Accelerometer and Gyroscope based IoT System for Activity Classification and Monitoring

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—IoT, Remote Patient Monitoring, Accelerometer, Gyroscope, Classification, Assisted Living

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 [2]. This shortage continues to affect rehabilitation facilities, senior living facilities, and other areas where patient movement is key to recovery and continued 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 [3]. 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 [4]. 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 [5]. These factors show that there is a need for reliable IoT based solutions that can provide continuous, accurate, and scalable monitoring, which can serve as the foundation for a real-time fall detection monitoring system. This project aims to build a reliable system which fulfils a real-world need, and provides a foundation for future work in this area.

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

- Acquire accelerometer and gyroscope data from an IoT device
- Transmit sensor data in real time to an IoT gateway
- Perform preprocessing at the gateway, including filtering, windowing, and feature extraction
- Push the data to a Cloud-based Machine Learning and Data Analytics platform for classification and storage
- Implement monitoring of the time-series data with configurable alarms based on healthcare needs

II. System Design

A. Design Requirements

There are several key requirements which must be considered when designing wearable IoT devices:

- Long operational period / battery life
- Medium-range communication - 10-30 meters including walls
- Reliable data transmission
- Accurate sensing - for activity classification, accurate motion sensing
- Comfort - wearables need to be small and lightweight

The device must be able to run for at least 24 hours, ideally much longer, between charges to avoid frequent patient disturbance and unnecessary additional work for healthcare workers. The device ideally needs to be able to communicate at medium-range including at least one wall in a hospital or senior living environment if it is to be an effective activity tracker. With the goal of the project being real-time classification and alerting, the data stream needs to be consistent and reliable, not reconnecting frequently or dropping packets. Additionally, the sensors employed for motion sensing, mainly accelerometers and gyroscopes, must be precise enough to allow the classifier to distinguish between different activities, however this is less of a concern as smartphones have had this technology for over a decade. The component selections outlined in Section III aim to fulfil or exceed all of these requirements where possible, whilst also outlining a suitable prototyping platform for the device.

B. Block diagram of the IoT system

The overall architecture of the IoT system is illustrated in Fig. 1. The future expansion of the project is represented by the blocks with dotted outlines.
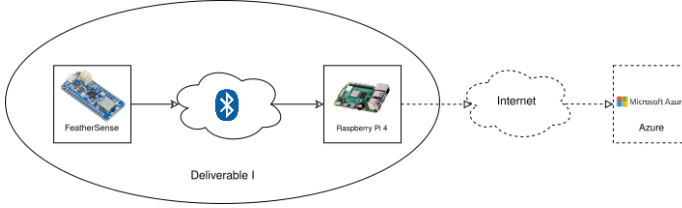

Fig. 1. Block diagram of the IoT system, showing the layered design from sensing to future cloud integration.

C. Simplified IoT architecture

The system follows a simplified three-layer IoT architecture. The Sensing and Perception Layer is composed of a wearable sensor node, a compact hip-worn device with a 6-axis IMU for motion capture, where 3 axes are provided by the accelerometer and 3 axes are provided by the gyroscope, giving access to detailed movement information. The device also features a microcontroller with an accompanying BLE module, a battery management system module, and a rechargeable battery.

The Gateway / Edge Compute Layer features an IoT Gateway with a BLE module for collecting data from multiple sensor nodes, internet connectivity (by WiFi or Ethernet), and a powerful system-on-chip capable of data filtering, windowing, and feature extraction in real-time for multiple wearable devices. It is intended to be a stationary gateway mounted in a central location where it can capture data from many devices, process it, and push it to a cloud data processing pipeline.

The Application Layer consists of a machine-learning and data analytics pipeline where a machine learning model classifies the current activity using the received data. The classification, time, and received data are saved in a time-series database for long-term storage, allowing healthcare professionals to access historical activity data for their patients, and configure alerting for patients who are idle for too long or who may be over-active during their recovery.

III. COMPONENT SELECTIONS

A. Sensing module

The BMI160 Low-Power 6-Axis IMU [6] combines a 3-axis accelerometer and a 3-axis gyroscope, both supporting $\pm2G$, $\pm4G$, $\pm8G$, and $\pm16G$ modes, configurable over its serial interface. It uses a simple I2C interface and can operate at voltages between 1.7 and 3.6v, consuming only 1mA when both the accelerometer and gyroscope are active at a 100Hz sample rate. It can be configured to operate at lower sample rates and uses just $180\mu A$ when in accelerometer-only mode at 100Hz. It also features intelligent power management and can go into a sleep state during rest, reactivating the gyroscope only when movement is detected, further reducing power consumption. Bosch advertises this device as an ideal choice for wearable IoT devices.

Altogether this sensor meets the ultra-low power requirement, has more than sufficient resolution at 16-bit, provides all essential data for the machine learning pipeline, and is a widely used option in the field.

The BMI270, also from Bosch SensorTec, is the next-generation model of the BMI160, providing more accuracy, self-calibration, and movement recognition, and while some of these features may be useful in this context, it does not justify the increased cost, significant increase in firmware complexity due to its boot-time configuration requirement, and higher power draw.

Similarly, the MPU-6050, a highly well-known IMU in industry, also elicits a 4x higher power consumption than the BMI160 in a larger package - at 3.9 mA, it is still a viable choice, however with the wide availability of the BMI160, it remains the ideal choice.

B. IoT Node

The Nordic nRF52810 microcontroller [7] is an ideal choice for powering the wearable IoT node. It features a 32-bit ARM Cortex-M4 processor running a 64 Mhz, and 192kB flash memory with 24kB of RAM. This is more than sufficient for simple data collection, packaging, and transmission. It also features integrated BLE 5.0 meaning an external module is not needed. It consumes a miniscule $2.0\mu A$ at 3.3 volts when idle, and peaks at just 4.6mA when transmitting or receiving data, making it an ideal choice for a low-power wearable IoT device. The chip also supports ANT, and has a 2.4Ghz transceiver which can be used to develop proprietary protocols if needed.

Another option which was considered is the STM32L031 - an excellent low power option which costs slightly less than the nRF52810, however, this microcontroller does not feature an on-board BLE module, thus ultimately raising the price and design complexity by requiring an external communication IC.

Other platforms like the Espressif ESP32-C3 are used widely in IoT projects, offering Wi-Fi as well as BLE connectivity, more processing power, and cryptography acceleration [8], they typically have a much higher power requirement, as much as 130mA while active. Using these in production comes with an increased cost for no benefit and vastly diminishes battery life.

The Adafruit FeatherSense nRF52840 [9] was chosen as a prototyping board for the project. This board provides a microcontroller from the same family as the production microcontroller, and 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 sensors correspond with the data requirements of the HARTH dataset [10], [11] powering the activity classifier in the Cloud IoT pipeline.

C. Gateway

For this project, the Raspberry Pi 4 serves as the IoT gateway. The Pi 4 Model B 4GB RAM variant features a

Broadcom BCM2711, a System-on-Chip containing a quad-core ARM Cortex-A72 processor. All Pi 4 Model Bs feature LPDDR4 memory running at 3200Mhz. It features on-board 2.4Ghz and 5.0Ghz WiFi, Gigabit Ethernet, Bluetooth 5.0 and BLE 5.0, and a Micro SD card slot. The device consumes a maximum of 15w under load, and takes up quite a small space. With the amount of processing available on-board, and at a cost of only $60 USD, the Pi 4 is an ideal choice for communicating with, and handling data from a large volume of IoT nodes. The Pi 4 runs a full Linux environment by default but can easily be customized to run a stripped back Linux distribution designed solely to run IoT Gateway functions. As it uses Linux, it natively supports Python for prototyping and C/C++ for production deployments as standard, with a wide variety of libraries available for BLE communication, MQTT for sending data to the cloud, edge machine learning libraries, and much more.

Additionally, the Raspberry Pi can be accessed remotely and easily serviced by means of re-imaging should anything go wrong, or if remote diagnostics / application support is required.

An industrial IoT gateway device such as a RAD IoT Gateway could run from hundreds to thousands of euro, and is completely overkill for a small-to-medium scale deployment like in a hospital environment or nursing home. Additionally, it comes with vendor lock-in and increased development times due to is proprietary nature.

D. Communication

Bluetooth Low Energy 5.0 (BLE 5) was selected for communicating between the IoT Device and the Gateway. The nRF52810 contains an integrated BLE 5 communication module, with a transmit power of up to 4 dBm. With this power, 10-30m range including obstacles can easily be done, at data rates of up to 2 Mbps if necessary. Additionally, BLE is designed for many-to-one communication, making it easy to connect many devices to the Raspberry Pi 4 IoT Gateway.

Other options considered were Zigbee, which is a very low power mesh-networking solution, however it requires a separate network coordinator IC not available on the nRF52810 or the Raspberry Pi. Additionally, it features much lower data rates, and a mesh network is not needed for a deployment of the proposed size.

WiFi HaLow was also considered initially due to its long range and configurable data rates, however it also requires external hardware and has a much higher power requirement than BLE 5, making it a less than ideal choice for a wearable device.

The Bluetooth Low Energy protocol is also natively supported by the FeatherSense nRF52840 board. The initial implementation of sensor data communication used BLE UART with Concise Binary Object Representation (CBOR) encoding - "The Concise Binary Object Representation (CBOR) is a data format whose design goals include the possibility of extremely small code size, fairly small message size, and extensibility without the need for version negotiation." [12]. Ulimately, the UART was too slow to meet the 50 Hz sampling target, and the system was reimplemented using BLE GATT at a data rate of 52 packets per second, where fixed length packets of six floating point values were used. This allowed stable data transmission of at least 50 samples per second for each each of the 6 axes of the IMU.

E. 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 inneficient. 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] Irish Nurses and Midwives Organisation, "Pre-budget submission 2025," https://www.inmo.ie/Portals/0/Documents/Public/Publications/INMO%20Pre%20Budget 2024, accessed: Oct. 21, 2025.

[3] 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.

[4] 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.

[5] 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/

[6] Bosch SensorTec, "Bmi160 datasheet," https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi160-ds000.pdf, accessed: Oct. 22, 2025.

[7] Nordic Semiconductor, "Nordic semiconductor nrf52810 system-on-chip," https://www.nordicsemi.com/Products/nRF52810, accessed: Oct. 22, 2025.

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

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

[10] 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.

[11] 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.

[12] Carsten Bormann, "Rfc 8949 concise binary object representation," https://cbor.io/, accessed: Oct. 22, 2025.