Compas'2023 : Parallélisme/ Architecture/ Système
LISTIC/USMB - Annecy France, 4-7 juillet 2023

A robust smartphone-based architecture for prolonged monitoring of gait

El Amine Bechorfa[a, b], Antoine Boutet[b], and Carole Frindel[a]

[a]Univ Lyon, INSA Lyon, CREATIS, Inserm
[b]Univ Lyon, INSA Lyon, Inria, CITI

Abstract
Gait analysis is important for evaluating neurological disorders such as Stroke and Parkinson's disease. Prior to the technological advancements, healthcare professionals had to rely on subjective assessments (i.e., human-based) of gait which were time consuming and not very reproducible. With the advent of IoT, objective (e.g., measurement-based) assessment methods, gait analysis can now be performed more accurately and effectively. It is worth noting, however, that there are still limitations to these objective methods, especially the lack of privacy-preserving continuous data collection. To overcome this limitation, we present in this paper a privacy-by-design monitoring application for post-stroke patients to evaluate their gait before, during, and after a rehabilitation program. Gait measurements were collected by a mobile application that captures spatiotemporal parameters continuously in background using the built-in smartphone accelerometer. Statistical techniques were then used to extract general indicators about the performed activity as well as some more specific gait metrics in real-time such as regularity, symmetry and step speed. These metrics are calculated based on the detected steps while patients are performing an activity. This two-level analysis provides valuable insights and statistical information about the activities performed by the patient. It is also a useful tool for practitioners to monitor the progression of neurological disorders. Ongoing work is underway to integrate learning models into the application which will be trained and personalized via a federated approach.

Index Terms : Smartphone-based system, Inertial sensors, Data collection, Software architecture, Gait analysis, Real-time patients monitoring, Signal Processing

1. Introduction

As the population ages, the prevalence of health issues like stroke is increasing [16, 6]. This has resulted in a growing need for continuous monitoring and follow-up of patients. While various services are available to provide such monitoring, they face significant challenges. For instance, delivering these services requires extensive human resources, material, and time. Moreover, personalized monitoring of individual patients is difficult, if not impossible, to achieve. Finally, long-term monitoring is costly for the healthcare system. A viable solution is to automate and personalize the patient follow-up process by providing relevant metrics and indicators for both the practitioner and the patient while maintaining data privacy.
Recent works have explored the use of built-in smartphone sensors to identify walking, counting steps and/or analyse gait events using several and different methods [8, 4] based either on

time domain such as Mean Crossing Counts, Dynamic Time Wrapping [14], Windowed Peaks Detection, etc or frequency domain such as Short Term Fourier Transform [13].
Patient monitoring applications, and gait analysis, in particular, offer a valuable solution for evaluating neurological disorders. They enable practitioners to track the evolution of gait parameters before, during, and after rehabilitation programs, allowing them to assess their effectiveness and make informed decisions about program development.
The paper is organized as follows. We start by reviewing background in patient monitoring in Section 2 before presenting an overview of our post-stroke monitoring application in Section 3. We then report preliminary results about the performance of our application to detection both dynamic activity and walking activity in Section 4 before concluding in Section 5.

2. Background

To monitor patients, the appropriate method must be selected based on the specific objective assessment. This can be achieved using reflective markers placed on various body parts (e.g., on feet, legs and pelvis) or through metrics computed from Inertial Measurement Units (IMUs). For instance, [15] focus on the use of IMUs with built-in accelerometer. Zijlstra et al. [17], on the other hand, use peaks detection to extract gait events from trunk accelerations, identifying local maxima and minima of the vertical acceleration signal to determine heel strikes and toe-offs, respectively. They also used a time normalization algorithm to create a standard gait cycle, which allowed for the calculation of spatio-temporal gait parameters such as step length and cadence. In [1], a threshold-based approach is proposed to detect initial contact and toe-off events from the acceleration signals, and the authors compare accelerometer-based gait event detection algorithms with force plate-based gold standard methods.
Ellis et al. [7] validate the accuracy of gait analysis using a smartphone-based mobile application called *SmartMOVE* for Parkinson's disease patients and healthy elderly individuals. The application includes Rhythmic Auditory Cueing, which is a technique used in gait rehabilitation where patients synchronize their steps to an external auditory cue. This technique shows promise as a potentially effective non-pharmacological intervention for personalized gait rehabilitation [5]. Some studies detail methods to develop a gait analysis platform. In [9], the author proposes the general approach of (1) activity identification, (2) event detection, and (3) analysis, using machine learning to identify activity.
There are a lot of wearable sensors and IoT-based monitoring applications, such as MSCopilot [1] which is a medical device for monitoring multiple sclerosis. It allows practitioners to evaluate the progression of patients at home and between appointments and to share data with their healthcare providers. However, collecting motion data can be lead to the inference of personal and potentially sensitive information or lead to re-identification [2, 11]. This privacy concern must be addressed to avoid becoming a barrier to user adoption. Despite the increasing adoption of IoT and the emergence of monitoring applications, few studies address the real-life and continuous use of this analysis while taking in account patient's private data.

3. Post-stroke Monitoring Application : An Overview

Our application is designed to meet the needs of a real-world medico-social use case with ARRPAC [2], a medico-social center part of the Hospices Civils de Lyon (HCL) that provides support for individuals with post-stroke disorders. ARRPAC's mission is to address the medical and

1. https ://www.mscopilot.com/
2. https ://gcsms-arrpac.fr/

social needs of their patients, and our application serves as a valuable tool to enhance the personalization and effectiveness of the support they provide. By partnering with ARRPAC, we aim to demonstrate the potential of our solution in improving the delivery of medico-social services. While there are valuable insights into the needs of such a solution, several challenges must be addressed. One major challenge is the need for continuous patient monitoring, which requires continuous processing and necessitates the development of sensor-probing strategies to minimize battery drain. Another challenge is maintaining patient privacy, as patients may be hesitant to adopt these technologies if they are not convinced of their privacy-preserving capabilities. Finally, a flexible presentation dashboard is required for both practitioners and patients. This dashboard should be easy to use and allow for the clear visualization of important information.

In this section, we provide an overview of our post-stroke monitoring application. We begin by describing the application architecture in section 3.1. We then move on to discuss data collection in section 3.2, followed by an explanation of the metrics and the dashboard in sections 3.3 and 3.4, respectively. Finally, we provide more detailed information on gait parameter extraction and algorithm validation in section 3.5, and implementation details in section 3.6.

3.1. Architecture

The design of our privacy-preserving framework comprises three main elements (Figure 1) : a client application running on the user's smartphone, the application server, and the hospital practitioner. To limit and minimize the exposition of sensitive information, the raw data stays on the smartphone and are not shared with the server. This data is instead computed locally on the smartphone and only specific metrics and gait indicators are send to the application server.
This data is instead computed locally on the smartphone and only specific metrics and gait indicators are send to the application server. This minimisation scheme reduces the possible inference attacks that an adversary can conduct on the stored data [10, 3] (i.e., from few metrics compared to raw data). In addition, the metrics of patients can also be encrypted before being uploaded and stored in a database of the server. The practitioner can then request the application server (and decipher in case the stored information are encrypted) to monitor patient through a dashboard implemented as a website providing a clear representation of the patients results and evolution.

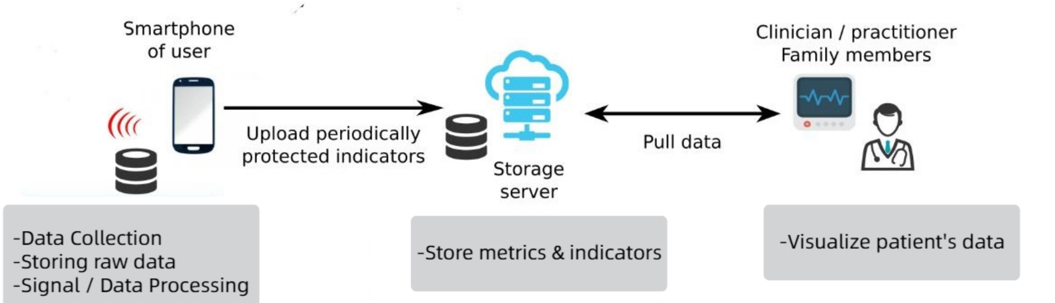


FIGURE 1 – The global architecture of our application

3.2. Data Collection

Data collection is a continuous process that runs in the background, even when the user is not actively using the application. It is important to note that this background data collection is designed to be lightweight in order to limit the impact on the battery consumption. To achieve that, the application collects data at a low sampling frequency until an activity is detected. If activity is detected, the application increases the sampling frequency. To detect an activity, the application computes the total acceleration of the subject along all three axes of the accelerometer. This was done by applying the magnitude formula, $|\mathbf{A}| = \sqrt{x^2 + y^2 + z^2}$, which provides a single value that represents the total intensity of motion at a given moment in time.
By analysing the total intensity of motion, our approach uses a Standard Deviation (std) threshold to identify periods of movement, while eliminating static activities. Specifically, values that exceed this threshold are considered to be periods of movement. Once a movement is identified, pattern detection is applied to further refine the walking activity detection. By filtering data based on the thresholding and the pattern detection, walking activity can be detected when the signal meets certain criteria, in our case, we set another threshold for a minimum duration of patterned signal. Appendix B reports the accuracy of this dynamic activity detection on the MotionSense dataset.

3.3. Metrics

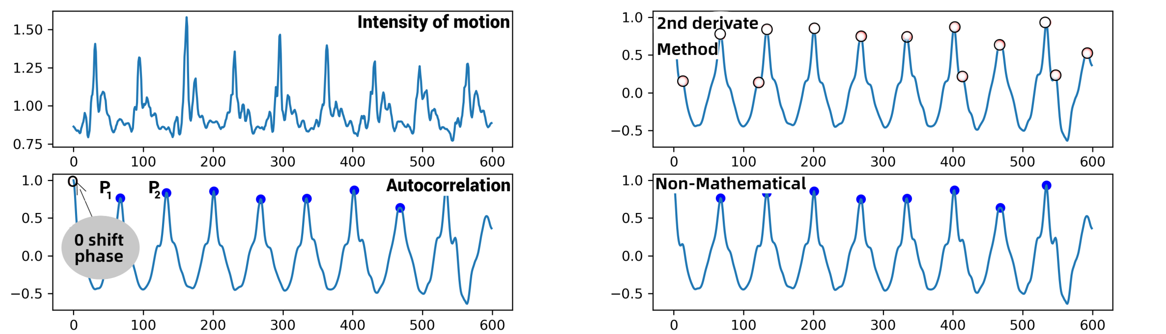
We provide two types of metrics. The first type is related to physical indicators for patients and their families, such as monitoring daily activity such as the number of steps and the type of activity (static vs dynamic). The second type is related to measurements used to quantify different aspects of human gait or gait analysis metrics for hospital practitioners, such as cadence, regularity, symmetry and step speed. To calculate these metrics, we use an autocorrelation function (explained in section 3.5), which allows us to extract many important gait parameters. As illustrated in figure 2a, the first peak of the autocorrelation signal P_1 represents the correlation between neighboring steps and the second peak P_2 correspond to the correlation between two consecutive strides. From this information, we assume that the ratio between two consecutive peaks (e.g., P_1/P_2) and two non-consecutive peaks represents symmetry and regularity, respectively.

3.4. Dashboards

We also developed dashboards to present the metrics to both patients (through the smartphone application) and the practitioners (through a web interface). The patient dashboard (Figure 5a) is highly customizable, with a vast library of metrics and daily activity indicators to choose from. Patients can select the considered indicators and arrange them in any order they desire. The second dashboard (Figure 6) in the other side, is dedicated to metrics related to gait for the practitioner. This includes analyzing walk dynamics, regularity, and symmetry. This ergonomic dashboard gives a lot of flexibility to the practitioners (e.g., visualizing results of the last month, week, etc). By providing practitioners with these gait-related metrics and their evolution over time, our application aims to improve the accuracy of diagnosis and enhance the effectiveness of treatment.

3.5. Steps Detection

Since human gait is a periodic motion, the metrics can be derived using the autocorrelation function of the series that represents the total intensity of the motion. An autocorrelation function measures the correlation between a value at time t_i and t_{i+1} of the same signal. It is important to note that the autocorrelation values exhibit a peak or a local maximum with each full



(a) Accelerometer magnitude $|\mathbf{A}|$ and the corresponding ACF signal

(b) Second derivative (no parameter) and threshold-based methods $((\Delta) = 1)$ for peaks detection

FIGURE 2 – Autocorrelation signal and peaks detection from MotionSense dataset

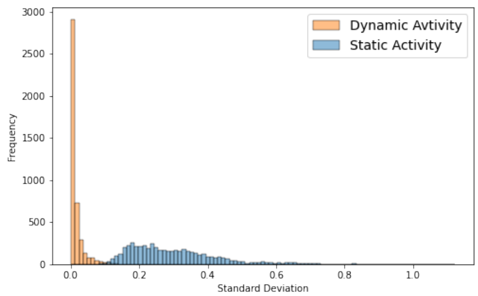
step. The following equation 1 represents a normalized and unbiased autocorrelation function :

$$\mathrm{ACF}(k) = \frac{1}{(N-k)(\sigma^2)} \sum_{i=1}^{N-k} (x_i - \mu)(x_{i+k} - \mu), \quad (1)$$

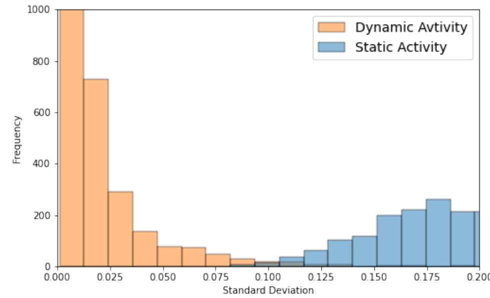
where k denotes the time shift, x_i represents the gait data at time i, μ is the mean of the gait data, N and σ^2 are the length and the variance of the gait data, respectively. The $\mathrm{ACF}(k)$ value indicates the degree of similarity between the gait data at time i and the gait data at time $i+k$, as a function of the time shift k. To normalize the values between the range $[-1;1]$, we are dividing by ν_i. Also, to avoid attenuation of the autocorrelation signal, we are dividing by $(N-k)$ instead of N to ensure that the number of terms in the numerator always equals to the denominator $(N-k)$.
Figure 2a shows the intensity of motion and its corresponding autocorrelation, while 2b displays the extracted samples. The red dots that appear as peaks in the autocorrelation signal were not detected using the mathematical zero-derivative based method (zero-crossings of first derivative) or the second derivative sign-changing technique. Instead, we employed the concept that a peak is characterized by the presence of lower points around it. To detect the peaks, we established an optimal threshold value (Δ) so we can prevent the detection of unwanted peaks in noisy signals (Figure 2b). If the difference between the point and its surroundings is greater than Δ, we identify that point as a peak. The process of detecting steps, beginning with the collection of data, is summarized in Algorithm 1 (Appendix A).

3.6. Implementation Details

The mobile application was developed in the Android Studio (Java) environment for devices running on Android 5.0 or higher. We utilized the Android Sensor Manager to access the device's sensors and the Room Database to store data locally. For the back-end implementation, we used the Flask framework in Python and the PostgreSQL relational database to store results on a remote secure server. We utilized a Toolkit and Object Relational Mapper called SQLAlchemy to access the database from our Flask API. Finally, we used React.js, a free JavaScript library developed by Facebook, to create reusable UI components for our web application dashboard, enabling the development of complex and interactive applications.



(a) std Distribution



(b) std Distribution Zoomed

FIGURE 3 – Distribution of the Standard Deviation (std) of the magnitude $|\mathbf{A}|$ of the accelerometer signal

4. Experiments

In this section, we assessed the process of distinguishing between static and dynamic activity types using the publicly available Motionsens dataset[3]. This dataset contains accelerometer data from smartphones (as well as other information like gender, height, etc) for 24 data subjects who performed 6 activities (such as walking, standing, etc.) in 15 trials. We categorized the activities into two groups : static and dynamic, and tested our activity detection method. To do this, we calculated the Standard Deviation (std) of the magnitude of our data $|\mathbf{A}|$ and obtained the distribution shown in Figure 3.

Table 1 summarizes the results of the two tasks to be performed on the signals. The first task involves recognizing when someone is walking, by using a threshold called the std, as explained in Section 3.2. The second task is about detecting each step while walking, as explained in Section 3.5. For each task, we tested different combinations of parameter values to assess how sensitive our method was to these parameters and to identify the optimal values.

It is essential to note that we processed windows of 150 samples with a sampling rate of 50 Hz, which corresponds to 3-second windows. In the first taks, we experimented with three different values of the standard deviation threshold. We aimed to maximize the accuracy and precision of our predictions by testing these values. In the second level, we manually set a peak detection threshold (Δ) for each output signal from the first level. We only assumed that someone was walking when we detected at least three peaks within a given window.

TABLE 1 – Accuracy and precision of first and second tasks when their parameters change

Task	Thresholds	TP	FP	TN	FN	Accuracy	Precision
Detection of static or dynamic activity	std = 0.38	1198	0	4357	3995	0.581	1.0
	std = 0.1	5176	50	4307	17	**0.992**	**0.990**
Detection of a walking activity	$\Delta = 0.6$, std= 0.38	1159	0	0	39	**0.967**	**1.0**
	$\Delta = 0.6$, std = 0.1	5007	4	46	169	**0.966**	**0.999**

3. https ://github.com/mmalekzadeh/motion-sense

Table 1 reports the accuracy of the detection on the MotionSense dataset according to different thresholds. Specifically, the detection if the activity is static or dynamic is firstly performed and then the detection of the walking activity is done on the data identifying as dynamic activity. Results show that a higher `std` threshold in the detection of the nature of the activity (i.e., static or dynamic) leads to higher accuracy. However, a lower `std` threshold means that most of the data are identified as dynamic resulting to a higher computational cost (and battery consumption). Nonetheless, applying our filters yielded promising results in both cases.

5. Conclusion and future work

The use of objective methods using IMUs has significantly improved the accuracy and effectiveness of evaluating neurological disorders such as Stroke and Parkinson's disease. However, privacy concerns and computational limitations remain significant challenges. To address this, the paper proposed a privacy-by-design monitoring application for post-stroke patients, which collects spatiotemporal parameters using the built-in smartphone accelerometer and extracts gait metrics in real-time. While this approach provides valuable insights into gait analysis and patient monitoring, future work is necessary to integrate more gait analysis metrics and other metrics that therapeutic experts consider beneficial. Additionally, federated learning is currently explored to further enhance the accuracy and effectiveness of activity type detection.

Annexes

A. Walking Identification Pseudo-Algorithm

Algorithm 1 Walking Identification Algorithm

Require: Accelerometer signal collected in background
Ensure: Walking identification flag

Variables : *magnitude*, `std`, *threshold*, *pattern*, *t*, *duration*
while listening to sensor **do**
    calculate *magnitude* of accelerometer signal
    calculate standard deviation (`std`) of *magnitude*
    if `std` > *threshold* **then**
        pattern detection
        if *pattern* exists and *duration* > *t* **then**
            Walking detected
            Start collecting data
            Perform Step Detection
            Perform Gait Analysis
        end if
    end if
end while

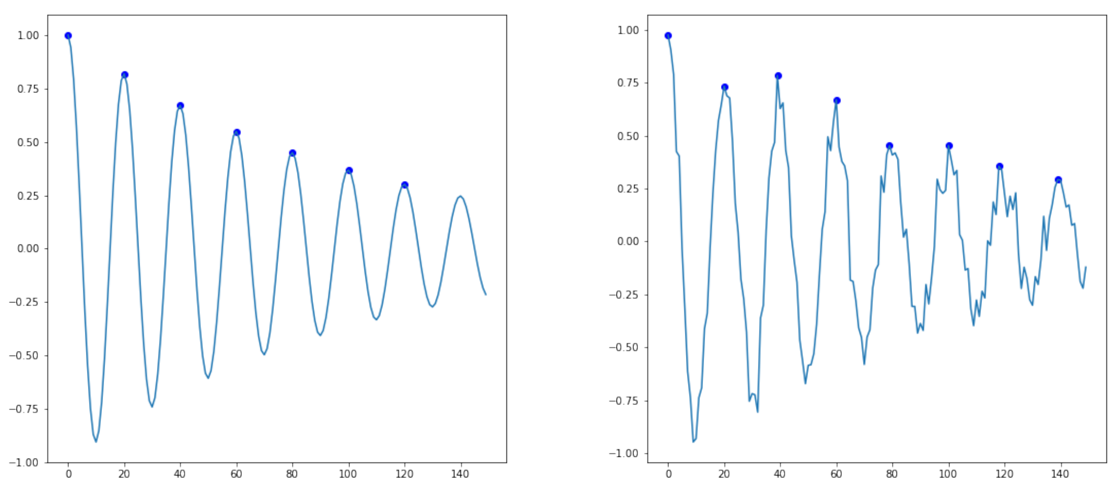
B. Steps Detection Algorithm Validation

We assessed the validity and accuracy of our algorithms using a damped cosine function $y(t)$, defined as :

$$y(t) = Ae^{-\alpha t}\cos(2\pi f t + \phi), \tag{1}$$

with manually fixed parameters and a defined number of samples representing time. The dependent variable $y(t)$ is a function of time t and is defined by the amplitude A, the decay constant α, the frequency f, and the phase angle ϕ of the cosine wave.
The objective was to use the peaks detection algorithm to identify the peaks and to determine a close estimate of the decay rate constant α using all the detected peaks. This was accomplished by calculating the decay rate constant using every two consecutive peaks P_k and P_{k+1}, as shown here :

$$\alpha(t) = \ln(P_k/P_{k+1})/(t_{k+1} - t_k). \tag{3}$$



(a) Peaks Detection Original Signal (b) Peaks Detection Noisy Signal

FIGURE 4 – Difference between the detected peaks in both noisy and clean signal

After adding a random noise N with a normal (Gaussian) probability distribution with mean μ and standard deviation σ to our signal (see Figure 4), we calculated the Signal-to-Noise Ratio (snr) in decibels (dB), for each peak A, as :

$$snr = 10\log_{10}(\frac{A}{\sigma}). \tag{2}$$

The results are summarized in Table 2. Note that for this example, we fixed the noise parameters $\sigma = 0.1$ and $\mu = 0$ according to [12], and used the parameters of the original damped cosine function ($\alpha = \frac{1}{100}$, $f = \frac{1}{20}$, $A = 1$, $\phi = 0$) and 150 samples representing the x axis. Using

these parameters, we obtained a noisy, patterned signal that we used to test the validity of our algorithm, with a known attenuation parameter.
Table 2 shows the peaks ($P_{2..8}$) distance (d) between the noisy and original signals, as well as the snr in dB. The signal amplitude, with 7 peaks (counting from the second one), for 150 samples, illustrates approximately 6,000 ms of walking with a sampling rate of 25 Hz (i.e., 1 sample = 40ms). Figure 4 shows that the difference between the peaks detected in the original signal and the noisy signal is acceptable. The d_y column shows the height difference between the peaks, while the d_x column shows the time difference between the peaks in seconds.

TABLE 2 – Peaks ($P_{2..8}$) Distance (d) Between Noisy and the Original Signal

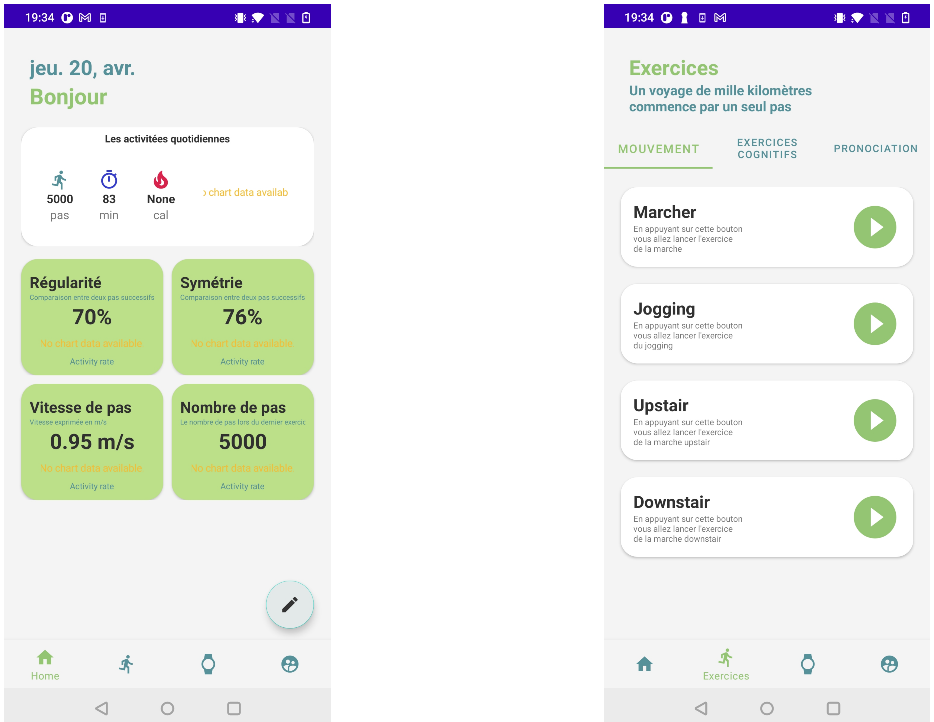
	P_2	P_3	P_4	P_5	P_6	P_7	P_8
snr (dB)	9.281	8.70	7.99	7.10	6.20	5.94	5.88
$d_y(height)$	0.129	0.027	0.042	0.144	0.099	0.164	0.098
$d_x(seconds)$	0.04	0.04	0.04	0.0	0.0	0.04	0.04

By analyzing the data presented in Table 2, we can conclude that the discrepancy between the detected peaks in the original signal and the noisy signal is within an acceptable range. Specifically, the performance fluctuates according to the peaks number and tends to deteriorate for the lowest peaks due to a lower SNR.

C. Dashboards/GUIs

Bibliography

1. Benson (L. C.), Clermont (C. A.), Watari (R.), Exley (T.) et Ferber (R.). – Automated accelerometer-based gait event detection during multiple running conditions. *Sensors*, vol. 19, n7, 2019, p. 1483.
2. Boutet (A.), Frindel (C.), Gambs (S.), Jourdan (T.) et Ngueveu (R. C.). – Dysan : Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks. – In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, ASIA CCS '21*, ASIA CCS '21, p. 672–686, New York, NY, USA, 2021. Association for Computing Machinery.
3. Boutet (A.), Frindel (C.), Gambs (S.), Jourdan (T.) et Ngueveu (R. C.). – DYSAN : Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks. – In *ACM ASIACCS 2021 - 16th ACM ASIA Conference on Computer and Communications Security*, Hong Kong (Virtuel), China, juin 2021.
4. Brajdic (A.) et Harle (R.). – Walk detection and step counting on unconstrained smartphones. – In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 225–234, 2013.
5. de Oliveira Gondim (I. T. G.), de Souza (C. d. C. B.), Rodrigues (M. A. B.), Azevedo (I. M.), de Sales (M. d. G. W.), Lins (O. G.) et al. – Portable accelerometers for the evaluation of spatio-temporal gait parameters in people with parkinson's disease : an integrative review. *Archives of Gerontology and Geriatrics*, vol. 90, 2020, p. 104097.
6. Dogra (S.), Dunstan (D. W.), Sugiyama (T.), Stathi (A.), Gardiner (P. A.) et Owen (N.). – Active aging and public health : Evidence, implications, and opportunities. *Annual review of public health*, vol. 43, 2022, pp. 439–459.



(a) Patient's customizable Dashboard (b) Patient's movement exercises

FIGURE 5 – Patient's level GUIs

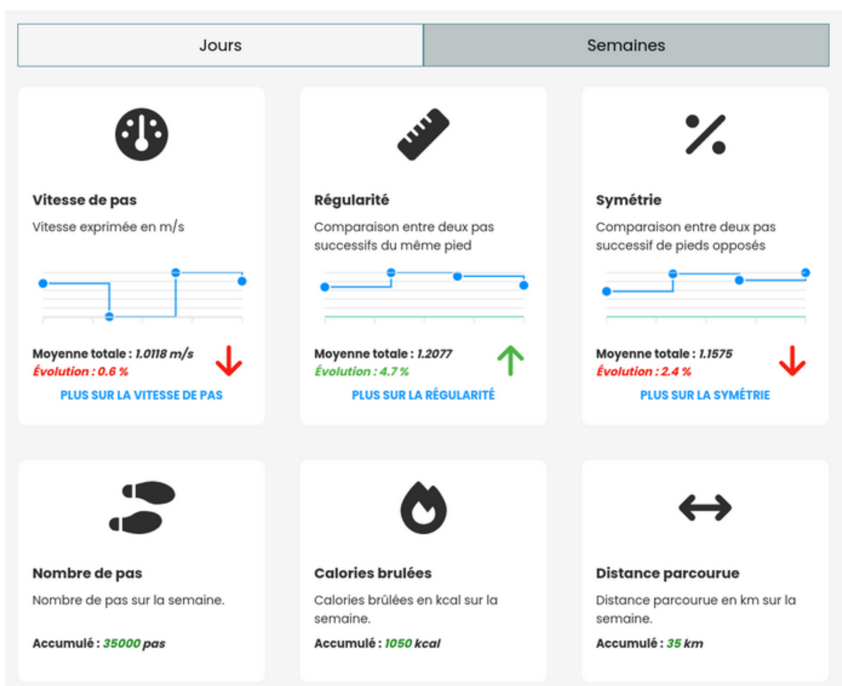


FIGURE 6 – Practitioner's dashboard : visualization of the evolution of the activity of the patients

7. Ellis (R. J.), Ng (Y. S.), Zhu (S.), Tan (D. M.), Anderson (B.), Schlaug (G.) et Wang (Y.). – A validated smartphone-based assessment of gait and gait variability in parkinson's disease. *PLoS one*, vol. 10, n10, 2015, p. e0141694.
8. Gard (P.), Lalanne (L.), Ambourg (A.), Rousseau (D.), Lesueur (F.) et Frindel (C.). – A secured smartphone-based architecture for prolonged monitoring of neurological gait. – In *Internet of Things (IoT) Technologies for HealthCare : 4th International Conference, HealthyIoT 2017, Angers, France, October 24-25, 2017, Proceedings 4*, pp. 3–9. Springer, 2018.
9. Gurchiek (R. D.), Choquette (R. H.), Beynnon (B. D.), Slauterbeck (J. R.), Tourville (T. W.), Toth (M. J.) et McGinnis (R. S.). – Open-source remote gait analysis : A post-surgery patient monitoring application. *Scientific reports*, vol. 9, n1, 2019, p. 17966.
10. Jourdan (T.), Boutet (A.), Bahi (A.) et Frindel (C.). – Privacy-Preserving IoT framework for activity recognition in personal healthcare monitoring. *ACM Transactions on Computing for Healthcare*, vol. 2, n1, novembre 2020, pp. 1–22.
11. Jourdan (T.), Boutet (A.), Bahi (A.) et Frindel (C.). – Privacy-preserving iot framework for activity recognition in personal healthcare monitoring. *ACM Trans. Comput. Healthcare*, vol. 2, n1, dec 2021.
12. Lăpădat (A. M.), Tiberius (C. C.) et Teunissen (P. J.). – Experimental evaluation of smartphone accelerometer and low-cost dual frequency gnss sensors for deformation monitoring. *Sensors*, vol. 21, n23, 2021, p. 7946.
13. Rougé (P.), Moukadem (A.), Dieterlen (A.), Boutet (A.) et Frindel (C.). – Generalizable features for anonymizing motion signals based on the zeros of the short-time fourier transform. *Journal of Signal Processing Systems*, vol. 95, n1, 2023, pp. 89–99.
14. Sankoff (D.) et Kruskal (J. B.). – Time warps, string edits, and macromolecules : the theory and practice of sequence comparison. *Reading : Addison-Wesley Publication*, 1983.
15. Sharif Bidabadi (S.), Murray (I.) et Lee (G. Y. F.). – Validation of foot pitch angle estimation using inertial measurement unit against marker-based optical 3d motion capture system. *Biomedical engineering letters*, vol. 8, 2018, pp. 283–290.
16. Yousufuddin (M.) et Young (N.). – Aging and ischemic stroke. *Aging (Albany NY)*, vol. 11, n 9, 2019, p. 2542.
17. Zijlstra (W.) et Hof (A. L.). – Assessment of spatio-temporal gait parameters from trunk accelerations during human walking. *Gait & posture*, vol. 18, n2, 2003, pp. 1–10.