A Smartphone-based Architecture for Prolonged Monitoring of Gait

Louise Bart, El Amine Bechorfa, Antoine Boutet, Jan Ramon, Carole Frindel

A Smartphone-based Architecture for Prolonged Monitoring of Gait

Louise Bart
Univ Lyon, INSA Lyon
CREATIS, Inserm
Lyon, France
louise.bart
@insa-lyon.fr

El Amine Bechorfa
Univ Lyon, INSA Lyon
Inria
Lyon, France
mohamed.bechorfa
@inria.fr

Antoine Boutet
Univ Lyon, INSA Lyon
Inria, CITI
Lyon, France
antoine.boutet
@insa-lyon.fr

Jan Ramon
Inria
Lille, France
jan.ramon
@inria.fr

Carole Frindel
Univ Lyon, INSA Lyon
CREATIS, Inserm
Lyon, France
carole.frindel
@insa-lyon.fr

***Abstract*—Gait analysis is important for evaluating neurological disorders such as stroke and Parkinson's disease. Traditionally, healthcare professionals had to rely on subjective assessments (i.e., human-based) of gait which were time consuming and not very reproducible. However, with the advent of IoT and indeed more 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 are collected by a mobile application that continuously captures spatiotemporal parameters in the background using the built-in smartphone accelerometer. Statistical techniques are then applied to extract general indicators about the performed activity, as well as some more specific gait metrics in real-time such as regularity, symmetry and walking speed. These metrics are calculated based on the detected steps while patients are performing an activity. Additionally, a deep learning approach based on an auto-encoder is implemented to detect abnormal activities in the gait of patients. These analyses provides both valuable insights and statistical information about the activities performed by the patient, and a useful tool for practitioners to monitor the progression of neurological disorders and detect anomalies. We conducted experiments using this application in real conditions to monitor post-stroke patients in collaboration with a hospital, demonstrating its ability to compute valuable metrics and detect abnormal events patient's gait.**

***Index Terms*—IoT, machine learning, privacy-by-design, mobile application, auto-encoder, anomaly detection, gait analysis, post-stroke rehabilitation**

I. Introduction

As the population ages, the prevalence of health issues like stroke is increasing [1], [2]. This trend has resulted in a growing demand for continuous monitoring and follow-up of patients. However, existing services face significant challenges. They require substantial human resources, materials, and time, Furthermore, achieving personalized monitoring for individual patients is often difficult, if not impossible. Lastly, 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 to both practitioner and patients while ensuring data privacy.

Recent works have explored the utilization of built-in smartphone sensors for identifying walking, step counting steps and gait analysing using various methods [3] based either on time domain [4] frequency domain [5]. Patient monitoring applications, with a focus on gait analysis, offer a valuable solution for evaluating neurological disorders. They empower practitioners to track the evolution of gait parameters before, during, and after rehabilitation programs. This capacity enables them to assess the effectiveness of interventions and make informed decisions regarding program development. However, concerns related to the intrusive nature of such monitoring and the potential breach of private life have limited the adoption of such solutions [6].

In this paper, we present a privacy-by-design monitoring application for post-stroke patients to evaluate their gait before, during, and after a rehabilitation programs, along with the ability to detect gait anomalies. To achieve this, the mobile application continuously captures spatiotemporal parameters in the background using the built-in smartphone accelerometer. We employ both signal processing and deep learning approaches to extract general and specific indicators related to gait and the performed activity, as well as to detect abnormal activities in the gait of patients.

To address privacy concerns, we ensure that raw data is processed exclusively on the user's device, with only specific gait indicators relevant to practitioners being transmitted to the application server. This minimisation scheme prevents the collection and centralisation of raw data, significantly reducing the risk of data breaches. Moreover, by sharing only a limited set of metrics and indicators, we substantially reduce the potential for inference attacks compared to using raw data [7], [8]. Our solution comprises a mobile application that collects information, calculates indicators, and presents metrics to patient through a customizable dashboard. Additionally, this mobile application securely transmits certain indicators to an application server, facilitating personalized patient monitoring by practitioners.

Furthermore, we conducted real-world experiments to monitor post-stroke patients in collaboration with an hospital. Our findings demonstrate that this mobile application can compute valuable gait metrics for both patients and practitioners, while

minimizing battery consumption.

The paper is organized as follows: we begin by reviewing literature in patient monitoring in Section II before providing an overview of our post-stroke monitoring application in Section III. We then present our results regarding the application's performance in step, activity and anomaly detection, gait metrics and battery consumption in Section IV before concluding in Section V.

II. Background

There are various existing applications designed for patient monitoring, each employing different methodologies.

Zijlstra et al. [9] used peaks detection to extract gait events from trunk accelerations. They identified local maxima and minima in the vertical acceleration signal to determine heel strikes and toe-offs. Additionally, they implemented a time normalization algorithm to create a standardized gait cycle, facilitating the calculation of spatio-temporal gait parameters.

In contrast, Benson et al. [10] proposed a threshold-based approach for detecting initial contact and toe-off events from acceleration signals. They also compared accelerometer-based gait event detection algorithms with force plate-based gold standard methods.

Ellis et al. [11] validated the accuracy of gait analysis using a smartphone-based mobile application for Parkinson's disease patients and healthy elderly individuals. This application included Rhythmic Auditory Cueing, a technique used in gait rehabilitation where patients synchronize their steps to an external auditory cue.

Furthermore, some studies focused on developing comprehensive gait analysis platforms. Gurchiek [12] presents a general approach encompassing activity identification, event detection and analysis, using machine learning to identify activity. Similarly, Gard et al. [13] have designed a versatile application that goes beyond conventional gait analysis. This application is capable of detecting conditions like intermittent claudication and gait asymmetry by computing parameters such as duty cycle and the ratio between odd and even peaks of autocorrelation from the vertical accelerometer signal.

In the field of wearable sensors and IoT-based monitoring applications, there are examples such as MSCopilot[1], designed for monitoring multiple sclerosis. It allows practitioners to assess patient progression at home and between appointments, facilitating data sharing with healthcare providers. However, it is essential to highlight that as patient data collection becomes more prevalent, privacy and security concerns come to the forefront. The collection of sensitive motion data, even for legitimate healthcare purposes, can potentially raise privacy concerns, including the risk of unauthorized access and misuse of personal information. Therefore, addressing these privacy and security issues becomes paramount in the development and adoption of patient monitoring solutions.

To automatically detect gait anomalies, Gerazov et al. [14] proposed a signal processing-based approach that identifies repeated and unusual patterns in time series. Specifically, this approach calculated distances between segments in time series locally and identifies significant patterns, enabling adaptation to each patient's gait.

Conversely, our solution relies on a deep learning approach, utilizing an auto-encoder to learn an individual's gait and its evolution over time, thereby facilitating the detection of abnormal activities.

III. Post-stroke Monitoring App

Our application has been meticulously designed to meet the requirements of a real-world medico-social use case with ARRPAC[2]. ARRPAC is a medico-social center part of the Hospices Civils de Lyon (HCL) that provides support for individuals dealing with with post-stroke disorders. ARRPAC's mission is to address the medical and social needs of their patients. Our application, in this context, serves as a tool to enhance the personalization and efficacy of the support they offer. Through our partnering with ARRPAC, our objective is to demonstrate the potential of our solution in enhancing the delivery of medico-social services.

However, it is imperative to acknowledge that while there is a wealth of insights regarding the requirements of such a solution, we are confronted with several significant challenges. One of the primary challenges entails the necessity for continuous patient monitoring, which requires ongoing data processing. Consequently, we must develop sensor-probing strategies to minimize battery consumption. Another challenge is preserving patient privacy, as the acceptance and adoption of these technologies hinge on patients being assured of the preservation of their privacy.

Furthermore, a flexible presentation dashboard is required for both practitioners and patients. This dashboard should be user-friendly, facilitating the clear and comprehensible visualization of important information.

In the subsequent sections, we provide an overview of our post-stroke monitoring application. We begin by describing the application's architecture in section III-A. We then explain data collection in section III-B, followed by a description of the metrics and the dashboard in sections III-D and III-F, respectively. Finally, we provide more detailed information on gait parameter extraction and algorithm validation in section III-C, and present the implementation specifics in section III-G.

A. Architecture

The architecture of our privacy-preserving framework comprises three main elements, as illustrated in Figure 1: a client application running on the user's smartphone, the application server, and the hospital practitioner. In order to minimize the exposition of sensitive information, raw data remains exclusively on the smartphone and is not transmitted to the server. Instead, this data is processed locally on the smartphone, and only specific metrics and gait indicators are forwarded to the application server.

[1]https://www.mscopilot.com/

[2]https://gcsms-arrpac.fr/

This minimisation approach significantly reduces the potential for inference attacks that could be conducted by adversaries on stored data [7], [8] (i.e., from few metrics compared to raw data). In addition, patient metrics can be encrypted before being uploaded and stored in the server's database.

Subsequently, the practitioner can request access to the application server (and decrypt the stored information, if encrypted) to monitor patients through a dashboard implemented as a website. This dashboard provides a clear and comprehensible representation of the patient results and progression.

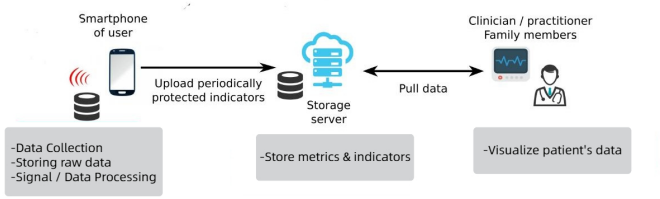


Fig. 1: Global application architecture

B. Data Collection

Data collection is a continuous process that runs in the background with a lightweight approach to minimize its impact on battery consumption. To achieve that, the application initially collects data at a low sampling frequency until the detection of an activity triggers an increase in the sampling frequency.

In order to detect activity, the application continuously samples the total acceleration of the subject along three axes of the accelerometer. This is done at regular intervals and involves computing the total intensity of motion using the magnitude formula, $|\mathbf{A}| = \sqrt{x^2 + y^2 + z^2}$. The calculated magnitude provides a single value that represents the overall intensity of motion at each measurement interval.

Our approach involves comparing this measured magnitude to a statistical threshold. Specifically, we calculate the average magnitude over a set number of previous measurements (e.g., over the last minute) and then compute the standard deviation (std) of these measurements. By doing so, we obtain an estimate of the typical variation in the measured magnitude. To identify periods of movement, we consider the magnitude of the current measurement and compare it to the average magnitude. If the current magnitude exceeds the average by a certain number of standard deviations, it is classified as a period of movement. Conversely, if it falls below this threshold, it is considered a period of relative inactivity.

After detecting movement, the application increases the sampling frequency and applies pattern recognition techniques to enhance the accuracy of identifying walking activity. In our approach, we have also established an additional criterion, a minimum duration threshold. This threshold specifies the minimum length of time a patterned signal must persist to be classified as walking activity. It helps to distinguish genuine walking from momentary fluctuations.

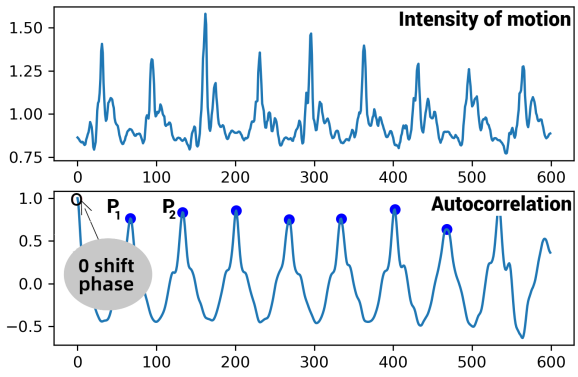


Fig. 2: Accelerometer magnitude $|\mathbf{A}|$ and the corresponding ACF signal

C. Steps Detection

Given that human gait constitutes a periodic motion, we derive metrics using the autocorrelation function (also noted ACF) of the series representing the total intensity of motion. The autocorrelation quantifies the correlation between a value at time t_i and t_{i+1} within the same signal. It is noteworthy that the autocorrelation values exhibit peaks or local maxima with each full step. Equation 1 illustrates a normalized and unbiased autocorrelation function:

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

where, k represents the time shift, x_i denotes the gait data at time i, μ is the mean of the gait data, N along with σ^2 denote the length and variance of the gait data, respectively. The ACF(k) value denotes the similarity between the gait data at time i and the gait data at time $i + k$, based on the time shift k. To normalize the values between the range $[-1; 1]$, we divide by σ^2. To prevent attenuation of the autocorrelation signal, we divide by $(N - k)$ instead of N, ensuring that the numerator's number of terms always matches the denominator's $(N-k)$. Figure 2 illustrates the intensity of motion and its corresponding autocorrelation function where peaks correspond to steps.

To automatically identify peaks (i.e., the steps), we detect the maximum value of the ACF signal centered in a time window, named peak detection interval Δ. This interval Δ is manually established during a calibration step when first launching the application. Compared to using mathematical zero-derivative-based methods or the second derivative sign-changing technique, the adopted approach discards meaningless peaks in case of the presence of noise in the ACF signal.

D. Metrics

We provide two distinct categories of metrics within our application. The first category pertains to physical indicators

intended for patients and their families. These metrics encompass the monitoring of daily activities, including the count of steps and the classification of activity types as static or dynamic [3].

The second category focuses on measurements used to quantify different aspects of human gait, tailored for hospital practitioners engaged in gait analysis. These metrics encompass cadence, regularity, symmetry and step speed. To compute these metrics, we employ an autocorrelation function (as explained in section III-C), which allows us to extract many essential gait parameters.

As illustrated in Figure 2, the initial peak of the autocorrelation signal P_1 represents the correlation between adjacent steps, while the second peak P_2 corresponds to the correlation between two consecutive strides. Leveraging this information, we deduce that the ratio between two consecutive peaks (e.g., P_1/P_2) and two non-consecutive peaks represent symmetry and regularity, respectively.

While the time required to complete a full step is symbolized by the distance between two consecutive peaks on the x-axis, we can then calculate the speed of a step (s) defined by $s = \frac{l}{P_{n+1} - P_n}$ by taking approximations of step length (l) based on the age of the user [4].

E. Anomaly detection

The detection of abnormal gait activities in patients is important, as these can often serve as early indicators of a deterioration in their condition. Promptly reporting this information to practitioners is crucial for providing better care. Anomalies in gait can manifest in various forms: 1) gait variability (stroke can result in motor control loss and reduced coordination), 2) altered gait timing (delays or early occurrences in different phases of the gait cycle) and 3) postural instability (struggle to maintain balance while walking).

Our approach hinges on the utilization of an auto-encoder. An auto-encoder is a type of artificial neural network employed for unsupervised learning. It comprises an encoder and a decoder. The encoder compresses input data into a lower-dimensional representation known as the latent space. Conversely, the decoder reconstructs the original data from the latent space. The primary objective of an auto-encoder is to minimize the disparity between the input and the output, compelling the model to acquire a concise representation of the data.

We train an auto-encoder locally on the smartphone to learn the patient's normal gait. Subsequently, we leverage this model to identify anomalies. Specifically, the trained model excels at accurately reconstructing signals associated with normal walking but struggles to correctly reconstruct abnormal data or data containing anomalies. Therefore, we employ the reconstruction error, equivalent to the training loss of the model, as an indicator for anomaly detection.

[3]We omit to specify these indicators for space reason.

[4]Low Pace: young adults (60-70 cm), older adults (50-60 cm), and Medium Pace: young adults (70-80 cm), older adults (60-70 cm).

To effectively identify anomalies, we introduce a threshold on the loss. This threshold is derived from the range of scores assigned by our model and is calculated as follows (Equation 2):

$$threshold = \mu(loss) + 2 * \sigma(loss), \tag{2}$$

where μ is the mean function and σ the standard deviation.

The threshold is established following the training step. The data is cut in time segments and a segment is identified as abnormal if the loss is greater than the threshold.

Fig. 3: Patient's customizable dashboard

F. Dashboards

We have developed dashboards to present metrics to both patients (through the smartphone application) and practitioners (through a web interface).

The patient dashboard (Figure 3) is highly customizable, with an extensive library of metrics and daily activity indicators to choose from. Patients can select the specific indicators they wish and arrange them in their preferred order.

Conversely, the second dashboard (Figure 4), designed for practitioners, is dedicated to gait-related metrics. It provides in depth analysis of walk dynamics, regularity, and symmetry. This user-friendly dashboard affords significant level of flexibility, allowing practitioners to visualize results over various time frames, such as the last month or week. By providing practitioners with these gait-related metrics and their longitudinal trends, our application aims to enhance diagnosis accuracy and treatment effectiveness.

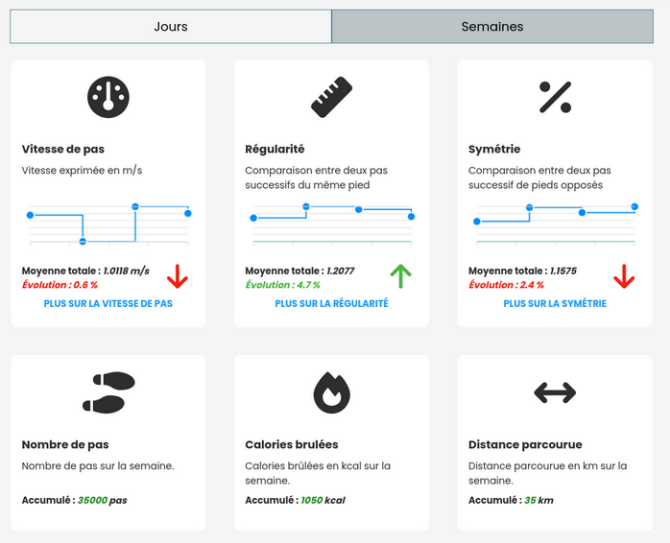


Fig. 4: Practitioner's dashboard: visualization of the patient's activities evolution

G. 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.

IV. Experiments

We have implemented and evaluated our monitoring solution. This section reports the main results on activity and step detection (Section IV-A), gaits metrics (Section IV-B), anomaly detection (Section IV-C), and battery consumption (Section IV-D).

A. Activity and Step Detection

In this section, we evaluate the capacity of our application to detect a dynamic activity and to detect a step. Firstly, we examined the process of differentiating between static and dynamic activity types using the publicly available MotionSense[5] dataset [15]. 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 classified these activities into two distinct groups: static and dynamic, and proceeded to test our activity detection method.

[5]https://github.com/mmalekzadeh/motion-sense

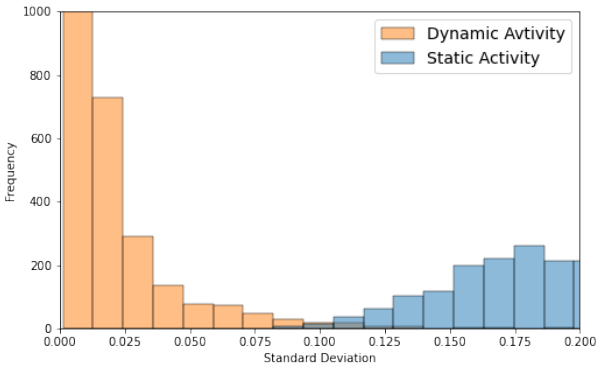


Fig. 5: Distribution of standard deviation (std) of magnitude $|\mathbf{A}|$ of the accelerometer signal

To accomplish this, we computed the standard deviation (std) of the magnitude of our data $|\mathbf{A}|$, and the resulting distribution is illustrated in Figure 5.

Table I then reports the precision (i.e., an accuracy assessment) and the recall (i.e., a completeness assessment) of the two detection tasks conducted on the signals. The first task involves identifying instances of dynamic activity, which is achieved through the utilization of a threshold on the standard deviation as described in Section III-B. The second task focuses on detecting individual steps, as outlined in Section III-C. For each task, we conducted experiments using various combinations of parameter values to gauge the sensitivity of our method to these parameters and pinpoint the most effective settings.

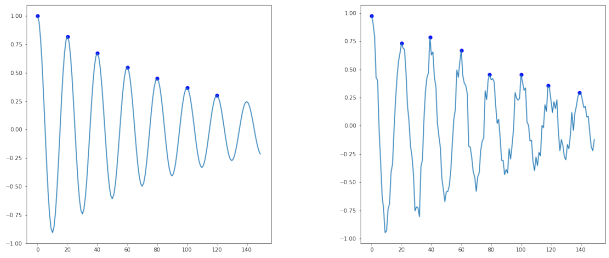
It is worth noting that we processed windows consisting of 150 samples, with a frequency rate of 50 Hz, corresponding to 3-seconds intervals. In the first task, we examined two distinct standard deviation (std) threshold values, with the goal of optimizing the precision and the recall of the prediction. In the second task, we established a peak detection interval (Δ) manually. The criterion for recognizing walking activity was the detection of at least three peaks within a given window of data.

Results indicate that a smaller std threshold for discriminating the nature of the activity and detect the dynamic activity leads to an ideal precision but exhibits a poor recall. Opting for a higher std threshold results in a very significant improvement of recall but slightly reduces the precision leading to wrongly classified data as dynamic activity. This miss-classification also incurs a greater computational cost and increased battery consumption, as the sampling frequency is increased to better analyse the signals for dynamic activities. Nevertheless, results report that the calibration of our thresholds yield to correctly detect both dynamic activities and the steps.

To better assess and validate the accuracy of our step detection, we employed a damped cosine function denoted as $y(t)$ to roughly simulate the autocorrelation function (ACF)

TABLE I: Effect of algorithm parameter variations on accuracy and precision in dynamic/static and step detection

Detection task	Thresholds	TP	FP	TN	FN	Precision	Recall
Dynamic activity	std = 0.38	1198	0	4357	3995	1.0	0.230
	std = 0.1	5176	50	4307	17	**0.990**	**0.996**
Steps	$\Delta = 0.6$, std= 0.38	1159	0	0	39	**1.0**	**0.967**
	$\Delta = 0.6$, std = 0.1	5007	4	46	169	**0.999**	**0.967**



(a) Original ACF signal (b) Noisy ACF signals

Fig. 6: Peaks detection in ACF signals

of the magnitude of the motion, which is defined as follows (Equation 3):

$$y(t) = Ae^{-\alpha t}\cos(2\pi ft + \phi), \tag{3}$$

with manually defined parameters and a specified number of samples representing time. The variable $y(t)$ depends on time t and is determined by the amplitude A, a decay constant α, the frequency f, and the phase angle ϕ of the cosine wave.

Figure 6(a) depicts the associated autocorrelation function (ACF). As shown on the figure, adding a decay constant makes it possible to evaluate thresholding in peak detection (i.e., the last oscillation of the signal is not detected as a peak).

To assess that we detect all the peaks of the signal, we calculate the constancy of the decay rate α on each pair of consecutive peaks detected, denoted P_k and P_{k+1} (Equation 4). We then validate that all peaks are correctly detected as these estimates correspond to the true value of α.

$$\alpha(t) = \ln(P_k/P_{k+1})/(t_{k+1} - t_k) \tag{4}$$

After adding random noise N to our original ACF signal, drawn from a Gaussian probability distribution with mean μ and standard deviation σ (Figure 6(b)), we calculated the Signal-to-Noise Ratio (SNR) in decibels (dB) for each peak A as follows (Equation 5):

$$SNR = 10\log_{10}(\frac{A}{\sigma}). \tag{5}$$

It is worth noting that, for this specific example, we kept the noise parameters fixed at $\sigma = 0.1$ and $\mu = 0$, as specified in [16], which experimentally evaluated instrumental noise on sensors. Additionally, we utilized the parameters from the original damped cosine function, including $\alpha = \frac{1}{100}$, $f = \frac{1}{20}$, $A = 1$, $\phi = 0$, and incorporated 150 samples representing the x-axis. With these settings, we generated a noisy, patterned signal that served as a test for the validity of our algorithm.

Table II provides both the SNR in dB and the distance (d) between the noisy and original ACF signals on the peaks ($P_{2..8}$). The ACF signal, which includes 7 peaks (counting from the second one), spans 150 samples, representing approximately 6,000 milliseconds of walking with a sampling rate of 25 Hz (equivalent to 1 sample every 40 milliseconds). The d_y column indicates the height difference between the peaks, while the d_x column represents the time difference between the peaks in seconds. Upon analyzing the data presented in Table II, we show that the variation between the detected peaks in the original signal and the noisy signal falls within an acceptable range. Figure 6(b) visually demonstrates that the difference between the peaks detected in the original and the noisy ACF signal remains acceptable.

B. Gait metrics

We conducted a real-life data collection experiment in which we simulated pathological walking using a splint. We instructed the participant to hold the smartphone naturally in their hand while walking in a straight line, both with and without the splint, at two different paces - a slow pace and a normal pace. Figure 7 and 8 depict the differences in regularity and symmetry between normal and pathological walking, while Figure 9 illustrates the speed we calculated for the same subject, with approximate step lengths between 0.65 - 0.75 meter based on age and pace. The results clearly demonstrate that our current pipeline effectively computes precise metrics for both typical and pathological gait patterns. This capability allows us to observe the evolution of regularity, symmetry, and walking speed during a recovery process.

C. Anomaly detection

In this section, we assess our application's capability to detect abnormal activities by setting a threshold on the loss of an auto-encoder trained to learn the patient's gait. Given the limited availability of anomaly data, we aim to evaluate our ability to distinguish between walking and running activities based on the loss metric.

Figure 10 illustrates the distribution of the auto-encoder loss for walking and running activities, using data from the MotionSense dataset. The results reveal distinct loss distributions for these two activities, allowing them to be clearly distinguished by the auto-encoder.

In order to test our anomaly detection in more realistic conditions, we collected gait data from four individuals with

TABLE II: Peak detection: SNR of the noisy ACF and distances between noisy and original ACF signals for peaks ($P_{2..8}$)

	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

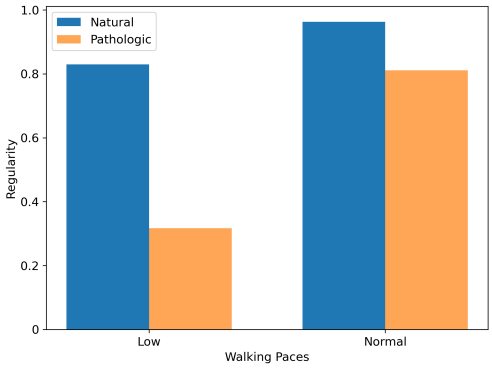


Fig. 7: Comparison of gait regularity between natural and pathological gait.

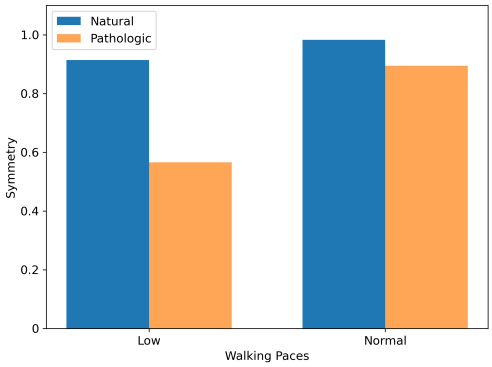


Fig. 8: Comparison of gait symmetry between natural and pathological gait.

Fig. 9: Difference in walking speed between natural and pathological gait with different paces.

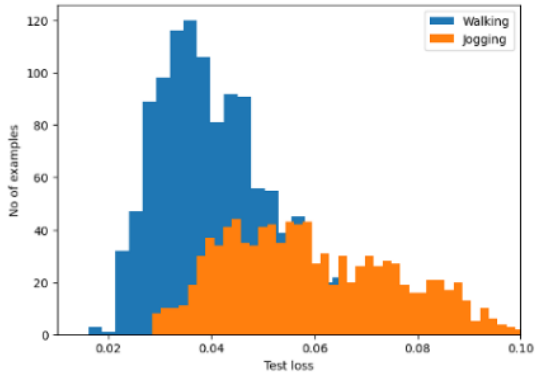


Fig. 10: Distribution of the loss of the auto-encoder for walking and running activities (MotionSense dataset)

and without the use of an ankle brace. We then cut the data into segments and identified the segments detected as an anomaly. The table III reports the ratio of segments detected as an anomaly on our normal (i.e., without splint) and abnormal (i.e., with splint) data. We considered two auto-encoder models, model A corresponds to a model trained on walking data from the MotionSense dataset, and model B which corresponds to model A re-trained with our normal data. Finally, we also compared the use of our threshold (denoted T2 from Equation 2) with another threshold T1 defined as $T1 = \mu(loss) + \sigma(loss)$.

First, results show that training the auto-encoder model on user data gives a higher anomaly detection rate than a model trained on data from a public dataset (approximately 40% more segments detected). Second, our threshold perfectly discriminates normal gait segments (i.e. 0% of segments detected) compared to T1 which falsely identifies more than 20% of abnormal segments.

D. Battery consumption

In this section, we assess the effect of the monitoring application on battery consumption.

Figure 11 illustrates the battery percentage evolution of a smartphone under two conditions: without monitoring (our baseline) and with a monitoring application that utilizes two different sampling frequencies (i.e., low and high frequencies) managed by SensorEventListener [6].

[6]SensorEventListener is a component of the Android Sensor framework, allowing Android applications to receive sensor events from various sensors on the device.

TABLE III: Detection of anomalies in normal and abnormal gait with different thresholds and models

	Thresholds	Normal Gait	Abnormal Gait
Model A	T1	21.4%	61.5%
	T2	**0%**	**30.7%**
Model B	T1	21.42%	92.30%
	T2	**0%**	**69.23%**

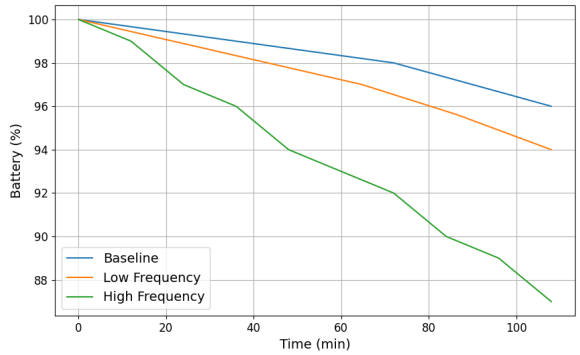


Fig. 11: Impact of active data collection on battery consumption

The results indicate that the application's impact on battery consumption is relatively modest. After approximately two hours of monitoring, using a low-frequency sampling rate results in only a 1% increase in battery usage compared to the baseline, while using a high-frequency sampling rate leads to a 5% increase. It is important to note that the high-frequency consumption observed here represents an upper bound, as this frequency (and associated processing) was enforced during the experiment. In typical usage, a high sampling frequency is employed only when an activity is detected (as described in Section III-B).

V. Conclusion and Future Work

The use of IoT-based objective methods has led to significant advancements in assessing neurological disorders. However, the field faces notable challenges related to privacy and computational limitations. In response, this paper introduces a novel solution – a privacy-focused monitoring application designed for post-stroke patients. This app utilizes smartphone accelerometers to gather spatiotemporal data and generate real-time gait metrics, offering valuable insights into patient monitoring and gait analysis. This has the potential to enhance the quality of care for stroke recovery patients.

The future directions include integrating more gait analysis metrics and parameters deemed important by experts to provide a comprehensive view of a patient's condition, enabling more precise diagnoses and tailored treatments. Additionally, the study hints at exploring federated learning, a decentralized approach to data analysis, to enhance activity type detection while safeguarding privacy.

References

[1] M. Yousufuddin and N. Young, "Aging and ischemic stroke," *Aging (Albany NY)*, vol. 11, no. 9, p. 2542, 2019.

[2] S. Dogra, D. W. Dunstan, T. Sugiyama, A. Stathi, P. A. Gardiner, and N. Owen, "Active aging and public health: Evidence, implications, and opportunities," *Annual review of public health*, vol. 43, pp. 439–459, 2022.

[3] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013, pp. 225–234.

[4] D. Sankoff and J. B. Kruskal, "Time warps, string edits, and macromolecules: the theory and practice of sequence comparison," *Reading: Addison-Wesley Publication*, 1983.

[5] P. Rougé, A. Moukadem, A. Dieterlen, A. Boutet, and C. Frindel, "Generalizable features for anonymizing motion signals based on the zeros of the short-time fourier transform," *Journal of Signal Processing Systems*, vol. 95, no. 1, pp. 89–99, 2023.

[6] F. Schuster, "Security and privacy concerns for iot adoption in the home domain: A user perspective," 2022.

[7] T. Jourdan, A. Boutet, A. Bahi, and C. Frindel, "Privacy-Preserving IoT framework for activity recognition in personal healthcare monitoring," *ACM Transactions on Computing for Healthcare*, vol. 2, no. 1, pp. 1–22, Nov. 2020. [Online]. Available: https://inria.hal.science/hal-03045108

[8] A. Boutet, C. Frindel, S. Gambs, T. Jourdan, and R. C. Ngueveu, "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, Jun. 2021. [Online]. Available: https://inria.hal.science/hal-02512640

[9] W. Zijlstra and A. L. Hof, "Assessment of spatio-temporal gait parameters from trunk accelerations during human walking," *Gait & posture*, vol. 18, no. 2, pp. 1–10, 2003.

[10] L. C. Benson, C. A. Clermont, R. Watari, T. Exley, and R. Ferber, "Automated accelerometer-based gait event detection during multiple running conditions," *Sensors*, vol. 19, no. 7, p. 1483, 2019.

[11] R. J. Ellis, Y. S. Ng, S. Zhu, D. M. Tan, B. Anderson, G. Schlaug, and Y. Wang, "A validated smartphone-based assessment of gait and gait variability in parkinson's disease," *PLoS one*, vol. 10, no. 10, p. e0141694, 2015.

[12] R. D. Gurchiek, R. H. Choquette, B. D. Beynnon, J. R. Slauterbeck, T. W. Tourville, M. J. Toth, and R. S. McGinnis, "Open-source remote gait analysis: A post-surgery patient monitoring application," *Scientific reports*, vol. 9, no. 1, p. 17966, 2019.

[13] P. Gard, L. Lalanne, A. Ambourg, D. Rousseau, F. Lesueur, and C. Frindel, "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*. Springer, 2018, pp. 3–9.

[14] B. Gerazov, E. Hadzieva, A. Krivosei, F. I. S. Sanchez, J. Rostovski, A. Kuusik, and M. Alam, "Matrix profile based anomaly detection in streaming gait data for fall prevention," 2023.

[15] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Privacy and utility preserving sensor-data transformations," *Pervasive and Mobile Computing*, vol. 63, p. 101132, 2020.

[16] A. M. Lăpădat, C. C. Tiberius, and P. J. Teunissen, "Experimental evaluation of smartphone accelerometer and low-cost dual frequency gnss sensors for deformation monitoring," *Sensors*, vol. 21, no. 23, p. 7946, 2021.