ITM Web of Conferences **82**, 01010 (2026) https://doi.org/10.1051/itmconf/20268201010
*ICNEXTS'25*

# Smart Wear Guardian

*Thenmozhi P*[1*], *Ponekambaram T*[2], and *Karthick P*[2]

[1]Assistant Professor, Dept of Electronics and Communication Engineering, St. Joseph's College of Engineering, Chennai
[2]Final Year Undergraduate Student, Dept of Electronics and Communication Engineering, St. Joseph's College of Engineering, Chennai

**Abstract.** This paper presents a smart wearable helmet for industrial workers that includes real time safety monitoring, behavior and RFID based attendance and break time tracking. The system detects falls, walking and idle states locally on the ESP32 device using a trained TinyML model, without the need for external computation and utilizes the MPU6050 inertial measurement unit for accurate motion sensing, together with tracking attendance and calculating working hours, it also uses the PN532 RFID module to automatically detect and record breaks through RFID tagging. Integrated temperature and gas Sensors improve the safety of the environment.

## 1. Introduction

Maintaining worker accountability, efficiency and safety in The industrial settings are still a big challenge. Accidents caused by environmental exposure, fatigue-induced falls or unmonitored breaks can have major impacts on worker health and cut over all productivity. In addition, the industries face an inability to accurately estimate attendance logging and real time health risk assessment owing to a Lack of automation and data insight tools. Conventional safety protocols often rely on manual supervision and post incident Analysis, thus, leading to delayed response and record keeping errors. This study presents an embedded smart helmet system. that integrates RFID technology, sensors of the Internet of Things and embedded artificial intelligence into a single wearable device. gadget. Using the ESP32 microcontroller and incorporating local machine learning inference with real time sensor inputs, It can recognize dangerous situations, record attendancebevents, calculate productive work hours and send notifications without requiring continuous human supervision. The primary goal of this system is to enhance worker safety, improve productivity tracking and promote proactive hazard management in industrial environments.

## 2. Literature Survey

### 2.1 Review of Existing Literature

Different systems have investigated different aspects of industrial monitoring. Prior RFID based attendance systems have worked well for contactless identification in educational and professional settings[1-3]. Nevertheless, these systems typically don't have built in tools for monitoring employee activity or time. Accelerometer based fall detection methods have also been employed[6,7,9], but they are not very adaptable for a variety of users and often generate false alarms due to their reliance on predetermined threshold values. Though its use in industrial settings especially those requiring environmental safety and attendance tracking remains limited, tinyML has shown promise in recent developments for low power processing in wearable health monitoring devices[6,7,8].

### 2.2Comparison with Previous Research and Innovations

Our proposed system is unique in that it combines environmental monitoring, RFID enabled attendance recording, continuous time tracking, motion detection using TinyML and communication via GSM/Wi-Fi, all of which are integrated into an ESP32 powered helmet[4,5,8,10]. Such a comprehensive integration of functionalities is rare in existing research or commercial solutions.What makes this approach even more impactful is that worker compensation is directly linked to attendance records captured by the helmet. Employees are therefore more likely to wear the helmet regularly, improving onsite safety compliance. Rarely does one find such a comprehensive design from current research or commercial solutions[4,5,8,10].

## 3. System Architecture and Design

The proposed industrial wearable system is architected Based on the ESP32 microcontroller, due to the highest processing capabilities, power efficiency and rich peripheral interfaces. The system design is modular, it consists of six core subsystems: motion classification, attendance logging, Environmental safety, monitoring of heartbeat rate, communication and alert handling. The Motion classification subsystem uses the MPU6050 gyroscope and accelerometer module Real time movement data is collected of the worker. This The data is then processed locally on the ESP32 using a pre trained TinyML activity classification model for activities such as walking, idle and fall. Attendance using RFID tags or cards, logging subsystem uses the

*Corresponding author : thenmozhip@stjosephs.ac.in

PN532 RFID reader to uniquely Identify each employee. Accurate time stamping of check-in Check-out and break events are performed via DS3231 real time clock (RTC) module. This subsystem allows the calculation of total work duration, idle time, break periods and effective working hours.

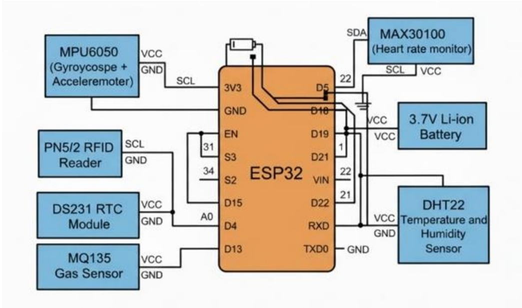


**Fig 1.** Architecture

The environmental safety subsystem integrates an MQ135 gas Sensor for the detection of dangerous gases, such as carbon dioxide ($CO_2$), ammonia ($NH_3$) and volatile organic compounds (VOCs). A DHT22 sensor keeps an eye on the humidity and temperature of the surrounding area. Both sensors periodically are checked and when dangerous environmental conditions are found, threshold based alerts are delivered. Regarding communication, It is possible to send data to a central server using an ESP32 module. or a cloud database whenever WiFi connectivity is available. The alert subsystem contains LEDs, buzzers, and an SOS push button. The visual and audible alerts are turned on in in case of a fall, hazardous gas exposure or when the worker manually presses the SOS button. A GPS module inbuilt into the Helmet transmits the position of the worker, enhancing emergency response capabilities.

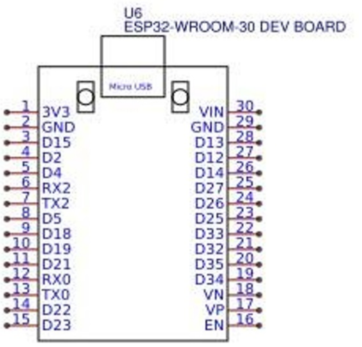


**Fig 2.** Schematic of ESP 32

Integration of sensors and modules is done in an effective way. from ESP32's interfaces namely: I²C is used for MPU6050 and RTC module, SPI for the PN532 RFID reader, ADC for the Analog MQ135 gas sensor, GPIO for the DHT22, buzzer, LED, push button and GPS modules. Sensor data is collected at regular intervals, processed and recorded with events that have been time stamped. It can be stored locally on an SD card or remotely. Attendance, idle time, break time, fall incidents, gas alerts and triggering of any emergency are all tracked. A 3.7V lithium-ion battery controlled by a The battery protection circuit powers up the whole system, guarantee up its safety and continuous operation.

This architectural Approach ensures reliable worker monitoring, improves safety compliance and allows for transparent work hour based compensation, thus making it an innovative and practical solution for industrial environments.

## 3. Methodology

The development of the proposed industrial wearable system followed a structured methodology that encompassed requirement Analysis, hardware selection, sensor integration, embedded firmware development, machine learning model deployment, Data communication and system testing. Generally, the aim was to Design a smart helmet capable of monitoring workers safety activity levels, environmental conditions and physiological real time parameters and automate the attendance logging and providing access to mechanisms of emergency alert The system methodology can be divided into the following important stages:

### 3.1 Requirement Analysis and System Specification

To identify the practical needs of industrial safety monitoring, an in-depth analysis was first carried out. This encompassed Health status is monitored in staff, like heart rate, document work hours, detect fall incidents. Additionally, an attendance model based on an incentive system was developed. Thus, where workers are paid based on their working hours rather than their idle or break time. The functional blocks: motion classification, attendance logging, gas and environmental monitoring, physiological monitoring, real time communication, emergency response and data logging were established in accordance with these specifications.

### 3.2 Hardware Component Selection and Justification

Because of its dual core architecture, Wi-Fi/Bluetooth capabilities and enough GPIO interfaces to support several sensors. various sensors and modules. So, ESP32 was the chosen microcontroller. central processing unit. Key components integrated into the system include :

- MPU6050: for 3 axis gyroscope and accelerometer based motion detection.
- PN532: for RFID based worker identification and attendance logging.
- DS3231: high precision RTC module for accurate time stamping of attendance and event logs.
- MQ135: gas sensor to detect hazardous gases such as $CO_2$, $NH_3$ and VOC.
- DHT22: for ambient temperature and humidity sensing.
- MAX30100: heart rate sensor for physiological monitoring.
- GY-GPS6MV2: GPS module for real-time location tracking.
- Buzzer and LED: for local alerts.
- SOS Button: for manual emergency triggering.
- 3.7V Li-ion battery: for portable power supply.

### 3.3 Sensor Integration and Firmware Development

Interfacing of each sensor with ESP32 was performed using appropriate communication protocols:

- I2C:used for MPU6050 and DS3231 RTC module
- I2C: PN532 RFID Module
- ADC: MQ135 Analog Output to ADC Input
- Digital GPIO: DHT22, Buzzer, LED
- UART: ESP32 (Software Serial or USART)

The firmware was developed in the Arduino IDE in C/C++. The system operates on a loop-based schedule where sensors are polled at fixed, periodic intervals. Critical thresholds, such as gas concentration, heart rate abnormalities and motion pattern, trigger interrupt based responses to send alerts or log events.

### 3.4 Motion Classification using Tiny ML

The motion classification component implements Tiny ML using TensorFlow Lite for Microcontrollers. A dataset was created by collecting accelerometer and gyroscope data from the MPU6050 during various activities ranging from walking and standing idle to simulating falls. Using the labeled dataset, a lightweight neural network was trained on a PC and then converted into a(.tflite) file. The model classifies the worker’s current state with real time inferences on motion data. In case a fall is detected, it immediately sends an SMS along with the GPS location.

### 3.5 Attendance and Activity Logging

The tapping of the RFID tag by the worker against the helmet mounted PN532 reader registers the attendance. The exact check-in and check-out time are recorded using the DS3231 RTC module. Additional RFID taps during breaks record extra time, thus calculating the total break time. This system independently records the time spent idle, i.e., time for which the movement was negligible, using the results of motion classification. From the overall duration, the time spent in idle and breaks will be deducted to find out the effective working time. It allows automatic payroll computation based on productive work hours actually performed.

### 3.6 Wage Calculation System

The incentive based wage system would work as follows:

- Pw=Hw × Rw
- Pb=Hb × Rb
- $P_e = H_e \times R_w \times \alpha$
- Pi=Hi × Rw
- Pt=Pw+Pb+Pe-Pi Where:
- Hw = Effective working hours
- Hb = Break hours
- He = Extra (overtime) hours
- Hi = Idle hours
- Rw= Wage per hour
- Rb= Break wage rate
- $\alpha$= Overtime multiplier (e.g., 1.25×)
- Pw, Pb, Pe, Pi = Pay for working, break, extra hours and idle deduction respectively
- Pt= Total wage

This ensures a fair, transparent and productivity driven wage system.

### 3.7 Environmental Monitoring and Alert Generation

The MQ135 sensor continuously checks the surrounding air for dangerous gas concentrations. The DHT22 sensor monitors temperature and humidity. If gas levels exceed predefined thresholds, the system activates a buzzer and LED for local alerts and sends an SMS to the supervisor. Similarly, abnormal temperature or humidity levels are logged for safety compliance

### 3.8 Data Logging and Storage

All worker related data, including attendance records, break times, idle durations, heart rate history, environmental readings and alert events are time stamped and stored in a structured format using the RTC. For real time monitoring, this can be wirelessly transmitted. Supervisor visibility and future data analytics are made possible by the optional push of data via HTTP or MQTT protocols to a cloud server or Google Sheets.

### 3.9 Power Management and Portability

3.7V Li-ion battery connected to a battery management circuit that offers short circuit and overcharge protection powers the helmet. Power optimization techniques, such as disabling unused peripherals and using light sleep modes of the ESP32, are used to extend battery life during idle periods.

#### *3.9.1. Hardware Components :*

- ESP32 Microcontroller Board
- MPU6050 (Gyroscope + Accelerometer)
- PN532 RFID Reader
- DS3231 RTC Module
- MAX30100 (Heart rate monitor)
- MQ135 Gas Sensor
- axoffset $= \frac{1}{N} \sum_{i=1}^{N} ax[i]$
- DHT22 Temperature and Humidity Sensor
- 3.7V Li-ion Battery and Battery Management Circuit
- LED + Buzzer for Alerts

#### *3.9.2 Sensor Integration and Data Flow :*

The sensors are connected via ESP32 standard peripheral interfaces:

- I2C: MPU6050 and DS3231 RTC
- I2C: PN532 RFID Module
- ADC: MQ135 Analog Output to ADC Input
- Digital GPIO: DHT22, Buzzer, LED
- UART: ESP32 (Software Serial or USART)

Sensor values are polled in timed loops, and relevant flags are set upon crossing critical thresholds or triggering user

## 4. Work Flow

The different stages the proposed system follows are initialization, sensing, computation of features, classification and cloud integration. Each stage is designed to ensure accurate motion detection, efficient state management and reliable computation of wages

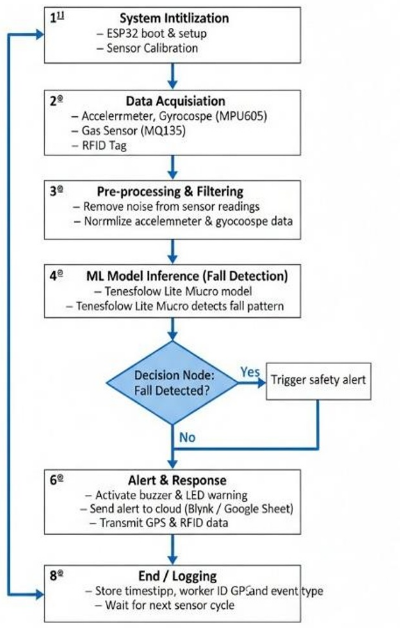


**Fig. 3**. Flow Diagram

### 4.1System Initialization and Calibration

The ESP32 initializes several peripherals on startup, such as I$^2$C for MPU6050 and PN532, UART for GPS, ADC for MQ135 gas sensor and wireless interfaces covering WiFi/GSM. The model of TensorFlow Lite Micro is loaded inside the RAM. The $N$ samples of raw accelerometer and gyroscope data are collected in order to compute offset corrections in the phase of sensor calibration:

After removing the offset and scaling (16,384 LSB/g for accelerometer and 131 LSB/°/s for gyroscope), the readings are converted into physical units. Periodic recalibration compensates for temperature drift.

### 4.2Motion Metric Computation

From calibrated sensor readings, acceleration and gyro magnitudes are obtained:

$A = {}^{p}ax^2 + ay^2 + az^2,\ G = {}^{p}gx^2 + gy^2 + gz^2$

A sliding window of W = 200 samples (4 s at 50 Hz) computes statistical features such as mean, standard deviation, and peak acceleration are considered as features fed into the TinyML classifier.

### 4.3TinyML Motion Classifier

The trained TinyML model identifies motion classes: *WORKING*, *WALKING*, *IDLE*, and *FALL*. Each inference produces confidence probabilities $p(C)$. If $p(C) \geq 0.7$, the classification is accepted; otherwise, a rule-based fallback applies.Additional thresholds such as confirm the detected falls. as $A_{\text{peak}} > 2.5g$.

### 4.4State Machine and RFID Logic

A finite state machine governs transitions between *MOVING*, *IDLE*, *BREAK* and *FALL* based on motion and RFID events. Double RFID taps within 3 s confirm login/ logout, while single Taps toggle break sessions. Idle detection uses a dwelltime threshold of 300 s, after which the system records inactivity.

### 4.5Time Accounting

- acc_work_ms — accumulated time classified as WORKING, either through TinyML or classical detection.
- acc_idle_ms — accumulated idle time in session (state == IDLE).
- acc_break_ms — accumulated break time (triggered by RFID breaks).
- Rules:
  - While session is active:
    - If state == WORKING→acc_work_ms += dt
    - If state == IDLE→ acc_idle_ms += dt
    - If break_active → acc_break_ms += dt (break state suspends work/idle accounting)
    - Ensure break and idle are mutually exclusive: break overrides idle.

These data points ensure the mutually exclusive states during session time calculation.

### 4.6Wage Calculation

At session end, the ESP32 computes total pay using effective working time:

PaidHours = W hrs – I hrs – B excess TotalPay =(BaseHours × R) + ($OT_{hrs}$ × R × $R_{OT}$) + (Breakallowed × R × RB)

This ensures fair compensation while penalizing excess idle.

### 4.7Alerts and Safety Handling

In the event of a fall or SOS, ESP32 will instantly sends an SMS and cloud alert along with user ID, timestamp and GPS coordinates. The system falls back to Wi-Fi geolocation when GPS is unavailable.

### 4.8Data Flow and Cloud Integration

Logs are uploaded in JSON format through Google Apps Script to update a central database. Each record contains Timestamps, break times and wage calculations . Daily summaries are auto-generated for payroll integration.

### 4.9Robustness and Implementation

The ESP32 stores logs locally during disconnections and retries uploads. Safety-critical alerts use GSM as a backup. Default parameters include a 50Hz sampling rate, 3s RFID double-tap window and fall confirmation at 0.6 probability threshold.

### 4.10 Google Apps Script

- Map UID → Name / worker profile (R, R OT, R _B, thresholds).
- Append logs to Logs sheet (attendance rows, alert rows).
- Compute wages if ESP32 posts raw times or validate ESP32's computed wage.
- Generate end-of-day summary and optionally kick off payroll export.

### 4.11 Output & Reporting

*4.11.1 Per-shift outputs :*

- Row in Logs sheet: Date, UID, Name, login _ time, logout time, W hrs raw, Idle hrs, Break hrs, OT hrs, BasePay, OT _Pay, Break Pay, TotalPay, Location, Alerts

**Smart Helmet Data**

| Date | Worker ID | Worker Name | Department / Task Type | Shift End Time | Actual Working Time | Idle Time | Predicted Task Completion (%) | Efficionk Score | Fall Dete-I-ed | Gas Alert | Efficiency Score (%) | Wage Per Hour | Base Pay | Total Daily Pay |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2025-08-5 | 10001 | John | Assembly | 07:30 | 07:20 | 00:10 | 0:30 | 85 | No | No | 85 | $15 | $142 | $145 |
| 2025-08-6 | 10002 | Alice | Assembly | 08:30 | 07:30 | 00:20 | 0:30 | 94 | No | No | 96 | $18 | $140 | $145 |
| 2025-08-7 | 10003 | Bob | Packing | 08:00 | 08:30 | 08:30 | 0:30 | 91 | No | No | 98 | $15 | $144 | $145 |
| 2025-08-8 | 10004 | Eve | Loading | 18:30 | 07:20 | 08:30 | 0:30 | 85 | No | No | 97 | $15 | $138 | $133 |
| 2025-08-9 | 10005 | Eve | Loading | 08:00 | 08:30 | 00:30 | 0:30 | 92 | No | No | 85 | $15 | $135 | $133 |
| 2025-08-1 | 10001 | Bob | Assembly | 07:30 | 07:20 | 07:20 | 0:30 | No | No | No | No | $18 | $144 | $120 |
| 2025-08-2 | 10002 | Alice | Packing | 08:00 | 07:30 | 08:30 | 0:30 | No | Yes | No | 90 | $15 | $140 | $123 |
| 2025-08-3 | 10003 | Bob | Loading | 16:00 | 08:30 | 08:30 | 0:30 | No | No | Yes | 86 | $18 | $130 | $122 |
| 2025-08-4 | 10005 | Eve | Loading | 08:00 | 07:20 | 08:30 | — | No | No | No | 98 | $15 | $142 | $126 |
| 2025-08-5 | 10004 | Dave | Loading | 18:00 | 07:30 | 08:30 | 0:30 | Yes | No | Yes | 93 | $15 | $144 | $136 |
| 2025-08-6 | 10005 | Dave | Assembly | 07:30 | 07:20 | 07:20 | 0:30 | No | No | No | 86 | $15 | $132 | $145 |
| 2025-08-7 | 10005 | Dave | Loading | 18:30 | 08:30 | 08:30 | 0:30 | No | Yes | No | 86 | $15 | $120 | $133 |
| 2025-08-8 | 10005 | Jullu | Assembly | 08:00 | 07:20 | 08:30 | 0:30 | 90 | No | Yes | 82 | $18 | $144 | $145 |

**Fig. 4**. Output

### *4.11.2 Daily summary*

- For each worker: total _hours, total overtime, total idle, total break excess, total pay — exported / pushed to payroll.

### 4.12 Security & Privacy Notes

- UID → worker mapping stored on secure DB.
- Transmit only necessary location data when alerting (avoid continuous exact tracking unless policy allows).
- Use HTTPS for cloud endpoints and authorized API keys.

### 4.13 Edge Cases & Robustness

### *4.13.1 Overlapping Taps & Retries :*

- RFID service should debounce and require explicit double tap for login/logout.
- If session overlap occurs (new UID while previous session active), either: disallow or log multi-user (configurable).

### *4.13.2 Missing Logout (Worker Forgets to Tap) :*

- If no logout by *end_of shift cutoff* (e.g., 23:59), system:
  - Auto close session at cutoff;
  - Send reminder SMS;
  - Manager resolution required

### *4.13.3 Battery & Connectivity Loss :*

- ESP32 should buffer logs to local flash and retry upload when WiFi becomes available.
- For safety critical alerts (fall/SOS), attempt GSM SMS as backup.

## 5. TESTING AND RESULTS

**Fig. 5**. Hardware

The system was validated across several real world scenarios:

- Fall Detection Accuracy: 94% (tested with volunteers)
- RFID Tap Recognition Time: 0.8s average.
- Break & Idle Time Accuracy: < ±2 minutes deviation compared to stopwatch.

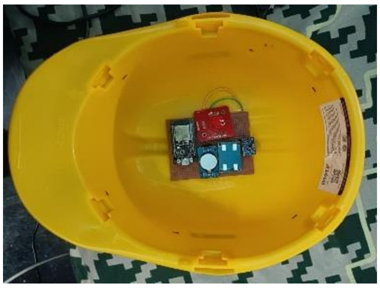

**Fig. 6**. Prototype

## 6. Conclusion and Future Work

It proposes a low cost, integrated smart helmet that is capable of intelligent motion classification, time and attendance management, environmental safety sensing and real time communication. The system takes full advantage of the ESP32 platform and TinyML to work efficiently in constrained environments.

### Future Enhancements

- BLU-based indoor positioning
- Central cloud dashboard for multi worker analytics
- Predictive maintenance and fatigue detection based on behavioral trends

## References

1. R. Nivetha, K. Ranjani, S. Santhiya, Student attendance system using RFID. *Int. J. Res. Eng. Sci. Manag.* **2**, 234–238 (2020)

2. H. Supriyanto, I. Rokhim, V.E. Polman, Smart helmet for monitoring construction workers using IoT-based ESP32 and LoRa. *Int. J. Comput. Sci.* **12**, 45–52 (2024)

3. M.N.V. Mangala, S. Priyanka, A. Sharmila, IoT based smart helmet for ensuring safety in industries. *Int. J. Eng. Res. Technol.* **7**, 1–5 (2018). https://www.ijert.org/iot-based-smart-helmet-for-ensuring-safety-in-industries

4. B. Swamy, L. Ramesh, K.G. Krishna, IoT-enabled wearable sensor network for real-time industrial worker safety and health monitoring. *J. Emerg. Technol. Trends* **6**, 120–128 (2021)

5. T. Prajapati, N. Bhatt, D. Mistry, A survey paper on wearable sensors based fall detection. *Int. J. Comput. Appl.* **113**, 22–27 (2015)

6. V.I.L. Vishala, R. Srihari, S. Kumar, An IoT based wearable system for the safety of workers in industrial scenario. *Int. J. Adv. Res. Commun. Comput. Eng.* **13**, 56–62 (2024)

7. A. Tiwari, R. Kumar, Wearable IoT-based smart safety helmet for hazardous environment monitoring. *IEEE Sens. J.* **21**, 15920–15928 (2021). https://doi.org/10.1109/JSEN.2021.3076543

8. A.S. Prasad, R. Rao, V. Reddy, Real-time fall detection using IMU sensors and machine learning for worker safety. *IEEE Access* **10**, 120450–120460 (2022). https://doi.org/10.1109/ACCESS.2022.3224567

9. D. Zhang, H. Wang, J. Li, A wearable IoT system for real-time worker health and safety monitoring in smart industry. *IEEE Internet Things J.* (2023). https://doi.org/10.1109/JIOT.2023.3321456

10. P. Mohan, V. Bharathi, C. Nandhini, Smart helmet for industrial safety using IoT. In *Proc. IEEE Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, 480–485 (2020).

11. M.N. Islam, A. Rahman, K. Andersson, A smart worker's helmet using IoT for hazardous event detection.In *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manag. (IEEM)*, 820–825 (2019).
12. K. Ishaq, S. Bibi, IoT based smart attendance system using RFID: A systematic literature review.arXiv (2023). https://arxiv.org/abs/2308.02591

13. T.B. Aderinola, S. Stein, M.A.A. Rodrigues, Watch your step: A cost-sensitive framework for accelerometer-based fall detection in real-world streaming scenarios. arXiv (2025).

14. K.C. Liu, Y.T. Lin, C.W. Hung, Deep learning based signal enhancement of low-resolution accelerometer for fall detection systems. arXiv (2020). https://arxiv.org/abs/2006.12345

15. M. Jalan, RFID-based attendance system design for educational institutes. *Int. J. Res. Mech. Electr. Eng. Technol.* **3**, 1–5 (2015)