Evaluating Stress Monitoring Pipelines at the Ultra-Edge: A Mobile Device-Based Study

Vanessa Gamero[1], **Sergio T. Kofuji**[1]

[1]Department of Electronic Systems Engineering
Escola Politécnica da Universidade de São Paulo

`vgamero@usp.br, kofuji@usp.br`

Abstract. *Continuous stress monitoring using pervasive sensing generates multimodal data streams that are processed through pipelines, which may be implemented across distributed architectures. In this context, smartphones can operate as ultra-edge nodes, enabling local processing of these data while reducing latency and data transmission to upper layers. However, there is limited empirical analysis of how these pipelines operate under continuous execution at the ultra-edge. This study addresses this gap by evaluating the feasibility of executing a stress monitoring pipeline at the ultra-edge through the quantification of its data volume and computational requirements under continuous execution. For this purpose, each pipeline stage is analyzed in terms of generated data volume and the associated computational requirements in CPU, memory, and energy usage. A baseline configuration is compared with alternative strategies to reduce data volume and computational load. Preliminary results show that input data rates are low; however, continuous processing introduces sustained computational demands that must be considered for real-world use. These results provide a baseline for evaluating processing costs in subsequent pipeline stages.*

1. Introduction

The widespread availability of sensors in personal devices has enabled the pervasive generation of multimodal data in everyday environments [Detweiler and Hindriks 2016]. These capabilities facilitate the study of human states across different domains, including mental well-being, where psychological stress is a relevant case. The response to psychological stress varies significantly among individuals and is influenced by physiological, behavioral, and environmental factors. This variability limits point-in-time stress evaluation, motivating continuous approaches based on physiological and behavioral data [Goodday and Friend 2019].

Computational pipelines for psychological stress inference process these data through multiple analytical stages, transforming them into structured representations to support the characterization of stress dynamics [Smets et al. 2018]. However, they are primarily designed for this purpose under offline settings, with limited studies evaluating real-world deployment through quantitative metrics of performance and resource consumption [Zhang et al. 2024].

In ultra-edge environments, where processing is performed directly on resource-limited personal devices, the deployment of these pipelines offers advantages such as reduced feedback latency, enhanced data privacy, and minimized data transmission to higher-level computing layers [Shi et al. 2016]. However, these benefits come at the cost of sustained computational processing and energy consumption, which must operate within the limited computational and energy resources of smartphones [Can et al. 2019].

These constraints highlight the importance of establishing an empirical basis to understand the impact of distributing pipeline stages across architectural layers on resource usage and system efficiency. To address this, the study evaluates the execution of a stress monitoring pipeline at the ultra-edge node on a smartphone across three processing stages. It quantifies, for each pipeline stage, the volume of generated data and the associated computational requirements in terms of CPU, memory, and energy usage. The results are compared between a baseline configuration and alternative configurations that incorporate two strategies to reduce data volume and processing load during local execution.

2. Related work

Psychological stress monitoring is typically implemented through computational pipelines that operate primarily on physiological data, complemented by behavioral and contextual information [Smets et al. 2018]. These data are continuously processed across successive pipeline stages, incurring computational cost and increasing data volume across stages [Zhang et al. 2024].

Several studies have supported stress monitoring through the development of mobile sensing frameworks [Ferreira et al. 2015] and the release of real-world datasets [Kang et al. 2023]. In these systems, mobile devices function as ultra-edge nodes responsible not only for data acquisition but also for early-stage processing [Garcia-Ceja et al. 2015, Landreani et al. 2019].

Recent work has explored stress inference at the edge through model optimization techniques [Jaiswal et al. 2024, Rachakonda et al. 2019], demonstrating the feasibility of performing local inference with reduced model size and energy consumption. However, these approaches focus primarily on the inference stage, overlooking preceding pipeline stages where continuous data transformation also incurs computational costs. In contrast, this study focuses on the characterization of pipeline execution at the ultra-edge, analyzing data flow and associated CPU, memory, and energy requirements under continuous local execution prior to inference.

3. Methodology

3.1. Multimodal Data Sources

This study is conducted under non-clinical conditions and considers physiological and contextual sources of data that are widely used in psychological stress monitoring. In this context, heart rate variability (HRV) is a widely used indicator of autonomic nervous system (ANS) activity and a proxy for stress-related responses.

The Polar H10 is a single-lead chest strap that provides electrocardiogram (ECG) data. This device provides raw ECG, heart rate (HR), and RR intervals, from which HRV metrics are derived [Polar Electro Oy 2021]. The validity of Polar H10 measurements for HRV applications is supported by the literature [Schaffarczyk et al. 2022].

Contextual information often used in stress monitoring includes user motion data to provides context for physiological measurements by characterizing motion-related noise and artifacts, as well as enabling the inference of basic user activities such as walking or running. This is analyzed either from raw accelerometer data [Weiss 2019] or through Google APIs. Complementary contextual data, such as location, is also obtained through these API [Google for Developers 2026]. Table 1 summarizes the data sources and the types of data provided by each source.

Table 1. Multimodal sources and provided data

Sources	**Data provided**			**Acquisition settings**
	Raw	Metrics	Inferred	
Polar H10	ECG	HR; RR	–	ECG F_s: 130 Hz
Accelerometer	3-axis accel.	–	–	F_s: 5, 10, 20 Hz
Activity API	–	–	Activity state	System-defined
Location API	–	–	Loc. coordinates	System-defined

Accel.: Acceleration; API: Application Programming Interface; Loc.: Location; F_s: Sampling frequency

Physiological and contextual data streams are integrated through a local hub in a smartphone application developed for this study. This hub is responsible for managing connections to the data sources, and use the AWARE framework [Ferreira et al. 2015] and the Polar H10 API. Prior to pipeline execution, the incoming data streams are aggregated and temporally synchronized using the timestamps provided by each data source to align samples across modalities into a common timeline.

3.2. Pipeline for psychological stress monitoring

The pipeline captures core data transformations across three stages: pre-processing, temporal segmentation, and feature extraction, prior to stress inference. Focusing on these stages enables the characterization of computational cost and data volume without unnecessary complexity. It operates on both raw sensor data and device- or API-computed data, enabling analysis of trade-offs between processing flexibility and computational cost.

The pre-processing stage includes representative ECG filtering configurations with varying computational demands, such as Butterworth bandpass, its combination with a notch filter, and wavelet-based denoising. Accelerometer data are filtered using a Butterworth low-pass filter, while derived data streams (e.g., HR and RR intervals) are filtered using physiological thresholds (30–220 bpm) and activity states based on confidence levels ($\geq$ 80%).

The temporal segmentation stage partitions data into time windows. For ECG, the window lengths follow ultra-short HRV analysis to support continuous monitoring [Castaldo et al. 2019], with R-peak detection used to derive RR intervals [Van Gent et al. 2019]. Accelerometer data are segmented using standard windowing approaches for activity recognition [Garcia-Ceja et al. 2015, Weiss 2019].

The feature extraction stage derives time- and frequency-domain representations to capture physiological and motion dynamics while reducing data dimensionality, enabling compact signal representation and efficient processing under continuous execution.

3.3. Pipeline management strategies

A quality evaluation step is introduced after temporal segmentation to ensure the reliability of subsequent feature extraction. Window quality is assessed using lightweight signal quality indicators (SQI), including data completeness, signal variance, physiological plausibility of derived measures, and motion-aware indicators derived from accelerometer data. Segments that do not meet these criteria are discarded prior to feature computation, preventing the propagation of unreliable data and reducing unnecessary processing overhead.

The second strategy reduces computational cost by adapting the temporal window length for feature computation based on ultra-short HRV durations reported in the literature [Castaldo et al. 2019]. A baseline 60 s window is used to estimate the root mean square of successive differences (RMSSD), a time-domain HRV metric capturing short-term variability. When HRV metrics remain stable and motion levels are low, the window length is reduced to 30 s to lower computational cost. If RMSSD subsequently indicates a potential stress response, the window is restored to 60 s to confirm whether the change reflects a sustained physiological pattern rather than a transient fluctuation.

3.4. Metrics of evaluation

Proportional Set Size (PSS): PSS is used as the primary metric to quantify memory usage. A baseline is established after application initialization with the pipeline inactive, and memory usage during execution is compared against it to estimate the incremental cost of each pipeline stage. Results are reported in absolute terms and normalized by the smartphone's total RAM for cross-device comparison.

CPU usage: CPU usage is measured using two complementary approaches. Under laboratory conditions, Android Studio Profiler is used to inspect CPU scheduling and thread activity during pipeline execution. For in situ evaluation, CPU usage is estimated at the process level using Android system APIs by measuring the elapsed CPU time between the start and end of each pipeline stage.

Data volume: Data volume is estimated using two approaches: first, derived from the acquisition parameters of each sensing modality; second, computed from the data received and generated at each pipeline stage. Results are reported in bytes per window and bytes per second to enable comparison across stages and configurations.

4. Results

4.1. Characterization of input data streams

Laboratory measurements were conducted to characterize the data streams received by the smartphone from the sensing sources before entering the pipeline. ECG stream from Polar H10 showed stable packets with 219 bytes of ECG payload plus a 10-byte header metadata. Accelerometer data are recorded as tri-axial samples, resulting in 20 bytes per measurement. The estimated input data volumes are summarized in Table 2.

These estimations indicate that the combined input data rate from ECG and accelerometer streams remains low, reaching approximately 0.8 KB/s under the highest configuration, equivalent to about 2.8 MB per hour or 67 MB per day. This volume depends

Table 2. Estimated input data volume of sensing sources

Source	Structure	Size	Rate	Data volume
Polar H10	packet (73 ECG samples + header metadata)	229 B/packet	∼1.78 packets/s	≈ 408 B/s
smartphone accelerometer	sample (timestamp + x,y,z)	20 B/sample	5 samples/s	100 B/s
			10 samples/s	200 B/s
			20 samples/s	400 B/s

on the selected sensing modalities and acquisition settings and may increase with additional or higher-rate sources; therefore, storage duration and retention strategies should be considered. Despite the low data rate, continuous data flow requires sustained processing, including filtering, segmentation, and feature extraction, which may introduce additional computational and energy demands that need to be evaluated.

5. Conclusions

This study explores the feasibility of executing a stress monitoring pipeline on the smartphone as the processing node. Although the acquisition stage appears not to represent a significant burden, continuous processing may introduce computational demands on the device that require further evaluation. Accordingly, the evaluation framework considers CPU usage, memory consumption, and data volume across pipeline stages.

6. Future Work

Future work will focus on evaluating the trade-offs introduced by performing quality evaluation at the ultra-edge, which reduces the propagation of invalid or noisy data and avoids unnecessary processing, despite the additional local computational cost. To support this analysis, the experimental evaluation will be extended through continuous data acquisition in real usage conditions, incorporating richer contextual data to assess their impact on pipeline behavior.

Acknowledgment

The authors would like to thank the financial support from the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors acknowledge the use of GPT-based AI assistance to improve text clarity.

References

Can, Y. S., Arnrich, B., and Ersoy, C. (2019). Stress detection in daily life scenarios using smart phones and wearable sensors: A survey. *Journal of biomedical informatics*, 92:103139.

Castaldo, R., Montesinos, L., Melillo, P., James, C., and Pecchia, L. (2019). Ultra-short term hrv features as surrogates of short term hrv: A case study on mental stress detection in real life. *BMC medical informatics and decision making*, 19(1):12.

Detweiler, C. A. and Hindriks, K. V. (2016). A survey of values, technologies and contexts in pervasive healthcare. *Pervasive and Mobile Computing*, 27:1–13.

Ferreira, D., Kostakos, V., and Dey, A. K. (2015). AWARE: Mobile Context Instrumentation Framework. *Frontiers in ICT*, 2.

Garcia-Ceja, E., Osmani, V., and Mayora, O. (2015). Automatic stress detection in working environments from smartphones' accelerometer data: a first step. *IEEE journal of biomedical and health informatics*, 20(4):1053–1060.

Goodday, S. M. and Friend, S. (2019). Unlocking stress and forecasting its consequences with digital technology. *npj Digital Medicine*, 2(1):1–5.

Google for Developers (2026). Overview of google play services. Last updated 2026-03-02 UTC.

Jaiswal, D., Mukhopadhyay, S., and Sharma, V. (2024). Tinystressnet: On-device stress assessment with wearable sensors on edge devices. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 166–171. IEEE.

Kang, S., Choi, W., Park, C. Y., Cha, N., Kim, A., Khandoker, A. H., Hadjileontiadis, L., Kim, H., Jeong, Y., and Lee, U. (2023). K-EmoPhone: A Mobile and Wearable Dataset with In-Situ Emotion, Stress, and Attention Labels. *Scientific Data*, 10:351.

Landreani, F., Faini, A., Martin-Yebra, A., Morri, M., Parati, G., and Caiani, E. G. (2019). Assessment of ultra-short heart variability indices derived by smartphone accelerometers for stress detection. *Sensors*, 19(17):3729.

Polar Electro Oy (2021). Polar ble sdk. `https://github.com/polarofficial/polar-ble-sdk`. GitHub repository, accessed 2026-03-07.

Rachakonda, L., Mohanty, S. P., Kougianos, E., and Sundaravadivel, P. (2019). Stress-lysis: A dnn-integrated edge device for stress level detection in the iomt. *IEEE Transactions on Consumer Electronics*, 65(4):474–483.

Schaffarczyk, M., Rogers, B., Reer, R., and Gronwald, T. (2022). Validity of the polar h10 sensor for heart rate variability analysis during resting state and incremental exercise in recreational men and women. *Sensors*, 22(17):6536.

Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646.

Smets, E., Rios Velazquez, E., Schiavone, G., Chakroun, I., D'Hondt, E., De Raedt, W., Cornelis, J., Janssens, O., Van Hoecke, S., Claes, S., Van Diest, I., and Van Hoof, C. (2018). Large-scale wearable data reveal digital phenotypes for daily-life stress detection. *npj Digital Medicine*, 1(1):1–10.

Van Gent, P., Farah, H., Van Nes, N., and Van Arem, B. (2019). Analysing noisy driver physiology real-time using off-the-shelf sensors: Heart rate analysis software from the taking the fast lane project. *Journal of Open Research Software*, 7(1).

Weiss, G. M. (2019). Wisdm smartphone and smartwatch activity and biometrics dataset. *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set*, 7(133190-133202):5.

Zhang, P., Jung, G., Alikhanov, J., Ahmed, U., and Lee, U. (2024). A reproducible stress prediction pipeline with mobile sensor data. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 8(3):1–35.