



**POLITECNICO**  
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

# Intelligent Event-Driven Orchestration for Multimodal Digital Twins: Integrating Environmental and Physiological Sensing in Shared Spaces

LAUREA MAGISTRALE IN COMPUTER SCIENCE AND ENGINEERING - INGEGNERIA INFORMATICA

Author: CLAUDIA GALANTINO

Advisor: PROF. MARCO DOMENICO SANTAMBROGIO

Co-advisor: SUSANNA BARDINI

Academic year: 2024-2025

---

## 1. Introduction

Today, people spend approximately 90% of their day in indoor environments, such as offices, schools, and homes [1]. Although modern architecture focuses on energy efficiency and aesthetic design, the invisible link between indoor conditions and human cognitive performance remains largely unaddressed. Emerging researches [2] suggests that even moderate levels of  $CO_2$  accumulation and air pollution can significantly affect cognitive functions, increase perceived stress, and reduce overall wellbeing. Despite the widespread adoption of wearables and IoT sensing, current implementations treat these data streams independently. The primary challenge is **data fragmentation**: physiological sensors and environmental IoT systems collect data separately, creating vast repositories of information that lack a unified context. The motivation behind this research lies in the belief that biometric data acquire a different meaning when coupled with the environmental context in which they were generated. The main objective of this thesis is to **develop a scalable framework** capable of **ingesting and synchronizing heterogeneous data streams** to monitor human-environment interaction. By leveraging

**Apache Kafka** as the core messaging backbone, the framework overcomes traditional synchronization latencies and ensures the high throughput required to define the technological **backbone for a human-centric Digital Twin**.

## 2. Proposed Framework

The proposed framework overcomes the limitations of independent monitoring systems [3] by introducing a unified, **Event-Driven Architecture for shared workspace**, where multiple independent users (providing unique physiological events) occupy one common environment (providing a single stream of ambient events). To bridge the gap between these heterogeneous sources, the framework ensures both **spatial and temporal coherence**. Rather than relying on traditional batch architectures [4], this architecture treats data as a continuous stream of notifications, enabling **low-latency processing** and **high fault tolerance** in distributed environments. Figure 1 provides a high-level overview of the proposed Event-Driven Architecture and its main components. It highlights how physiological events, from multiple users, and ambient events, from the shared environment, are ingested, synchronized, and orchestrated to

maintain a coherent digital shadow.

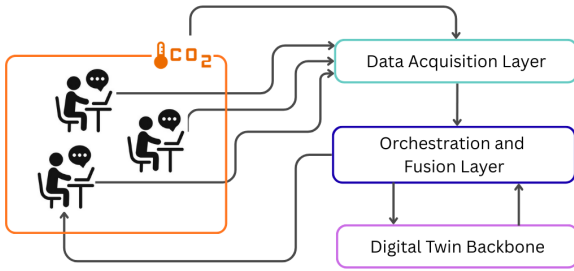


Figure 1: High-level overview of the proposed Event-Driven Architecture, showing the data acquisition layer (wearable and environmental sensors), the orchestration and fusion layer (Kafka-based stream coordination and context-aware mapping), and the resulting digital-twin backbone used to maintain a synchronized user-environment state.

## 2.1. The Intelligent Orchestrator and Context-Aware Logic

The logical core of the system is the Intelligent Orchestrator, which acts as the coordinating and decision-making entity. Unlike simple automation, the orchestrator maintains a global vision of the system, ensuring that data dependencies are respected and that workflows are triggered only when specific conditions are met. The framework's context awareness is achieved through a shift from "data-ready" to "**context-triggered**" logic. The Orchestrator evaluates the incoming event stream and manages the lifecycle of the user's digital representation. To reach this objective, the process begins with **identity mapping** where the system translates user identification across the pipeline to ensure that physiological data is always correctly contextualized by the specific room the user is working from. The system employs **presence validation** system to optimize computational resources. In fact, data fusion and analytical modules are only activated when a user is confirmed to be in a monitored room. Moreover, the Orchestrator acts as an **intelligent producer**, publishing trigger events to machine-learning modules only when it detects significant patterns, such as spikes  $CO_2$  coinciding with variations in heart rate.

## 2.2. Multimodal Data Fusion

The framework addresses the  $N$ -to-1 mapping problem: in a shared workspace, multiple independent users (providing unique physiological events) occupy one common environment (providing a single stream of ambient events). To bridge the gap between these heterogeneous sources, the framework ensures both spatial and temporal coherence:

- **Spatial Coherence:** By leveraging a real time state-store (Redis), the system routes common environmental metrics only to the profiles of users currently present within that specific space.
- **Temporal Coherence:** To resolve the asynchronicity between real-time IoT sensors and bursty wearable data, the system employs a **backward synchronization strategy**. Each physiological sample is associated with the last valid environmental state recorded immediately prior to that timestamp, ensuring the biometric response is correlated with the exact environmental stimuli present at that moment.

By combining physiological metrics like heart rate and respiration with environmental indicators such as  $CO_2$ , TVOC, temperature and humidity, the system builds a holistic representation of the human-environment interaction. In its current state, the framework serves as a functional digital shadow of the user's wellbeing. It provides the necessary **backbone for a Digital Twin** by maintaining a synchronized virtual state, laying the infrastructure for future bidirectional interaction and predictive actuation.

## 3. Implementation

The framework is implemented as a distributed architecture composed of isolated, containerized microservices managed via Docker Engine and Docker Compose. This setup ensures system resilience through a **restart: always** policy, preventing total system failure if an individual module crashes. The infrastructure is deployed on an Ubuntu 24.04.3 LTS Virtual Machine with 32 CPU cores and 64 GB of RAM to handle high-throughput bursts from multiple concurrent users.

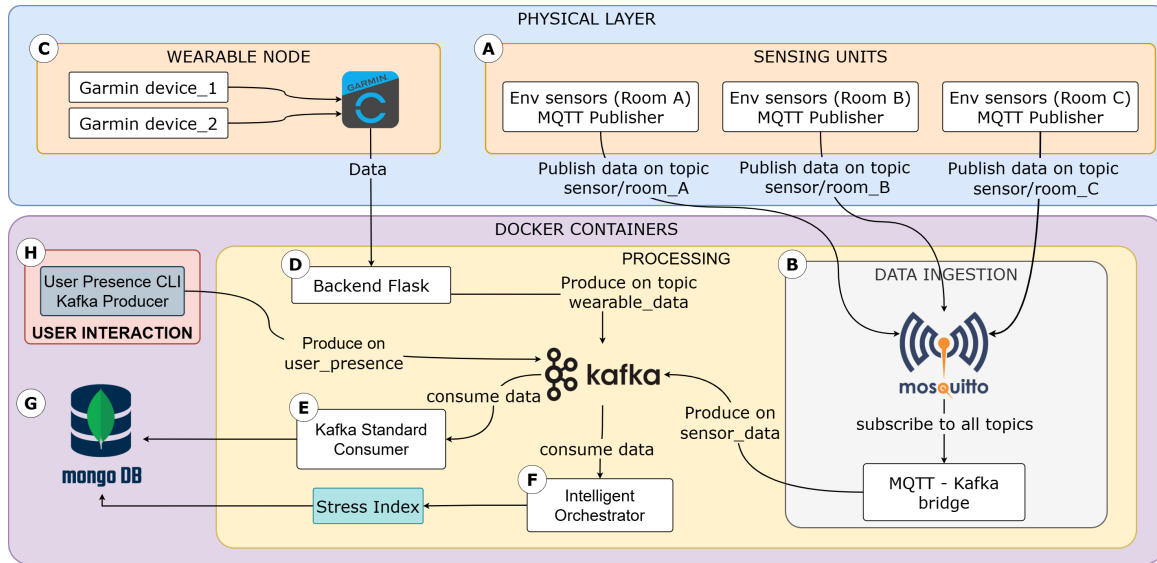


Figure 2: System architecture of the distributed microservice-based framework. The diagram shows the end-to-end data flow from the environmental sensing layer (A-B) (MQTT-enabled Raspberry Pi nodes) and the wearable layer (C-D) (Garmin Health API/webhooks) through the messaging backbone (E-F) (Apache Kafka topics for sensor, wearable, and presence streams), up to the persistence layer (G) (MongoDB) and the real-time state store (H) (Redis) used for user-to-room mapping and orchestration.

### 3.1. Data Ingestion and Messaging

As shown in Figure 2, the ingestion layer serves as the bridge between the physical world and the digital pipeline, handling two distinct data streams. The first layer is the **Environmental Layer**, composed of *sensing units* based on Raspberry Pi 4 that act as edge gateways. Each unit is equipped with DHT22 (temperature and humidity) and SGP30 ( $CO_2$  and TVOC) sensors and functions as an MQTT publisher, sending measurements every 20 minutes to an **Eclipse Mosquitto broker** to optimize power and network usage. The second layer is the **Wearable Layer**. Physiological data from Garmin Vivoactive 5 watches is captured via the Garmin Health API. Since Garmin data is bursty and cannot be pulled autonomously, the system relies on a **Flask-based ingestion gateway** to receive HTTPS POST notifications via secure tunneling. These streams are then unified by the **message brokering** layer. Here, **Apache Kafka** acts as the central backbone, decoupling producers from consumers. It manages three primary topics (`sensors_data`, `wearable_data`, and `user_presence`), each divided into three partitions to enable parallel processing and horizontal scalability.

### 3.2. Data Processing: The Intelligent Orchestrator

The processing is divided between two components. The "standard" consumer that is in charge of collecting raw data for a reliable storage, and the Orchestrator. The Intelligent Orchestrator is a specialized consumer-producer entity that transforms raw events into synchronized states. It focuses on three main features. The first is **state management**; the orchestrator maintains the real-time location of users by consuming events from the `user_presence` topic-triggered by a custom Command Line Interface (CLI) and storing the *User-to-Room* mapping in Redis. To reconcile high-frequency physiological bursts (e.g., heart rate every 15s) with sparse environmental telemetry (every 20 mins), the Orchestrator performs **backward synchronization**. It queries MongoDB to retrieve the last valid environmental state recorded immediately prior to each biometric sample's timestamp. The last main feature provided by the orchestrator is **context-aware triggering**, it operates as a decision-engine, activating computationally expensive machine-learning jobs only when specific criteria are met, such as confirmed user presence within monitored working hours (09:00-18:00).

### 3.3. Persistence Management

For long-term storage, the system utilizes **MongoDB**, which provides the schema flexibility required for evolving IoT JSON payloads. To minimize disk I/O overhead and enhance throughput, the "standard" consumer, through Batch Insert Strategy, buffers incoming Kafka messages and performs a single write operation every  $N$  messages. Persistent Docker Volumes are mapped to local directories on the VM for the MongoDB data, Kafka logs, and Orchestrator outputs, ensuring no data is lost during container updates or system restarts.

## 4. Experimental Validation

The experimental validation involved monitoring participants across different environments (Home, Lab, and Library) over approximately 30 days. The configuration details are listed in Table 1.

Parameter	Value – Description
<b>Participants</b>	3 Users
<b>Wearable Devices</b>	3 Garmin Vivoactive 5
<b>Monitored Rooms</b>	3 (Home, Lab, Library)
<b>Sensing Units</b>	1 Raspberry Pi 4 <i>sensing unit</i>
<b>Observation Period</b>	≈ 30 days
<b>Study Sessions</b>	09:00 – 18:00 (Active Monitoring)
<b>Sleep Sessions</b>	23:00 – 07:00
<b>Target Sampling</b>	Minute-level aggregation

Table 1: Overview of the experimental configuration used for framework validation.

### 4.1. Feature Engineering

The core challenge was to reconcile the data streams and address individual physiological differences, we implement a multi-stage feature pipeline. The process aims to transform asynchronous telemetry into a synchronized dataset ready for supervised learning. The raw data is first aggregated to a **minute-level granularity** to create a consistent temporal axis. Given the high sparsity of wearable metrics compared to the dense heart rate samples (≈ 20-minute intervals), a Forward-Fill (FFILL) strategy is applied to environmental and physiological states. Sleep data were integrated to personalize the model for each user, accounting for the subjective nature of physiological metrics.

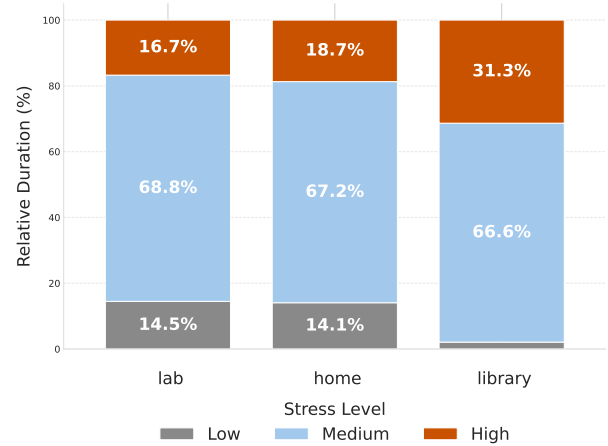


Figure 3: Heatmap highlights how the temporal distribution of stress episodes changes across the monitored environments (Home, Lab, Library), enabling a quick comparison of the time slots in which stress tends to be more frequent or more contained.

### 4.2. Statistical Results

The analysis on the data confirmed a correlation between our surroundings and user stress levels. In Figure 3, the heatmap provides an overall view of the linear dependencies among variables, highlighting that  $CO_2$  and TVOC tend to be positively associated with higher stress values. While the home and lab environments share similar "Low" stress profiles, the library presents a unique case. In fact, the user experienced higher stress due to higher temperature (≈ 29 °C) and increased  $CO_2$  levels (≈ 520 ppm), caused by the higher number of people present. Figure 4 complements the aggregate analysis by reporting an example of feature-specific behavior. By looking at the home environment, we see a weak negative correlation ( $r = -0.295$ ) between temperature and stress. As the temperature slightly increased, the stress scores showed a tendency to decrease. This suggests that even minor thermal adjustments can influence comfort, although fluctuations throughout the day indicate that temperature is just one piece of a much larger puzzle. The plot supports the intuition that comfort-related changes may be reflected in the physiological response. In fact, the study user experienced higher levels of stress when the room temperature was not comfortable. Although colder environments may increase focus, they can also increase stress (e.g., due to dis-

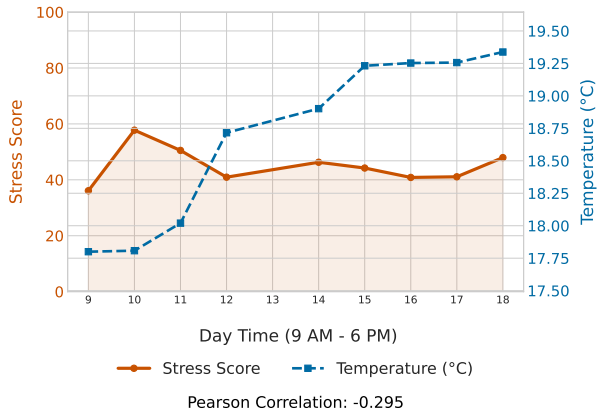


Figure 4: Scatter plot illustrating the relationship between temperature and stress in the Home environment.

comfort and shivering).

### 4.3. Stress Classification Proof of Concept

The predictive evaluation aimed to classify user stress levels into binary categories (Low vs. High) using a 5-minute-ahead forecast window, i.e., predicting future stress from the current multimodal sensor readings. The target variable was obtained by applying a binary split to `stress_score` with threshold set to 45, which yields a balanced class distribution: samples with `stress_score`  $\geq 45$  were labeled as “High Stress” (Class 1), while samples below the threshold were labeled as “Baseline/Low Stress” (Class 0). This horizon enables proactive interventions (e.g., automated ventilation or “break” alerts) before stress levels become critical. We benchmarked a set of widely used classifiers with different inductive biases, including tree-based ensembles (Random Forest and Extra Trees), gradient boosting methods (LightGBM and XGBoost), and a distance-based baseline (K-Nearest Neighbors). Among the evaluated approaches, Gradient Boosting models (LightGBM and XGBoost) achieved the best performance, with an accuracy of 80% and an F1-score of 0.79. A comparative summary of the main classifiers and their scores is presented in Table 2. In addition to physiological signals (with heart rate remaining a primary predictor), environmental metrics ( $CO_2$  and temperature) and sleep-history indicators improved both interpretability and predictive reliability. As

Model	Accuracy	F1 Score
<b>LGBMClassifier</b>	<b>0.80</b>	<b>0.79</b>
XGBClassifier	0.79	0.79
RandomForestClassifier	0.78	0.78
ExtraTreesClassifier	0.78	0.77
KNeighborsClassifier	0.77	0.77

Table 2: Classifier performance comparison on the binary stress classification task using a 5-minute prediction horizon. Accuracy and F1-score are reported to assess both overall correctness and the balance between precision and recall. LightGBM and XGBoost achieved the best performance.

shown in Figure 5, the importance profiles indicate that physiological features remain the dominant predictors, while environmental and sleep-related variables provide complementary information. In particular, environmental metrics can contribute to increasing or decreasing the estimated stress level by contextualizing the physiological response.

### 4.4. Architectural Efficiency

The Event-Driven Architecture proved to be highly scalable and resilient, maintaining performance stability even under the fluctuating operational loads typical of human-centric monitoring. By leveraging Kafka-based partitioning and message keys, the system achieved significant parallelism and guaranteed the consistent event ordering required for the precision of the backward synchronization logic. Furthermore, the intelligent orchestrator achieved important resource optimization by adopting context-triggered logic, activating the computationally expensive machine learning pipeline and fusion modules only during validated user presence windows. This selective processing ensured that the digital shadow remained semantically relevant while effectively preventing the processing of uncorrelated telemetry. Finally, to ensure long-term reliability, the persistence layer utilized a Batch Insert strategy to enhance MongoDB throughput, while Kafka’s native event persistence provided a robust mechanism for state recovery through stream replaying.

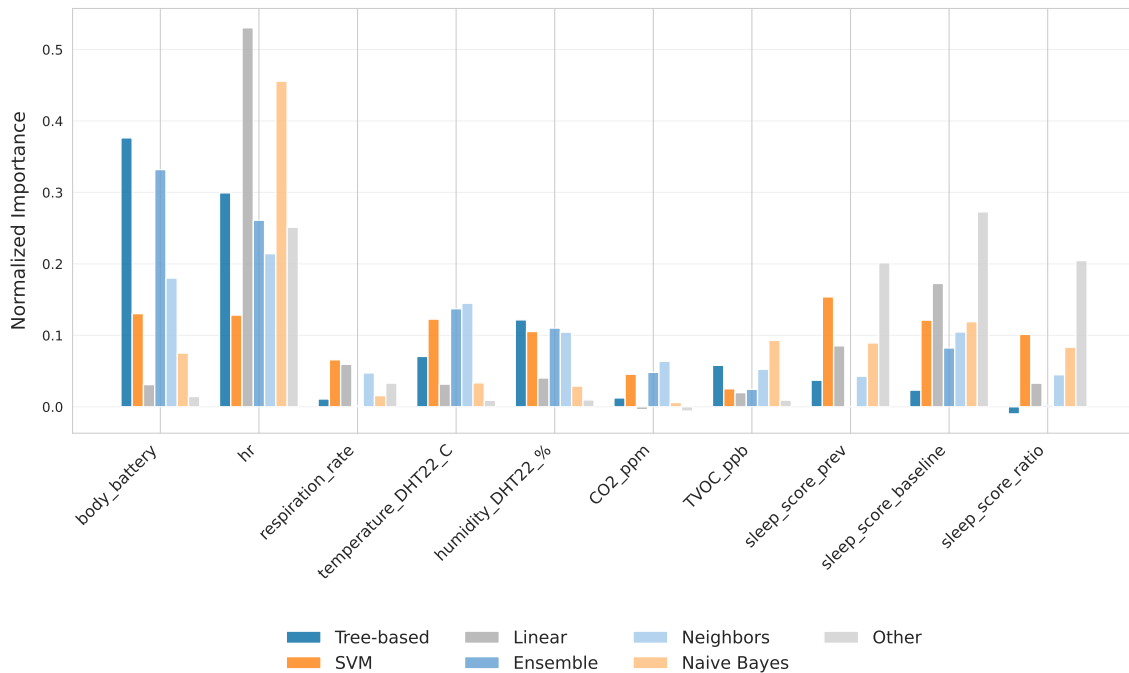


Figure 5: Feature usage by model family. It summarizes how different algorithm families leverage the input variables, highlighting the dominant role of physiological features and the contribution of environmental signals and sleep-related metrics in the final decision.

## 5. Conclusions

This thesis presents a scalable Event-Driven Architecture that synchronizes high-frequency biometric data from Garmin wearables with low-frequency environmental telemetry through a central Intelligent Orchestrator. Using Apache Kafka and MQTT, the framework overcomes data fragmentation and temporal mismatches, enabling a robust backbone for a human-centric Digital Twin. Experimental results show a correlation between indoor air quality and physiological stress, with tree-based machine learning models achieving a prediction accuracy of approximately 80%. The system’s microservice based design and use of schema-less storage ensure horizontal scalability and a wearable-agnostic approach, allowing for the integration of diverse sensing devices without altering the core pipeline. Although the framework currently functions as a reactive platform, future developments will focus on enabling bidirectional interaction for automated environmental actuation, such as HVAC control, while incorporating deep learning and federated learning to enhance both predictive power and data privacy. Finally, user interaction will be improved by developing a dedicated mobile application to provide real-

time, personalized feedback directly to the user.

## References

- [1] Mohamed Sazif Mohamed Subri, Kadir Arifin, Muhamad Faiz Aiman Mohd Sohaimin, and Azlan Abas. “The parameter of the Sick Building Syndrome: A systematic literature review”. In: *Helicon* 10.12 (June 2024), e32431. ISSN: 2405-8440. DOI: 10.1016/j.helicon.2024.e32431.
- [2] Bowen Du, Marlie C. Tandoc, Michael L. Mack, and Jeffrey A. Siegel. “Indoor CO2 concentrations and cognitive function: A critical review”. en. In: *Indoor Air* 30.6 (2020), pp. 1067–1082. ISSN: 1600-0668. DOI: 10.1111/ina.12706.
- [3] Luan Lazzari and Kleinner Farias. *Uncovering the Hidden Potential of Event-Driven Architecture: A Research Agenda*. arXiv:2308.05270 [cs]. Aug. 2023. DOI: 10.48550/arXiv.2308.05270.
- [4] Ritesh Kumar. “Event-Driven Architectures for Real-Time Data Processing: A Deep Dive into System Design and Optimization”. In: 9 (Mar. 2023), pp. 1–18. DOI: 10.5281/zenodo.15026990.