes accelerometer samples and processes them using the core step-counting logic. The wrapper keeps track of the cumulative step count and returns it after each sample is processed. This wrapper simplifies the integration of the algorithm with the Espruino platform by abstracting the underlying details of filtering and state machines, making it easy to use in real-time applications on wearable devices.
  6. **Output.** The algorithm continuously tracks the number of steps based on the filtered accelerometer data. Each valid step is added to the total count, and the step history ensures that steps are not counted multiple times if they occur too quickly or irregularly.

### 3.7.5. Autocorrelation algorithm

The autocorrelation algorithm is based on the work by Neraj Bobra [34], which performs step counting by analyzing the accelerometer signal through an autocorrelation process. The sequence of operations includes filtering, mean removal, autocorrelation calculation, and peak evaluation within the autocorrelation function to identify valid steps. The complete process is presented in Figure 3.19:

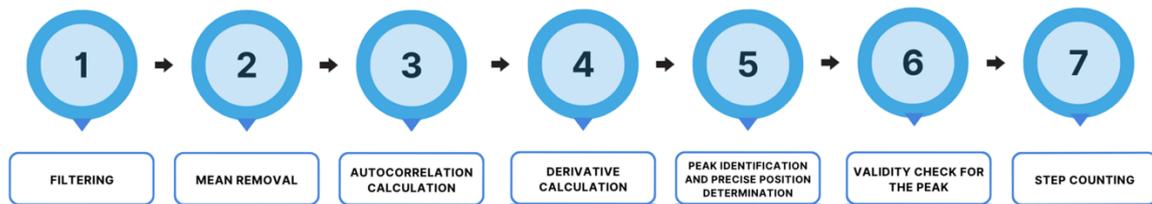


Figure 3.19 - Autocorrelation algorithm

The algorithm works as follows:

- 1. Low-Pass Filtering.** The first step involves applying a low-pass filter to the accelerometer signal to reduce high-frequency noise and preserve the components relevant to step movement. The *lowpass\_buffer()* function takes the magnitude of the raw accelerometer signal buffer as input and returns a filtered buffer using an *LPFilter* structure.
- 2. Mean Removal.** The mean of the signal is removed using the *remove\_mean()* function. This centers the signal around zero, improving the accuracy of the subsequent autocorrelation and reducing the impact of low-frequency variations that are irrelevant for step counting. Figure 3.20 shows an example of how the signal, within a window of 50 tuples, changes after the magnitude calculation, filtering and averaging removal stage.

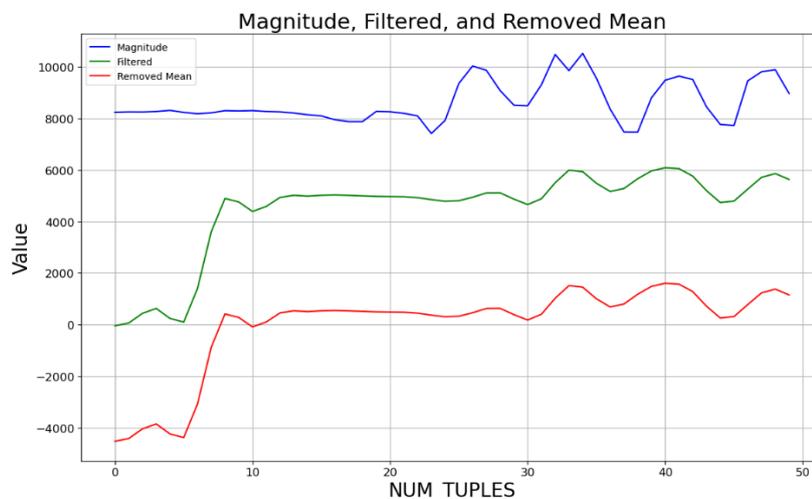


Figure 3.20-Magnitude, Filtered and Removed mean

3. **Autocorrelation Calculation.** The `autocorr()` function calculates the autocorrelation of the filtered signal using a predefined buffer with a number of lags set by the constant `NUM_AUTOCORR_LAGS`. In this case, `NUM_AUTOCORR_LAGS` is set to 16, chosen to cover a range of typical walking frequencies. This approach assumes a sampling frequency of 12.5 Hz (e.g., 1000 ms / approximately 80 ms sampling period). This lag choice is sufficient to cover frequency ranges from walking to running, where the maximum step frequency can be up to around 3 steps per second (333 ms per step, corresponding to 4 lags).
4. **Derivative Calculation.** To detect the first positive peak, the derivative of the autocorrelation function is calculated using the `derivative()` function. A derivative filter with the constant `DERIV_FILT_LEN = 7` is used, applying the coefficients stored in `deriv_coeffs`. The derivative is used to identify sign changes indicating peaks in the autocorrelation signal, specifically transitions from positive to negative values. Figure 3.21 shows an example of the peak present in the autocorrelation signal being indicated by the change of sign of the derivative, from positive to negative.

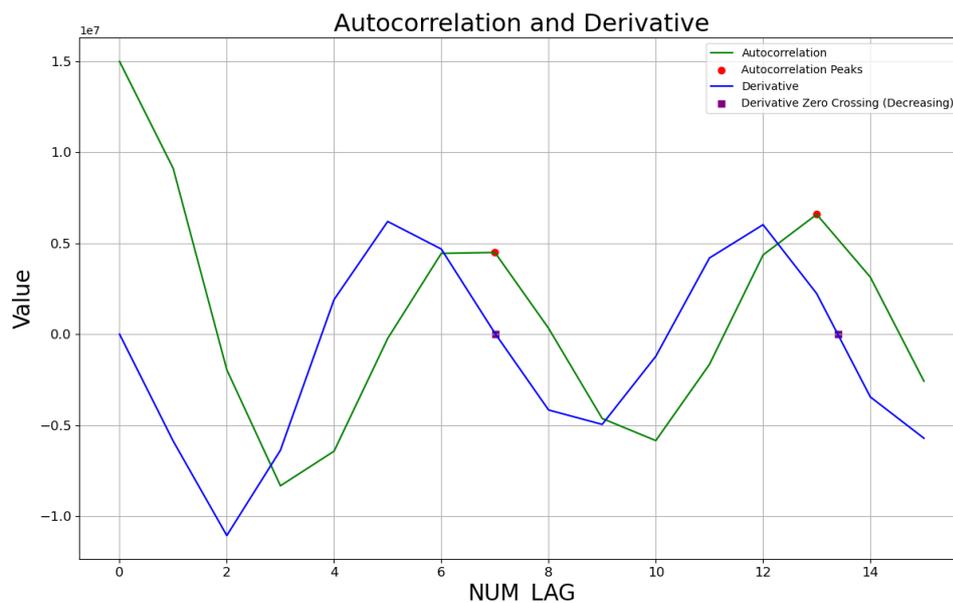


Figure 3.21- Peak detection through derivative phase

5. **Peak Identification and Precise Position Determination.** The function searches for the first peak in the derivative that meets the positive-to-negative sign transition criteria starting from the constant `FIRST_AUTOCORR_PEAK_LAG`, set to 4, to avoid non-representative peaks (e.g., noise). The exact peak position is refined with `get_precise_peakind()`, which verifies that surrounding values are lower than the current peak value.

6. **Validity Check for the Peak.** To ensure the detected peak is significant enough and truly represents a step, the code checks several parameters using the `get_autocorr_peak_stats()` function:
- **Number of Points in Descent and Ascent:** the code verifies that there are at least two points with negative and positive slope (`AUTOCORR_MIN_HALF_LEN = 2`) to the right and left of the peak, respectively, ensuring that the peak is well-defined.
  - **Amplitude Delta Between Peak and Valley:** the amplitude difference between the peak and the minimum points (valleys) to the left and right of it is calculated. This difference must exceed a threshold (`AUTOCORR_DELTA_AMPLITUDE_THRESH = 1e6`) to avoid considering very small peaks or noise.
7. **Step Counting.** If all the above conditions are met, the peak is considered valid for step counting. The step count is calculated by dividing the sampling frequency (`SAMPLING_RATE`) times the time window length (`WINDOW_LENGTH`) by the peak position (`peak_ind`), which represents the period of the signal.

Table 3.4 presented the values of the parameters used in the defines, along with an explanation of their selection.

Define	Value	Explanation
NUM_TUPLES	50	Specifies the number of samples collected over an interval of 4 seconds. Together with <code>SAMPLING_RATE</code> , it determines the size of the data buffers and window for autocorrelation calculations.
WINDOW_LENGTH	(NUM_TUPLES / SAMPLING_RATE)	Length of the time window in seconds, based on the number of samples and sampling rate. Here, it is set to 4 seconds.
NUM_AUTOCORR_LAGS	16	The choice ensures that the autocorrelation function captures a full step cycle, providing enough coverage for step frequencies ranging from 1

		to 3 steps per second, with some additional lags to improve accuracy and handle noise at a 12.5 Hz sampling rate
FIRST_AUTOCORR_PEAK_LAG	4	The choice for the first feasible autocorrelation lag corresponds to the minimum number of samples required to cover a 333 ms step cycle (step rate of 3 step per second) at a 12.5 Hz sampling rate (333 ms / 80 ms per sample $\approx 4$ ).

Table 3.4 - Defines used in Autocorrelation algorithm

### 3.7.6. Fast Fourier Transform algorithm

In the case of the Fast Fourier Transform (FFT) algorithm, the code implements an algorithm for step counting by analyzing the accelerometer signal through the FFT. The sequence of operations includes data collection, segmentation into time windows, analysis of the dominant frequency within each window, and calculation of the total number of steps. The entire workflow is demonstrated in Figure 3.22:



Figure 3.22 - Fast Fourier Transform algorithm

Below is a detailed explanation of the key steps involved in the process.

1. **Data Preparation.** The first phase of the process involves collecting acceleration data. The raw signal, consisting of accelerations along the x, y, and z axes, is combined into a magnitude of acceleration, calculated as the square root of the sum of the squares of the accelerations along the three axes. The data is then stored in a circular buffer of length `WINDOW_LEN` (32 samples), which is read and updated continuously using a cyclic index, `signal_buffer_next_i`. Figure 3.23 shows the magnitude of acceleration as the sample changes within sliding window.

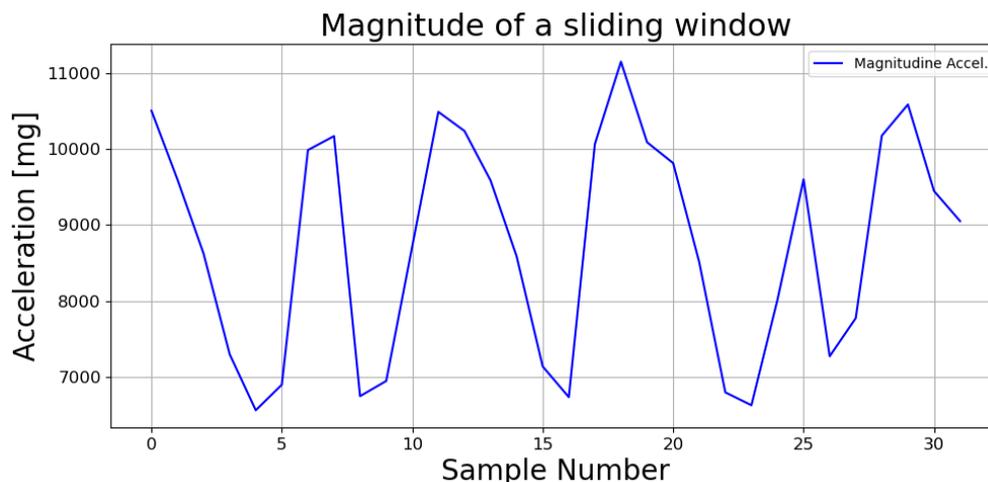


Figure 3.23- Magnitude phase in FFT algorithm

2. **Segmentation into Time Windows.** The signal is segmented into time windows of length *WINDOW\_LEN* (32 samples). Each window is analyzed separately, and it moves by *WINDOW\_STEP* (12 samples) at a time. This means each window overlaps with the previous one, allowing for a smooth transition between consecutive analyses.
3. **Detection of Significant Movement.** After each new sample is added to the buffer, the code calculates the magnitude of the signal in each window and checks whether the difference between the maximum and minimum values in the window exceeds a threshold defined by *MOVEMENT\_DETECTION\_THRESHOLD* (1500). If the difference exceeds the threshold, the system considers that there has been a significant movement, indicating a potential step.
4. **Fast Fourier Transform.** When significant movement is detected, the code prepares the data for frequency analysis by applying the FFT. Each window of data is converted into an array of complex numbers, where the real values (accelerations) are used as the real part, and the imaginary part is set to zero. The FFT is then applied to this data window via the FFT function, which performs the Fourier transformation on the signal and computes the dominant frequency in the frequency domain.
5. **Identification of Dominant Frequency.** Once the FFT is performed, the code analyzes the output to find the dominant frequency. The FFT output is a series of complex values representing the various frequencies in the signal. The code examines frequency indices between *MIN\_FREQ\_FFT\_I* (2) and *MAX\_FREQ\_FFT\_I* (7), corresponding to the frequency range typically associated with steps. The magnitude of each FFT component is calculated, and the index corresponding to the frequency with the highest magnitude is selected as the dominant frequency. Figure 3.24 shows the magnitude of the frequency components of a sliding window. In the highlighted window, the index with the highest magnitude corresponds to 5. Figure 3.25 shows the dominant frequency detected in consecutive

windows, which provides insights into changes in stepping frequency (faster/slower steps).

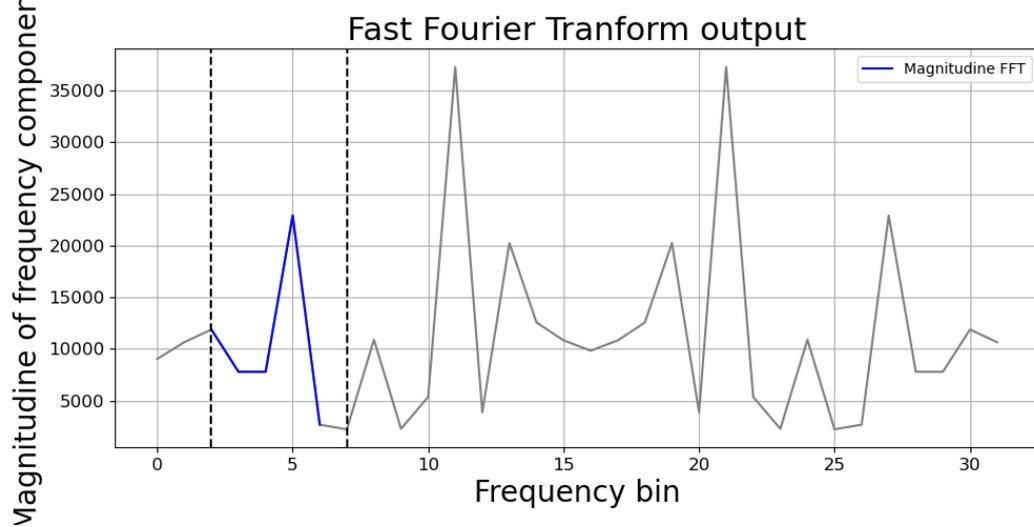


Figure 3.24 -Magnitude of the frequency components between the frequency bins [2;7]

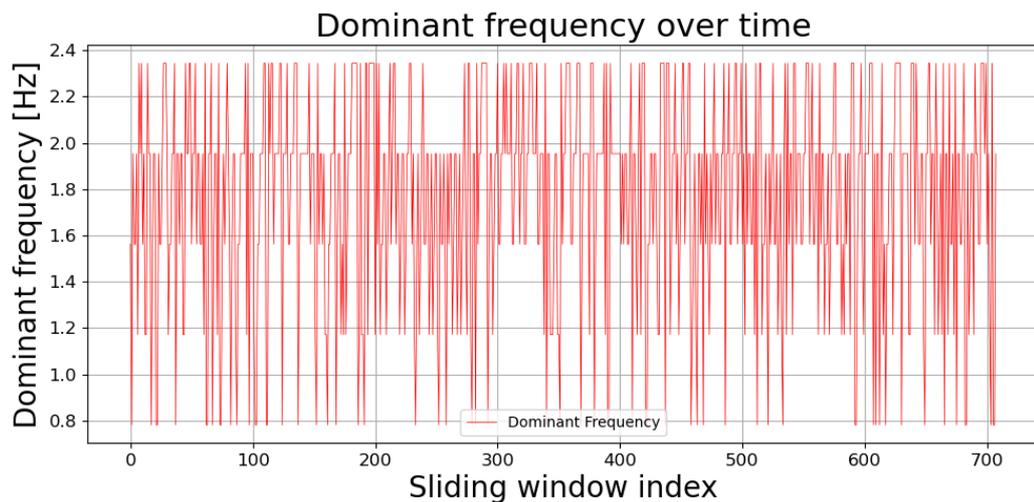


Figure 3.25-Dominant frequency over time

6. **Step Count Calculation.** Once the dominant frequency is identified, the code uses this frequency to calculate the number of steps. The dominant frequency (in Hz) is multiplied by the time represented by the window step size `WINDOW_STEP` and divided by the sampling frequency `SAMPLING_FREQ` to determine the number of steps detected in that period. This value is then accumulated in the variable `total_steps` to maintain the running total step count.
7. **Return Total Step Count.** After processing each window, the total step count is updated and returned by the function `fft_stepcount_totalsteps()`. This value represents the cumulative number of steps detected up to that point.

Table 3.5 presented the values of the parameters used in the defines, along with an explanation of their selection.

Define	Value	Explanation
WINDOW_LEN	32	The sliding window contains 32 samples of the acceleration signal. This value is chosen to be a power of 2, which facilitates the implementation of FFT, as it is more efficient when the input size is a power of 2.
WINDOW_STEP	12	The number of samples the window moves at each step. This value allows for overlap between consecutive windows and provides a time resolution of approximately 1 second with sampling frequency of 12.5 Hz.
MIN_FREQ_FFT_I	2	These indices define the frequency range to focus the FFT analysis on. These values are chosen to encompass the frequency range typically associated with human steps, ranging from 1 Hz to about 3 Hz. For N=32 and Fs=12.5 Hz, the indices are calculated as follows: $i = \frac{f_i * 32}{12.5}, \text{ for } i = 0,1, \dots, 31$

Table 3.5 - Defines used in FFT algorithm

### 3.7.7. Oxford Algorithm

The Oxford algorithm is an implementation of a step detection system designed for wearable devices in healthcare applications. It is based on the open-source algorithm [35] developed by Anna Brondin and Marcus Nordström at Malmö University. The system efficiently processes accelerometer data to detect and count steps, with a focus on optimizing performance for embedded systems. The whole sequence is displayed in Figure 3.26:



Figure 3.26 - Oxford algorithm

The algorithm works as follows:

1. **Pre-processing stage.** Data from the accelerometer's three orthogonal axes ( $x$ ,  $y$ ,  $z$ ) are processed to compute the magnitude:

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$

The implementation also includes a time scaling mechanism to ensure the time values are adjusted to a consistent scale, in milliseconds, for the processing of the samples. The time for each sample is updated cumulatively by adding the delta time, which is adjusted by the scaling factor. The variable *cumulative\_time* tracks the total time in milliseconds.

2. **Filtering stage.** A low-pass FIR filter is applied to the accelerometer data to remove high-frequency noise, focusing on the relevant signal frequencies (below 3 Hz). This stage implements a fixed-point FIR filter, optimized for embedded systems, which processes data in batches and outputs filtered magnitudes for further analysis. Figure 3.27 shows an example of how the accelerometer signal changes after the pre-processing and filtering phase.

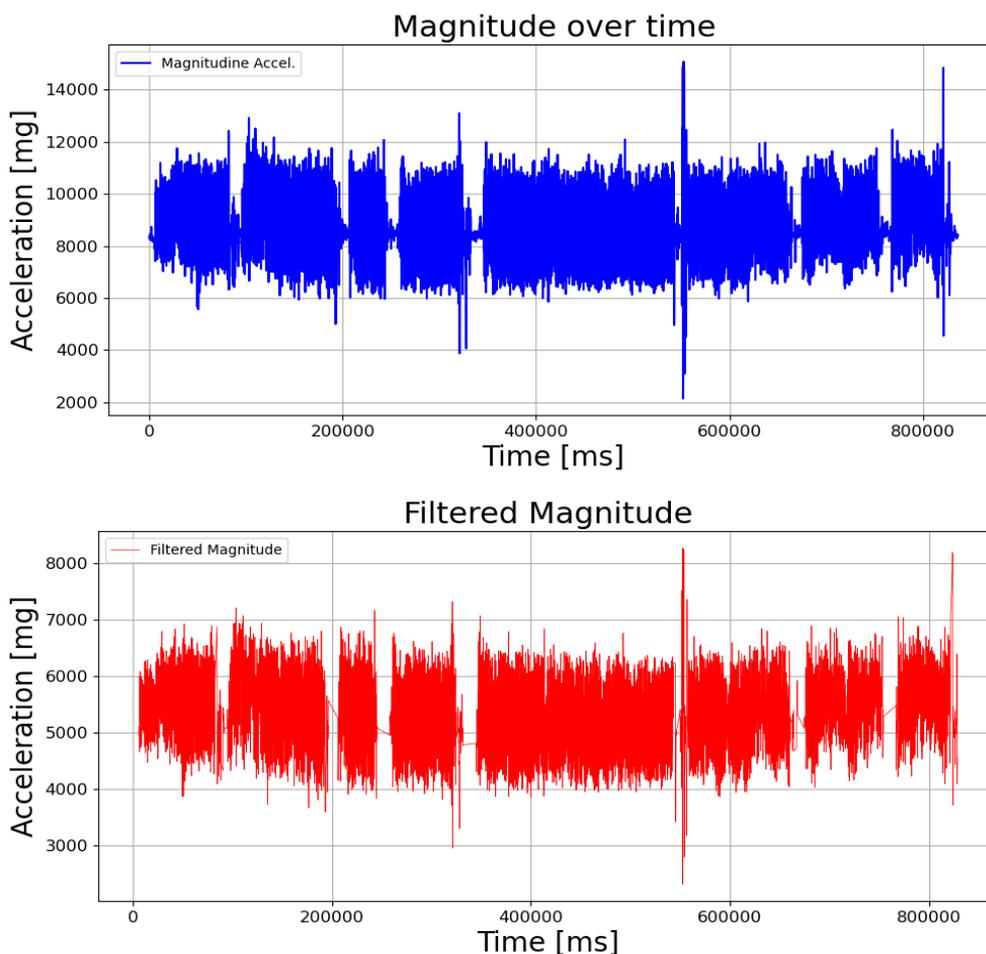


Figure 3.27- Pre-processing and filtering stages in Oxford algorithm

3. **Scoring stage.** This stage identifies peaks in the signal by accentuating local variations within a moving window of data points. This process calculates the average difference between the magnitude of the central point in the window and the magnitudes of the other points in the window, dividing the window into two parts: left (before the midpoint) and right (after the midpoint).

**Left and right differences.** The algorithm starts by selecting the central point in the window, denoted as  $M$  (the midpoint), and then iteratively calculates the difference between  $M$ 's magnitude and each point's magnitude in the left and right half of the window. If we let  $x_i$  represent each magnitude within the window and  $x_m$  the magnitude at the midpoint, then:

$$diffLeft = \sum_{i=0}^{midpoint-1} (x_M - x_i)$$

This represents the sum of differences for the left side, where each value before the midpoint is subtracted from  $x_m$ . Similarly:

$$diffRight = \sum_{j=midpoint+1}^{windowSize-1} (x_M - x_j)$$

This represents the differences for the right side of the window.

**Scoring peak calculation.** Once the differences for the left and right sides are obtained, the scoring peak *scorePeak* is calculated by summing *diffLeft* and *diffRight* and dividing by the total number of comparisons in the window, which is  $windowSize-1$ :

$$scorePeak = \frac{diffLeft + diffRight}{windowSize-1}$$

This division provides an average difference, effectively emphasizing points that differ most from the central magnitude, hence identifying peaks.

4. **Detection phase.** The detection phase identifies significant peaks by comparing each new data point's magnitude against a dynamically updated mean and standard deviation. As each data point is processed, the algorithm recalculates the mean and standard deviation based on the previous values, allowing it to detect outliers that indicate peaks. Initially, the algorithm accumulates enough data points (at least 15) to establish a baseline mean and standard deviation. With each new data point, it then updates these values incrementally, without needing a full recalculation from scratch. The key detection rule checks whether a data point's magnitude exceeds a threshold set relative to the mean and standard deviation:

$$dataPoint.magnitude - mean > \sigma \times threshold\_int + \frac{\sigma}{threshold\_frac}$$

Here, *threshold\_int* and *threshold\_frac* control the sensitivity of the peak detection. When a data point meets this condition, it's marked as a peak, added to the output buffer, and forwarded to the next processing stage. This approach ensures that only significant changes in the data trigger a detection, making the system more robust against minor fluctuations.

5. **Post-processing phase.** This phase is responsible for ensuring that detected peaks are spaced sufficiently apart in time to be considered as valid steps. This is done by checking the time difference between consecutive data points and discarding those that occur too close together. The time threshold ensures that closely spaced peaks are filtered out, preventing the detection of multiple steps for a single movement. If the current data point has a higher magnitude than the previous one and does not meet the time threshold, the algorithm keeps the newer data point, as it may indicate a stronger signal. The detection logic can be described by the following condition:

$$dataPoint.time - lastDataPoint.time > timeThreshold$$

If the time difference is greater than the threshold, the data point is considered valid, and the step callback is invoked.

In Table 3.6 presented the values of the parameters used in the defines, along with an explanation of their selection.

Define	Value	Explanation
FILTER_TAP_NUM	15	Number of filter coefficients (15). Defines the FIR filter's complexity and behavior.
filter_taps[]	12	Array of FIR filter coefficients used to pass low frequencies (0-3 Hz) and attenuate higher ones (4-6.25 Hz)
MOTION_THRESHOLD	1500	Threshold for detecting significant movement
WINDOW_SIZE	5	A window size of 5 allows detecting steps quickly, with 5 samples at 12.5 Hz (one every 0.08 s), making it responsive to changes while handling up to 3 steps per second (around 333 ms).
TIME_THRE	300	Maximum time between steps (300 ms). Used to prevent counting too rapid steps as separate events.

Table 3.6 - Defines used in the Oxford algorithm.

### 3.8. Movement Detection Phase

The movement detection phase operates by analyzing the magnitude of acceleration data over a brief, rolling time window. This phase uses a buffered moving average approach, which maintains a running history of recent acceleration magnitudes. By continuously updating and assessing this buffer, the algorithm can identify when significant movement occurs.

This provides a reliable means of distinguishing movement from stationary periods, serving as a preliminary filter before more computationally intensive processing. By rejecting periods with minimal variation, the detection phase reduces the workload on subsequent stages of the algorithm, thereby conserving computational resources. It is particularly useful in wearable devices where CPU efficiency is paramount.

The detection phase also minimizes false-positive detections from minor shifts or noise in the signal, improving the accuracy of step-counting and other activity recognition algorithms. This approach to movement detection ensures that only meaningful acceleration changes, such as those from walking, running, or other physical activities, trigger further processing, making it an essential step in step-counting algorithms designed for wearable devices.

The algorithm processes acceleration data in the following steps:

1. **Buffer initialization and management.** The detection buffer is initialized with a defined length, *DETECTION\_BUFFER\_LEN*, which is set to hold approximately one second of data at a sampling frequency of 12.5 Hz. This length allows the detection algorithm to assess a recent history of accelerometer readings, capturing both minor and major changes in motion.
2. **Updating the buffer.** For each new sample, the algorithm adds the latest magnitude value to the buffer. When the buffer is full, the oldest data point is removed, allowing the buffer to maintain a fixed size. This process ensures that only the most recent acceleration values contribute to the movement analysis.
3. **Movement detection criteria.** The detection logic calculates the range of magnitudes within the buffer by finding the difference between the maximum and minimum values. If this range exceeds a predefined threshold (*DETECTION\_THRESHOLD*), movement is detected. The threshold value has been experimentally determined to minimize false positives and accurately reflect true movement patterns. If the difference between the maximum and minimum values is below the threshold, the sample is classified as non-movement.

## 3.9. Performance metrics

The benchmark algorithm generates two types of output files:

1. **Summary Table:** This file provides a high-level summary of results for each input file, detailing the total step count calculated by each algorithm, the corresponding number of computational cycles (used to assess the computational cost), and the reference step count obtained from the gold standard algorithm.
2. **Detailed Step Count Files:** For each input file, a separate output file is generated with detailed results. The first column contains the timestamps in milliseconds, while subsequent columns show the cumulative step counts calculated by each algorithm at those specific times, allowing for time-matched comparisons.

In parallel, an additional file is created for each accelerometer data file from the IMU-based device, containing timestamps in milliseconds and the cumulative step count calculated by the gold standard algorithm.

Since the Bangle.js smartwatch and the IMU-based accelerometer operate at different sampling frequencies (12.5 Hz for the Bangle.js and 100 Hz for the IMU-based device) and were started at different times, timestamp alignment was necessary. This alignment synchronizes the data from both devices, allowing for a direct comparison of step counts from each algorithm against the gold standard.

To provide a comprehensive assessment of the step-counting algorithms, a detailed approach was implemented, aimed at analyzing how each method performs across a variety of real-world activity scenarios. This involved carefully segmenting the data into discrete, manageable intervals and calculating specific error metrics for each algorithm. By structuring the analysis in this way, the results would reflect not only the overall accuracy of each algorithm but also reveal its adaptability and consistency under different conditions.

The data regarding both the number of steps calculated by the various algorithms and regarding the reference, were therefore separated into 30-second frames as a means of gaining a fine-grained, accurate overview of the behavior of each step-counting algorithm. This may, in turn, allow for insight into how each algorithm performs over short time intervals representative of real-world changes in activities and variations in movements. By analyzing these shorter segments, it was possible to get a reliable estimate of the actual steps taken during each period, minimizing the risk of cumulative errors that can occur when looking only at a final total. Segmenting in this manner also made it easier to observe any

variability in algorithm performance and how well each method adapted to dynamic situations, such as transitioning from rest to high-intensity activity. After segmentation, each window from the algorithm-generated file was compared with the corresponding window from the reference file using performance metrics, which are described in the following sections.

### 3.9.1. Error Calculation and Metrics

For each 30-second segment, different error metrics were calculated as a means of quantifying exactly how accurately each of the algorithms was counting steps:

- **Mean Absolute Error (MAE).** The MAE is the average difference between the algorithm's step count and the reference step count from the IMU. It provides a simple measure of overall accuracy. The mean MAE and its standard deviation allowed both the accuracy and consistency of each algorithm to be quantified.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i|$$

where  $n$  is the total number of observations (e.g., 30-second windows),  $y_i$  represents the actual (reference) step count in the  $i$ -th observation, and  $\bar{y}_i$  is the algorithm's estimated step count for the  $i$ -th observation.

- **Median error and IQR (Interquartile Range).** The median error is robust, being less affected by extreme values, while the IQR provides the range of the middle 50% of errors as an indication of variability.

$$Median\ Error = median(y_i - \hat{y}_i)$$

$$IQR = Q_3 - Q_1$$

$Q_1$  is the first quartile (25<sup>th</sup> percentile) of the error distribution, representing the value below which 25% of the data fall;  $Q_3$  is the third quartile (75<sup>th</sup> percentile), representing the value below which 75% of the data fall.

- **Limits of Agreement LOA:** computed as the mean error  $\pm$  1.96 standard deviations of the error, yielding a range within which most of the differences between the algorithm and the reference step counts fall. This gives insight into each algorithm's potential range of error.

$$LOA = mean\ error \pm 1.96 * standard\_deviation$$

The code used to calculate these metrics iterated over each of the segmented files, comparing the output of each algorithm with the IMU reference step counts. These errors were then summarized in a table that could convey a clear picture of how each algorithm performed across different activities.

### 3.9.2. Bland-Altman

To further verify the agreement between the step count of each algorithm and the reference of the IMU, a Bland-Altman analysis was performed. A Bland-Altman plot provides a visualization of the agreement of each algorithm with reference data for every moving window of 30 seconds; it shows possible biases and variabilities in error.

The Bland-Altman method is a statistical approach used to assess the level of agreement between two different measurement methods, often employed when comparing a new method to a standard or reference method. The process involves calculating the difference between the measurements obtained from each method for each sample, which reflects how much the new method deviates from the reference.

The average of these differences, known as the mean difference, provides an indication of any systematic bias between the two methods. If the mean difference is close to zero, it suggests there's no consistent over- or underestimation by one method relative to the other. However, a significant mean difference would indicate systematic bias.

To further assess agreement, the Bland-Altman method defines limits of agreement, calculated with the formula reported in the previous section. These limits outline a range within which 95% of the differences are expected to lie, assuming the differences follow a normal distribution. This range helps to understand the degree of variability in the differences and whether they are acceptable for practical purposes.

The Bland-Altman method is commonly visualized with a plot, where the differences between measurements are on the y-axis, and the average of each measurement pair is on the x-axis. A horizontal line at the mean difference represents the bias, while two additional lines indicate the upper and lower limits of agreement. This plot is useful for detecting patterns, such as whether the differences are consistent across the range of values or vary depending on the size of the measurement.

If the limits of agreement are narrow and there is no systematic pattern, it can be concluded that the two methods have good agreement.

### 3.9.3. Box plot

A boxplot is a graphical summary that represents key statistics in a dataset. It highlights the central 50% of values, or interquartile range (IQR), as well as any outliers, providing a clear view of the data's distribution, spread, and symmetry. This is particularly useful in identifying variability and comparing multiple groups. Key components of a boxplot are:

1. **Box (IQR).** The box in a boxplot spans from the first quartile (Q1) to the third quartile (Q3), representing the middle 50% of the data, known as the IQR. This value helps capture the range of values around the median, filtering out the influence of extreme values.
2. **Median (Q2).** The line inside the box represents the median (Q2), the 50<sup>th</sup> percentile, which divides the dataset in half. The position of the median within the box indicates skewness: a centered median line indicates a roughly symmetric distribution; if the median line is closer to Q1, the data is right-skewed (positive skew); if the median is closer to Q3, the data is left-skewed (negative skew)
3. **Whiskers.** The whiskers extend from the edges of the box and capture the data within a defined range. The lower whisker extends from Q1 to the smallest data point within 1.5 times the IQR below Q1, calculated as:

$$\text{Lower Bound} = Q1 - 1.5 \times IQR$$

The upper whisker extends from Q3 to the largest data point within 1.5 times the IQR above Q3:

$$\text{Upper Bound} = Q3 + 1.5 \times IQR$$

4. **Outliers,** Data points that fall beyond the lower and upper bounds are called outliers and are marked as individual points beyond the whiskers. Outliers are defined by lower outliers and upper outliers, which are computed as in the following formulas, respectively:

$$\text{Data points } x < Q1 - 1.5 \times IQR$$

$$\text{Data points } x > Q3 + 1.5 \times IQR$$

The height of the box, so the IQR, and the length of the whiskers indicate the spread of the data. A larger IQR or longer whiskers indicate more variability.

The position of the median within the box and the relative lengths of the whiskers reveals symmetry or skewness.

## 4 Results and Discussion

This section provides an in-depth analysis across various metrics, including accuracy, computational efficiency, and adaptability to different movement types and environments.

### 4.1. Bland-Altman analysis

The main purpose of the Bland-Altman plot is to display the individual differences in the step count from each algorithm compared to the reference IMU step count over different activities. It helps to assess any systematic biases, detects a trend of the magnitude of the errors, and intuitively provides the performance regarding each algorithm's reliability for a wide range of different activity types.

#### 4.1.1. Interpretation of Bland-Altman Plot Components

Each of the following Bland-Altman plots includes:

- **Mean difference (bias).** The black dashed line represents the mean difference between the algorithm and the reference IMU count. A positive mean difference reflects overestimation by the algorithm whereas a negative reflects underestimation.
- **Limits of agreement (LoA).** The red and green dotted lines represent the limits of agreement, computed as the mean difference  $\pm 1.96$  times the standard deviation (SD) of the differences. The limits of agreement provide the range within which 95% of the differences fall. A narrower LoA denotes higher agreement, while a wider LoA is indicative of greater variability in the error.
- **Distribution of data and variability of activities.** Each point in the following plot represents the difference in step count in a 30 second window, color-coded by activity type, *i.e.*, 'light task', 'complete rest', 'indoor activity', 'climbing stairs', and 'outdoor walk'. Data points concentrated around the mean difference represent stable performance, while wide dispersion suggests higher variability. Color-coding is useful since it lets us immediately see whether some activities come to introduce systematic biases or errors into the algorithm's counting.

Table 4.1 summarizes the results obtained for each algorithm in terms of mean difference, standard deviation, upper LoA, and lower LoA. The total table with MAE, median error (MED), LoA values for each activity, subject, algorithm are in appendix A1. Moreover, Appendix A2 presents the Mean

Percentage Error values calculated over 30-second time windows for each activity and algorithm.

Algorithm	Mean Difference	Standard Deviation	Upper LoA	Lower LoA
Dummy	2,90	15,28	32,85	-27,06
BangleSimple	-10,34	16,36	21,72	-42,40
Espruino	-11,29	18,02	24,04	-46,61
Oxford	3,12	15,57	33,64	-27,41
Autocorrelation	-11,55	17,93	23,59	-46,69
FFT	-4,80	14,12	22,87	-32,47

Table 4.1 - Mean difference, standard deviation, upper LoA, lower LoA of each algorithm



Figure 4.1 - Bland-Altman plot for Dummy algorithm

As shown in Figure 4.1 the Dummy algorithm has a positive bias 2.90 steps, suggesting it tends to slightly overestimate the step count compared to the reference. The upper and lower LoAs are at +32.85 and -27.06, as shown in Table 4.1, respectively, indicating a broad range of variability in its estimates. The wide LoA indicates high inconsistency in the Dummy algorithm's step counting accuracy. The relatively large range suggests that the Dummy algorithm lacks adaptability, particularly in dynamic activity types, such as 'indoor activity' and

'outdoor walk'. Its fixed-rate assumption results in errors that vary significantly across different activity levels, as it does not account for contextual changes in movement.

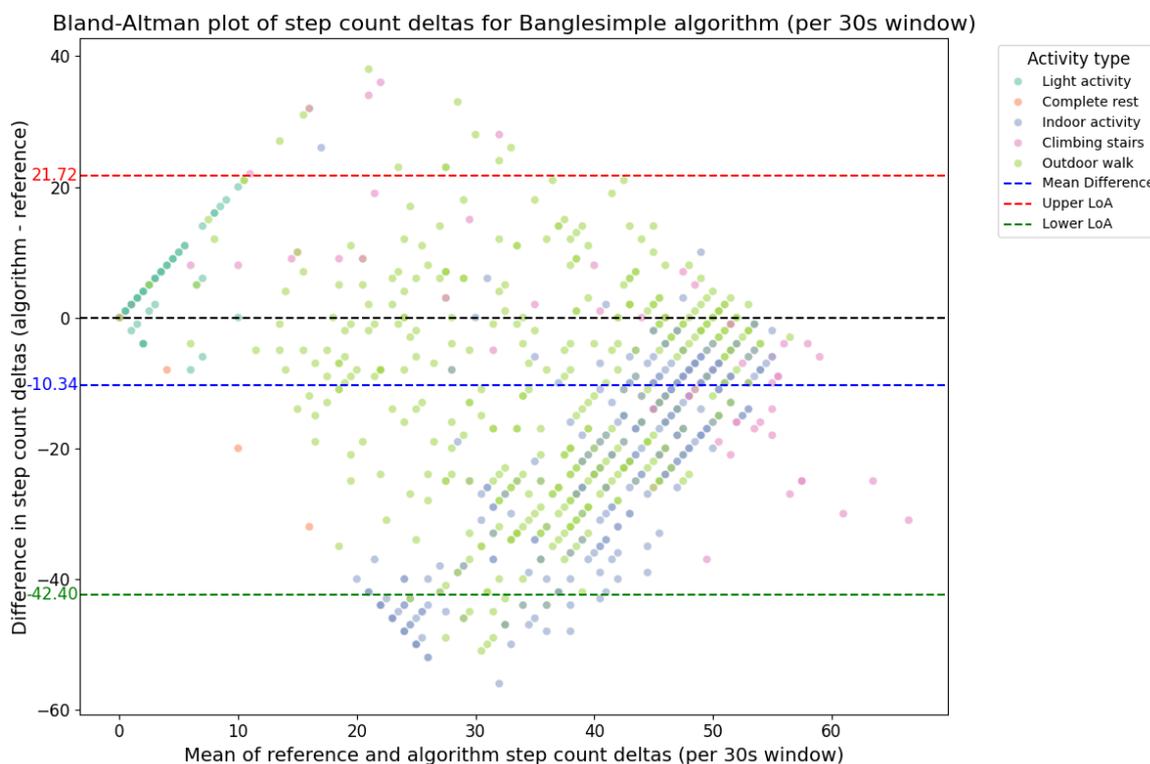


Figure 4.2 - Bland-Altman plot for BangleSimple algorithm

The mean difference for Bangle-Simple, shown in Figure 4.2, is  $+10.34$ , indicating a tendency toward overestimation relative to the reference. The upper and lower LoAs are  $21.72$  and  $-42.40$ , respectively, yielding an asymmetrical range around the bias line. The wider lower LoA suggests greater potential for underestimating steps in certain conditions, particularly where step patterns are less regular, such as during 'climbing stairs' and 'indoor activity'. The occasional underestimation points to limitations in handling non-periodic or abrupt movement patterns, likely due to a simpler approach in detecting step changes without sophisticated filtering.



Figure 4.3 - Bland-Altman plot for Espruino algorithm

The mean difference for Espruino, shown in Figure 4.3, is approximately -11.29, indicates that the Espruino algorithm generally underestimates step counts relative to the reference. The LoAs are wider, at +24.04 and -46.61, reflecting a substantial spread in step count accuracy. This wide range suggests variability in performance across different activity types.

The negative bias implies that Espruino might miss steps in certain scenarios, particularly in activities with subtle or complex movements (*e.g.*, 'indoor activity' and 'climbing stairs'), where detection may be more challenging. The wide LoA further suggests that Espruino's accuracy is inconsistent across varied activities, potentially making it less reliable in applications where precise step counting is critical across diverse conditions.

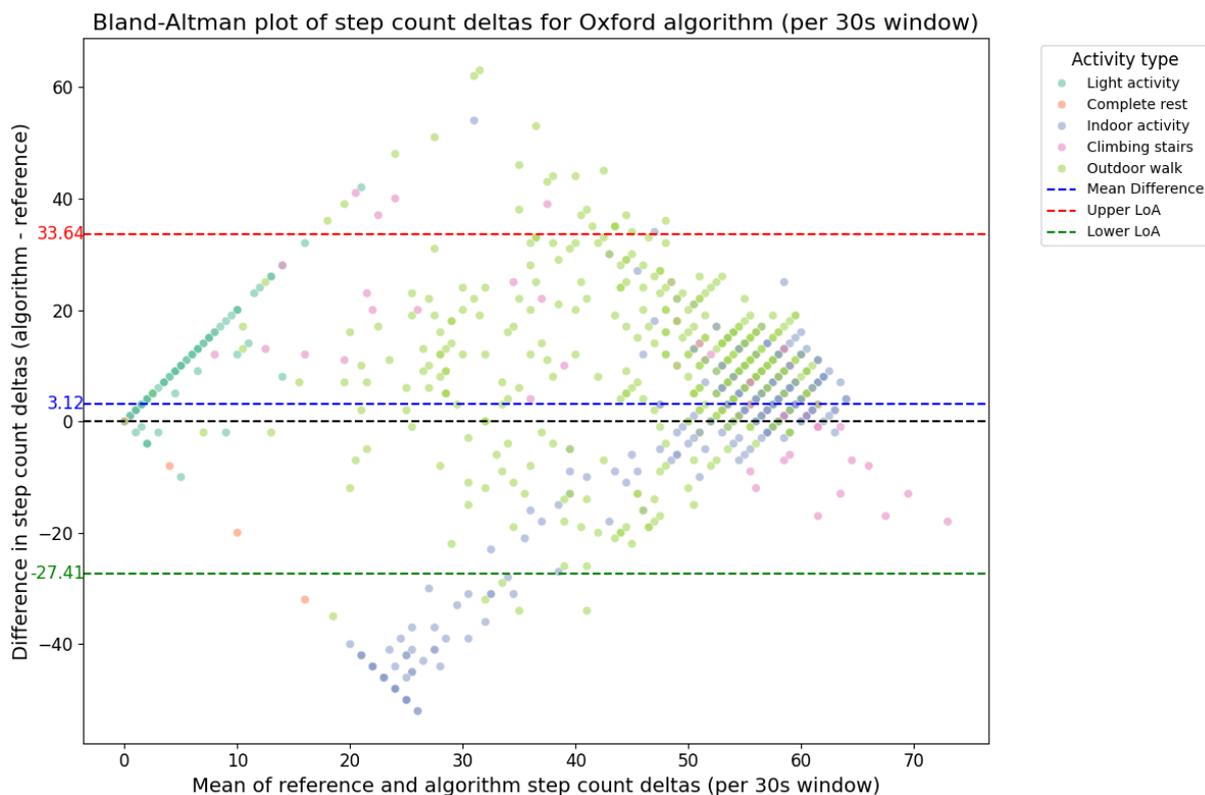


Figure 4.4 - Bland-Altman plot for Oxford algorithm

The Oxford algorithm, shown in Figure 4.4, Oxford shows a lower bias at around 3.12, indicating a relatively balanced performance compared to the reference. The LoAs are slightly narrower than those of Espruino, at +33.64 and -27.41, suggesting that Oxford has moderate variability, with a performance range like the Dummy algorithm but with less overestimation.

The Oxford algorithm's near-neutral bias suggests a balanced approach to step counting that can perform well across different activities. However, the relatively wide LoA reveals some variability in accuracy, especially in activities with irregular movement patterns like 'indoor activity' and 'outdoor walk'. The balance in bias may result from its state-machine logic, which can adapt reasonably well across different types of movements but still lacks fine-tuning for extreme conditions

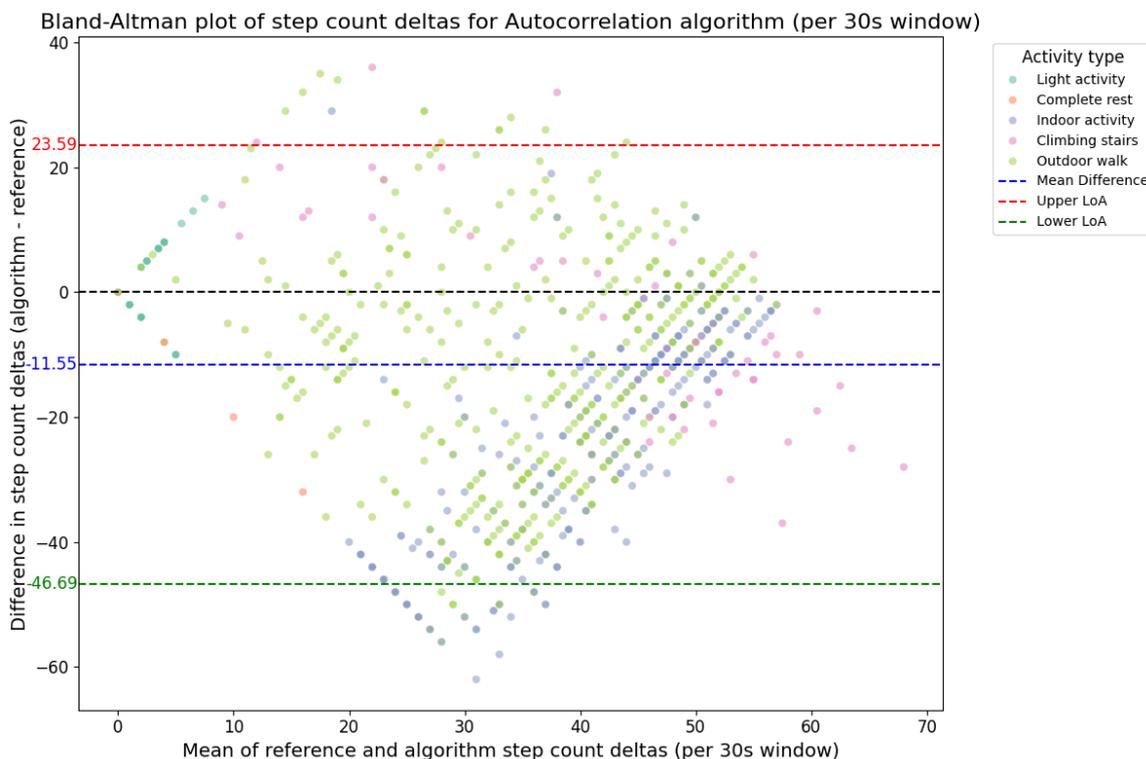


Figure 4.5 - Bland-Altman plot for Autocorrelation algorithm

As shown in Figure 4.5, the mean difference in the case of the Autocorrelation algorithm is -11.55, indicating a negative bias. This means that the Autocorrelation algorithm underestimates the number of steps on average compared to the reference. This algorithm has one of the broadest ranges for LoAs, with an upper LoA at +23.59 and a lower LoA at -46.69, reflecting substantial variability in accuracy. The high variability and negative bias imply that while the Autocorrelation algorithm may function adequately in specific scenarios, it may struggle to provide accurate step counts consistently across diverse activity types. The tendency to underestimate steps suggests it may miss steps in low-intensity or irregular movement patterns.

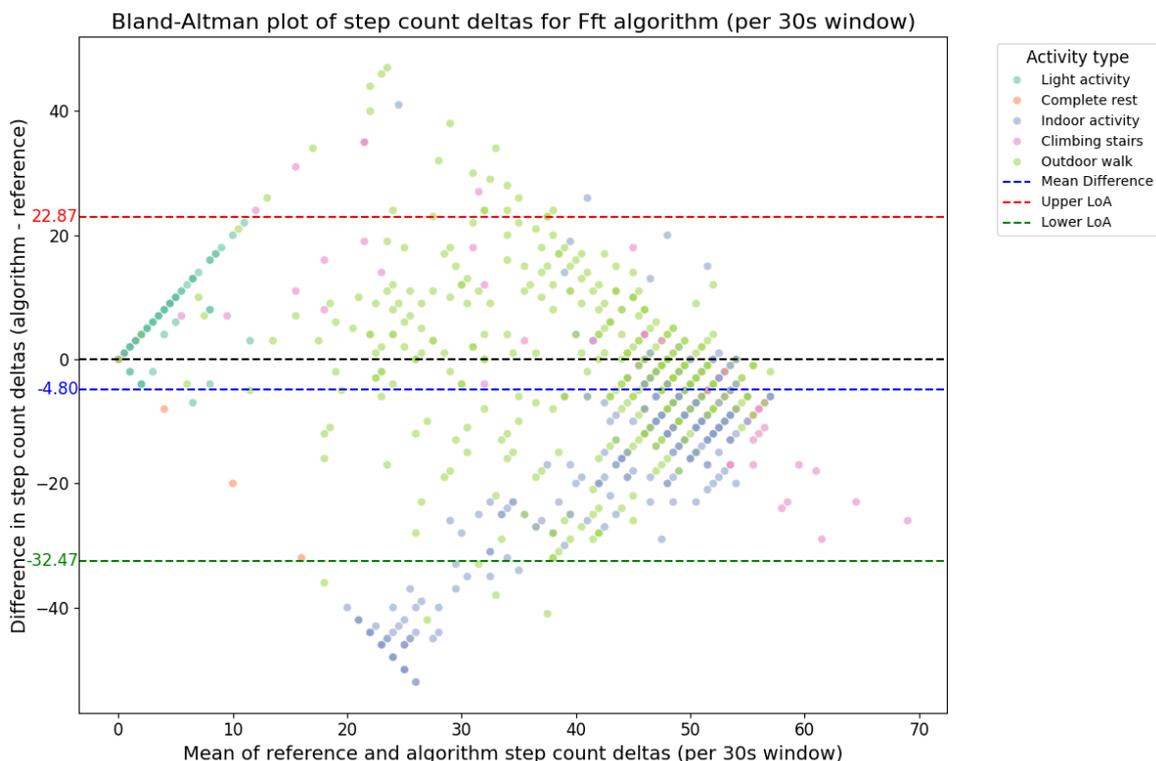


Figure 4.6 - Bland-Altman plot for FFT algorithm

The FFT-based algorithm, represented in Figure 4.6, has a slight negative bias around  $-4.80$ , indicating a minor tendency to underestimate steps compared to the reference. The LoAs are relatively tight, with the upper LoA at  $+22.87$  and the lower LoA at  $-32.47$ . This relatively narrow range indicates lower variability than most other algorithms.

FFT's narrow LoAs and slight negative bias indicate that it offers consistent performance across activities, with fewer extreme deviations in accuracy. This is likely due to its frequency-based approach, which allows it to filter noise and recognize periodic movement patterns effectively. The slight underestimation could stem from its reliance on detecting frequency components, which may lead to minor missed steps in non-periodic or abrupt movements. Overall, the FFT-based algorithm's consistency suggests it may be the most reliable option for varied conditions, especially in activities with predictable movement patterns like 'outdoor walk'.

## 4.2. Computational cost of the algorithms

In addition to evaluating accuracy, a critical aspect of implementing step-counting algorithms in wearable devices is their computational cost. Devices like smartwatches and fitness trackers operate under constrained processing power and energy consumption. Therefore, it is essential that step-counting algorithms are not only accurate but also efficient in terms of CPU cycles, to reduce battery consumption and enable real-time operation.

The computational cost of each step-counting algorithm was measured in CPU cycles to understand the processing load each algorithm imposes. The results show significant variation in computational requirements among the algorithms.

To evaluate the computational cost of each algorithm, we calculated the sum of CPU cycles required to process all input files. The resulting values are shown in Table 4.2:

Algorithm	Sum CPU Cycles
Dummy	633497
BangleSimple	612986
Espruino	512598
Oxford	6480319
Autocorrelation	648345
FFT	10169942

*Table 4.2 - CPU cycles for each algorithm*

### 4.3. Box plot analysis

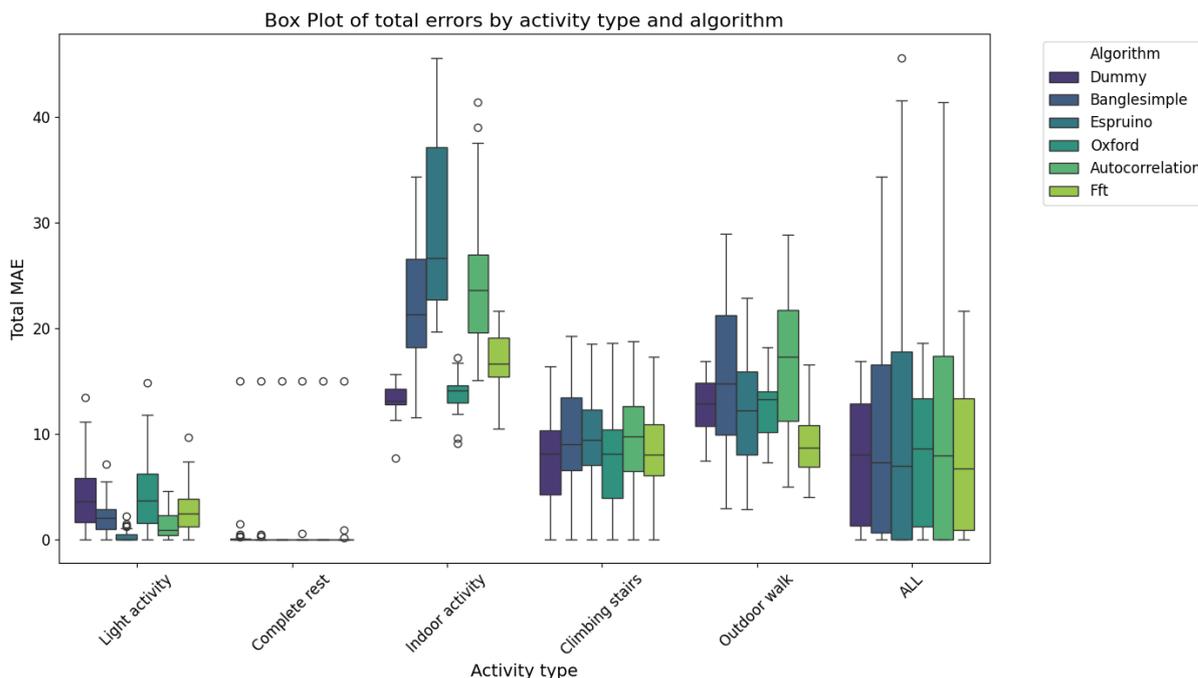


Figure 4.7 - Box plot of each algorithm and of each activity

This box plot, shown in Figure 4.7, illustrates the performance of estimation algorithms in terms of total MAE across various activity types: 'light activity', 'complete rest', 'indoor activity', 'climbing stairs', 'outdoor walk', and a cumulative aggregate ('all'). The box plot structure allows for an in-depth analysis of the error distribution for each algorithm, enabling us to assess not only the median and variability of the error but also the presence of outliers, which may indicate problematic or extreme cases.

The results vary between the different activities as follows:

- **'Light activity'**. Light activity is characterized by the absence of actual steps, with only arm movements being detected as steps. The algorithms exhibit relatively short boxes with centered medians, indicating a symmetric error distribution and low variability (*i.e.*, small IQR). The consistency in performance across algorithms and the absence of significant outliers suggest that low-intensity activity is a stable, manageable scenario for all approaches. A reduced IQR implies that the model generalizes well in low-activity conditions, where signal variations are minimal, keeping the MAE consistently low. For this activity, the worst-performing algorithms are the Dummy and Oxford algorithms.
- **'Complete rest'**. In this category, the boxes are highly compressed, with some isolated outliers representing rare error spikes. Low median values and consistent distribution indicate that this activity is straightforward and lacks significant variations. The ability of the algorithms to maintain

low MAE in resting conditions suggests a high level of robustness when handling static or minimally varying data. The absence of large variations in the input data makes this activity easy to manage for all algorithms, confirming their ability to operate with stationary data.

- **‘Indoor activity’**. The indoor activity introduces significant variability in errors, as evidenced by the wider IQR and numerous outliers. For the treadmill activity, the autocorrelation, Bangle-Simple, and Espruino algorithms exhibit higher average MAE values, along with greater variability compared to the other algorithms. In contrast, the Oxford and dummy algorithms show the best performance in terms of lower MAE and stability. The FFT algorithm falls in between, with moderate MAE and variability levels
- **‘Climbing stairs’**. In this category, the error distribution is more uniform and contained. The median is centered within the boxes of almost all algorithms, and the IQR is not excessively wide, indicating that the algorithms maintain a consistent error level. Climbing stairs implies constant vertical and repetitive variation in the signal.
- **‘Outdoor walk’**. This activity shows considerable variability in errors. The boxes are elongated, and there are outliers, especially for algorithms like Bangle-Simple and Autocorrelation, which struggle to maintain consistency in outdoor environments where conditions can be dynamic and unpredictable. In this activity, it emerges that the FFT algorithm is the one with the lowest average MAE values among the various tested algorithms, as well as having lower variability, meaning it exhibits less dispersion around the central values.
- **‘All’**. When considering the error across all activities together, both the IQR and the presence of outliers increase significantly. This indicates greater overall variability in algorithm performance, with Bangle-Simple, Espruino and Autocorrelation algorithms exhibiting a more compact error distribution compared to others. FFT and Dummy, on the other hand, show better generalizability, managing to keep stable and relatively low errors across a wide range of activities, likely due to more robust and flexible model architecture. The overall aggregate highlights the difficulty some algorithms face in maintaining consistent performance across diverse activities.

## 4.4. Discussion

Through examining both results and practical implications, the discussion aims to guide the choice of step-counting algorithms best suited for wearable applications, especially in cases where resources are constrained, or specific user needs are prioritized.

#### 4.4.1. Bland-Altman discussion

In comparing the consistency and variability across algorithms, FFT, Dummy and Oxford stand out for their relatively narrow LoAs, indicating a higher level of accuracy and stability in step counts across different activities. This consistency suggests that the algorithms maintain a reliable performance, even as activity types change, with FFT showing particularly robust adaptability. In contrast, Autocorrelation and Espruino exhibit wider LoAs, which reflects greater variability and suggests less dependable step count accuracy. The higher variability in these algorithms implies that their performance may fluctuate more widely, making them less reliable in diverse activity settings.

Regarding the bias, the Dummy and Oxford algorithms show a mild positive tendency, meaning they tend to overestimate step counts slightly compared to the reference. Autocorrelation, Espruino and Bangle-simple algorithms, instead, show a high negative bias, meaning they tend to underestimate steps. This underestimation may result in missed steps, particularly in activities where subtle movements or complex patterns are present. On the other hand, FFT algorithm shows a slight negative bias, indicating a balanced approach that is less likely to result in significant over or underestimation of steps. Overall, the best performing algorithm in this analysis was the FFT algorithm, followed by the Oxford and Dummy algorithms. A more detailed analysis of each algorithm's performance across different activities is provided in Section 4.4.3

#### 4.4.2. Computational costs discussion

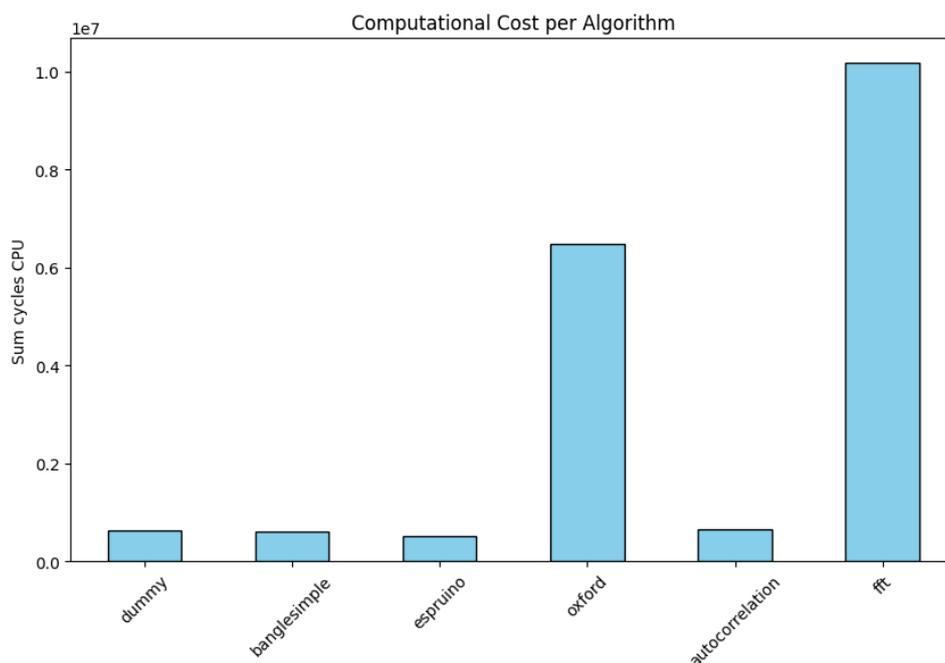


Figure 4.8 - Computational cost of each algorithm

The analysis of computational costs across various step-counting algorithms highlights a fundamental trade-off between efficiency and accuracy, a crucial consideration for wearable devices where processing power and battery life are limited. The results, shown in Figure 4.8, show that algorithms already integrated into the Bangle.js, Espruino and BangleSimple, maintain a high level of efficiency with relatively low CPU cycle consumption, establishing them as benchmarks for resource-conscious applications.

In contrast, more complex algorithms, such as Oxford and FFT, offer additional accuracy but require considerably higher computational resources. Oxford emerges as a middle-ground option, consuming more cycles than Espruino and BangleSimple but offering enhanced step-counting accuracy, which may justify its use in applications where precision is paramount. The elevated computational cost of Oxford and FFT algorithms makes them less practical for continuous use in most wearables, although they show potential in specialized contexts where accuracy can be prioritized over efficiency.

#### 4.4.3. Box plot discussion

In general, the FFT, dummy, and Oxford algorithms show nearly identical performance when evaluating their overall effectiveness across the different activities. This suggests that these algorithms are relatively consistent in handling various types of movement, demonstrating a level of adaptability that makes them suitable for general use in step counting. However, when considering their performance in specific activities tested through the protocol, noticeable differences emerge.

For less dynamic activities such as rest and light activity, the performance of the algorithms tends to be quite similar. In these contexts, the simpler motion patterns result in relatively low variability in the error values, and all the algorithms can maintain sufficient accuracy. However, a closer inspection reveals that the Dummy and Oxford algorithms show slightly worse results compared to the FFT.

In contrast, when the participant is walking on a treadmill ('indoor activity'), the Oxford and Dummy algorithms demonstrate superior performance. The higher accuracy in these specific contexts may be attributed to the fact that these algorithms are better suited to environments with more controlled, predictable movement patterns. Indoor activities tend to involve more repetitive motions, and these algorithms may be more effective at identifying these regularities, resulting in fewer errors compared to more complex outdoor activities.

On the other hand, the FFT algorithm proves to be the most effective for dynamic outdoor activities ('outdoor walk'). These types of activities, which often involve rapid and unpredictable movements, are more challenging for many step-counting algorithms.

In conclusion, while the overall performance of the FFT, Dummy, and Oxford algorithms is comparable across a range of activities, the true differences lie in their ability to adapt to specific conditions. The Oxford and Dummy algorithms excel in indoor settings, where the movement is more predictable, while the FFT algorithm outperforms the others in dynamic, outdoor conditions. These findings suggest that choosing the most appropriate algorithm depends heavily on the nature of the activity, and that further optimization of these algorithms could enhance their performance in more specialized settings.

## 5 Conclusion and Future developments

### 5.1. Conclusion

The analysis of the Bland-Altman plots, specifically focusing on the limits of agreement and mean differences, reveals that the FFT algorithm outperforms the other algorithms in terms of accuracy, followed by the Oxford and Dummy algorithms. The FFT algorithm demonstrates the tightest limits of agreement, indicating superior consistency and reliability in step counting across various activities. This consistency suggests that the FFT algorithm is highly adaptable and maintains its accuracy even as the activity type changes.

When examining the performance of these algorithms across specific activities, it becomes clear that the real difference between them lies in how they handle different types of movement. The FFT, Espruino, Autocorrelation, and Bangle-Simple algorithms perform well in low-intensity, less dynamic activities, such as 'complete rest' and 'light activity'. In these types of activities, the motion is less variable, and the algorithms can produce relatively accurate results with little fluctuation in error values. However, when focusing on specific performance nuances, the Dummy and Oxford algorithms perform slightly worse in these contexts, potentially due to their inability to handle small, subtle movements effectively. The slight overestimations seen in these algorithms may contribute to a less precise representation of step counts in low-activity scenarios.

On the other hand, in more controlled environments such as indoor treadmill activities, the Oxford and Dummy algorithms emerge as the best performers, showing superior accuracy compared to other algorithms. This can be attributed to the nature of indoor activities, which often involve repetitive and predictable movement patterns. The FFT algorithm, while still performing reasonably well indoors, shows less adaptability to the repetitive nature of treadmill walking ('indoor activity') when compared to Oxford and Dummy.

In the case of 'outdoor walk', however, the FFT algorithm outperforms the others. Outdoor activities typically involve more unpredictable and rapid movements,

presenting a greater challenge to step counting algorithms. The FFT algorithm, with its frequency-based approach that decomposes signals into frequency components, is better equipped to handle the noise and variability inherent in these types of dynamic environments. This makes FFT particularly well-suited to real-life outdoor walking, where the movement is more irregular and varied.

Given these results, the FFT algorithm stands out as the best overall choice for step counting across a broad range of scenarios. The choice of FFT is justified by its balanced performance across different activity types. Its performance in low-intensity activities, such as rest and light activity, is critical, as most of the population tends to lead a sedentary lifestyle. In fact, statistics show that approximately 60-85% of adults globally do not meet the recommended levels of physical activity (WHO, 2020), with a significant portion engaging in sedentary behaviors for prolonged periods. In these scenarios, it is more crucial for an algorithm to minimize false positives, incorrectly detecting steps, rather than false negatives, where steps might be missed.

Moreover, because many individuals lead sedentary lives, the algorithm must be robust enough to handle noise, especially in low-intensity activities. In such contexts, the signal can be prone to small, irrelevant fluctuations, and a less robust algorithm might mistakenly interpret these as movement. Therefore, it is crucial that the algorithm remains accurate even when the activity is minimal, ensuring that noise does not compromise the signal. This is why the FFT algorithm's ability to effectively manage noise in low-intensity activities is particularly valuable, offering a more reliable and consistent performance across various activity levels.

While the FFT algorithm performs exceptionally well in many contexts, it does come with the drawback of high computational costs. This is a significant limitation for real-time applications, especially in wearable devices such as smartwatches where battery life and processing power are constrained. Therefore, a key area for future research will be optimizing the FFT algorithm to reduce its computational load. Possible improvements could include refining the frequency-domain processing or exploring alternative methods for real-time processing that can maintain the algorithm's accuracy while making it more efficient.

In conclusion, while the FFT algorithm stands out for its accuracy, stability, and adaptability across various activities, the high computational cost remains a challenge that needs to be addressed. Nonetheless, its overall performance, particularly in dynamic, outdoor activities, justifies its selection as the best algorithm among those tested. With further optimization, FFT could become a viable and reliable choice for continuous, real-time activity tracking in wearable devices, making it a strong candidate for future developments in health and fitness monitoring technologies.

## 5.2. Future Development

Given the depth of analysis undertaken herein, a number of potential future directions now appear that could help further extend and deepen the insights developed in this study. First, using the comparative analysis done with step-counting algorithms, a new, optimized algorithm is to be developed specifically with integration into Bangle.js in mind. The best algorithm would balance computational efficiency with accuracy, taking the best elements of different current methods. Given this very encouraging performance of the FFT approach, it might form one very firm basis for further refinement into a robust step-counting method that can handle a wide range of real-world activities while keeping processing demands low.

Another possible future direction could be the expansion of the comparison scope to include heart rate measurement algorithms in addition to step-counting, since heart rate is another important vital metric for physical activity and health monitoring. This would therefore include the methodological comparison of heart rate algorithms in an effort to determine which algorithms perform best for wearable applications, much like the step-counting methods compared in this study. This would take such imbedded heart rate tracking in the Bangle.js to more reliable heights, hence a comprehensive health monitoring device.

Future work could also be directed to the collection of larger datasets, mainly recording for a longer duration and the number of activities. Activities involving running introduce other movement dynamics, which may provide new insights into algorithm performance during higher intensities. The extension of the dataset would allow for a more in-depth study of the robustness of the algorithms and therefore would enable the tuning and validation of algorithms for real-world applications.

Future designs may incorporate adaptive algorithms that learn with time from the peculiar patterns of single individuals. This will allow much finer-tuned algorithm accuracy, especially for users having non-standard walking patterns due to factors like age or mobility assistance devices. Adaptive algorithms can also make the Bangle.js much more resistant to awkward moves and interruptions, thereby raising the level of accuracy during unconstrained, day-to-day usage.

The wearable could adopt a platform of cloud computing for periodic updates or improvements in its algorithm library. By doing so, this device provides periodic optimization of its algorithms, either through new data or through user feedback, to maintain the accuracy of step-count and heart rate when activity patterns change or new movement types are encountered.

These are some of the lines of development that might be explored in the future and could make the Bangle.js a more useful, accurate, and adaptable health-monitoring device, thus positioning it effectively as a real-time solution for users with diversified needs and levels of activity.

## Bibliography

- [1] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, 2008.
- [2] [Online]. Available: <https://www.espruino.com/Bangle.js2>.
- [3] A. Shcherbina, «Accuracy in Wrist-Worn, Sensor-Based Measurements of Heart Rate and Energy Expenditure in a Diverse Cohort,» *Journal of Personalized Medicine*, 2017.
- [4] S. K. N. Niël Henk Conradie, «Digital sovereignty and smart wearables: Three moral calculi for the distribution of legitimate control over the digital,» *Journal of Responsible Technology*, vol. Volume, n. 12, 2022.
- [5] E. S. Raymond, *The Cathedral and the Bazaar*, O'Reilly Media, 1999.
- [6] «<https://www.who.int/news-room/fact-sheets/detail/physical-activity>,» [Online].
- [7] T. R. Y. J. L. S. N. J. Montes J, «Step Count Reliability and Validity of Five Wearable Technology Devices While Walking and Jogging in both a Free Motion Setting and on a Treadmill.,» *Int J Exerc Sci.*, 2020.
- [8] F. Y. G. J. B. M. J. G. A. P. M. D. S. A. M. L. e. T. W. B. Modave, «Mobile Device Accuracy for Step Counting Across Age Groups.,» niversity of Florida, Department of Health Outcomes and Policy, Gainesville, FL, United States, 2017.
- [9] H. G. S. P. H. S. Abu-Faraj ZO, «Human Gait and Clinical Movement Analysis,» in *Encyclopedia of Electrical and Electronics Engineering*, New York, USA, John Wiley & Sons,, 2015, pp. 1-34.
- [10] T. Marinelli, «Cammino e biomeccanica,» 2015. [Online]. Available: <https://www.cristianfrancavilla.it/analisi-del-cammino/>.
- [11] C. Francavilla, in "*Analisi del cammino.*
- [12] J. M. B. Jacquelin Perry, «Gait Analysis: Normal and Pathological Function,» *Sports Sci Med*, 2010.
- [13] C. Tudor-Locke, «Taking Steps toward Increased Physical Activity: Using Pedometers To Measure and Motivate.,» in *President's Council on Physical Fitness and Sports*, Washington, DC., 2002.
- [14] C. I. P. K. K. B. C.J. Minns Lowe, «Measuring step count: why it is important not to assume measures are reliable,» in *Physiotherapy*, 2021, pp. Volume 110, 1-4.

- [15] S. C. U. t. ADXL367, «Analog devices,» [Online]. Available: <https://www.analog.com/media/en/technical-documentation/app-notes/an-2554.pdf>.
- [16] N. E.-S. Maan Khedr, «A Smartphone Step Counter Using IMU and Magnetometer for Navigation and Health Monitoring Applications,» *Inertial Sensors for Positioning and Navigation*, vol. 17, 2017.
- [17] S. Ferrante, *slides*.
- [18] G. J. K. D. N. C. P. E. Z. N. E. M. Coravos A, «Digital Medicine: A Primer on Measurement,» *Digit Biomark*, 2019.
- [19] [Online]. Available: <https://www.transparencymarketresearch.com/digital-health-market.html>. .
- [20] T. C. W. S. T. E. W. W. N. C. Wattanapisit A, «Can mobile health apps replace GPs? A scoping review of comparisons between mobile apps and GP tasks,» *BMC Med Inform Decis Mak*, 2020.
- [21] M. A. B. R. L. W. P. R. F. K. Carroll JK, «Who Uses Mobile Phone Health Apps and Does Use Matter? A Secondary Data Analytics Approach,» *Med Internet Res.*, 2017.
- [22] C. M. O. G. Y. C. C.-L. K. C. T. a. S. A. S. Dario Salvi, «Mobistudy: mobile-based, platform-independent, multi-dimensional data collection for clinical studies. In Proceedings of the 11th International Conference on the Internet of Things,» in *Association for Computing Machinery*, New York, NY, USA, , 2021.
- [23] M. A. Tavakoli Golpaygani A, «Future of Wearable Health Devices: Smartwatches VS Smart Headphones,» *J Biomed Phys Eng*, 2021.
- [24] Z. L. X. L. M. H. S. a. M. P. S. Mohan Babu, «Wearable Devices: Implications for Precision Medicine and the Future of Health Care,» *Annual Review of Medicine*, 2024.
- [25] K. R. F. a. J. Torous, «The Opportunity and Obstacles for Smartwatches and Wearable Sensors,» *IEEE Pulse*, 2019.
- [26] M. H. S. Q. K. H. S. S. T. S. Masoumian Hosseini M, «Smartwatches in healthcare medicine: assistance and monitoring; a scoping review.,» *BMC Med Inform Decis Mak*, 2023.
- [27] L. D. C. A. C. E. V. d. B. Hans Stuyck, «Validity of the Empatica E4 wristband to estimate resting-state heart rate variability in a lab-based context,» *International Journal of Psychophysiology*, vol. 182, pp. 105-118, 2022.
- [28] M. V. T. M. P. H. R. A. K. A. C. & A. E. A. McConnell, «Mobile Health Advances in Physical Activity, Fitness, and Atrial Fibrillation: Moving Hearts.,» in *Circulation*.

- [29] T. W. M. M. A. A. Mishra, «Pre-symptomatic detection of COVID-19 from smartwatch data.,» in *Nature Biomedical Engineering*, 2020.
- [30] S. JS, H. JT, B. KA, H. N e C. JA., «Peroxiredoxin 1 plays a primary role in protecting pancreatic  $\beta$ -cells from hydrogen peroxide and peroxyxynitrite.,» *Am J Physiol Regul Integr Comp Physiol.*, 2020.
- [31] S.Lucchesini, «Hardware and firmware development of RespirHò: awearable device for continuos respiratory monitoring,» 2021.
- [32] F. Dotti, « Integration and Validation of a Wearable Body Sensor Network for Personal Monitoring during Static and Dynamic Activities,» 2023.
- [33] «Github Espruino,» [Online]. Available: <https://github.com/espruino/Espruino/blob/c1824f087b2618fb53711482c49323da4b6e4d67/libs/misc/stepcount.c#L152>.
- [34] «Autocorreletion,» [Online]. Available: [https://github.com/nerajbobra/embedded\\_pedometer](https://github.com/nerajbobra/embedded_pedometer).
- [35] «Oxford step counter,» [Online]. Available: <https://github.com/Oxford-step-counter/C-Step-Counter>.]



## List of Figures

Figure 2.1 - Gait cycle .....	13
Figure 2.2 - Mobistudy- Use case .....	21
Figure 3.1 - Lower part of the case .....	28
Figure 3.2 - Upper part of the case .....	29
Figure 3.3 - Bangle.js smartwatch.....	31
Figure 3.4 - IMU- based device inside the case .....	33
Figure 3.5 - Smartphone application associated with the IMU-based unit.....	34
Figure 3.6 - Polar streaming web application.....	34
Figure 3.7 -Bangle.js streaming web application.....	35
Figure 3.8 - Acquisition protocol .....	36
Figure 3.9 -Outdoor route.....	37
Figure 3.10 - Step counting from the IMU-based device .....	38
Figure 3.11- Light activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d) .....	40
Figure 3.12- Complete Rest: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d) .....	41
Figure 3.13- Indoor activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d) .....	42
Figure 3.14- Stairs Activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d) .....	43
Figure 3.15- Outdoor Activity: Acceleration Components (a), Acceleration Magnitude (b), Filtered Magnitude (c), Peak Detection (d) .....	44
Figure 3.16 - Dummy algorithm .....	46
Figure 3.17 - Bangle algorithm.....	47
Figure 3.18 - Espruino algorithm.....	48
Figure 3.19 - Autocorrelation algorithm .....	50
Figure 3.20-Magnitude, Filtered and Removed mean.....	50
Figure 3.21- Peak detection through derivative phase .....	51
Figure 3.22 - Fast Fourier Transform algorithm.....	53
Figure 3.23- Magnitude phase in FFT algorithm.....	54
Figure 3.24 -Magnitude of the frequency components between the frequency bins [2;7].....	55
Figure 3.25-Dominant frequency over time.....	55
Figure 3.26 - Oxford algorithm .....	56
Figure 3.27- Pre-processing and filtering stages in Oxford algorithm.....	57
Figure 4.1 - Bland- Altman plot for Dummy algorithm.....	66
Figure 4.2 - Bland- Altman plot for BangleSimple algorithm .....	67
Figure 4.3 - Bland- Altman plot for Espruino algorithm .....	68
Figure 4.4 - Bland- Altman plot for Oxford algorithm.....	69
Figure 4.5 - Bland- Altman plot for Autocorrelation algorithm.....	70

Figure 4.6 - Bland- Altman plot for FFT algorithm.....	71
Figure 4.7 - Box plot of each algorithm and of each activity .....	73
Figure 4.8 - Computational cost of each algorithm.....	75

## List of Tables

Table 2.1 – Tasks of a general practitioner that can be potentially performed by mHealth apps .....	20
Table 2.2 – Comparison of the major smartwatches .....	24
Table 3.1 - IMU-based device's components .....	28
Table 3.2 - Population characteristics .....	33
Table 3.3 - State machine .....	49
Table 3.4 - Defines used in Autocorrelation algorithm .....	53
Table 3.5 - Defines used in FFT algorithm .....	56
Table 3.6 - Defines used in the Oxford algorithm. ....	59
Table 4.1 - Mean difference, standard deviation, upper LoA, lower LoA of each algorithm .....	66
Table 4.2 - CPU cycles for each algorithm .....	72

## List of Abbreviations

WHO: World Health Organization  
IMU: Inertial Measurement Units  
GPS: Global Positioning System  
AI: Artificial Intelligence  
ML: Machine Learning  
CAGR: Computed Annual Growth Rate  
mHealth: Mobile Health  
EHRs: Electronic Health Records  
PPG: Photoplethysmography  
LED: Light-Emitting Diode  
Bpm: Beats per Minute  
SpO2: Blood Oxygen Saturation  
ECG: Electrocardiograms  
HR: Heart Rate  
PCB: Printed Circuit Board  
ANT: Adaptive Network Topology  
ABS: Acrylonitrile Butadiene Styrene  
WBANs: Wireless Body Area Networks  
LCD: Liquid Crystal Display  
BLE: Bluetooth Low Energy  
SD: Secure Digital  
FIR: Finite Impulse Response  
DC: Direct Current  
FFT: Fast Fourier Transform  
CPU: Central Processing Unit  
MAE: Mean Absolute Error  
IQR: Interquartile Range  
LoA: Limits of Agreement  
LA: Light Activity

## 6 Appendix A

This Appendix contains additional material referred to the Materials and Methods section.

### 6.1. A.1 Metrics Calculation

Subject	Activity type	Dummy_mae	Dummy_med	Dummy_loa	Banglesimple_mae	Banglesimple_med	Banglesimple_loa
S01	Light activity	3.91 ± 3.08	4.00 ± 1.25	-2.13, 9.95	2.09 ± 1.81	2.00 ± 1.00	-1.46, 5.65
S01	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S01	Indoor activity	15.67 ± 18.59	4.00 ± 3.75	-50.07, 46.07	21.72 ± 13.26	-18.00 ± 7.50	-52.77, 15.10
S01	Climbing stairs	4.50 ± 5.26	1.00 ± 2.75	-10.84, 15.84	7.75 ± 7.76	-2.00 ± 4.88	-25.27, 18.77
S01	Outdoor walk	13.78 ± 11.26	8.00 ± 8.00	-16.69, 38.61	9.93 ± 8.99	-2.00 ± 5.25	-27.72, 25.06
S02	Light activity	1.36 ± 2.46	0.00 ± 0.75	-3.46, 6.19	0.73 ± 1.42	0.00 ± 0.25	-2.06, 3.51
S02	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S02	Indoor activity	13.11 ± 15.80	2.00 ± 3.75	-44.04, 35.59	27.56 ± 16.50	-33.00 ± 14.50	-59.90, 4.79
S02	Climbing stairs	7.50 ± 9.00	1.00 ± 4.25	-19.66, 26.66	16.50 ± 17.94	-13.00 ± 14.75	-53.44, 23.44
S02	Outdoor walk	10.70 ± 9.27	7.00 ± 1.88	-14.97, 31.03	21.77 ± 11.93	-25.50 ± 10.50	-46.17, 3.43
S03	Light activity	5.73 ± 8.05	3.00 ± 2.00	-10.05, 21.51	2.73 ± 5.24	1.00 ± 1.25	-7.54, 12.99
S03	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S03	Indoor activity	12.89 ± 18.51	-2.00 ± 3.25	-49.81, 30.26	32.72 ± 12.24	-32.50 ± 5.50	-56.72, -8.73
S03	Climbing stairs	2.25 ± 3.86	0.00 ± 1.12	-6.47, 9.97	10.00 ± 8.98	-10.50 ± 6.50	-27.60, 7.60
S03	Outdoor walk	7.45 ± 6.88	2.50 ± 3.62	-14.39, 22.49	12.40 ± 7.58	-9.50 ± 6.62	-31.96, 14.96
S04	Light activity	6.82 ± 4.31	7.00 ± 3.00	-1.63, 15.26	3.36 ± 2.06	4.00 ± 0.75	-0.68, 7.41
S04	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S04	Indoor activity	11.33 ± 14.04	2.50 ± 2.75	-38.85, 30.85	24.72 ± 11.69	-22.00 ± 5.00	-47.63, -1.82
S04	Climbing stairs	5.80 ± 6.72	3.00 ± 6.00	-7.38, 18.98	3.60 ± 6.99	0.00 ± 0.00	-17.36, 11.76
S04	Outdoor walk	14.84 ± 9.14	12.00 ± 4.75	-4.16, 33.32	8.48 ± 7.33	-1.00 ± 6.00	-23.05, 21.18
S05	Light activity	13.45 ± 10.46	15.00 ± 6.75	-7.05, 33.96	7.09 ± 6.41	6.00 ± 3.75	-5.47, 19.65
S05	Complete rest	1.43 ± 1.81	0.00 ± 1.50	-2.12, 4.98	0.43 ± 0.79	0.00 ± 0.25	-1.11, 1.97
S05	Indoor activity	12.83 ± 17.94	1.00 ± 3.25	-48.27, 34.60	21.33 ± 14.15	-20.00 ± 6.38	-49.07, 6.40
S05	Climbing stairs	10.50 ± 19.69	1.00 ± 5.75	-28.09, 49.09	13.25 ± 14.55	-4.50 ± 8.88	-36.75, 44.25
S05	Outdoor walk	12.90 ± 11.90	6.00 ± 4.00	-19.03, 38.52	10.97 ± 9.37	-5.00 ± 3.50	-29.98, 26.75
S06	Light activity	1.80 ± 2.94	0.00 ± 1.38	-3.96, 7.56	1.00 ± 1.33	0.00 ± 1.00	-1.61, 3.61
S06	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S06	Indoor activity	14.72 ± 16.02	2.00 ± 13.25	-47.71, 34.49	20.17 ± 14.59	-15.50 ± 11.00	-48.75, 8.42
S06	Climbing stairs	9.00 ± 9.31	-1.00 ± 5.00	-27.35, 27.35	14.00 ± 12.54	-8.00 ± 11.00	-43.55, 25.55
S07	Light activity	1.64 ± 3.04	0.00 ± 0.50	-4.70, 7.61	0.73 ± 1.56	0.00 ± 0.00	-2.66, 3.75
S07	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S07	Indoor activity	15.11 ± 13.06	8.00 ± 4.12	-38.63, 40.85	11.50 ± 14.29	-6.50 ± 2.50	-39.84, 17.95

S07	Climbing stairs	14.50 ± 18.23	3.00 ± 8.75	-35.03, 53.03	14.25 ± 12.61	-4.00 ± 9.12	-40.87, 40.37
S07	Outdoor walk	15.17 ± 9.93	14.00 ± 3.62	-10.34, 37.34	7.17 ± 8.20	0.00 ± 4.25	-19.63, 22.90
S08	Light activity	8.36 ± 7.12	10.00 ± 5.75	-7.41, 23.05	5.27 ± 4.54	6.00 ± 3.00	-6.90, 15.26
S08	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S08	Indoor activity	13.67 ± 16.78	2.00 ± 7.25	-46.93, 36.04	18.61 ± 14.94	-15.50 ± 7.12	-48.87, 12.98
S08	Climbing stairs	5.00 ± 5.77	0.00 ± 2.50	-16.00, 16.00	9.25 ± 11.00	-2.00 ± 5.62	-32.85, 22.35
S08	Outdoor walk	9.00 ± 6.86	6.00 ± 4.00	-8.20, 23.91	17.86 ± 9.55	-19.50 ± 7.38	-37.52, 2.52
S09	Light activity	0.30 ± 0.95	0.00 ± 0.00	-1.56, 2.16	0.10 ± 0.32	0.00 ± 0.00	-0.52, 0.72
S09	Complete rest	0.17 ± 0.41	0.00 ± 0.00	-0.63, 0.97	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S09	Indoor activity	12.78 ± 14.49	2.00 ± 4.00	-40.80, 35.02	16.39 ± 14.00	-13.00 ± 7.38	-43.83, 11.05
S09	Climbing stairs	8.75 ± 7.18	3.00 ± 5.88	-23.49, 24.99	8.75 ± 11.56	-4.50 ± 6.62	-31.84, 15.34
S09	Outdoor walk	10.67 ± 7.46	8.00 ± 2.38	-9.15, 27.42	14.40 ± 8.31	-16.00 ± 7.25	-33.12, 6.72
S10	Light activity	6.00 ± 8.49	1.00 ± 6.25	-10.63, 22.63	3.20 ± 4.44	0.50 ± 3.00	-5.51, 11.91
S10	Complete rest	0.43 ± 1.13	0.00 ± 0.00	-1.79, 2.65	0.29 ± 0.76	0.00 ± 0.00	-1.20, 1.77
S10	Indoor activity	14.11 ± 17.34	3.00 ± 7.12	-49.06, 34.84	19.83 ± 15.42	-15.00 ± 7.00	-50.07, 10.40
S10	Climbing stairs	9.50 ± 11.47	6.00 ± 4.75	-12.99, 31.99	7.50 ± 10.15	-0.50 ± 4.00	-21.42, 28.42
S10	Outdoor walk	16.28 ± 13.17	6.00 ± 9.50	-28.59, 46.38	16.41 ± 11.75	-13.00 ± 12.00	-44.74, 25.15
S11	Light activity	4.10 ± 4.93	1.50 ± 3.88	-5.57, 13.77	2.00 ± 2.98	0.50 ± 1.38	-3.84, 7.84
S11	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S11	Indoor activity	14.44 ± 15.27	6.00 ± 3.50	-43.49, 39.71	17.72 ± 14.43	-14.00 ± 5.50	-46.01, 10.57
S11	Climbing stairs	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S11	Outdoor walk	14.32 ± 9.73	10.00 ± 7.25	-22.05, 38.50	22.05 ± 12.79	-22.50 ± 8.25	-47.11, 3.02
S12	Light activity	0.90 ± 2.02	0.00 ± 0.00	-3.07, 4.87	0.30 ± 0.67	0.00 ± 0.00	-1.02, 1.62
S12	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S12	Indoor activity	13.06 ± 17.37	-1.50 ± 8.50	-47.76, 24.76	22.00 ± 14.36	-19.50 ± 5.75	-50.14, 6.14
S12	Climbing stairs	15.25 ± 18.93	0.00 ± 7.62	-45.50, 54.00	19.25 ± 17.08	-5.00 ± 12.12	-56.12, 53.62
S13	Light activity	5.55 ± 4.18	4.00 ± 3.00	-4.56, 14.56	2.55 ± 2.58	2.00 ± 1.25	-4.43, 8.07
S13	Complete rest	0.29 ± 0.76	0.00 ± 0.00	-1.20, 1.77	0.29 ± 0.76	0.00 ± 0.00	-1.20, 1.77
S13	Indoor activity	12.78 ± 16.52	2.00 ± 2.75	-45.35, 34.46	25.56 ± 16.54	-27.50 ± 15.00	-57.97, 6.85
S13	Climbing stairs	16.40 ± 15.73	6.00 ± 11.50	-29.29, 51.69	15.40 ± 13.52	0.00 ± 9.00	-42.29, 43.49
S13	Outdoor walk	16.77 ± 17.72	4.00 ± 6.88	-34.60, 53.96	28.95 ± 15.25	-28.00 ± 15.00	-67.17, 17.62
S14	Light activity	3.50 ± 4.14	2.00 ± 3.12	-4.62, 11.62	1.70 ± 1.89	1.00 ± 1.50	-2.00, 5.40
S14	Complete rest	15.00 ± 14.00	-14.00 ± 8.50	-42.44, 12.44	15.00 ± 14.00	-14.00 ± 8.50	-42.44, 12.44
S14	Climbing stairs	9.50 ± 8.06	10.00 ± 5.25	-6.30, 25.30	6.25 ± 4.86	2.50 ± 4.38	-16.21, 17.71
S14	Outdoor walk	16.87 ± 12.46	7.00 ± 10.25	-31.24, 46.27	14.74 ± 11.87	-4.00 ± 10.25	-41.92, 24.44
S15	Light activity	3.60 ± 4.06	0.00 ± 3.12	-8.78, 11.98	2.50 ± 2.22	0.00 ± 2.00	-6.18, 7.18
S15	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S15	Indoor activity	13.44 ± 14.86	-7.00 ± 8.38	-44.28, 24.06	18.78 ± 13.64	-15.50 ± 7.75	-45.50, 7.95
S15	Climbing stairs	5.50 ± 5.26	1.00 ± 3.25	-15.62, 16.62	9.75 ± 10.72	-3.00 ± 6.38	-33.30, 21.80
S16	Light activity	11.09 ± 9.80	12.00 ± 6.75	-8.12, 30.30	5.45 ± 5.09	5.00 ± 3.25	-4.52, 15.42
S16	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S16	Indoor activity	12.78 ± 19.17	-1.50 ± 3.50	-50.85, 31.73	34.33 ± 13.15	-36.00 ± 6.00	-60.10, -8.57

S16	Climbing stairs	$3.50 \pm 5.07$	$-0.50 \pm 2.00$	-9.87, 13.87	$6.00 \pm 4.24$	$-3.00 \pm 4.50$	-17.06, 14.06
S16	Outdoor walk	$12.87 \pm 12.65$	$2.00 \pm 8.38$	-33.04, 37.70	$21.20 \pm 13.94$	$-21.50 \pm 13.00$	-52.13, 14.13
S17	Light activity	$0.80 \pm 1.69$	$0.00 \pm 0.00$	-4.11, 2.51	$1.20 \pm 2.70$	$0.00 \pm 0.00$	-6.49, 4.09
S17	Complete rest	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00
S17	Indoor activity	$7.67 \pm 10.79$	$1.00 \pm 3.00$	-28.93, 20.93	$30.22 \pm 12.36$	$-30.00 \pm 8.50$	-54.45, -6.00
S17	Climbing stairs	$10.25 \pm 12.84$	$6.00 \pm 4.38$	-14.92, 35.42	$10.25 \pm 6.95$	$-6.00 \pm 8.12$	-28.92, 23.42
S17	Outdoor walk	$11.33 \pm 5.84$	$10.00 \pm 2.25$	-3.36, 24.56	$20.37 \pm 10.37$	$-24.00 \pm 6.75$	-40.69, -0.04
S18	Light activity	$2.33 \pm 4.64$	$0.00 \pm 1.50$	-6.75, 11.42	$1.11 \pm 2.32$	$0.00 \pm 0.50$	-3.43, 5.65
S18	Complete rest	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00
S18	Indoor activity	$12.11 \pm 16.64$	$1.50 \pm 3.75$	-45.30, 30.63	$16.78 \pm 15.26$	$-11.00 \pm 4.88$	-47.94, 16.60
S18	Climbing stairs	$3.40 \pm 4.56$	$0.00 \pm 1.00$	-10.43, 12.43	$6.60 \pm 8.23$	$0.00 \pm 7.00$	-23.35, 11.75
S18	Outdoor walk	$9.86 \pm 4.86$	$9.00 \pm 2.50$	0.34, 19.39	$2.93 \pm 2.99$	$1.00 \pm 2.50$	-5.87, 9.25
S19	Light activity	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00
S19	Complete rest	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00	$0.00 \pm 0.00$	$0.00 \pm 0.00$	0.00, 0.00
S19	Indoor activity	$14.50 \pm 14.03$	$6.00 \pm 4.25$	-41.90, 38.01	$13.83 \pm 14.91$	$-8.50 \pm 1.88$	-43.05, 15.39
S19	Climbing stairs	$11.25 \pm 9.22$	$12.00 \pm 5.62$	-6.81, 29.31	$6.75 \pm 8.73$	$3.50 \pm 5.12$	-11.81, 24.31
S19	Outdoor walk	$11.68 \pm 6.85$	$10.00 \pm 6.00$	-12.42, 29.48	$7.42 \pm 5.98$	$0.00 \pm 5.25$	-18.93, 19.04
S20	Light activity	$3.56 \pm 7.88$	$0.00 \pm 1.50$	-11.88, 18.99	$2.33 \pm 5.29$	$0.00 \pm 0.50$	-8.04, 12.70
S20	Complete rest	$0.00 \pm \text{nan}$	$0.00 \pm 0.00$	nan, nan	$0.00 \pm \text{nan}$	$0.00 \pm 0.00$	nan, nan
S20	Indoor activity	$11.39 \pm 16.17$	$1.00 \pm 3.00$	-43.38, 30.38	$31.72 \pm 18.46$	$-40.00 \pm 16.38$	-67.90, 4.45
S20	Climbing stairs	$0.00 \pm \text{nan}$	$0.00 \pm 0.00$	nan, nan	$0.00 \pm \text{nan}$	$0.00 \pm 0.00$	nan, nan
S20	Outdoor walk	$11.33 \pm 6.39$	$9.00 \pm 3.50$	-8.14, 27.11	$26.93 \pm 14.13$	$-28.00 \pm 10.00$	-55.64, 2.53

Subject	Activity type	Espruino_mae	Espruino_med	Espruino_loa	Oxford_mae	Oxford_med	Oxford_loa
S01	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	3.64 ± 3.38	3.00 ± 1.50	-3.00, 10.27
S01	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S01	Indoor activity	30.33 ± 14.44	-26.50 ± 12.38	-68.02, 15.58	17.22 ± 18.27	3.50 ± 5.75	-51.05, 48.60
S01	Climbing stairs	8.25 ± 7.50	-3.00 ± 5.62	-25.95, 18.45	4.00 ± 5.48	1.50 ± 2.75	-8.10, 15.10
S01	Outdoor walk	10.00 ± 9.97	0.00 ± 7.25	-26.83, 28.90	13.33 ± 10.41	8.00 ± 9.25	-14.04, 36.27
S02	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	1.45 ± 2.58	0.00 ± 1.00	-3.61, 6.52
S02	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S02	Indoor activity	25.94 ± 13.88	-20.00 ± 10.00	-53.15, 1.26	14.61 ± 15.50	3.50 ± 7.50	-45.45, 37.78
S02	Climbing stairs	17.50 ± 16.13	-10.50 ± 14.00	-54.79, 30.79	7.50 ± 10.15	-0.50 ± 4.00	-21.42, 28.42
S02	Outdoor walk	15.93 ± 10.08	-14.00 ± 5.25	-39.45, 12.38	12.07 ± 9.41	8.00 ± 3.38	-14.59, 33.26
S03	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	5.73 ± 7.24	3.00 ± 2.50	-8.46, 19.92
S03	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S03	Indoor activity	41.61 ± 15.82	-49.50 ± 5.25	-72.62, -10.61	12.78 ± 18.75	-2.00 ± 3.50	-50.09, 30.98
S03	Climbing stairs	10.50 ± 8.81	-10.50 ± 4.50	-27.77, 6.77	3.50 ± 4.04	0.50 ± 2.00	-11.97, 9.97
S03	Outdoor walk	12.70 ± 8.52	-10.50 ± 9.38	-33.51, 15.31	7.25 ± 6.34	3.50 ± 4.12	-14.41, 21.31
S04	Light activity	0.09 ± 0.30	0.00 ± 0.00	-0.50, 0.68	7.45 ± 4.52	9.00 ± 2.50	-1.41, 16.32
S04	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S04	Indoor activity	22.83 ± 14.72	-18.50 ± 7.25	-51.68, 6.01	9.61 ± 14.40	-1.00 ± 3.50	-38.13, 25.13
S04	Climbing stairs	2.60 ± 3.78	-1.00 ± 1.50	-10.01, 4.81	5.20 ± 4.97	3.00 ± 4.00	-4.54, 14.94
S04	Outdoor walk	6.68 ± 6.60	0.00 ± 3.50	-18.13, 18.96	12.74 ± 9.74	12.00 ± 5.75	-6.58, 31.93
S05	Light activity	1.18 ± 2.64	0.00 ± 0.00	-3.99, 6.35	14.82 ± 12.06	15.00 ± 7.25	-8.83, 38.47
S05	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.57 ± 0.98	0.00 ± 0.50	-1.34, 2.48
S05	Indoor activity	39.39 ± 16.09	-44.00 ± 9.12	-73.06, -4.61	14.11 ± 17.61	1.00 ± 4.50	-48.97, 37.42
S05	Climbing stairs	12.25 ± 15.06	-2.50 ± 7.38	-34.30, 43.80	10.75 ± 19.52	1.50 ± 5.38	-27.50, 49.00
S05	Outdoor walk	12.23 ± 10.44	-5.00 ± 5.00	-34.54, 27.51	13.48 ± 11.71	8.00 ± 5.25	-18.57, 39.08
S06	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	1.50 ± 3.10	0.00 ± 0.88	-4.58, 7.58
S06	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S06	Indoor activity	22.56 ± 15.25	-20.00 ± 13.00	-53.53, 9.53	14.61 ± 16.95	2.00 ± 13.50	-49.29, 34.07
S06	Climbing stairs	12.50 ± 13.28	-6.00 ± 9.25	-41.60, 23.60	11.50 ± 10.41	-3.00 ± 7.25	-33.07, 33.07
S07	Light activity	0.18 ± 0.60	0.00 ± 0.00	-1.36, 1.00	1.55 ± 2.91	0.00 ± 0.50	-4.53, 7.26
S07	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S07	Indoor activity	26.61 ± 18.66	-31.50 ± 17.62	-63.62, 10.84	16.72 ± 12.89	10.50 ± 6.25	-39.74, 44.08
S07	Climbing stairs	17.00 ± 14.00	-6.00 ± 11.50	-48.20, 46.20	14.25 ± 17.21	3.00 ± 8.62	-34.52, 51.02
S07	Outdoor walk	6.70 ± 7.40	-0.50 ± 3.88	-19.75, 19.68	14.97 ± 10.03	13.50 ± 4.38	-10.87, 37.34
S08	Light activity	1.27 ± 3.13	0.00 ± 0.00	-7.41, 4.87	8.82 ± 7.93	10.00 ± 5.50	-7.93, 24.84
S08	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S08	Indoor activity	30.17 ± 15.71	-29.50 ± 14.00	-66.33, 10.44	15.61 ± 16.13	3.00 ± 11.25	-48.37, 38.71
S08	Climbing stairs	10.00 ± 10.80	-2.50 ± 6.25	-33.85, 23.85	5.50 ± 6.14	0.50 ± 3.00	-15.47, 18.47
S08	Outdoor walk	12.89 ± 7.43	-11.00 ± 5.75	-29.27, 5.20	9.11 ± 7.28	7.50 ± 6.12	-7.74, 24.39
S09	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S09	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S09	Indoor activity	24.61 ± 16.51	-19.50 ± 13.88	-57.42, 8.64	12.33 ± 14.81	2.00 ± 4.00	-41.17, 33.84
S09	Climbing stairs	9.00 ± 11.05	-5.50 ± 4.25	-30.65, 12.65	9.75 ± 7.68	4.00 ± 6.88	-22.42, 28.92
S09	Outdoor walk	8.00 ± 7.32	-3.00 ± 4.12	-23.76, 16.96	9.53 ± 8.10	7.50 ± 4.88	-10.76, 26.89
S10	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	6.70 ± 9.99	1.50 ± 6.38	-12.88, 26.28
S10	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S10	Indoor activity	19.67 ± 15.71	-12.50 ± 8.38	-50.47, 11.13	14.56 ± 17.12	3.50 ± 7.00	-49.13, 36.02
S10	Climbing stairs	7.25 ± 11.24	1.00 ± 4.12	-18.43, 29.93	9.25 ± 12.84	4.50 ± 5.38	-15.92, 34.42

S10	Outdoor walk	15.31 ± 11.77	-5.00 ± 13.50	-41.01, 34.39	15.59 ± 13.08	6.00 ± 13.00	-28.49, 45.04
S11	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	4.30 ± 5.12	2.00 ± 4.50	-5.74, 14.34
S11	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S11	Indoor activity	21.56 ± 16.59	-15.00 ± 11.88	-55.28, 13.73	14.50 ± 15.46	4.50 ± 5.38	-43.69, 40.24
S11	Climbing stairs	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S11	Outdoor walk	12.00 ± 9.94	-10.50 ± 6.25	-33.46, 13.10	12.82 ± 10.29	8.00 ± 6.50	-22.51, 36.51
S12	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.40 ± 1.26	0.00 ± 0.00	-2.08, 2.88
S12	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S12	Indoor activity	37.78 ± 15.97	-44.50 ± 11.62	-69.07, -6.49	13.28 ± 17.85	-1.00 ± 8.25	-49.07, 27.18
S12	Climbing stairs	18.50 ± 14.98	-6.50 ± 12.50	-52.93, 48.93	14.00 ± 17.42	0.50 ± 7.25	-40.18, 50.18
S13	Light activity	2.18 ± 3.28	0.00 ± 1.00	-8.60, 6.42	6.09 ± 4.70	5.00 ± 3.25	-4.42, 15.88
S13	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S13	Indoor activity	21.72 ± 13.64	-19.00 ± 5.38	-48.46, 5.02	15.33 ± 16.03	6.50 ± 4.12	-47.27, 39.50
S13	Climbing stairs	14.20 ± 13.39	0.00 ± 7.50	-37.70, 42.90	18.60 ± 15.53	10.00 ± 12.50	-32.01, 55.61
S13	Outdoor walk	22.86 ± 13.52	-17.00 ± 11.25	-59.12, 33.40	18.23 ± 18.53	7.00 ± 8.50	-32.72, 57.53
S14	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	3.60 ± 4.58	1.50 ± 3.12	-5.37, 12.57
S14	Complete rest	15.00 ± 14.00	-14.00 ± 8.50	-42.44, 12.44	15.00 ± 14.00	-14.00 ± 8.50	-42.44, 12.44
S14	Climbing stairs	3.75 ± 4.79	0.00 ± 1.88	-13.58, 11.08	8.75 ± 9.07	7.50 ± 5.88	-9.03, 26.53
S14	Outdoor walk	18.71 ± 12.95	-17.00 ± 11.25	-47.13, 13.71	17.97 ± 12.66	10.00 ± 10.75	-30.94, 48.68
S15	Light activity	1.00 ± 1.70	0.00 ± 0.75	-4.33, 2.33	3.10 ± 4.61	0.00 ± 1.12	-9.73, 11.93
S15	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S15	Indoor activity	36.94 ± 14.06	-43.00 ± 8.00	-64.51, -9.38	11.89 ± 15.68	-1.00 ± 6.12	-43.50, 26.17
S15	Climbing stairs	8.50 ± 13.18	0.50 ± 4.50	-35.20, 24.20	6.50 ± 6.95	0.50 ± 3.50	-20.05, 20.05
S16	Light activity	0.27 ± 0.90	0.00 ± 0.00	-1.50, 2.05	11.82 ± 9.92	14.00 ± 7.00	-7.62, 31.26
S16	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S16	Indoor activity	45.61 ± 17.49	-51.00 ± 3.62	-81.30, -9.25	13.67 ± 19.46	1.00 ± 4.25	-52.31, 35.42
S16	Climbing stairs	11.25 ± 7.80	-6.50 ± 8.88	-32.35, 23.85	3.75 ± 4.99	1.50 ± 2.62	-7.41, 13.91
S16	Outdoor walk	21.40 ± 12.97	-18.00 ± 11.00	-52.34, 16.34	13.97 ± 12.04	3.50 ± 9.62	-32.78, 39.11
S17	Light activity	1.40 ± 3.27	0.00 ± 0.00	-7.81, 5.01	1.40 ± 3.27	0.00 ± 0.00	-7.81, 5.01
S17	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S17	Indoor activity	37.33 ± 13.06	-37.00 ± 8.12	-62.92, -11.74	9.11 ± 10.91	2.00 ± 3.75	-29.98, 25.75
S17	Climbing stairs	9.75 ± 7.14	-5.00 ± 7.38	-27.22, 24.72	9.75 ± 11.76	7.00 ± 7.62	-13.30, 32.80
S17	Outdoor walk	20.00 ± 10.48	-23.00 ± 6.75	-41.11, 1.51	13.93 ± 7.19	13.50 ± 3.88	-2.62, 29.42
S18	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	2.44 ± 5.27	0.00 ± 1.00	-7.89, 12.77
S18	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S18	Indoor activity	24.44 ± 17.58	-24.00 ± 17.25	-61.45, 15.68	13.17 ± 16.99	-1.00 ± 5.00	-47.39, 31.72
S18	Climbing stairs	7.60 ± 10.78	0.00 ± 7.50	-28.74, 13.54	3.60 ± 4.62	0.00 ± 0.50	-11.57, 12.37
S18	Outdoor walk	2.86 ± 2.94	2.00 ± 2.00	-4.02, 8.92	8.38 ± 5.94	8.00 ± 3.00	-3.26, 20.02
S19	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S19	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S19	Indoor activity	20.56 ± 16.97	-15.00 ± 16.00	-53.83, 12.72	13.56 ± 14.86	3.00 ± 4.50	-42.41, 36.63
S19	Climbing stairs	6.25 ± 6.13	4.00 ± 5.12	-10.38, 19.88	10.25 ± 8.66	10.50 ± 5.12	-6.71, 27.21

S19	Outdoor walk	6.47 ± 6.41	1.00 ± 3.25	-16.99, 18.99	10.11 ± 6.17	6.00 ± 5.25	-13.16, 26.22
S20	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	3.78 ± 8.53	0.00 ± 1.50	-12.93, 20.49
S20	Complete rest	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S20	Indoor activity	36.17 ± 12.39	-38.00 ± 7.38	-60.45, -11.88	14.22 ± 15.16	5.00 ± 3.75	-44.37, 36.81
S20	Climbing stairs	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S20	Outdoor walk	8.81 ± 7.46	-6.00 ± 4.25	-25.02, 10.65	13.26 ± 6.63	12.00 ± 3.75	-7.58, 30.24

Subject	Activity type	Autocorrelation_mae	Autocorrection_med	Autocorrection_loa	fft_mae	fft_med	fft_loa
S01	Light activity	0.73 ± 2.41	0.00 ± 0.00	-4.00, 5.45	2.36 ± 2.29	2.00 ± 1.00	-2.13, 6.86
S01	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S01	Indoor activity	23.44 ± 14.95	-20.00 ± 14.12	-58.05, 17.61	16.61 ± 16.13	-9.50 ± 2.12	-51.18, 27.07
S01	Climbing stairs	6.50 ± 6.03	-1.50 ± 4.00	-18.87, 18.87	9.00 ± 7.07	-4.00 ± 6.50	-26.83, 19.83
S01	Outdoor walk	11.19 ± 9.44	0.00 ± 7.00	-29.15, 28.86	9.33 ± 8.94	-2.00 ± 5.50	-24.80, 26.28
S02	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.73 ± 1.27	0.00 ± 0.50	-1.77, 3.22
S02	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S02	Indoor activity	26.61 ± 17.66	-30.50 ± 17.12	-61.22, 8.00	15.11 ± 15.68	-11.50 ± 6.50	-45.84, 15.62
S02	Climbing stairs	14.00 ± 12.73	-8.50 ± 11.25	-43.62, 24.62	10.25 ± 7.23	-6.00 ± 8.12	-29.67, 21.17
S02	Outdoor walk	28.90 ± 15.49	-31.50 ± 13.12	-60.67, 3.94	8.77 ± 7.29	-5.00 ± 3.75	-24.78, 18.72
S03	Light activity	0.73 ± 2.41	0.00 ± 0.00	-4.00, 5.45	3.09 ± 4.95	1.00 ± 2.00	-6.61, 12.79
S03	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S03	Indoor activity	32.17 ± 18.70	-33.50 ± 15.12	-68.82, 4.49	21.67 ± 15.35	-19.00 ± 6.12	-52.56, 10.12
S03	Climbing stairs	8.50 ± 7.33	-6.50 ± 7.50	-25.80, 13.80	5.25 ± 4.27	-5.50 ± 2.38	-13.62, 3.12
S03	Outdoor walk	13.75 ± 9.91	-8.00 ± 5.25	-37.67, 24.57	9.05 ± 5.39	-6.50 ± 5.88	-23.50, 14.40
S04	Light activity	1.09 ± 2.59	0.00 ± 0.00	-3.98, 6.16	4.82 ± 3.37	5.00 ± 2.00	-1.79, 11.43
S04	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S04	Indoor activity	23.61 ± 17.78	-22.00 ± 15.50	-59.02, 12.47	15.22 ± 14.28	-9.50 ± 6.25	-43.20, 12.76
S04	Climbing stairs	2.40 ± 3.05	-1.00 ± 2.00	-8.38, 3.58	3.20 ± 3.56	0.00 ± 1.50	-10.54, 8.94
S04	Outdoor walk	11.13 ± 9.42	1.00 ± 6.75	-27.63, 29.89	7.26 ± 7.06	1.00 ± 5.00	-14.77, 22.32
S05	Light activity	3.45 ± 5.43	0.00 ± 3.00	-7.19, 14.10	9.64 ± 7.31	8.00 ± 5.00	-4.69, 23.97
S05	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.86 ± 1.21	0.00 ± 0.75	-1.52, 3.24
S05	Indoor activity	21.83 ± 15.19	-17.00 ± 11.25	-51.61, 7.95	19.33 ± 14.41	-15.50 ± 4.75	-50.56, 16.11
S05	Climbing stairs	16.50 ± 11.12	-11.00 ± 13.75	-47.16, 34.16	13.25 ± 15.13	-4.00 ± 8.62	-36.81, 45.31
S05	Outdoor walk	17.35 ± 11.76	-8.00 ± 10.25	-46.49, 28.55	10.84 ± 10.04	-4.00 ± 7.50	-29.47, 28.95
S06	Light activity	0.80 ± 2.53	0.00 ± 0.00	-4.16, 5.76	1.20 ± 2.57	0.00 ± 0.38	-3.84, 6.24
S06	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S06	Indoor activity	20.22 ± 15.88	-13.50 ± 11.88	-51.35, 10.90	15.61 ± 16.76	-9.00 ± 11.88	-48.46, 17.24
S06	Climbing stairs	13.00 ± 16.79	-1.50 ± 7.25	-48.21, 34.21	14.25 ± 11.95	-6.00 ± 10.12	-43.52, 31.02
S07	Light activity	0.82 ± 2.14	0.00 ± 0.00	-3.96, 4.87	1.27 ± 2.00	0.00 ± 0.50	-3.43, 5.24
S07	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S07	Indoor activity	15.11 ± 13.88	-11.00 ± 5.88	-42.71, 13.16	10.50 ± 14.78	-5.50 ± 3.75	-39.73, 20.07
S07	Climbing stairs	12.50 ± 9.95	-4.50 ± 8.50	-36.39, 31.39	12.75 ± 11.79	-3.50 ± 8.12	-36.17, 37.67
S07	Outdoor walk	7.37 ± 7.72	0.00 ± 4.50	-18.56, 22.76	7.43 ± 7.61	-0.50 ± 4.88	-18.59, 22.66
S08	Light activity	4.55 ± 3.93	0.00 ± 3.75	-9.40, 13.40	6.00 ± 4.98	7.00 ± 4.00	-6.13, 16.67
S08	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S08	Indoor activity	20.72 ± 16.64	-11.00 ± 13.50	-56.05, 18.83	17.50 ± 15.36	-9.50 ± 7.88	-50.43, 21.21
S08	Climbing stairs	9.00 ± 7.79	-4.00 ± 6.50	-27.83, 18.83	9.50 ± 10.21	-3.50 ± 6.50	-32.05, 20.05
S08	Outdoor walk	17.25 ± 12.29	-14.00 ± 11.75	-41.61, 7.39	6.82 ± 5.37	-4.50 ± 4.25	-19.18, 10.26
S09	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.10 ± 0.32	0.00 ± 0.00	-0.52, 0.72
S09	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S09	Indoor activity	17.44 ± 15.44	-10.50 ± 12.50	-47.70, 12.81	15.22 ± 14.40	-11.50 ± 6.88	-45.43, 19.43
S09	Climbing stairs	9.00 ± 11.17	-4.00 ± 6.50	-32.11, 17.11	8.00 ± 10.30	-3.00 ± 5.50	-29.27, 16.27

S09	Outdoor walk	15.83 ± 10.40	-16.00 ± 9.38	-40.73, 15.99	7.13 ± 5.64	-3.50 ± 4.25	-19.85, 14.65
S10	Light activity	2.00 ± 3.30	0.00 ± 1.88	-4.47, 8.47	4.30 ± 6.22	1.00 ± 4.00	-7.89, 16.49
S10	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S10	Indoor activity	24.56 ± 14.59	-21.50 ± 11.62	-53.14, 4.03	16.67 ± 16.37	-10.00 ± 5.12	-48.75, 15.42
S10	Climbing stairs	10.50 ± 8.54	-7.00 ± 8.75	-31.57, 18.57	8.00 ± 10.95	-1.00 ± 4.50	-22.59, 30.59
S10	Outdoor walk	21.76 ± 12.90	-16.00 ± 12.50	-54.28, 20.83	13.24 ± 11.57	-2.00 ± 10.00	-36.15, 33.25
S11	Light activity	2.90 ± 4.86	0.00 ± 3.00	-6.63, 12.43	3.00 ± 4.06	0.50 ± 2.75	-4.95, 10.95
S11	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S11	Indoor activity	19.00 ± 14.43	-13.00 ± 6.00	-49.26, 13.93	15.78 ± 14.86	-10.00 ± 3.62	-46.44, 17.99
S11	Climbing stairs	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S11	Outdoor walk	15.73 ± 14.54	-7.00 ± 10.75	-46.63, 20.81	8.64 ± 10.04	-0.50 ± 5.38	-28.61, 22.24
S12	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.40 ± 0.84	0.00 ± 0.00	-1.25, 2.05
S12	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S12	Indoor activity	25.17 ± 15.39	-23.00 ± 13.00	-55.34, 5.01	21.56 ± 15.01	-16.00 ± 9.00	-50.98, 7.87
S12	Climbing stairs	18.75 ± 16.28	-5.50 ± 12.12	-53.81, 52.31	17.25 ± 16.07	-4.00 ± 10.62	-49.91, 50.41
S13	Light activity	3.09 ± 3.83	0.00 ± 1.00	-10.01, 9.65	3.64 ± 3.47	3.00 ± 2.50	-4.27, 10.82
S13	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.14 ± 0.38	0.00 ± 0.00	-0.60, 0.88
S13	Indoor activity	27.39 ± 18.34	-32.00 ± 18.75	-63.33, 8.55	15.89 ± 15.74	-10.50 ± 4.00	-46.74, 14.96
S13	Climbing stairs	13.20 ± 10.89	0.00 ± 9.00	-37.38, 34.18	13.80 ± 12.85	0.00 ± 10.00	-31.45, 43.05
S13	Outdoor walk	28.14 ± 16.08	-30.50 ± 11.00	-65.86, 15.95	16.59 ± 14.16	-5.00 ± 10.75	-44.42, 42.15
S14	Light activity	0.50 ± 1.58	0.00 ± 0.00	-2.60, 3.60	1.70 ± 2.31	1.00 ± 1.00	-2.83, 6.23
S14	Complete rest	15.00 ± 14.00	-14.00 ± 8.50	-42.44, 12.44	15.00 ± 14.00	-14.00 ± 8.50	-42.44, 12.44
S14	Climbing stairs	5.00 ± 5.60	0.50 ± 2.75	-13.89, 16.89	5.00 ± 6.16	1.50 ± 3.25	-11.04, 18.04
S14	Outdoor walk	19.45 ± 14.39	-16.00 ± 15.25	-52.62, 23.14	13.97 ± 10.60	-4.00 ± 10.25	-37.61, 30.32
S15	Light activity	1.00 ± 1.70	0.00 ± 0.75	-4.33, 2.33	2.60 ± 2.55	0.00 ± 2.25	-6.63, 7.83
S15	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S15	Indoor activity	25.17 ± 16.56	-24.50 ± 15.25	-57.62, 7.29	20.44 ± 13.97	-21.50 ± 9.25	-47.82, 6.93
S15	Climbing stairs	12.25 ± 8.58	-7.50 ± 9.88	-35.42, 24.92	9.50 ± 9.68	-4.00 ± 6.75	-31.25, 19.25
S16	Light activity	3.91 ± 3.91	0.00 ± 4.00	-5.79, 12.16	7.36 ± 6.30	9.00 ± 6.00	-4.98, 19.71
S16	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S16	Indoor activity	41.39 ± 16.65	-50.00 ± 8.12	-74.02, -8.76	21.33 ± 15.85	-18.00 ± 8.88	-54.64, 14.86
S16	Climbing stairs	11.00 ± 8.08	-5.00 ± 8.00	-31.16, 27.16	7.00 ± 4.83	-4.50 ± 5.75	-20.16, 14.16
S16	Outdoor walk	26.80 ± 15.73	-28.00 ± 8.75	-60.99, 10.59	15.40 ± 10.35	-8.00 ± 6.75	-41.04, 27.17
S17	Light activity	1.40 ± 3.27	0.00 ± 0.00	-7.81, 5.01	1.10 ± 2.42	0.00 ± 0.00	-5.85, 3.65
S17	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S17	Indoor activity	37.56 ± 14.81	-41.50 ± 6.88	-66.59, -8.52	17.00 ± 10.26	-15.00 ± 5.62	-37.10, 3.10
S17	Climbing stairs	11.50 ± 14.15	-2.50 ± 7.00	-32.15, 41.15	7.75 ± 7.59	-2.50 ± 5.12	-21.57, 24.07
S17	Outdoor walk	17.33 ± 11.92	-16.00 ± 10.62	-41.78, 8.31	5.50 ± 4.60	-3.00 ± 2.50	-15.17, 12.84
S18	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	1.44 ± 3.00	0.00 ± 0.50	-4.44, 7.33
S18	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S18	Indoor activity	17.61 ± 16.34	-10.50 ± 9.00	-51.05, 18.50	18.94 ± 14.19	-14.00 ± 5.75	-49.21, 14.65
S18	Climbing stairs	6.40 ± 7.13	0.00 ± 7.00	-21.57, 11.97	6.60 ± 7.89	0.00 ± 6.50	-22.96, 12.16

S18	Outdoor walk	4.97 ± 4.56	1.00 ± 4.00	-9.78, 14.89	3.97 ± 3.09	-1.00 ± 2.50	-10.37, 9.47
S19	Light activity	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S19	Complete rest	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00, 0.00
S19	Indoor activity	16.44 ± 15.07	-12.50 ± 9.88	-46.13, 13.46	12.83 ± 15.49	-8.00 ± 4.50	-43.20, 17.53
S19	Climbing stairs	8.50 ± 8.39	7.00 ± 3.25	-7.94, 24.94	6.25 ± 8.66	2.00 ± 4.12	-13.37, 23.87
S19	Outdoor walk	7.11 ± 5.40	0.00 ± 7.00	-18.69, 16.69	6.05 ± 5.12	-1.00 ± 4.00	-16.41, 15.04
S20	Light activity	0.89 ± 2.67	0.00 ± 0.00	-4.34, 6.12	2.56 ± 5.94	0.00 ± 0.50	-9.09, 14.20
S20	Complete rest	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S20	Indoor activity	39.06 ± 12.92	-42.00 ± 6.12	-64.38, -13.73	17.22 ± 14.83	-13.50 ± 7.38	-46.44, 12.22
S20	Climbing stairs	0.00 ± nan	0.00 ± 0.00	nan, nan	0.00 ± nan	0.00 ± 0.00	nan, nan
S20	Outdoor walk	23.19 ± 12.25	-26.00 ± 10.00	-47.62, 1.54	6.15 ± 5.68	-2.00 ± 5.00	-17.58, 15.21

## 6.2. A.2 Mean Percentage Error in 30 seconds time window for each activity

Algorithm	Light activity - Reference Steps	Light activity - Steps Range	Light activity - Mean Error (%)
Dummy	0.00	0.00 - 24.00	nan
Banglesimple	0.00	0.00 - 16.00	nan
Espruino	0.00	0.00 - 1.00	nan
Oxford	0.00	0.00 - 26.00	nan
Autocorrelation	0.00	0.00 - 8.00	nan
Fft	0.00	0.00 - 18.00	nan

Algorithm	Complete rest - Reference Steps	Complete rest - Steps Range	Complete rest - Mean Error (%)
Dummy	0.00	0.00 - 0.00	nan
Banglesimple	0.00	0.00 - 0.00	nan
Espruino	0.00	0.00 - 0.00	nan
Oxford	0.00	0.00 - 0.00	nan
Autocorrelation	0.00	0.00 - 0.00	nan
Fft	0.00	0.00 - 0.00	nan

Algorithm	Climbing stairs - Reference Steps	Climbing stairs - Steps Range	Climbing stairs - Mean Error (%)
Dummy	28.70	0.00 - 60.00	42.34
Banglesimple	28.70	0.00 - 51.00	30.90
Espruino	28.70	0.00 - 52.00	26.29
Oxford	28.70	0.00 - 65.00	40.20
Autocorrelation	28.70	0.00 - 58.00	35.89
Fft	28.70	0.00 - 51.00	30.72

Algorithm	Indoor activity - Reference Steps	Indoor activity - Steps Range	Indoor activity - Mean Error (%)
Dummy	49.18	0.00 - 61.00	25.25
Banglesimple	49.18	0.00 - 54.00	48.37
Espruino	49.18	0.00 - 58.00	58.82
Oxford	49.18	0.00 - 67.00	26.22
Autocorrelation	49.18	0.00 - 56.00	55.40
Fft	49.18	0.00 - 59.00	34.54

<b>Algorithm</b>	<b>Outdoor walk - Reference Steps</b>	<b>Outdoor walk - Steps Range</b>	<b>Outdoor walk - Mean Error (%)</b>
Dummy	38.59	0.00 - 61.00	40.21
Banglesimple	38.59	0.00 - 54.00	37.05
Espruino	38.59	0.00 - 55.00	26.95
Oxford	38.59	0.00 - 69.00	40.03
Autocorrelation	38.59	0.00 - 56.00	36.88
Fft	38.59	0.00 - 55.00	20.85

