Accelerometer and Gyroscope based IoT System for Activity Classification

Séamus Knightly
1MECE1
University Of Galway
Galway, Ireland
s.knightly1@universityofgalway.ie

Seán Kelly
1MECE1
University Of Galway
Galway, Ireland
s.kelly178@universityofgalway.ie

Abstract—**This document is a model and instructions for LaTeX. YOLKS**
Index Terms—**THINGS, GO, HERE, PLEASE.**

I. INTRODUCTION

A. Problem Statement

1) Current Situation: Real-time monitoring of human activity has valuable applications in healthcare, helping to improve patient care, reduce the workload on healthcare staff, and lower costs while enhancing quality of life [1]. The INMO has stated that the nursing shortage which began in 2007 continues to worsen into 2025. This shortage continues to affect rehabilitation facilities, senior living facilities, and other areas where patient movement is key to recovery and general health.

2) Need for IoT System: Despite this potential, many current monitoring solutions remain limited. Systems such as pendant alarms, deployed through initiatives like the Seniors Alert Scheme, require manual activation by the user in the event of a fall [2]. This dependency on user input makes them less reliable in emergency situations. Automatic fall detection systems still face challenges with false alarms and missed detections. One study reported that an accelerometer based prototype detected only 80% of real world falls, with false alarms occurring approximately once every 40 hours of use [3]. These technological limitations are compounded by staff shortages in the health and social care sector, which place additional strain on staff morale and reduce care quality [4]. As a whole these factors show that there is a need for better IoT based solutions that can provide continuous, accurate, and scalable monitoring.

3) Aim and Objectives: The aim of this project is to design and implement an IoT system for activity monitoring using inertial sensors. The objectives are to:

- Acquire accelerometer and gyroscope data using a Feathersense board.
- Transmit sensor data in real time via BLE to an IoT gateway.
- Perform preprocessing at the gateway, including feature extraction (e.g. jerk, magnitude).
- Prepare the data pipeline for integration with machine learning models as part of Deliverable II.

II. SYSTEM DESIGN

A. Block diagram of the IoT system

The overall architecture of the IoT system is illustrated in Fig. 1.

Fig. 1. Block diagram of the IoT system, showing the layered design from sensing to cloud integration.

[TODO: Put block diagram figure here.]

B. Simplified IoT architecture

The IoT system follows a layered architecture. At the perception layer, the FeatherSense board captures motion data from its built-in accelerometer and gyroscope. This data is transmitted via the communication layer, using the BLE GATT protocol, to the IoT gateway (Raspberry Pi 4). The gateway performs preprocessing of the data streams.

[@Sean: maybe put in a sentence here about what is done there, very brief of course.]

For Deliverable I, cloud integration is not implemented. However, the block design includes a future planned cloud layer for storage, machine learning, and analytics.

III. COMPONENT SELECTIONS

A. Sensor Node

The Adafruit FeatherSense nRF52840 [5] was provided as the designated sensor node. This board integrates multiple sensors (e.g., accelerometer/gyroscope, magnetometer, microphone, temperature, humidity, and pressure sensors) along with native BLE support. For this project, only the accelerometer and gyroscope were used, as these correspond directly with the HARTH dataset [6], [7] chosen for activity recognition. The other onboard sensors were not relevant to the project aims and were therefore excluded.

Although platforms like the Espressif ESP32 DecKitC are used widely in other IoT projects, offering Wi-Fi and BLE connectivity [8], they typically require an external intertial measurement unit (IMU) if accelerometer or gyroscope sensing is needed. This comes with an increased cost, complexity

and board count. The FeatherSense nRF52840 on the other hand comes with an integrated LSM6DS3 IMU, native BLE, and USB & Batter charging support in one package. This makes it more suitable for the project's requirements for a self contained inertial sensor node.

B. Gateway

The Raspberry Pi 4 served as the IoT gateway. This device was issued for this project, and it has the sufficient computing capacity, integrated BLE support, and compatibility with Python's data acquisition and processing libraries. **[@Sean: again, if you want to add anything here. I think from the brief maybe we do need to do some comparison, like what I did inthe last paragraph]**

C. Communication

BLE was the protocol used, as it is the only wireless option natively supported by the FeatherSense nRF52840 board. An initial implementation used BLE UART with CBOR encoding. However this was too slow. To meet the 50 Hz sampling target, the system was reimplemented using BLE GATT, transmitting fixed length packets of six floating point values. This achieved stable operation at the required frequency.

D. Power

For prototyping, the FeatherSense was powered via USB, which ensured stability and ease of development. The board supports battery operation, but this was not implemented for Deliverable I, which focused on demonstrating data acquisition and transmission. In a full deployment scenario, low-power optimisation and battery usage would become essential for continuous wearable operation.

[@Sean: IDk I can look at the code and see if there is anything we can do to help with power consumption, just to say yeah in future we can do this. Not sure fully what to say on this point.]

IV. PROTOTYPE DESIGN PLAN

A. Subsystems

The prototype will be structured into four main subsystems:

- **Sensing:** inertial data acquisition from the FeatherSense accelerometer and gyroscope.
- **Communication:** wireless transmission via BLE, using GATT as the chosen protocol.
- **Edge Processing:** real-time data handling and basic feature extraction on the Raspberry Pi 4.
- **Visualisation:** plotting of data streams to confirm correct reception and interpretation.

This breakdown will ensure that each component of the system can be developed and validated independently before integration.

B. Integration Plan

The system will be developed incrementally. The sensor subsystem will first be validated by confirming raw inertial readings on the FeatherSense. Next, BLE transmission will be tested in isolation before final integration. Finally, the gateway will be configured to receive and interpret packets, after which end to end data flow from sensing to visualisation will be verified.

[@Sean: idk really, if you have any thoughts on the above two sections.]

C. Verification Plan

Verification activities will be designed to ensure that the prototype meets Deliverable I requirements. The main checks will include:

- Confirming that BLE transmission achieves the target 50 Hz sampling rate.
- Ensuring data packets are correctly structured and fully decoded at the gateway.
- Validating that received sensor values fall within expected physical ranges (e.g., acceleration values near ±9.8 m/s^2 for gravity).
- Measuring packet loss rates to assess communication reliability.

D. Verification Plan

Verification activities will be designed to ensure that the prototype meets Deliverable I requirements. The main checks will include:

- Confirming that BLE transmission achieved the target 50 Hz sampling rate.
- Ensuring data packets were correctly structured and fully decoded at the gateway.
- Validating that received sensor values were within expected physical ranges (e.g., acceleration values near ±9.8 m/s^2 for gravity).

[@Sean: again, some more stuff here maybe on what we are doing with regards verification. Will need to test packet loss. This verification section is just chatgpt nonsense atm]

V. IMPLEMENTATION

A. Feathersense Node Setup

The sensor node was implemented using the Adafruit FeatherSense nRF52840 board. The LSM6DS3 IMU was configured to sample at 50Hz, in line with the publicly available HARTH dataset. Sensor readings (acceleration in m/s^2 and angular velocity in deg/s) were acquired in real time through the Adafruit CircuitPython libraries.

B. Communication Performance

An initial implementation used the Nordic UART Service over BLE, with payloads encoded in CBOR format. This implementation was functional, but very inefficient. The structured CBOR packets were relatively large, so the transmission was slower and there was an increased coding and decoding

overhead on both device and gateway. As a result, the effective sampling rate was limited to approximately 11 Hz, and partial reads often caused incomplete or dropped data. The performance was too poor to be considered as an option, and further measurements of latency or packet loss were not taken in this configuration.

To address the issues found with the UART method, the system was reimplemented using a custom GATT service with two vendor-defined UUIDs (one for the service and one for the sensor characteristic). Instead of transmitting dictionary structures, the new approach packed six 32 bit floating point values (ax, ay, az, gx, gy, gz) into a fixed length 24 byte payload. This enabled stable streaming at the target rate of 50 Hz without the packet corruption or decoding delays observed previously.

[TODO: Record and report packet loss statistics in current implementation.]

VI. Implementation of Gateway Device

[@Sean: I'll leave this to you]

A. Data collection on Raspberry Pi

connection with Feathersense via BLE GATT.

B. Streaming Visualisation

live plotting of received sensor data.
(Screenshots/graphs of plots go here.)

VII. Edge/Fog Processing

Jerk calculation (derivative of acceleration).
Magnitude of accel/gyro.
Other features relevant for ML (you can mention RMS, variance, etc. if planned).
Show small plots of raw vs processed features.

A. Discussion & Conclusion

[@Sean: this once we've done the rest of course]
Summary: successful acquisition, transmission, preprocessing.
Limitations: still simulated/early prototype, real-world testing needed.
Next steps (for Deliverable II): deploy ML in cloud, integrate with Power BI.

References

[1] F. Serpush, M. B. Menhaj, B. Masoumi, and B. Karasfi, "Wearable sensor-based human activity recognition in the smart healthcare system," *Computational Intelligence and Neuroscience*, vol. 2022, p. 12, Feb 2022.

[2] H. S. E. (HSE), "Seniors alert scheme," https://www.hse.ie/eng/services/list/4/olderpeople/carersrelatives/community-support-for-older-people-.html, 2024, accessed: Oct. 19, 2025.

[3] M. Kangas, R. Korpelainen, I. Vikman, L. Nyberg, and T. Jämsä, "Sensitivity and false alarm rate of a fall sensor in long-term fall detection in the elderly," *Gerontology*, vol. 61, no. 1, pp. 61–68, 2015.

[4] E. Malone, "Staff shortages and public perception leading to healthcare worker burnout, fórsa research suggests," *The Irish Times*, May 2025, accessed: Oct. 19, 2025. [Online]. Available: https://www.irishtimes.com/health/2025/05/15/healthcare-burnout-forsa/

[5] Adafruit Industries, "Adafruit feather nrf52840 sense - overview," https://learn.adafruit.com/adafruit-feather-sense/overview, accessed: Oct. 19, 2025.

[6] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "Harth: A human activity recognition dataset for machine learning," *Sensors*, vol. 21, no. 23, p. 7853, Dec 2021.

[7] K. Bach, A. Logacjov, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "A machine learning classifier for detection of physical activity types and postures during free-living," *Journal for the Measurement of Physical Behaviour*, vol. 1, no. aop, pp. 1–8, Dec 2021.

[8] Maker Advisor, "Esp32 development boards review and comparison," https://makeradvisor.com/esp32-development-boards-review-comparison/, 2023, accessed: Oct. 19, 2025.