# Wearables for Health (W4H) Toolkit for Acquisition, Storage, Analysis and Visualization of Data from Various Wearable Devices

Arash Hajisafi, Maria Despoina Siampou, Jize Bi, Luciano Nocera, Cyrus Shahabi
*Dept. of Computer Science, University of Southern California, Los Angeles, CA, USA*
{hajisafi, siampou, jizebi, nocera, shahabi}@usc.edu

***Abstract*—The Wearables for Health Toolkit (W4H Toolkit) is an open-source platform that provides a robust, end-to-end solution for the centralized management and analysis of wearable data. With integrated tools and frameworks, the toolkit facilitates seamless data acquisition, integration, storage, analysis, and visualization of both stored and streaming data from various wearable devices. The W4H Toolkit is designed to provide medical researchers and health practitioners with a unified framework that enables the analysis of health-related data for various clinical applications. We provide an overview of the system and demonstrate how it can be used by health researchers to import and analyze a wide range of wearable data and perform data analysis, highlighting the versatility and functionality of the system across diverse healthcare domains and applications.**

***Index Terms*—wearable sensors, health monitoring, health analytics, visualization**

## I. Introduction

Wearable data is emerging as a crucial health and disease information source, capturing personal, behavioral, social, and environmental health-related factors. While wearables have been primarily used for activity tracking and fitness applications, more recently, these devices have found increasing utility in many healthcare applications, including health monitoring, clinical care, remote clinical trials, drug delivery, and disease characterization [1]–[4]. Yet, the comprehensive and continuous collection of full-day data from individuals renders their storage, management, and analysis challenging, highlighting the need for robust solutions to effectively handle the vast and continuous streams of information generated by wearable devices.

In response to the challenges posed by the rapid evolution of wearable technology, we introduce the Wearables for Health Toolkit (W4H Toolkit[1]), an open-source platform that leverages modern data technologies to manage large and continuously evolving wearable datasets efficiently. The W4H Toolkit is designed to empower health professionals and researchers with a unified platform that provides seamless access to wearable and essential analytical tools, thereby fostering innovation in wearable health data use across a wide range of healthcare sectors and applications. The platform is also relevant to developers who wish to develop and test new clinical applications leveraging wearable clinical trial data. The primary objectives of the toolkit are two-fold: first, to serve as a unified data repository capable of accommodating various types of wearables and health data (e.g., demographics), and second, to provide robust data query, streaming, and offline analytics coupled with intuitive data visualization capabilities.

To showcase our system's capabilities, we will present three distinct scenarios. Specifically, we will demonstrate how users *(i)* set up the toolkit and import their data from different wearable devices into the platform, *(ii)* leverage the system's querying and visualization utilities on historical data, and *(iii)* perform real-time analytics on streaming data. These scenarios provide a comprehensive system overview and demonstrate capabilities pertinent to various healthcare applications.

## II. W4H Toolkit Overview

### A. Modes of Operation

The W4H Toolkit provides two distinct utilization options: a *Hosted Solution* and a *User-Managed Solution*. Both alternatives can be fully HIPAA compliant, supporting secure authentication, data encryption, secure transmissions, access control, and audit logs. Regardless of the chosen mode, the toolkit integrates seamlessly into the W4H ecosystem, facilitating the import of offline data and the connection of live wearable data streams.

**Hosted Solution.** We have deployed the W4H instance on our own cluster and made it available for testing to health researchers individually. We plan to provide access to the larger research community in the near future.

**User-Managed Solution.** This is a Dockerized instance that users can easily install and configure to run on their own systems. This allows users to customize data security policies and implement additional HIPAA compliance measures to fit their use case.

### B. GeoMTS Abstraction

To address the complexities of wearable data and the variety of devices, we introduce a novel abstraction named Geospatial Multivariate Time Series (GeoMTS). This conceptualization views wearables as multiple geotagged time series. We introduce a robust schema designed for the efficient storage and retrieval of GeoMTS data using open-source database platforms. Additionally, we implement analytical methods, including outlier detection and accelerated computation of aggregate statistics, applicable to both stored and streamed GeoMTS data. This abstraction unifies the storage, analysis,

[1] The source code for the W4H Toolkit along with all its modules is available for access on GitHub: https://github.com/USC-InfoLab/w4h-integrated-toolkit.

and visualization of wearable data, regardless of the device the data originated from.

### C. GeoMTS-Enabled Operational Tools

To enable comprehensive operations encompassing data ingestion, storage, analysis, and visualization of wearable data, we have developed a set of tools building upon the GeoMTS data structure, as delineated below:

*1) StreamSim:* StreamSim, tailored to work consistently with GeoMTS data, fills the existing gap for a tool that is capable of accurately simulating real-time data streaming, especially in scenarios where actual data sources are unavailable. It enables researchers and developers to mimic real-time streaming of tabular data, facilitating the testing of applications and systems in controlled environments to ensure robustness and efficiency.

*2) W4H ImportHub:* W4H ImportHub is designed to integrate offline datasets, like CSV files, into the W4H platform, aligning with the GeoMTS schema and data structures. It aids in the smooth import of data while offering automatic attribute mapping to ensure data consistency and integration with the structure and standards of the W4H ecosystem.

*3) Wearable Device APIs:* We have developed libraries to fetch wearable data from the vendor servers, where it is first synced. These libraries are available for the major activity trackers in the market, including *pyGarminAPI*, *pyFitbitAPI*, *pyAppleWatchAPI* (available soon).

*4) Integrated Analytics Dashboard:* The Integrated Analytics Dashboard, a core component of the W4H toolkit, serves as a robust interface for GeoMTS data extraction, presentation, and analysis from wearables. This dashboard facilitates the analysis of both streaming and stored data conceptualized as GeoMTS, ensuring timely and actionable insights for users.

*5) Approximate Aggregate Queries on GeoMTS:* W4H platform provides the option of answering approximate aggregate queries on GeoMTS within specific time ranges, to enhance query performance. To this end, a compressed representation of the data is stored in the database. Here, we utilize the Fast Fourier Transform (FFT) for data transformation, storing only the first few coefficients. At query time, the Inverse Fast Fourier Transform (IFFT) is performed to reconstruct the time-domain signal from the stored coefficients. Transforming the data using such representation accelerates query processing for aggregates such as `AVERAGE` and `SUM`, which can be answered approximately through integration during the signal reconstruction phase. This ensures that the query answering time remains constant even as the query range expands. Other types of queries, like `MIN` and `MAX`, are directly identified within the specified interval of the reconstructed signal. This approximation technique assumes that the data have been generated at regularly spaced time intervals. To that extent, answering `COUNT` queries can also be approximated, leveraging the known sampling ratio. This simple yet effective approach is easily integrated into the W4H database as a Postgres extension.

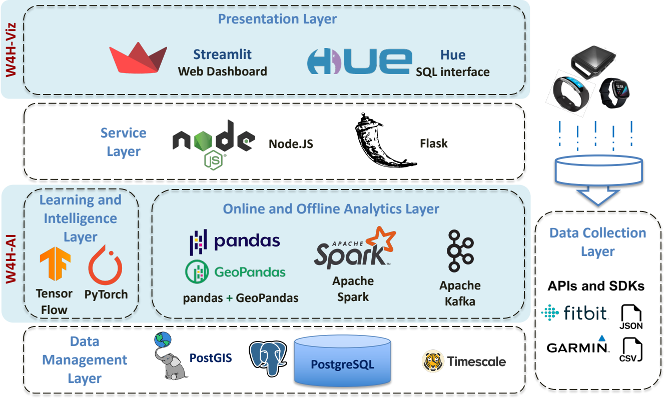


Fig. 1. W4H Layered Architecture

## III. W4H Architecture

### A. Overview

As illustrated in Fig. 1, W4H adopts a modular design represented by a layered architecture, where each layer is designed with a distinct objective that contributes to the overall functionality and effectiveness of the system. This design approach prioritizes modularity, fostering flexibility and maintainability while enabling a clear separation of concerns [5]. The layered design also facilitates a systematic approach to address the diverse requirements of wearable data management, analytics, and visualization, all orchestrated around the central abstraction of the GeoMTS data.

The layered architecture (Fig. 1) utilizes modern data components, such as Apache Spark [6] and Streamlit [7] and comprises six layers: and comprises six layers: (1) the Data Collection Layer for data acquisition, (2) the Data Management Layer for data storage and conversion to GeoMTS, (3) the Online and Offline Analytics Layer for streaming and batch analytics, (4) the Learning and Intelligence Layer for deploying Machine Learning (ML) and Deep Learning (DL) models, (5) the Service Layer for API integrations, and (6) the Presentation Layer for data visualization and interaction. In the remainder of this section we detail each layer of the W4H layered architecture.

### B. Data Collection Layer

The Data Collection Layer is pivotal for ingesting wearable data from various devices, addressing two major challenges: integrating a wide range of data formats and accommodating streaming data feeds. The former encompasses a myriad of data shapes and formats, e.g., CSV and JSON, each exhibiting a unique structure that necessitates a mapping to the W4H database schema for successful ingestion. ImportHub (Section II-C2) is engineered to address this challenge.

On the other hand, the real-time data feeds present a dynamic facet, demanding a robust infrastructure capable of handling live data streams. For this, we leverage Apache Kafka, a scalable message queue, alongside micro-services tailored for each real-time feed, such as those streaming from a Garmin watch, ensuring a smooth, real-time data ingestion process. The libraries pyGarminAPI and pyFitbitAPI, as discussed in Section II-C3, play a crucial role in this layer by

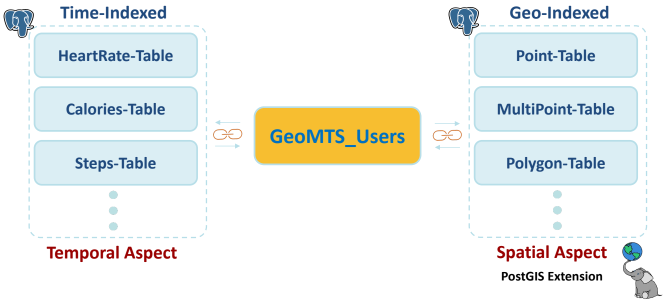


Fig. 2. Conceptual Framework of Geospatial Multivariate Time Series (GeoMTS) Abstraction

enabling seamless interaction with popular wearable device APIs.

### C. Data Management Layer

The Data Management Layer is crucial, serving as a bridge for integrating the collected data from the Data Collection Layer into the GeoMTS format, which is central to the efficacy of the entire W4H architecture. At the core of this layer is the conversion of the ingested data into the GeoMTS format, facilitated by the PostgreSQL database, augmented with the TimescaleDB [8] extension for temporal management, PostGIS [9] for spatial management, and our FFT PostgreSQL extension for efficient answering of range aggregate queries on time series data (Section II-C5). This integration ensures the convenient accessibility of GeoMTS data, laying a solid foundation for subsequent analytics.

The GeoMTS abstraction, illustrated in Fig. 2 separates each data source into distinct features, such as heart rate, calories, steps for wearable data, or trajectories for spatial time-series data. Each feature is then represented as a GeoMTS table, embodying a triplet of user ID, timestamp, and value. This abstraction accommodates every GeoMTS dataset, resulting in a standardized representation. The time dimension of temporal tables is indexed using TimescaleDB, while the spatial tables are indexed using PostGIS. Central to this schema is the table encapsulating subject (user) information, which serves as a mapping hub, linking each feature to its respective user using the user ID column and connecting all GeoMTS tables.

### D. Online and Offline Analytics Layer

This layer is engineered to provide analytical capabilities on the GeoMTS data, addressing the challenges of stream analytics, which necessitate fast, efficient processing. It incorporates Apache Kafka for scalable real-time message queuing, facilitating the smooth handling of streaming data. To tackle the demanding requirements of real-time analytics, Apache Spark, with its in-memory cluster computing, provides high-speed analytics, ensuring rapid insights extraction from continuous data streams. Moreover, for a more straightforward analysis of GeoMTS tabular data in an offline setting, pandas [10] and GeoPandas [11] are employed, providing an intuitive yet powerful framework for data manipulation and geospatial analysis, respectively. This combination of tools empowers the W4H Toolkit with a versatile analytical capability for both real-time and offline needs.

### E. Learning and Intelligence Layer

Recent advancements in machine learning (ML) and deep learning (DL) have significantly benefited health applications [12]. The Learning and Intelligence Layer in the W4H Toolkit is designed for the swift deployment of these models on GeoMTS data, which offers a structured, clean, and consistent data format. This setup not only makes data readily available but also minimizes the preprocessing overhead commonly encountered in machine learning pipelines, thus accelerating the model training and deployment process.

### F. Service Layer

The Service Layer, utilizing Flask and Node.js, acts as a gateway for accessing GeoMTS data and analytics results, forming a bridge between the core analytical components and the Presentation Layer. It furnishes APIs for easy access to data within the W4H platform and seamlessly channels analytics results to the dashboard for visualization and further interaction, ensuring a coherent flow of information to facilitate user engagement and data exploration.

### G. Presentation Layer

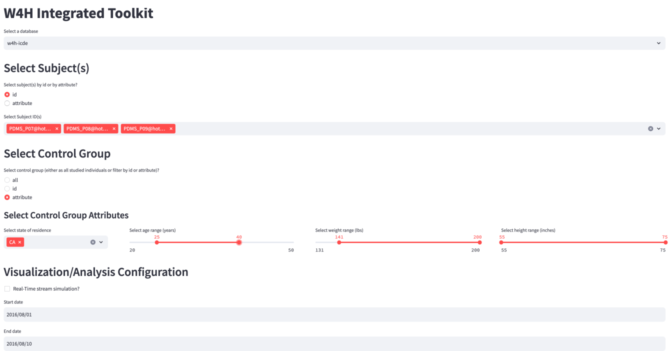
The Presentation Layer serves as the user interface to the W4H Toolkit, encapsulating visualization, interaction, and direct data access functionalities. It hosts an analytical dashboard created using Streamlit for real-time and offline wearable data analytics, alongside a SQL interface through Hue for direct interaction with GeoMTS data. The dashboard provides real-time outliers detection, comparative analytics between different groups of subjects and a control group, health insights, and more. Additionally, this layer furnishes the front end for other tools like StreamSim for simulating streaming data and ImportHub for data integration, enhancing the user experience and accessibility of these tools within the W4H platform.

## IV. Demonstration Scenarios

In this section, we discuss three demonstration scenarios encapsulating the end-to-end functionality of the W4H Toolkit from a user-centric perspective. These scenarios showcase the systematic process of importing data from various wearable devices, deriving insightful analytics from offline datasets, and harnessing real-time data from wearable devices for immediate insights and outlier detection (a video demonstration is available at: https://youtu.be/67a8kuMjSAU).

### A. Instantiating W4H Instance and Importing Offline Health Data from Various Devices

In this demonstration scenario (see the accompanying video), we illustrate the ease of setting up a W4H instance and integrating offline health data from multiple wearable devices, leveraging the built-in ImportHub tool. We present datasets collected from two different devices, one from Fitbit and one from Microsoft Band, along with their corresponding subject data files in CSV formats. Despite their varying formats, we demonstrate how W4H ImportHub maps these datasets to the W4H database schema, ensuring seamless transformation to GeoMTS format and ingestion of data.



W4H Integrated Toolkit

Select a database

w4h-icde

Select Subject(s)

Select subject(s) by id or by attribute?

● id
○ attribute

Select Subject ID(s)

PDMS_P07@hot... × PDMS_P08@hot... × PDMS_P09@hot... ×

Select Control Group

Select control group (either as all studied individuals or filter by id or attribute)?

○ all
○ id
● attribute

Select Control Group Attributes

Select state of residence: CA ×

Select age range (years): 25 – 40 (20 – 50)

Select weight range (lbs): 141 – 200 (131 – 200)

Select height range (inches): 55 – 75 (55 – 75)

Visualization/Analysis Configuration

□ Real-Time stream simulation?

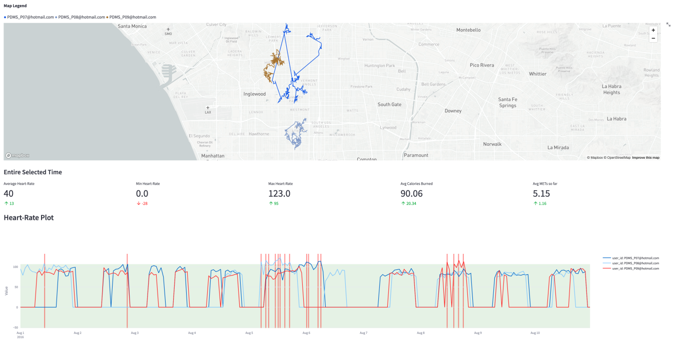
Start date

2016/08/01

End date

2016/08/10

Fig. 3. Query Selection in W4H Dashboard



Map Legend

● PDMS_P07@hotmail.com ● PDMS_P08@hotmail.com ● PDMS_P09@hotmail.com

Entire Selected Time

| Average Heart-Rate | Min Heart-Rate | Max Heart-Rate | Avg Calories Burned | Avg METs so far |
|---|---|---|---|---|
| 40 | 0.0 | 123.0 | 90.06 | 5.15 |
| ↑ 13 | ↓ -28 | ↑ 95 | ↑ 20.34 | ↑ 1.16 |

Heart-Rate Plot

Value
100
50
0
-50
Aug 1
2016
Aug 2
Aug 3
Aug 4
Aug 5
Aug 6
Aug 7
Aug 8
Aug 9
Aug 10
user_id: PDMS_P07@hotmail.com
user_id: PDMS_P08@hotmail.com
user_id: PDMS_P09@hotmail.com

Fig. 4. Results for Query in W4H Dashboard

We provide a self-contained Docker image for the system, enabling attendees to run the image to instantiate a W4H instance effortlessly. Through this, we showcase the straightforward process of importing data using the ImportHub tool. This demonstration aims to provide the audience with a hands-on insight into the simplicity and efficiency of setting up a W4H instance. Moreover, it highlights the ease with which offline health data can be integrated into the W4H platform, irrespective of its format and device.

### *B. Offline Health Analytics*

This scenario showcases the analytics capabilities of the W4H Dashboard on offline GeoMTS data already ingested into the system. Utilizing the datasets imported in the previous step, we show how users can specify queries through the dashboard. This includes selecting subjects based on IDs or attributes, choosing a control group for comparative health metric analysis, defining a date range, and specifying finer detailed target times for investigation.

We emphasize the system's speed in visualizing insights for the selected query, demonstrating various analytics such as health statistics comparison between selected subjects and a control group, and outlier detection in health metrics. Through this demonstration, the audience will grasp how the W4H Dashboard facilitates a diverse range of analytics on offline data, thus aiding in enhanced health understanding and informed decision-making. Fig. 3 shows a sample query selection, and Fig. 4 depicts the results for that query in the W4H Dashboard.

### *C. Real-Time Outlier Detection and Analytics*

In this scenario, we showcase the real-time analytics and outlier detection capabilities of the W4H Dashboard, complemented by the integrated StreamSim tool (Section II-C1), which facilitates the simulation of data streams from historical data in the absence of real-time feeds. Initiating with the data imported in the prior step, we show the process of configuring StreamSim to simulate a data stream from this historical data. Following this setup, we demonstrate the W4H system's capability to ingest this simulated data stream in real-time, generating health insights, computing statistics, and identifying outliers as data flows in. This scenario echoes W4H's analytical capabilities in a real-time setting, showcasing the comparative statistics and visualizations between selected subjects and a control group. Through this demonstration, the audience will comprehend the real-time analytic capabilities of the W4H Toolkit and how it can be utilized to derive immediate insights for real-time decision-making.

## V. Conclusion

The W4H Toolkit is an end-to-end solution for managing and analyzing health-related wearable data and translating data into actionable healthcare insights. With our future work, we plan to extend the toolkit to support additional data types and streams, introduce more customized options, and expand the Learning and Intelligence Layer to include evaluation mechanisms for comparing the performance of deployed ML/DL models. Through these additions, we aim to promote a more data-driven approach to healthcare.

## VI. Acknowledgments

Research supported by the National Institute of Health (NIH) under grant 5R01LM014026. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NIH.

## References

[1] M. T. Bianchi, "Sleep devices: wearables and nearables, informational and interventional, consumer and clinical," *Metabolism*, vol. 84, pp. 99–108, 2018.

[2] D. Johansson, K. Malmgren, and M. Alt Murphy, "Wearable sensors for clinical applications in epilepsy, parkinson's disease, and stroke: a mixed-methods systematic review," *Journal of neurology*, vol. 265, pp. 1740–1752, 2018.

[3] M. Lang, "Beyond fitbit: a critical appraisal of optical heart rate monitoring wearables and apps, their current limitations and legal implications," *Alb. LJ Sci. & Tech.*, vol. 28, p. 39, 2017.

[4] J. M. e. a. Pevnick, "Wearable technology for cardiology: An update and framework for the future," *Trends Cardiovasc. Med.*, vol. 28, no. 2, pp. 144–150, 2018.

[5] C. Anastasiou, J. Lin, C. He, Y.-Y. Chiang, and C. Shahabi, "Admsv2: A modern architecture for transportation data management and analysis," in *ACM SIGSPATIAL '19*, 2019, pp. 25–28.

[6] M. e. a. Zaharia, "Apache spark: A unified engine for big data processing," *Comm. of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[7] Streamlit, "Streamlit: A faster way to build and share data apps," https://www.streamlit.io/, 2023, version 1.24.1.

[8] I. Timescale, "Timescaledb: An open-source time-series sql database optimized for fast ingest and complex queries," 2021, https://www.timescale.com/.

[9] P. P. S. Committee *et al.*, "Postgis, spatial and geographic objects for postgresql," 2018. [Online]. Available: https://postgis.net

[10] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[11] K. J. et al., "geopandas/geopandas: v0.8.1," Jul. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3946761

[12] A. Hajisafi, H. Lin, Y.-Y. Chiang, and C. Shahabi, "Dynamic gnns for precise seizure detection and classification from eeg data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2024.