

Extracting and Processing Raw PPG Signals from Consumer Smartwatches for Real-Time Atrial Fibrillation Screening

Lely A. Luengas-Contreras^{1,*}, Esperanza Camargo-Casallas¹, Nancy Olarte², Jonathan Torres¹ and Miguel Angel Padilla³

¹Universidad Distrital Francisco José de Caldas, Bogota, Colombia

²Universidad Militar Nueva Granada, Bogota, Colombia

³Universidad Nacional Autónoma de México, Mexico D.F., Mexico

Abstract

Atrial fibrillation (AF) represents the most prevalent sustained arrhythmia and a primary antecedent of stroke and sudden cardiac death. It is also challenging to identify in a clinic due to its intermittent nature. Photoplethysmography (PPG) sensors integrated into consumer smartwatches could enable continuous screening; however, most manufacturers provide only derived metrics such as heart rate, and the deep learning models reported in the literature are typically validated on stored datasets rather than on live-captured signals. We developed a communication interface that retrieves raw PPG samples from wearables with diverse hardware, streams them to an Android application, processes them in real time, and supplies them to pre-trained AF detection models. Two data acquisition methods were employed: the Samsung Health Sensor SDK on Wear OS, which sampled the green channel at 100Hz, and a direct BLE GATT connection to InfiniTime firmware devices at 25Hz. Both streams are converted into a single internal format and processed through DC removal, band-pass filtering, initial sample discarding, and a median/MAD normalization chain. AF-flagged episodes are encrypted with AES-256 GCM using keys stored in the Android Keystore, and the app is protected by biometric authentication. Tests on both device families demonstrated continuous data acquisition, stable plotting, and real-time inference. The system is intended for research and screening purposes, not as a medical device.

Keywords

Photoplethysmography, Atrial fibrillation, Smartwatch, Bluetooth Low Energy, Wearable devices, Signal preprocessing, Mobile health, Data encryption

1. Introduction

Cardiovascular disease causes about 17.9 million deaths worldwide each year, and a large share of these fatalities occur before age 70 [1]. Roughly half of these patients experience sudden cardiac death [2, 3]. A study conducted in several Colombian clinics found that, of 331 patients with AF, 93.96% had more than one associated comorbidity, mainly high blood pressure (74.32%), chronic heart failure (53.1%), coronary artery disease (30.21%) and chronic kidney disease (26.28%) [4]. Atrial fibrillation (AF) is one of the arrhythmias most closely associated with these outcomes: the atria cease to contract in an organized fashion, leading to blood stasis and a higher likelihood of clot formation. Because AF often occurs in a paroxysmal manner, a standard ECG during a clinic visit may miss it. This limitation has driven research toward extended, unobtrusive monitoring that employs signal processing, portable devices, and smartphone applications based on photoplethysmography (PPG) or electrocardiography (ECG).

ICAIFW 2026: Workshops at the 9th International Conference on Applied Informatics 2026, October 13–17, 2026, Orlando, United States

*Corresponding author.

✉ lalenguasc@udistrital.edu.co (L. A. Luengas-Contreras); ecamargoc@udistrital.edu.co (E. Camargo-Casallas); nancy.olarte@unimilitar.edu.co* (N. Olarte); jrortresc@udistrital.edu.co (J. Torres); miguel.padilla@icat.unam.mx (M. A. Padilla)

0000-0002-3600-4666 (L. A. Luengas-Contreras); 0000-0002-6320-4049 (E. Camargo-Casallas); 0000-0001-7029-4006 (N. Olarte); 0000-0002-0335-9639 (J. Torres); 0000-0002-1389-2074 (M. A. Padilla)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Current market applications exhibit two main shortcomings. Many rely on the phone camera and require manual initiation, resulting in intermittent recordings, while others operate exclusively within the ecosystem of a single manufacturer, thereby preventing continuous monitoring or early alerts for the majority of users [5]. Additionally, a methodological gap exists: most published deep learning models are trained and evaluated on pre collected, cleaned datasets, leaving their performance on live streams from commercial wrist devices largely undocumented. The DIGITI research group at Universidad Distrital Francisco José de Caldas has been developing deep learning models that detect atrial fibrillation patterns in photoplethysmography (PPG) data [5]. These models require raw PPG segments rather than the aggregated heart rate values typically reported by watches, and this requirement constitutes the bottleneck addressed in this paper: each manufacturer independently determines whether raw data can be accessed, and devices that permit access employ varying formats, sampling rates, and transport mechanisms.

This paper presents a communication interface that links consumer smartwatches to a mobile processing pipeline and poses the question: How can raw PPG data from smartwatches with varying hardware be extracted, merged into a single stream, conditioned in real time, and supplied to an AF detection model while keeping the underlying biomedical data confidential? The contributions include a survey of raw data access across seven manufacturers, an acquisition and preprocessing pipeline that handles both a proprietary SDK and a direct BLE link identically once the data reaches the phone, the first deployment of the DIGITI models on live data instead of stored data, and an at-rest protection scheme for stored episodes. Table 1 compares the prior work with the proposed approach [5].

2. Theoretical Framework

2.1. Photoplethysmography

This is a non invasive optical method that records pulse waves as they propagate through the body. An LED illuminates a patch of tissue, and a photodetector measures the amount of light that returns, either reflected when the detector is adjacent to the emitter or transmitted when it is positioned on the opposite side. Blood volume variation refers to changes in light absorption within the vessels located just beneath the skin [6].

The intensity can be divided into two parts:

$$x(t) = x_{DC}(t) + x_{AC}(t) \quad (1)$$

The DC term represents the mean blood volume within the measured region plus the absorption from adjacent tissue, whereas the AC term corresponds to the cardiac pulsation, which is the source of heart rate and its variability [7]. Figure 1 displays both components on a real wrist PPG trace. Since atrial fibrillation manifests as an irregular sequence of pulses, this study focuses on the AC component and the waveform shape rather than on absolute amplitude, which explains the design of the conditioning chain in Section 3.4.

2.2. PPG sensors, BLE, and preprocessing

The wrist-worn PPG sensors operate in reflectance mode, where one or more LEDs illuminate the skin and a photodiode records the light that is not absorbed [15]. Green light is less influenced by venous blood; therefore, it typically yields a cleaner wrist signal than red or infrared light. The specific channel that is activated, its sampling frequency, and whether raw samples are exposed are determined solely by the firmware and the manufacturer's permissions, making device selection an engineering decision rather than a simple purchase choice [16].

Bluetooth Low Energy was introduced in version 4.0 of the Bluetooth specification, operates in the 2.4 GHz band with adaptive frequency hopping, and is designed for small packets sent at short intervals using minimal power [17, 18], matching the traffic pattern of a continuous PPG stream. It can also encrypt and authenticate the link, but only when pairing mode is enabled [19].

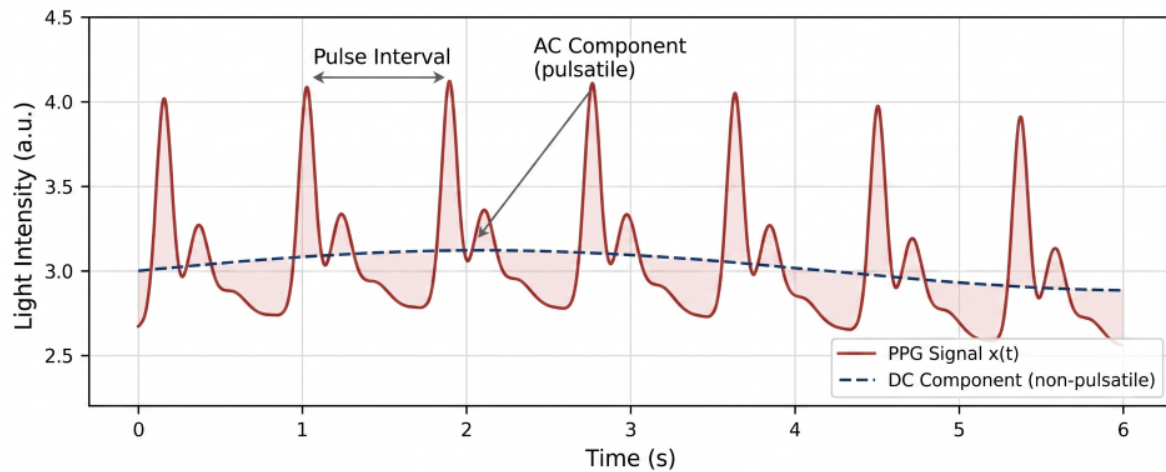


Figure 1: Components of the PPG signal. The dashed line represents the non-pulsatile DC component, and the shaded area represents the pulsatile AC component, from which the pulse interval is derived.

Table 1

Relationship between previous studies and proposed interface.

Reference	Focus	Relationship with the proposal
Alòs and Puig-Ribera [8]	Review of wearables and mHealth apps	Shows the heterogeneity of outcomes in mHealth interventions, motivating the testing of the system under real-world conditions.
López et al. [9]	Psychiatric monitoring wearable via Wi-Fi	Precedes the use of a server/app that receives sensor data and generates alerts.
Wearable B1 (RCS) [10]	EDA and heart rate wearable	Confirms that an in-house prototype can approach the accuracy of a certified device.
Iqbal et al. [11]	Portable telehealth for heart failure	Contributes the app layout featuring separate views for patients and clinical staff.
Shen et al. [12]	AF detection in ambulatory PPG using deep learning	Confirms that PPG is viable for AF screening; the interface developed here delivers the real-time signal.
Kwon et al. [13]	CardioTracker ring with deep learning	Supports the use of 10–30 s windows instead of sample-by-sample analysis.
Bashar et al. [14]	AF detection on smartwatch	Supports the exclusion of unstable initial segments and robust normalization.
Luengas et al. [5]	AF detection from PPG (DIG-ITI)	The interface is the acquisition layer that feeds the models.

Raw PPG samples cannot be used directly from the sensor; they must be cleaned by removing baseline drift, applying band-pass filtering, rejecting motion artifacts, and normalizing amplitude [7]. Normalization is particularly critical because the same processing pipeline must accommodate two devices with markedly different dynamic ranges, and the classifier depends on waveform shape rather than the raw magnitude supplied by a specific sensor. Since a raw PPG trace constitutes sensitive personal data, its protection must be integrated into the design from the outset; BLE encrypts only wireless transmission, so once the data reaches the phone, it still requires access control and encryption at rest [19].

3. Methodology

The system comprises four components (see Figure 2): a wearable device equipped with a PPG sensor, a wireless communication interface, a mobile application, and a signal processing module that prepares the data and executes the DIGITI group-built detection models.

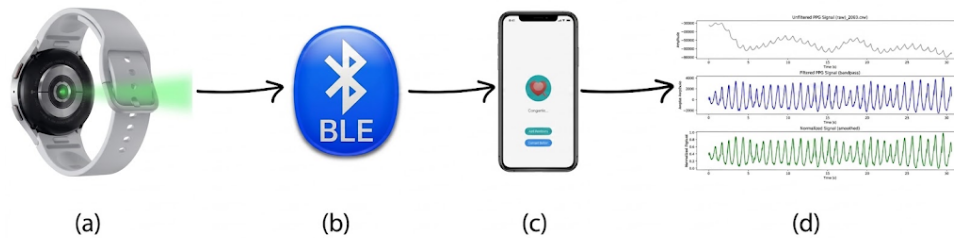


Figure 2: Overview of the proposed system. (a) A wearable device with a PPG sensor. (b) Wireless transmission of the signal over BLE. (c) Reception and management of data using the mobile application. (d) Preprocessing, visualization, and analysis using DIGITI models.

The work went through four stages. First, we survey which wearables actually expose raw PPG samples and determine the transport protocol that most of them share. Second, selecting a development environment and build acquisition for each device family. Third, the incoming streams are merged and conditioned in real time. Fourth, wiring in the detection models and protection mechanisms.

3.1. Protocol, environment, and access across manufacturers

Classic Bluetooth was ruled out early because of power draw, and Wi-Fi was ruled out because it wastes bandwidth on small packets sent continuously. BLE is the most effective in terms of power use, small payloads at short intervals, broad support among wearables, and reliable delivery [20].

Android Studio was the obvious development environment: it is the official Android IDE, documents its BLE APIs, costs nothing to license, and its language, Kotlin, made it easier to integrate two device families with very different access models [19, 20].

Seven manufacturers were checked to determine whether they exposed unprocessed sensor readings, as summarized in Table 2. Hardware is rarely the limiting factor because most current wrist devices carry a PPG sensor. However, the amount of data that reaches a third-party developer varies greatly among brands. Only Samsung devices running Wear OS from the Galaxy Watch 4 onward and the open-source PineTime allowed raw PPG extraction under the conditions of this project, so those two became the reference platforms.

3.2. Acquisition on Samsung and InfiniTime devices

Figure 3 shows both acquisition paths. On Samsung devices, the application first requests biomedical sensor permissions, then initializes the Samsung Health Sensor SDK and configures the sensor to on-demand mode at 100Hz, matching the expected sampling rate of the detection model. We evaluated the three channels of the BioActive sensor, as illustrated in Figure 4: the green channel produced the highest signal-to-noise ratio, whereas the red and infrared channels exhibited a pronounced DC component and drift, and the continuous operation of the green emitter introduced additional noise. Consequently, the green channel was selected, and a listener attached to the tracker relays the sensor values to the phone in real time.

InfiniTime devices lack an SDK, so the connection is managed manually: scan for nearby BLE devices, establish a GATT connection, identify the characteristic that carries the readings, and enable notifications so packets continue to arrive without repeated polling. During development, the connection used the device’s MAC address directly, which is fast; a scan-and-filter approach would be safer in a production build for reliable operation.

Table 2

Availability of raw PPG data for third-party developers by the manufacturer.

Manufacturer	Access to the raw PPG	Implications for the project
Huawei [21]	Granted only to companies after approval.	Not available for an academic project.
Garmin [22]	Not exposed even with the SDK.	Excluded.
Fitbit [23]	Heart and resting heart rates.	Insufficient for detection models.
Xiaomi	No documented health services.	Excluded.
Withings [24]	Devices only available in Europe/US.	Not usable in Colombia.
Samsung [25]	Green/red/infrared channels, free registration.	Selected as the main platform.
PineTime [26]	Direct sensor access over BLE.	Selected as a low-cost alternative.

3.3. Unification of incoming streams

Samsung devices transmit samples in JSON format, which are then converted into a standard structure. InfiniTime devices transmit binary packets that a decoding module transforms into the same structure. Both streams converge in a single shared input channel, the convergence point. This arrangement allows samples to be temporally ordered, buffered into windows required by the detection model, and maintains a separation between acquisition and processing, ensuring that adding a new device requires only a decoder, with no downstream modifications.

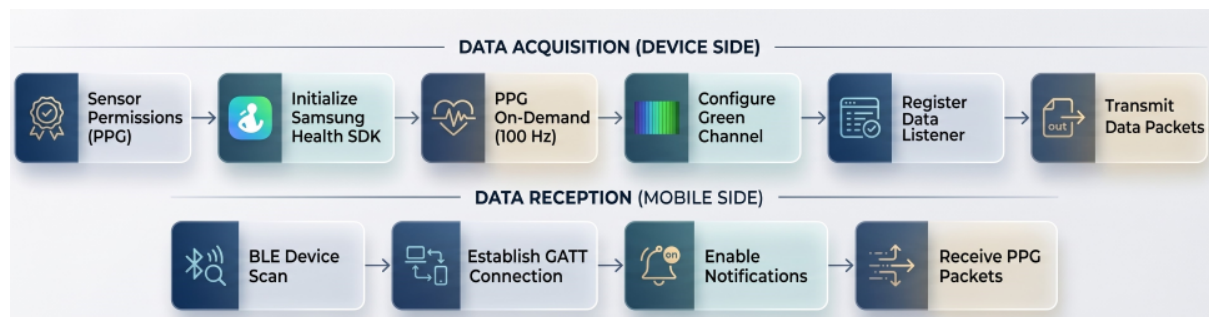


Figure 3: Acquisition of raw data on the wearable device. Top: Samsung Health Sensor SDK. Bottom: direct BLE link on devices running the InfiniTime firmware.

3.4. Real-time preprocessing

The conditioning chain is shown in Figure 5. First, a first-order high-pass filter strips out the DC component:

$$y[n] = \alpha (y[n-1] + x[n] - x[n-1]), \quad \alpha = \frac{1}{1 + 2\pi f_c / f_s} \quad (2)$$

The coefficient α comes from the cut-off frequency and the sampling rate of the source. This step cuts the baseline drift caused by the sensor shifting against the skin, without touching the pulsatile part of the signal.

Next, a band-pass filter, built as a second-order IIR section,

$$y[n] = \sum_{i=0}^2 b_i x[n-i] - \sum_{j=1}^2 a_j y[n-j] \quad (3)$$

with coefficients that depend on the device: 0.5–8 Hz for Samsung at 100 Hz, 0.5–4 Hz for InfiniTime at 25 Hz. Both bands stay inside

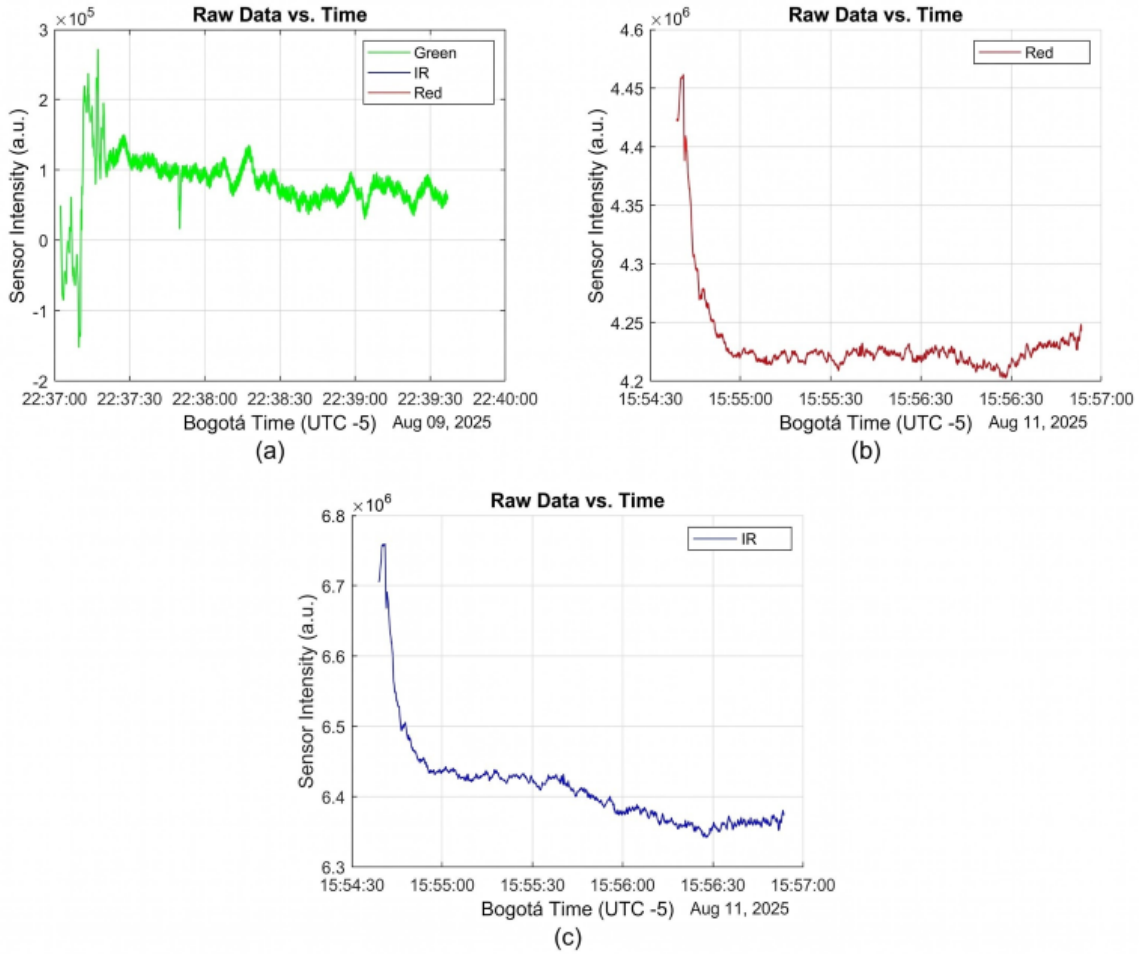


Figure 4: Raw traces of the three channels of the BioActive sensor. (a) Green. (b) Red. (c) Infrared. The red and infrared traces are dominated by the DC component and by a marked drift.

$$f_L \leq f \leq f_H < \frac{f_s}{2} \quad (4)$$

so the physiological content survives without aliasing at the lower rate of the second device.

When a measurement starts, the signal spikes while the sensor and the acquisition pipeline are still settling, so the first samples get dropped

$$N_d = \lceil t_e \cdot f_s \rceil \quad (5)$$

The stabilization time was found by testing and works out to 300 discarded samples on Samsung and 80 on PineTime, both covering roughly the same first couple of seconds.

Last comes normalization, shared by both devices, scaling each sample against the median and the median absolute deviation of a sliding window. For a window, the dispersion around its median is

$$\text{MAD}_w = \text{median}_{i \in w} (|x_i - \tilde{x}_w|) \quad (6)$$

and each sample turns into

$$x_{\text{norm},i} = \frac{x_i - \tilde{x}_w}{\max(\text{MAD}_w, \varepsilon)} \quad (7)$$

where ε is a small constant that keeps the division from blowing up if the window happens to be flat. This keeps outliers from dominating, centres the signal at zero, and puts every window on roughly the same scale; the resulting trace is a statistical transform for the model, not a physiological measurement.

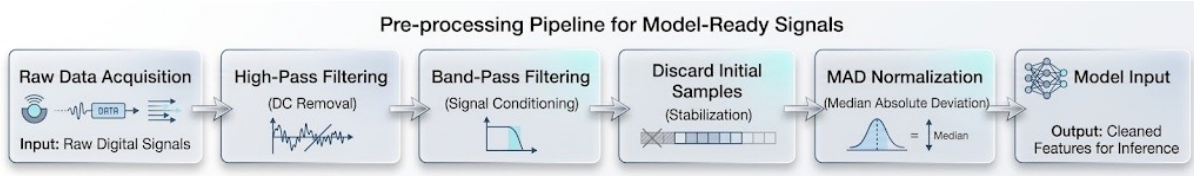


Figure 5: Block diagram of real-time signal conditioning chain before inference.

3.5. Integration of the detection models

Two models, previously trained by the DIGITI group, were integrated: one for 100 Hz signals and another for 25 Hz signals. Both models were converted from Keras to TensorFlow Lite, packaged with the project, and loaded during runtime. Each normalized window is input to the model, which outputs a probability for each class, as shown below:

$$\hat{y} = \begin{cases} \text{FA}, & p_{\text{FA}} > \tau \\ \text{RS}, & \text{otherwise} \end{cases} \quad (8)$$

with the threshold applied to the AF probability. Inference runs continuously over sliding windows as data comes in, and test datasets with examples of both classes were bundled in to check that the models actually behaved correctly once dropped into the mobile environment.

3.6. Protect acquired data

Four mechanisms address this issue. Accessing the application requires authentication via the Android application programming interface (API), biometrics, or device credentials. The only data stored on the device—AF episodes—are encrypted with Google Tink using AES-256 in GCM mode on small blocks, ensuring that encryption occurs during file writing [27]. Keys are stored in the Android Keystore, and an HKDF scheme generates a distinct subkey for each file [28]. All of this resides in the application’s internal directory, which is inaccessible to other installed applications.

Figure 6 presents the resulting workflow: on the wearable, it begins with a permission request and concludes with packaging the samples. Each incoming window is filtered, normalized, processed by the model, and displayed on the phone, and only windows identified as AF are stored and encrypted.

4. Results

The results are arranged according to the system’s data flow sequence: acquisition, conditioning, inference, and protection. Table 3 enumerates the parameters that vary between the two device families.

Both paths kept a continuous stream flowing to the phone. The raw Samsung signal, Figure 7(a), exhibits noticeable drift, though the cardiac oscillations are present, just partly buried in that drift. Once the high-pass stage runs, Figure 7(b), the signal sits around zero, and its amplitude drops from the order of 10^5 to about 3×10^4 , and the periodic pattern becomes obvious. PineTime behaves the same way at a much smaller scale, Figure 7(c)–(d): values near 1780 turn into oscillations of about 60 units, though the lower sample density caps how sharp the waveform can look.

Band-pass filtering clearly resolves the pulse on both devices, Figure 8(a)–(b): Samsung shows well-defined cycles around 1×10^4 , PineTime settles into a homogenized amplitude of approximately four units, which is lower resolution but still sufficient to work with. Figure 8(c)–(d) shows what MAD normalization does to both: amplitude evens out and the trace breaks into the processing windows, and the jumps visible between one window and the next come from the normalization itself, not from anything physiological.

The interaction flow was observed end-to-end: biometric authentication gates entry Figure 9(a), then the device selection screen appears in Figure 9(b), and the sensor permissions are requested on the

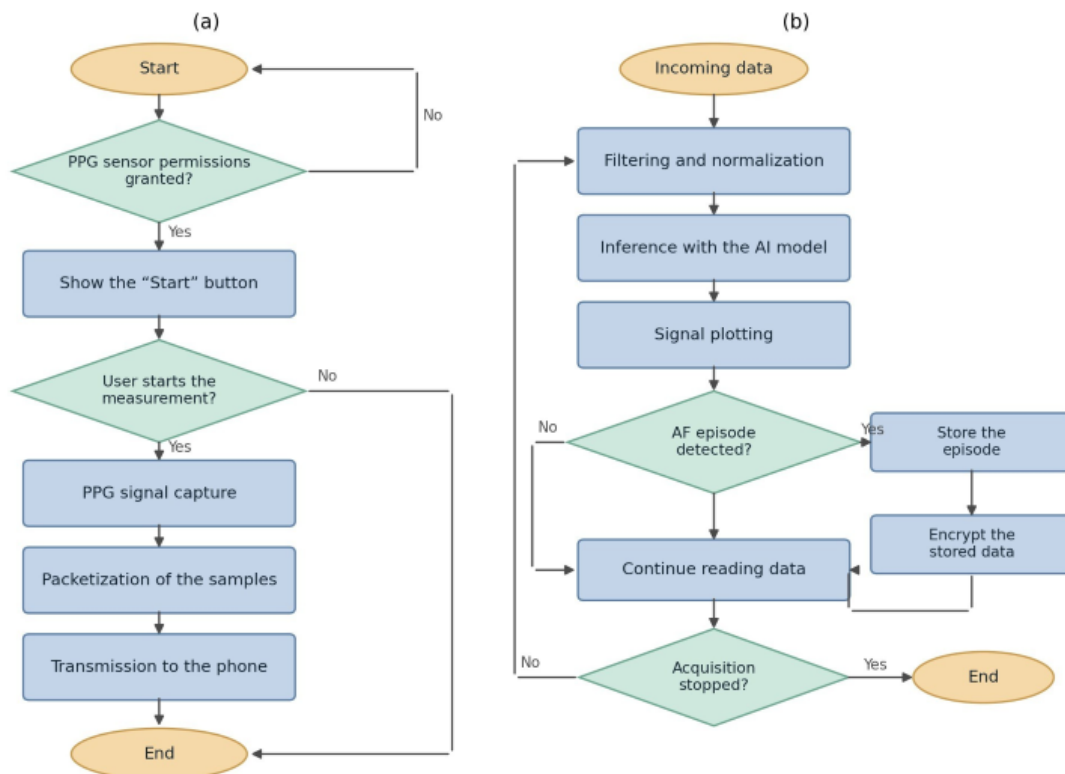


Figure 6: Application operation flow. (a) Wearable module. (b) When an episode is detected, the mobile module, including the storage and encryption branches, is executed.

Table 3

Device-dependent parameters of the acquisition and preprocessing chain.

Parameter	Samsung Galaxy Watch (Wear OS)	PineTime (InfiniTime)
Access mechanism	Samsung Health Sensor SDK	Conexión BLE/GATT directa
Sampling rate	100 Hz	25 Hz
Selected channel	Green	Green
Input format	JSON	Paquetes binarios
Band-pass filter	0.5–8 Hz, IIR second order	0.5–4 Hz
Discarded initial samples	300	80
Raw amplitude	Order of 10^5	Approximately 1780 units
Amplitude after DC removal	Around 3×10^4	Approximately 60 units
Amplitude after band-pass	Order of 1×10^4	Approximately 4 units
Detection model	Trained at 100 Hz	Trained at 25 Hz

watch. The main screen displays the connection status, live plot, and output of the detection module Figure 10(a). When an episode is flagged, that segment is logged (Figure 10(b)) and remains available in the encrypted history (Figure 10(c)). Both devices exhibited a small lag between acquisition and plotting, caused by data packing on the watch and processing stages on the phone. Hand checking the stored files confirmed that they are not readable as plain text and can only be opened through the decryption path built for them.

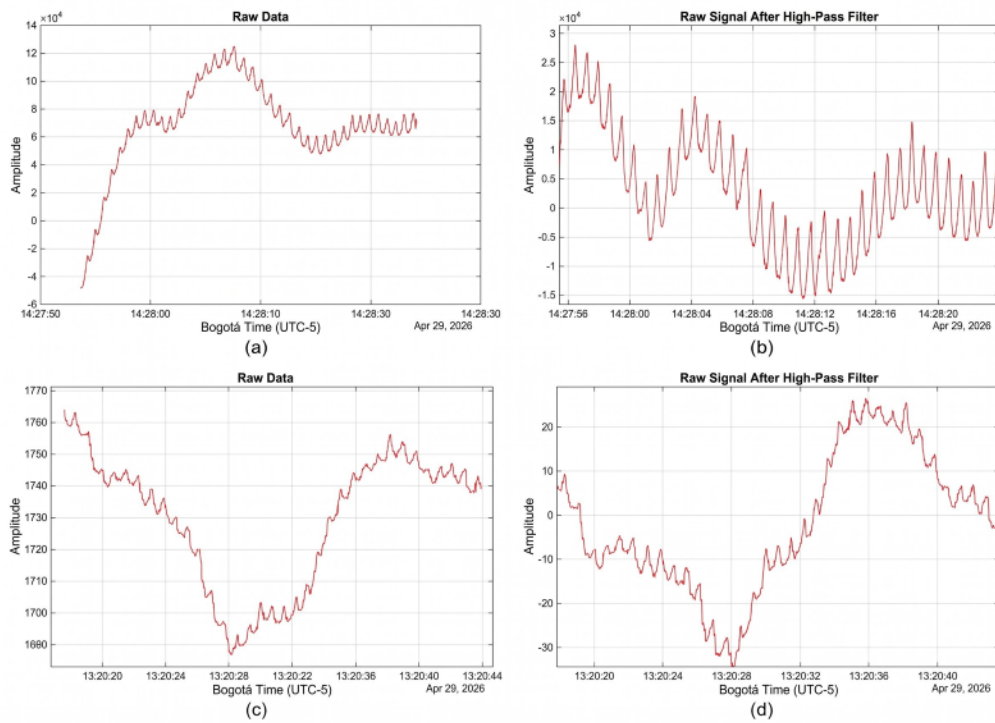


Figure 7: Raw and DC-removed signals. (a)-(b) Samsung device at 100 Hz. (c)-(d) PineTime device at 25 Hz.

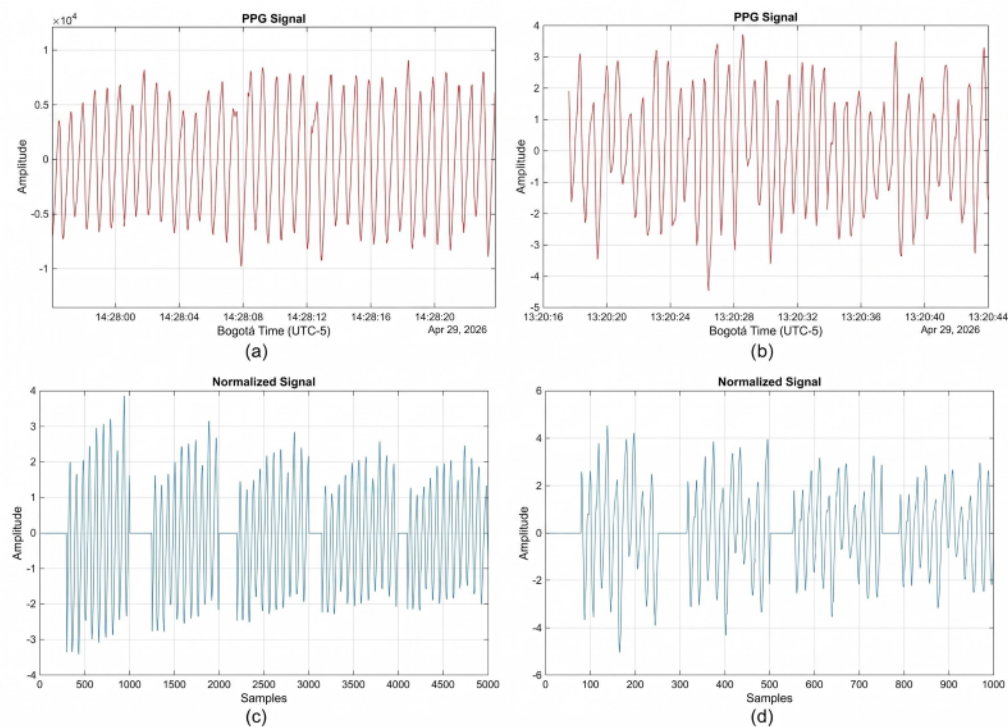


Figure 8: Conditioned signals. (a) Band-pass filter (Samsung). (b) Band-pass filtering, PineTime. (c) Normalized (Samsung). (d) Normalized (PineTime).

5. Discussion

BLE turned out to be the right call for the wearable-phone link, both for battery life and for how widely it is supported, letting the same transport serve a proprietary Wear OS stack and open-source

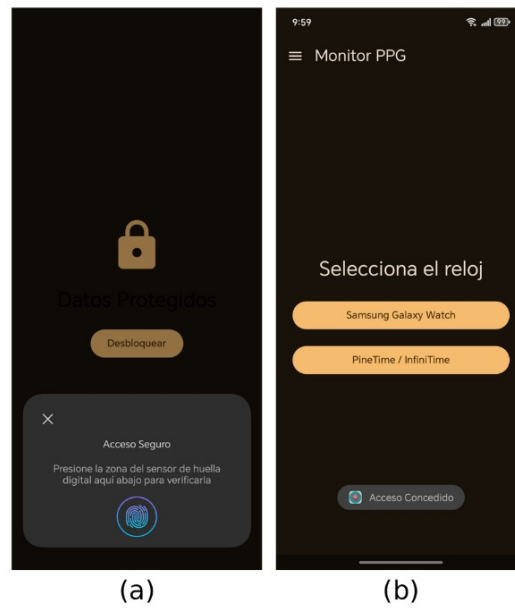


Figure 9: System access. (a) Biometric authentication. (b) Selection of the wearable device.

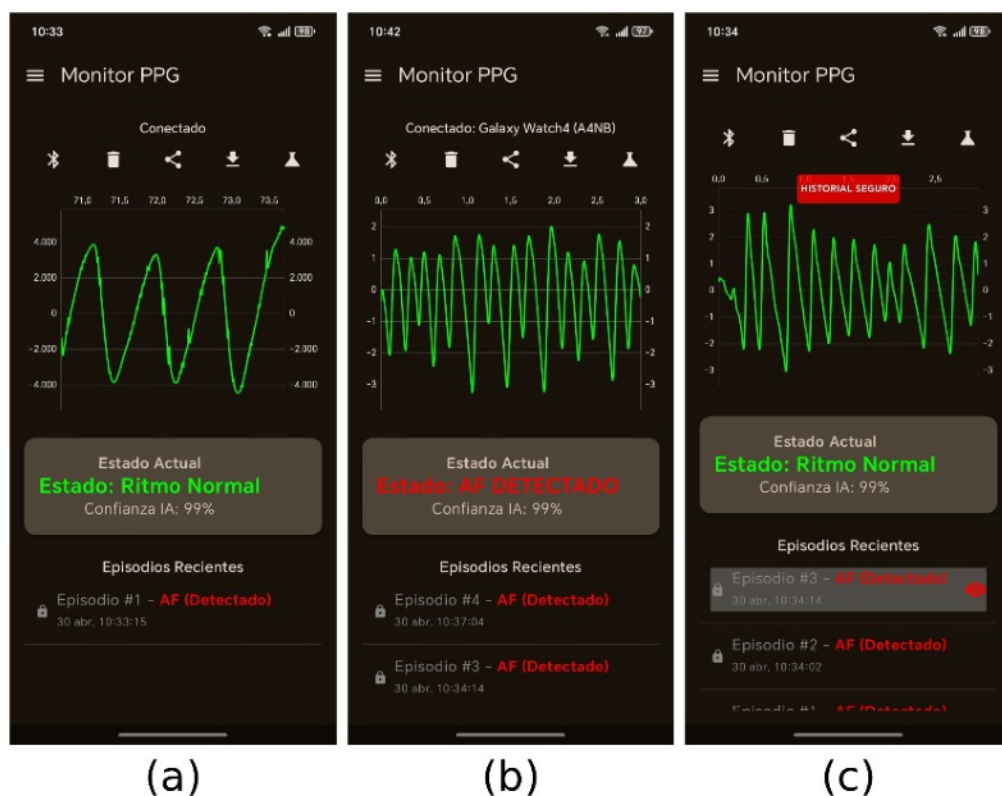


Figure 10: Mobile application during tests. (a) Real-time monitoring using a connected Galaxy Watch. (b) Episode classified as AF. (c) Encrypted history of stored episodes.

firmware without changes. The real obstacle was not technical but contractual: Table 2 shows that most manufacturers do not expose unprocessed measurements. Therefore, covering more brands would mean that the research group would be negotiating licenses, not writing more code.

The device-independent system converts both input formats into one shared representation before doing anything else with the data: storage, filtering, normalization, and inference all run on the

same structure regardless of the source of the samples. Thus, a new device only needs a decoder module. However, the preprocessing could not be fully identical because the difference in sampling rates forced different band-pass ranges and a different number of samples to be discarded for each device. Running the DIGITI models on live data rather than stored recordings is a step beyond what had been validated before, since it exposes them to drift, saturation transients, and the timing quirks of an actual wrist device. A few limitations remain worth naming: the MAD normalization creates discontinuities between windows that are not physiological, the acquisition-to-display delay builds up across the chain, PineTime's lower sampling rate caps waveform resolution, and functional integration, not diagnostic accuracy, which would require a controlled population and a reference standard.

6. Conclusions

This paper presents a communication interface that retrieves raw PPG data from smartwatches with varying hardware, consolidates it into a single stream, processes it in real time, and supplies it to DL models for atrial fibrillation detection. It operates on Samsung devices via the Health Sensor SDK and on InfiniTime devices via a direct BLE connection, after which both pathways merge into the same processing pipeline.

The architecture that separates the wearable from the phone, decoupling data acquisition from processing, and adjusting preprocessing to each sampling rate, enables multiple devices to operate within a single application. Biometric access control and encryption at rest treat the data as sensitive from the outset rather than as an afterthought. Future research should broaden manufacturer coverage and investigate normalization methods that eliminate window discontinuities, reduce processing latency, and run a clinical study to actually measure diagnostic performance. The platform is built for research and screening, not as a medical device.

Acknowledgments

The authors express their gratitude to Sistema Universitario Estatal (SUE D.C.), Colombia. They also thank the Universidad Distrital Francisco José de Caldas and the Universidad Militar Nueva Granada.

Funding information: Product derived from the research project titled “*Desarrollo de una aplicación para la detección temprana de fibrilación auricular a partir de técnicas de procesamiento digital de señales aplicado a datos de fotopletismografía (PPG)*”. Project funded by the *Sistema Universitario Estatal (SUE)*, the Research Office at the Universidad Distrital Francisco José de Caldas (ODI), and the Office of the Vice President for Research at the Universidad Militar Nueva Granada for the 2025-2 and 2026-1 funding periods, under the internal project codes PPEE-CV SUE DC 02 2023 and EXT-ING-4259.

Disclosure of Interests: The authors have no competing interests to declare that are relevant to the content of this article.

Declaration on Generative AI

During the preparation of this work, the authors used Trinka for grammar and spelling checks, and to paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] World Health Organization, Enfermedades cardiovasculares, 2024. URL: <https://www.who.int/es/health-topics/cardiovascular-diseases>, consultado el 18 de octubre de 2024.

- [2] Fundación Española del Corazón, Muerte súbita, 2024. URL: <https://fundaciondelcorazon.com/informacion-para-pacientes/enfermedades-cardiovasculares/muerte-subita.html>, consultado el 25 de septiembre de 2024.
- [3] C. X. Wong, et al., Epidemiology of sudden cardiac death: global and regional perspectives, *Heart, Lung and Circulation* 28 (2019) 6–14. doi:10.1016/j.hlc.2018.08.026.
- [4] J. A. Castro-Clavijo, et al., Prevalencia de fibrilación auricular en pacientes hospitalizados por medicina interna, *Revista Colombiana de Cardiología* 27 (2020) 557–563. doi:10.1016/j.rccar.2019.01.007.
- [5] L. A. Luengas-C., et al., Desarrollo de una aplicación para la detección temprana de fibrilación auricular a partir de técnicas de procesamiento digital de señales aplicado a datos de fotopletismografía (PPG), Informe Técnico, Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia, 2025.
- [6] J. Allen, Photoplethysmography and its application in clinical physiological measurement, *Physiological Measurement* 28 (2007) R1–R39. doi:10.1088/0967-3334/28/3/R01.
- [7] M. Elgendi, On the analysis of fingertip photoplethysmogram signals, *Current Cardiology Reviews* 8 (2012) 14–25. doi:10.2174/157340312801215782.
- [8] F. Alòs, A. Puig-Ribera, Uso de wearables y aplicaciones móviles (mhealth) para cambiar los estilos de vida desde la práctica clínica en atención primaria: una revisión narrativa, *Atención Primaria Práctica* 3 (2021) 100122. doi:10.1016/j.appr.2021.100122.
- [9] G. López, J. Forero, J. A. Gómez, Diseño de sistema wearable de monitoreo de pacientes psiquiátricos para reducción de eventos adversos, in: LACCEI, Bogotá D.C., Colombia, 2022.
- [10] Wearable para monitoreo de ritmo cardíaco y actividad electrodérmica, *Research in Computing Science* 149 (2020).
- [11] S. M. A. Iqbal, et al., A wearable telehealth system for the monitoring of parameters related to heart failure, *Heliyon* 10 (2024) e26841. doi:10.1016/j.heliyon.2024.e26841.
- [12] Y. Shen, M. Voisin, A. Aliamiri, A. Avati, A. Hannun, A. Ng, Ambulatory atrial fibrillation monitoring using wearable photoplethysmography with deep learning, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, NY, USA, 2019, p. 1909–1916. URL: <https://doi.org/10.1145/3292500.3330657>. doi:10.1145/3292500.3330657.
- [13] S. Kwon, et al., Detection of atrial fibrillation using a ring-type wearable device (CardioTracker) and deep learning analysis of photoplethysmography signals, *Journal of Medical Internet Research* 22 (2020) e16443. doi:10.2196/16443.
- [14] S. K. Bashar, et al., Smartwatch based atrial fibrillation detection from photoplethysmography signals, in: 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, Berlín, 2019. doi:10.1109/EMBC.2019.8856928.
- [15] F. Alòs, A. Puig-Ribera, Uso de wearables y aplicaciones móviles (mhealth) para cambiar los estilos de vida desde la práctica clínica en atención primaria: una revisión narrativa, *Atención Primaria Práctica* 3 (2021) 100122. doi:<https://doi.org/10.1016/j.appr.2021.100122>, telemedicina.
- [16] V. Nagarajan, J. Vignesh, Spot and continuous monitoring of heart rate, *IET Signal Processing* (2017).
- [17] S. M. Iqbal, M. A. Leavitt, G. Pedilus, I. Mahgoub, W. Asghar, A wearable telehealth system for the monitoring of parameters related to heart failure, *Heliyon* 10 (2024) e26841. URL: <https://www.sciencedirect.com/science/article/pii/S240584402402872X>. doi:<https://doi.org/10.1016/j.heliyon.2024.e26841>.
- [18] Z. Balas, et al., Research on the behaviour of Bluetooth Low Energy protocol in the heart rate monitoring application, *Procedia Computer Science* 225 (2023) 63–69. doi:10.1016/j.procs.2023.09.092.
- [19] Android Developers, Bluetooth de bajo consumo, 2025. URL: <https://developer.android.com/develop/connectivity/bluetooth/ble/ble-overview?hl=es-419>, consultado el 30 de octubre de 2025.
- [20] M. Căsar, et al., A survey on Bluetooth Low Energy security and privacy, *Computer Networks* 205

- (2022) 108712. doi:10.1016/j.comnet.2021.108712.
- [21] Huawei Developers, Introduction to Health Service Kit, 2025. URL: https://developer.huawei.com/consumer/en/doc/HMSCore-Guides/total_description-0000001417300754, consultado el 30 de octubre de 2025.
 - [22] Garmin Developer, Garmin Health SDKs, 2025. URL: <https://developer.garmin.com/health-sdk/overview/>, consultado el 30 de octubre de 2025.
 - [23] Google for Developers, Tipos de datos de la API de Google Health, 2025. URL: <https://developers.google.com/health/data-types?hl=es-419>, consultado el 30 de octubre de 2025.
 - [24] Withings Developer, Available health data, 2025. URL: <https://developer.withings.com/developer-guide/v3/data-api/all-available-health-data>, consultado el 30 de octubre de 2025.
 - [25] Samsung Developer, Samsung Health Sensor SDK, 2025. URL: <https://developer.samsung.com/health/sensor/overview.html>, consultado el 15 de julio de 2025.
 - [26] PINE64, PineTime, 2025. URL: <https://pine64.org/devices/pinetime/>, consultado el 30 de octubre de 2025.
 - [27] Google, What is Tink?, 2024. URL: <https://developers.google.com/tink/what-is?hl=es-419>.
 - [28] Android Developers, Android Keystore system, 2025. URL: <https://developer.android.com/privacy-and-security/keystore?hl=es-419>.