International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

IOT- Based Elderly Fall Detection and Health Monitoring System

Aisha Abdul Shukoor, Sinchana K R, Shreya Gajipur
Under the Guidance of:
Dr. G. V. Sowmya B.E., M.Tech, Phd
Department of Information Science and Engineering
JNN College of Engineering, Shivamogga, Karnataka, India- 577201

Abstract— The IoT Based Elderly Fall Detection and Health Monitoring System Using Smart Footwear is a system where the elderly individual can be monitored for their movements, health, environment to ensure safety. Because the elderly are more susceptible to falls, they could suffer serious injury so having a fall detection system would help. This system uses smart footwear and IoT technology to monitor and detect if someone is about to fall. The system contains a variety of sensors such as an MPU6050 accelerometer and gyroscope, MAX30102 heart rate and blood oxygen sensor, Force Sensitive Resistor (FSR) to measure pressure of footwear, HC-SR04 ultrasonic obstacle detection and Light Dependent Resistor (LDR) to monitor light levels. The data is then processed by an Arduino Uno using threshold values and logic to determine if there is a risk of falling. The information is then sent via Wi-Fi using an ESP8266 NodeMCU to a web application written in Flask. The dashboard contains information about the sensors, system status and fall detection. A buzzer has been added to give local alerts to the user. This project shows how smart footwear, sensors, embedded systems and IoT can be combined to monitor the elderly for safety. It allows the elderly to be identified as having a risk quickly, and can be monitored remotely.
Index Terms- Fall Detection, Smart Footwear, Internet of Things (IoT), Wearable Sensors, Elderly Healthcare, Health Monitoring, Real-Time Monitoring.

I. INTRODUCTION

Elderly people are more at risk of falling. This can cause severe injuries, reduce their mobility and restrict their independence. Falls should be detected as soon as possible so that assistance can be provided. Thanks to recent advances in wearable sensors and the Internet of Things (IoT), automated fall detection systems can be implemented to monitor the elderly. These systems monitor movement and detect unusual activities. Issues with these devices include detecting false alarms, detecting falls accurately and ensuring reliable communication. The false alarms can easily be switched off manually by the user via false press button which switches off all the events that happen after a fall is detected. Reliable monitoring systems help caregivers respond quickly during emergencies.

Wearable fall detection systems commonly use accelerometers and gyroscopes to identify sudden changes in body movement and orientation. Different techniques such as threshold-based algorithms, machine learning techniques can be used to differentiate between falls and normal activities from the accelerometer data. IoT can be used to relay the sensor data and notify caregivers. These sensor locations, energy consumption and personal movement can affect how well the system works. Accurate detection and communication can help increase the safety of the elderly.

This project proposes an IoT-Based Elderly Fall Detection and Health Monitoring System Using Smart Footwear. The system integrates an MPU6050 accelerometer and gyroscope to monitor movement and orientation, along with a MAX30102 sensor for pulse related measurements. An FSR sensor measures pressure, an HC-SR04 ultrasonic sensor detects distance, and an LDR measures light intensity. An Arduino Uno processes sensor readings, while an ESP8266 NodeMCU provides Wi-Fi connectivity for remote monitoring. These components collect relevant movement, environmental, and health related data.

The proposed system provides fall detection, local alerts, and remote notifications to caregivers through IoT communication. The system will also detect falls and notify the local area using the buzzer and LEDs. Push buttons support user interaction and alert control. Embedding the sensors within footwear provides an alternative to wearable devices. The prototype will be tested to see how well it detects falls, false alarms, reliable communication and usability. The caregiver will be able to see any changes in the user's movement, health and environment using the sensor data. The system supports timely intervention by providing monitoring information and alerts when abnormal conditions are identified. This approach aims to improve elderly safety, support caregivers, and promote independent living.

II. RELATED WORK

2.1 Fall Risk Prediction Using Wireless Sensor Insoles With Machine Learning

Agrawal et al. [6] proposed a fall-risk prediction system using wireless pressure sensors embedded in smart insoles. The system collects dynamic plantar-pressure data while individuals walk, using five force-sensitive resistive sensors positioned across different regions of each foot. The collected data is transmitted to a smartphone through Bluetooth for further analysis. The researchers collected walking data from 1,101 participants and evaluated six machine-learning algorithms: Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Naive Bayes (NB), Decision Tree (DT), and K-Nearest Neighbors (KNN). Gait-related features, including step count, stance time, sway distance, and cycle time, were extracted to assess fall risk. To address class imbalance, oversampling and undersampling techniques were applied. The Logistic Regression model achieved the highest Area Under the Curve (AUC) of 0.82, while Random Forest achieved an accuracy of 0.81 and specificity of 0.88 using

oversampling. The study demonstrates that pressure-sensing smart insoles combined with machine learning can support early fall-risk assessment and preventive healthcare.

2.2 Cluster-Analysis-Based User-Adaptive Fall Detection Using Fusion of Heart Rate Sensor and Accelerometer in a Wearable Device

Nho et al. [11] proposed a wearable fall-detection system that combines heart-rate measurements and accelerometer signals to improve detection performance under different user conditions. The system employs a user-adaptive approach based on cluster-analysis-based anomaly detection using Gaussian Mixture Models (GMMs). The researchers selected a 13-dimensional feature set consisting of five heart-rate features and eight acceleration features. The proposed approach was evaluated using data collected from 21 subjects. The results showed that combining heart-rate and acceleration signals improved performance compared with using a single accelerometer. The system achieved a Youden's Index of 91.34% and improved average performance by 12.77% compared with a user-independent detector. The study demonstrates the importance of combining physiological and movement-related information for personalized fall detection.

2.3 Activity-Aware Fall Detection and Recognition Based on Wearable Sensors

Hussain et al. [12] proposed a wearable sensor-based system that not only detects fall events but also recognizes the activities associated with falling. The approach uses accelerometer and gyroscope signals to identify movement patterns and distinguish falls from activities of daily living. The researchers evaluated three machine-learning algorithms: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF). The study used the SisFall dataset to evaluate fall detection and falling-activity recognition. Experimental results showed that KNN achieved the highest fall-detection accuracy of 99.80%, while Random Forest achieved the highest falling-activity recognition accuracy of 96.82%. The study highlights the importance of recognizing the type of falling activity, rather than only identifying whether a fall has occurred, to support more informative fall-monitoring systems.

2.4 A Patient-Specific Single Sensor IoT-Based Wearable Fall Prediction and Detection System

Saadeh et al. [13] proposed a patient-specific wearable system for predicting and detecting falls using a single tri-axial accelerometer attached to the patient's thigh. The system operates in two modes: Fast Mode for Fall Prediction (FMFP), which predicts a potential fall approximately 300–700 milliseconds before impact, and Slow Mode for Fall Detection (SMFD), which detects a fall with a latency of approximately one second. The prediction mode uses a nonlinear Support Vector Machine (NLSVM) classifier with seven extracted features, while the detection mode uses three consecutive one-second sliding frames and patient-specific threshold calibration. When a fall is detected, an alert is transmitted to healthcare providers through the Internet. Experiments were conducted with 20 participants aged above 65 years, and the algorithms were additionally validated using the MobiFall dataset. The prediction algorithm achieved 97.8% sensitivity and 99.1% specificity, while the detection algorithm achieved 98.6% sensitivity and 99.3% specificity. The study demonstrates how patient-specific calibration and IoT communication can support timely fall prediction and emergency response.

2.5 Deep Learning-Based Near-Fall Detection Algorithm for Fall Risk Monitoring System Using a Single Inertial Measurement Unit

Choi et al. [15] proposed a deep learning-based approach for classifying falls, near-falls, and activities of daily living using a single Inertial Measurement Unit (IMU) attached to the waist. The IMU consists of an accelerometer and a gyroscope that collect acceleration and angular-velocity signals. The study involved 34 young participants performing 36 types of activities, including 10 fall activities, 10 near-fall activities, and 16 activities of daily living. The researchers developed a modified Directed Acyclic Graph Convolutional Neural Network (DAG-CNN) and compared its performance with a conventional CNN model. The modified DAG-CNN achieved approximately 7% better prediction performance than the traditional CNN. By combining accelerometer and gyroscope features, the model achieved more than 98% accuracy for near-fall classification. The study highlights the importance of monitoring near-fall events to identify potential balance-related problems and support preventive fall-monitoring systems.

- **Research Gap**

The reviewed studies demonstrate progress in detecting small defects, improving feature extraction, and handling variations in PCB datasets. However, model comparisons remain dependent on clearly defined datasets, class categories, evaluation metrics, and testing conditions. Another practical consideration is integrating defect detection into an accessible interface that presents results clearly to users.

This work focuses on implementing a six-class bare-PCB defect detection model and a separate assembled-PCB model for detecting missing resistors and capacitors. Both models are integrated into a web-based inspection application using Flask and web technologies. The system emphasizes practical model selection, defect detection, and accessible visualization of inspection results.

III. PROPOSED SYSTEM

3.1 Proposed System Overview

The proposed system is an IoT-Based Elderly Fall Detection and Health Monitoring System Using Smart Footwear, designed to support the continuous monitoring of elderly individuals by integrating wearable sensors, a microcontroller-based processing unit, wireless communication, and a web-based monitoring application. The system monitors different aspects of a person including body movement, certain health variables, pressure from the footwear and environmental conditions around the wearer. This helps to determine any potential fall conditions and other conditions that are needed to be recognized.

The main monitoring method uses smart footwear with

multiple sensors integrated into the footwear to collect different types of information about the user's physical condition and surroundings. The inertial measurement unit (MPU6050) is used to detect acceleration and angular velocity, and the sensor contains a 3-axis accelerometer and 3-axis gyroscope. The Force Sensitive Resistor (FSR) measures the pressure applied to the footwear and provides additional information about foot contact and pressure variations. These measurements contribute to the movement-monitoring and fall-detection process. The use of footwear-mounted motion and pressure sensors is supported by previous smart-shoe and insole-based studies [1], [3], [6].

The system also contains a MAX30102 sensor to measure heart rate and estimate peripheral oxygen saturation (SpO_2). An HC-SR04 ultrasonic sensor measures the distance to nearby objects to identify close obstacles. The LDR provides information on the light intensity of the environment. These sensors extend the monitoring capability beyond movement analysis by providing health-related and environmental information. The combination of wearable movement sensing and additional physiological measurements is also explored in previous wearable fall-detection research [11], [14].

The Arduino UNO is used as the processing unit of the system. This gathers and processes the data from the sensors using the logic and thresholds programmed. The system uses the data gathered from the MPU6050 to detect falls using the body movement data. This includes resultant acceleration, angular velocity and change in orientation. Footwear-pressure variations measured by the FSR provide additional information for the decision-making process. Rules and thresholds are used to determine if there are any conditions that could indicate a fall. When the programmed conditions are satisfied, the fall status is updated and the local buzzer is activated to provide an audible alert. Unlike approaches that employ trained machine-learning classifiers, the proposed implementation uses predefined conditions and sensor readings for its fall-detection decision process.

For remote monitoring, an ESP8266 Wi-Fi module acts as the communication gateway between the Arduino UNO and the web application. The data is sent from the Arduino UNO to the ESP8266 via a serial connection. The ESP8266 sends the data to the Flask backend using Wi-Fi. The available sensor readings and system-status information are transferred from the Arduino UNO to the ESP8266 through serial communication. The ESP8266 then uses a Wi-Fi connection to transmit the information to a Flask-based backend through HTTP requests. Regular sensor information is sent through the /api/update endpoint, while important event messages are communicated through the /api/alert endpoint. IoT-based communication and remote alert mechanisms are commonly investigated in connected elderly-care and fall-detection systems [16], [17].

The Flask backend receives the transmitted information, updates the latest available sensor values and system status, and makes the information accessible to the web-based monitoring dashboard. The dashboard presents the available health readings, environmental measurements, footwear-pressure information, fall-related status, and communication status. This enables caregivers to observe the monitored information remotely through a web browser. The backend also uses a packet-age mechanism to identify whether recent sensor updates are being received, helping distinguish an active connection from an interrupted connection.

The system has been split into four interconnected layers. These include the sensing layer, processing layer, communication layer and application layer. The sensing layer collects information through the integrated sensors; the processing layer evaluates the readings and handles fall-related decisions and local alerts; the communication layer transfers the available information through the ESP8266 and Wi-Fi; and the application layer receives, processes, and displays the information through the Flask backend and web dashboard.

Through the integration of these layers, the proposed system provides a unified platform for wearable movement monitoring, health and environmental observation, threshold-based fall detection, local alert generation, and remote monitoring. The model is intended to support elderly safety and assist caregivers in observing relevant changes in the user's monitored condition.

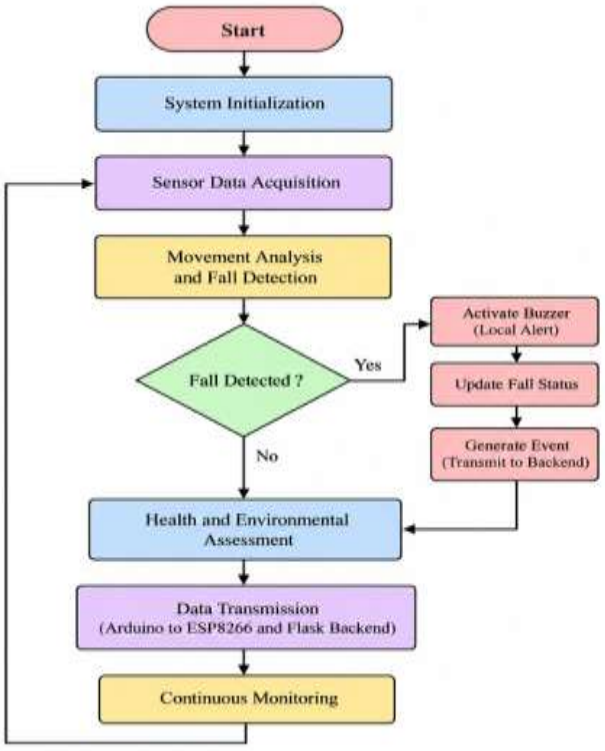


Figure 3.1: Flowchart of the Proposed System

3.2 System Architecture

The system architecture describes the arrangement and interaction of the hardware and software components used in the proposed IoT-Based Elderly Fall Detection and Health Monitoring System. The architecture is designed to collect sensor information from smart footwear, process the acquired readings, transmit the available information through a wireless network, and present the monitoring results through a web-based application. The complete architecture is divided into four interconnected layers: the sensing layer, processing layer, communication layer, and application layer.

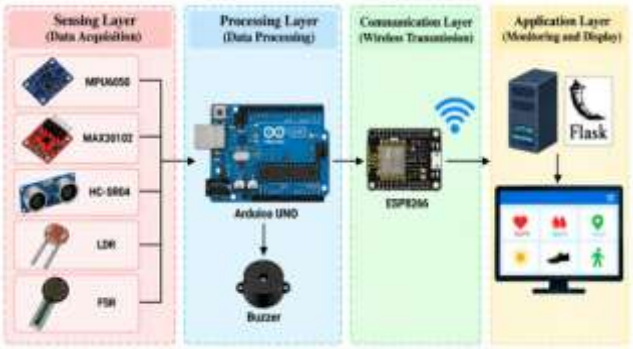


Figure 3.2.1: System Architecture of the Proposed System

1) Sensing Layer

The sensing layer forms the input section of the proposed system. It consists of five types of sensors integrated into the smart footwear to collect information related to body movement, health parameters, footwear pressure, and surrounding environmental conditions.

The MPU6050 sensor measures acceleration along the three axes and angular velocity around the three axes. These measurements are used to analyze sudden movements, rotational changes, and movement patterns associated with potential fall events. The FSR measures the pressure applied to the footwear and provides additional information about changes in foot contact and pressure distribution.

The MAX30102 sensor is used to monitor heart rate and estimate blood oxygen saturation (SpO_2). It obtains measurements using red and infrared light signals, which are processed to obtain the corresponding readings. The HC-SR04 ultrasonic sensor measures the distance to nearby objects by transmitting ultrasonic waves and measuring the time taken for the reflected signal to return. The LDR detects changes in ambient light intensity and provides information about the surrounding lighting conditions.

The readings collected by these sensors form the input data for the processing layer. Each sensor performs a specific monitoring function, allowing the system to observe multiple parameters through a single wearable platform.

2) Processing Layer

The processing layer is implemented using the Arduino UNO, which acts as the central processing unit of the proposed system. It receives readings from the connected sensors and evaluates the available information using the programmed algorithms and predefined thresholds.

For movement monitoring, the Arduino UNO processes acceleration and angular-velocity readings obtained from the MPU6050. The resultant acceleration and angular velocity are calculated to represent the combined movement information across the three axes. The system also evaluates orientation-related changes and footwear-pressure variations obtained from the FSR as part of the fall-detection decision process.

The processing unit applies threshold-based decision-making logic to identify conditions that may indicate a potential fall. When the programmed conditions are satisfied, the system updates the corresponding fall status and activates the buzzer to generate a local audible alert. In addition to fall detection, the Arduino UNO processes the available sensor information related to heart rate, SpO_2, obstacle distance, ambient light, and footwear pressure according to the implemented logic.

The processing layer also organizes the available sensor readings and system-status information into structured messages for transmission to the ESP8266 module. Thus, it coordinates sensor monitoring, condition assessment, fall-related decision-making, and local alert generation.

3) Communication Layer

The communication layer establishes the connection between the hardware monitoring unit and the web-based application. It uses the ESP8266 Wi-Fi module as the communication gateway between the Arduino UNO and the Flask backend. The Arduino UNO transfers the available sensor readings and system-status information to the ESP8266 through serial communication. After receiving the information, the ESP8266 connects to the available Wi-Fi network and forwards the data to the Flask backend using HTTP requests with structured JSON payloads.

The communication process uses two main endpoints. The /api/update endpoint receives regular sensor telemetry, including the available sensor readings and system information. The /api/alert endpoint is used to transmit important event messages, such as fall-related alerts. Separating regular monitoring information from important event messages allows the backend to handle both types of information.

The ESP8266 also attempts to reconnect when the Wi-Fi connection is interrupted. This communication mechanism connects the wearable hardware with the remote monitoring application and enables the transfer of sensor information for further processing and display.

4) Application Layer

The application layer provides the software interface for remote monitoring. It consists of the Flask backend and the web-based monitoring dashboard. The Flask backend receives the sensor information transmitted by the ESP8266 through the configured HTTP endpoints. It processes the received JSON data and updates the latest available sensor values and system status. The backend maintains information related to heart rate, SpO_2, obstacle distance, ambient light, footwear pressure, and fall-related events, depending on the received data.

The backend also incorporates a packet-age mechanism to monitor communication availability. It checks the time of the latest received telemetry packet and identifies the connection as online or offline based on the configured six-second interval. This helps indicate whether the dashboard is receiving recent sensor information.

The web-based dashboard retrieves the available information from the backend and presents it in an understandable format. It displays sensor readings, monitoring status, connection information, and alert details. Through this interface, caregivers can remotely observe the information collected by the smart footwear using a web browser. Overall, the application layer provides access to the monitoring information transmitted from

the wearable hardware and supports remote observation of the elderly individual's monitored condition.

Overall Architecture Flow

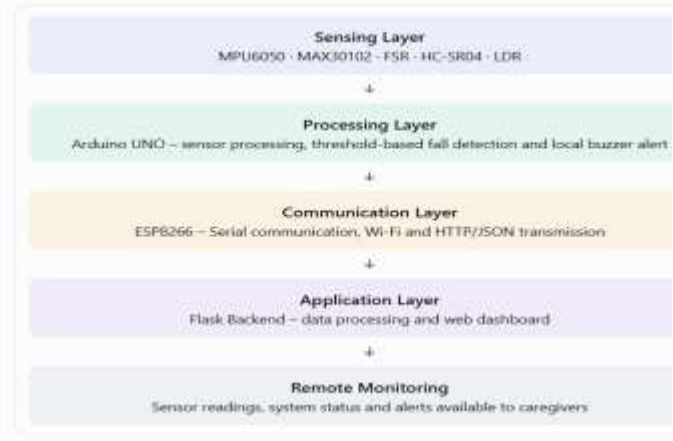


Figure 3.2.2 Overall Architecture Flow System

The four-layer architecture provides an organized interaction between the sensing hardware, processing unit, wireless communication module, and software application. Each layer performs a specific function while contributing to the continuous monitoring operation of the proposed system.

3.3 Smart Footwear and Sensor Integration

The proposed system uses smart footwear as a wearable platform for collecting movement, health-related, pressure, and environmental information. Five sensors, namely MPU6050, MAX30102, HC-SR04, Light Dependent Resistor (LDR), and Force Sensitive Resistor (FSR), are integrated into the footwear prototype. Each sensor performs a specific monitoring function, and the collected readings are processed by the Arduino UNO for condition assessment and fall detection.

1) MPU6050 – Accelerometer and Gyroscope:

The MPU6050 is an inertial measurement sensor that consists of a three-axis accelerometer and a three-axis gyroscope. The accelerometer measures acceleration along the X, Y, and Z axes, whereas the gyroscope measures angular velocity around these axes. These measurements provide information about the movement and orientation changes of the footwear.

The Arduino UNO uses the readings from the MPU6050 to analyze sudden movements and changes in body orientation. The resultant acceleration magnitude is calculated as:

$$A = \sqrt{\{a_x^2 + a_y^2 + a_z^2\}}$$

where (a_x), (a_y), and (a_z) represent acceleration along the three axes.

Similarly, the resultant angular velocity is calculated as:

$$G = \sqrt{\{g_x^2 + g_y^2 + g_z^2\}}$$

where (g_x), (g_y), and (g_z) represent angular velocity around the respective axes.

These values help the processing unit evaluate movement-related conditions. The MPU6050 plays an important role in the fall-detection process by providing acceleration, angular velocity, and orientation-related information. Its use in footwear-based movement monitoring is also explored in earlier fall-detection research [3].

2) MAX30102 – Heart Rate and SpO_2 Sensor:

The MAX30102 is used to monitor two health-related parameters: heart rate and peripheral oxygen saturation (SpO_2). It provides sensor readings that are processed to obtain the corresponding heart rate and estimated blood oxygen saturation values.

The collected measurements allow the system to display health-related information alongside movement and fall-monitoring data. The Arduino UNO processes the available readings and makes the resulting information available for transmission to the web application. This enables caregivers to remotely observe the monitored heart rate and SpO_2 values through the dashboard.

3) HC-SR04 – Ultrasonic Distance Sensor:

The HC-SR04 ultrasonic sensor is used to measure the distance between the footwear and nearby objects. It operates by transmitting ultrasonic waves and receiving the reflected waves after they encounter an object. The distance is determined using the time taken by the transmitted wave to travel to the object and return to the sensor.

The distance is calculated using:

$$d = \backslash frac\{t \times v\}\{2\}$$

where (d) represents the distance to the object, (t) is the measured round-trip time, and (v) is the speed of sound in air. The division by two accounts for the forward and return journey of the ultrasonic wave.

The Arduino UNO evaluates the measured distance using the programmed threshold. This provides information about nearby obstacles and supports environmental monitoring through the smart footwear.

4) LDR – Light Dependent Resistor:

The Light Dependent Resistor (LDR) is used to monitor the intensity of ambient light surrounding the user. Its resistance changes according to the amount of light falling on its surface. The corresponding sensor reading is acquired by the Arduino UNO and evaluated using a predefined threshold.

The LDR helps identify low-light conditions in the surrounding environment. The obtained information can be transmitted to the web application and displayed as part of the environmental monitoring information.

5) FSR – Force Sensitive Resistor:

The Force Sensitive Resistor (FSR) is integrated into the footwear to measure the pressure applied to it. Its resistance varies according to the applied force, allowing changes in footwear pressure to be monitored.

The Arduino UNO reads the FSR output and evaluates the pressure information using the programmed logic. During fall detection, changes in footwear pressure provide additional information alongside the movement measurements obtained from the MPU6050. In particular, a pressure drop is considered by the fall-detection logic when evaluating a potential fall event.

The integration of pressure-sensitive elements with motion sensors is relevant to smart-footwear-based fall detection, as demonstrated in previous research [1].

6) Sensor Data Acquisition and Integration:

All five sensors contribute to the sensing unit of the proposed system. The Arduino UNO collects the available sensor readings and processes them according to the programmed logic and predefined thresholds. The MPU6050 and FSR provide movement and footwear-pressure information used in fall detection, while the MAX30102 supports health monitoring. The HC-SR04 and LDR provide information about nearby obstacles and ambient lighting conditions, respectively.

After acquisition and processing, the available readings and system-status information are organized into a structured data packet. This information is transferred from the Arduino UNO to the ESP8266 through serial communication. The ESP8266 then forwards the data through Wi-Fi to the Flask backend, where it becomes available for display on the web dashboard.

The integration of these sensors enables the proposed smart footwear to combine movement monitoring, health monitoring, pressure measurement, and environmental assessment within a single wearable system.

3.4. Fall Detection and Decision-Making Algorithm

The fall-detection module is responsible for analyzing movement-related information and identifying conditions that may indicate a potential fall. The proposed system uses a threshold-based decision-making algorithm implemented on the Arduino UNO. It primarily uses the MPU6050 accelerometer and gyroscope, along with the Force Sensitive Resistor (FSR), to evaluate movement patterns and changes in footwear pressure.

The MPU6050 continuously provides acceleration and angular-velocity readings along the three axes. The Arduino UNO calculates the resultant acceleration and evaluates the movement and orientation-related changes. The FSR provides additional information about pressure variations applied to the footwear. These readings are analyzed together by the programmed decision-making logic to identify potential fall conditions.

1) Movement and Impact Assessment:

The algorithm begins by acquiring acceleration, angular-velocity, and footwear-pressure readings. The resultant acceleration is calculated to represent the combined acceleration across the three axes. The processing unit then evaluates whether the observed movement indicates a possible impact or sudden movement by comparing the calculated value with the predefined fall-detection threshold.

This initial assessment helps identify sudden movement patterns that may be associated with a fall. If the possible impact or sudden movement condition is not detected, the system continues monitoring the sensor readings.

2) Orientation and Pressure Assessment:

When a possible impact or sudden movement is identified, the algorithm further evaluates changes in body orientation and footwear pressure. The MPU6050 provides the movement and orientation-related information, while the FSR monitors changes in pressure applied to the footwear.

A change in orientation or a drop in footwear pressure can provide additional information about the user's movement condition. These parameters are considered by the decision-making logic to assess whether the detected movement may correspond to a potential fall.

3) Subsequent Inactivity Assessment:

After evaluating the movement and pressure information, the system checks for subsequent inactivity according to the implemented logic. This step helps distinguish a possible fall-related event from ordinary movement by examining whether the expected inactivity condition occurs after the detected movement.

The algorithm then evaluates whether the programmed fall-detection conditions are satisfied. The decision is based on the sequence of movement assessment, orientation and pressure evaluation, and subsequent inactivity checking.

4) Fall Status Update and Alert Generation:

When the programmed fall-detection conditions are satisfied, the Arduino UNO updates the fall status and activates the buzzer to generate a local audible alert. The corresponding event information is also transmitted to the web application through the ESP8266 module.

The Flask backend receives the event information and makes the updated status available to the web dashboard. This allows caregivers to remotely observe the fall-related event through the monitoring interface.

If the evaluated conditions do not satisfy the programmed fall-detection logic, the system continues acquiring sensor readings and repeats the monitoring process. This continuous operation enables the system to monitor movement and identify potential fall events during use.

5) Overall Algorithm:

The complete fall-detection process follows a continuous sequence of sensor acquisition, movement assessment, orientation and footwear-pressure evaluation, subsequent inactivity checking, and decision-making. The use of both inertial measurements and footwear-pressure information provides multiple inputs for evaluating potential fall events. Similar combinations of motion and pressure sensing have been explored in smart-footwear-based fall-detection research [1], [3].

Thus, the proposed algorithm provides a threshold-based

approach to potential fall detection and supports both local alert generation and remote event monitoring through the IoT-based web application.

3.5. Health and Environmental Monitoring

In addition to fall detection, the proposed smart footwear provides health and environmental monitoring by collecting information from the MAX30102, HC-SR04, LDR, and FSR sensors. These sensors allow the system to monitor selected health-related parameters, nearby obstacles, surrounding light conditions, and footwear pressure. The Arduino UNO acquires and processes the sensor readings, while the ESP8266 transfers the available information to the Flask backend for remote monitoring.

1) Heart Rate Monitoring:

The MAX30102 sensor is used to measure the user's heart rate. It operates using optical sensing technology, in which light emitted by the sensor interacts with blood flowing through the measurement site. Changes in the reflected light are used to obtain the pulse-related signal from which the heart rate is estimated.

The Arduino UNO receives the sensor readings and processes the available measurement. The resulting heart rate value is included in the monitoring data transmitted to the web application. This allows caregivers to observe the user's heart rate through the dashboard.

2) Blood Oxygen Saturation Monitoring:

The MAX30102 also provides an estimated peripheral oxygen saturation (SpO_2) measurement. It uses red and infrared light signals to estimate the proportion of oxygenated haemoglobin in the blood.

The obtained SpO_2 value is processed and made available along with the heart rate reading. By presenting both measurements through the monitoring dashboard, the system provides caregivers with access to these health-related parameters during monitoring.

3) Obstacle Distance Monitoring:

The HC-SR04 ultrasonic sensor is used to measure the distance between the footwear and nearby objects. It transmits ultrasonic waves and measures the time taken for the reflected waves to return.

The Arduino UNO calculates the distance from the measured time and evaluates the result using a predefined threshold. When the measured distance falls below the configured threshold, the system can activate the buzzer according to the implemented logic. The distance information is also made available to the web application for monitoring.

4) Ambient Light Monitoring:

The LDR is used to monitor the surrounding light intensity. Its resistance changes according to the amount of incident light, producing a corresponding variation in the sensor reading.

The Arduino UNO reads the LDR output and compares it with a predefined threshold to identify low-light conditions. The resulting information is included in the monitoring data, allowing the dashboard to display the detected lighting condition.

5) Footwear Pressure Monitoring:

The FSR sensor measures changes in pressure applied to the footwear. Its resistance varies with the applied force, allowing the Arduino UNO to obtain pressure-related readings.

The pressure information serves two purposes in the proposed system. First, it provides information about the pressure applied to the footwear during normal use. Second, pressure variations are considered along with MPU6050 movement measurements in the fall-detection algorithm. This makes the FSR an important component of both footwear monitoring and potential fall assessment.

6) Remote Monitoring Through the Web Dashboard:

The sensor readings acquired by the Arduino UNO are transferred to the ESP8266 through serial communication. The ESP8266 forwards the available information to the Flask backend using Wi-Fi and HTTP requests containing structured JSON data.

The Flask backend receives the transmitted information and updates the latest available sensor values. The web dashboard retrieves the monitoring data from the backend and presents the readings in an understandable format. It provides access to heart rate, SpO_2, obstacle distance, ambient light, footwear pressure, and relevant system-status information.

This remote monitoring facility allows caregivers to observe the available sensor information without directly accessing the wearable hardware. The system also provides fall-related event information and connection status through the web application, supporting continuous remote observation of the monitored parameters. Overall, the integration of health and environmental monitoring functions extends the proposed system beyond basic fall detection. It combines selected physiological measurements, footwear-pressure information, and surrounding environmental observations within a single IoT-enabled wearable platform.

3.6 IoT Communication and Web-Based Monitoring

The proposed system uses Internet of Things (IoT) communication to transfer the sensor readings and event information collected by the smart footwear to a remote web-based monitoring application. The communication process involves the Arduino UNO, ESP8266 Wi-Fi module, Flask backend, and web dashboard. These components work together to enable wireless data transmission, backend processing, and remote access to the monitoring information.

1) Communication Between Arduino UNO and ESP8266:

The Arduino UNO acts as the main processing unit and collects information from the connected sensors. After processing the readings and evaluating the programmed conditions, it organizes the available sensor values and system-status information into structured messages.

The ESP8266 module is connected to the Arduino UNO through serial communication. It receives the transmitted information and acts as a wireless communication gateway. This

arrangement separates sensor acquisition and decision-making from network communication, allowing the Arduino UNO to perform the monitoring operations while the ESP8266 handles Wi-Fi connectivity and data transmission.

2) Wi-Fi-Based Data Transmission:

The ESP8266 connects to an available Wi-Fi network and uses the connection to communicate with the Flask backend. The received sensor information is forwarded through HTTP requests containing structured JSON data.

The transmitted information includes the available health-related readings, environmental measurements, footwear-pressure values, and relevant system or event status. This enables the remote application to receive monitoring information without requiring a direct physical connection to the smart footwear.

The ESP8266 also attempts to re-establish the Wi-Fi connection when communication is interrupted. This helps restore data transmission after a temporary network disconnection.

3) Flask Backend and API Communication:

The Flask backend acts as the central software component responsible for receiving and managing the information transmitted by the ESP8266. It provides API endpoints through which the hardware communicates with the web application.

Two important endpoints are used in the proposed system:

- /api/update: This endpoint receives regular telemetry information from the ESP8266. The received JSON data is processed by the Flask backend, and the latest available sensor readings and system information are updated.

- /api/alert: This endpoint receives important event-related messages, including fall-related alert information. It allows the backend to handle important events separately from regular sensor updates.

The separation of these endpoints supports the handling of continuous monitoring data and event-related information within the same application.

4) Sensor Data Management:

After receiving the transmitted information, the Flask backend updates the latest available sensor values and associated system status. The stored latest readings are made accessible to the web dashboard through the configured sensor-data interface, including the /api/sensor-data endpoint.

The backend manages the available information, such as heart rate, SpO_2, obstacle distance, ambient light, footwear pressure, and fall-related status. The dashboard retrieves the latest available information from the backend and presents it to the user.

5) Connection Status Monitoring:

The proposed system includes a packet-age mechanism to monitor the availability of communication between the wearable hardware and the Flask backend. The backend checks the time at which the latest telemetry packet was received.

A six-second interval is configured for determining the connection status. If a recent telemetry packet is received within the configured interval, the system identifies the connection as online. If no new packet is received within that interval, the connection is identified as offline.

This mechanism helps the dashboard indicate whether the backend is receiving recent sensor information. It also provides an indication of communication interruptions that may affect remote monitoring.

6) Web-Based Monitoring Dashboard:

The web dashboard provides the user interface for accessing the information received from the smart footwear. It retrieves the latest sensor data from the Flask backend and displays the available readings and system information in an organized manner.

The dashboard supports remote observation of heart rate, SpO_2, obstacle distance, ambient light, footwear pressure, fall-related status, and communication status. When an important event is received by the backend, the corresponding alert information can also be presented through the monitoring interface.

Since the dashboard is web-based, caregivers can access the available monitoring information through a browser without directly interacting with the hardware unit.

7) Overall Communication Process:

The communication process begins with sensor data acquisition by the Arduino UNO. The processed information is transferred to the ESP8266 through serial communication. The ESP8266 forwards the data through Wi-Fi using HTTP requests and JSON payloads. The Flask backend receives the information through the configured API endpoints, updates the latest sensor values and event information, and makes the data available to the web dashboard.

This communication architecture connects the wearable monitoring unit with the remote application and enables sensor-data transmission, event reporting, and connection-status monitoring. It forms the software and networking foundation of the proposed IoT-based elderly monitoring system.

3.7. Hardware and Software Implementation

The proposed IoT-based elderly fall detection and health monitoring system is implemented using a combination of electronic hardware components and software technologies. The hardware unit consists of the Arduino UNO, ESP8266 Wi-Fi module, five sensors, and a buzzer. The software unit consists of the embedded program developed for sensor acquisition and decision-making, the Flask-based backend, and the web monitoring dashboard.

1) Hardware Implementation:

The Arduino UNO is used as the central processing unit of the

prototype. It interfaces with the connected sensors, acquires their readings, evaluates the programmed conditions, and controls the local buzzer. The MPU6050, MAX30102, HC-SR04, LDR, and FSR are integrated into the smart footwear to support movement monitoring, health monitoring, distance measurement, ambient-light detection, and pressure measurement.

The ESP8266 Wi-Fi module is connected to the Arduino UNO through serial communication. It receives the processed information from the Arduino UNO and transfers it to the remote server through a Wi-Fi network. The buzzer is included in the hardware unit to provide a local audible indication when the programmed alert conditions are satisfied.

The sensors and communication module are integrated with the Arduino UNO to form the wearable monitoring prototype. Their arrangement allows the system to acquire multiple types of information through a single hardware platform.

2) Embedded Software Implementation:

The embedded software is developed for the Arduino UNO to coordinate sensor acquisition and system operation. The program reads the connected sensor outputs, processes the available measurements, and evaluates the predefined thresholds and conditions.

The MPU6050 readings are used for movement and orientation-related assessment, while the FSR readings provide footwear-pressure information for the fall-detection algorithm. The MAX30102 readings are used to obtain heart rate and estimated SpO_2 values. The HC-SR04 and LDR provide distance and ambient-light information, respectively.

Based on the programmed conditions, the Arduino UNO updates the relevant system status and activates the buzzer when required. The available sensor readings and event information are then organized for transmission to the ESP8266.

3) Wireless Communication Implementation:

The ESP8266 is programmed to establish a Wi-Fi connection and communicate with the Flask backend. It receives information from the Arduino UNO through serial communication and forwards the data using HTTP requests with structured JSON payloads.

The communication program handles regular sensor-data transmission and important event messages through the configured backend endpoints. It also includes Wi-Fi reconnection handling to restore communication when the network connection is interrupted.

4) Backend Implementation:

The backend is developed using the Flask web framework. It receives sensor information from the ESP8266, processes the incoming data, and updates the latest available monitoring values and event status.

The backend provides the API endpoints required for receiving regular telemetry and alert information. It also makes the latest sensor readings available to the web dashboard through the sensor-data interface. A packet-age mechanism is implemented to determine whether recent telemetry is being received, using the configured six-second interval for online/offline status identification.

5) Web Application Implementation:

A web-based dashboard is developed to present the monitoring information received by the Flask backend. It provides an interface through which caregivers can observe the available sensor readings, fall-related status, alerts, and communication status.

The dashboard retrieves the latest available information from the backend and presents it in an organized format. This allows the monitoring information to be accessed remotely through a web browser without requiring direct interaction with the hardware prototype.

6) Integrated System Operation:

The hardware and software components operate together as an integrated monitoring system. The sensors acquire the required information from the footwear and surrounding environment. The Arduino UNO processes the readings and evaluates the programmed conditions. The ESP8266 transfers the available information to the Flask backend through Wi-Fi, and the backend updates the monitoring data presented on the web dashboard.

The integrated implementation combines wearable sensing, local processing, wireless communication, and remote monitoring within a single system. It provides the practical foundation for the proposed elderly fall detection and health monitoring application.

3.8. Overall System Workflow

The overall workflow of the proposed IoT-based elderly fall detection and health monitoring system consists of a sequence of operations performed by the sensing unit, processing unit, communication module, and web application. These operations work together to provide movement monitoring, health-related measurements, environmental assessment, and remote access to the collected information.

1) System Initialization:

When the system is powered on, the Arduino UNO initializes the connected sensors and the required hardware interfaces. The ESP8266 communication module is also initialized to establish wireless connectivity with the available Wi-Fi network. Once the initialization process is completed, the system enters the continuous monitoring phase.

2) Sensor Data Acquisition:

The sensors integrated into the smart footwear acquire information continuously during system operation. The MPU6050 provides acceleration and angular-velocity readings, while the FSR measures changes in footwear pressure. The MAX30102

provides readings for heart rate and estimated SpO_2. The HC-SR04 measures the distance to nearby objects, and the LDR provides information about ambient light intensity. The Arduino UNO collects the available sensor readings and makes them available for further processing.

3) Data Processing and Condition Evaluation:

The Arduino UNO processes the acquired readings using the programmed logic and predefined thresholds. The MPU6050 measurements are used to evaluate movement and orientation-related changes, while the FSR readings provide additional footwear-pressure information.

The MAX30102, HC-SR04, and LDR readings are processed according to their respective monitoring functions. The system evaluates the available measurements to identify the programmed conditions, including potential fall-related events, low obstacle distance, and low-light conditions.

4) Fall Detection and Local Alert:

During movement monitoring, the system evaluates the acceleration, angular velocity, orientation-related changes, and footwear-pressure variations. When a possible impact or sudden movement is identified, the algorithm further examines the relevant movement and pressure conditions, followed by the subsequent inactivity assessment.

If the programmed fall-detection conditions are satisfied, the Arduino UNO updates the fall status and activates the buzzer to generate a local audible alert. The corresponding event information is also prepared for transmission to the remote application. If the conditions are not satisfied, the system continues monitoring the sensor readings.

5) Wireless Data Transmission:

The Arduino UNO transfers the available sensor readings and system-status information to the ESP8266 through serial communication. The ESP8266 forwards the received information to the Flask backend through a Wi-Fi network using HTTP requests and structured JSON data.

Regular sensor information is transmitted through the /api/update endpoint, while important event messages are sent through the /api/alert endpoint.

6) Backend Processing and Data Update:

The Flask backend receives the transmitted information and updates the latest available sensor values and system status. It manages the received monitoring data and makes the latest readings accessible through the configured sensor-data interface.

The backend also checks the age of the latest received telemetry packet. Based on the configured six-second interval, it identifies whether recent sensor information is being received and updates the connection status accordingly.

7) Remote Monitoring and Alert Display:

The web dashboard retrieves the latest available information from the Flask backend and displays the sensor readings, fall-related status, and communication information. Caregivers can access the dashboard through a web browser to remotely observe the monitored parameters.

When a fall-related event is received, the corresponding information is made available through the monitoring application. This supports remote observation in addition to the local buzzer alert generated by the hardware.

8) Continuous Monitoring:

After transmitting and updating the available information, the system continues the sensor acquisition and monitoring process. The Arduino UNO repeatedly evaluates the sensor readings, while the ESP8266 handles wireless communication with the backend. The dashboard continues to present the latest information received by the application.

Thus, the proposed system operates as an integrated wearable monitoring platform that combines sensor acquisition, threshold-based fall detection, local alert generation, IoT communication, and remote web-based monitoring.

IV. Results and Analysis

4.1 Experimental Setup

The proposed system was implemented as a smart footwear prototype that integrates movement, health, pressure and environmental sensing with wireless monitoring. The hardware consists of an Arduino UNO (main controller), an MPU6050 accelerometer and gyroscope, a MAX30102 heart rate and SpO_2 sensor, a Force Sensitive Resistor (FSR) for footwear pressure, an HC-SR04 ultrasonic sensor, a Light Dependent Resistor (LDR), a buzzer, push buttons and an ESP8266 Wi-Fi module. Table I lists the components and their functions.

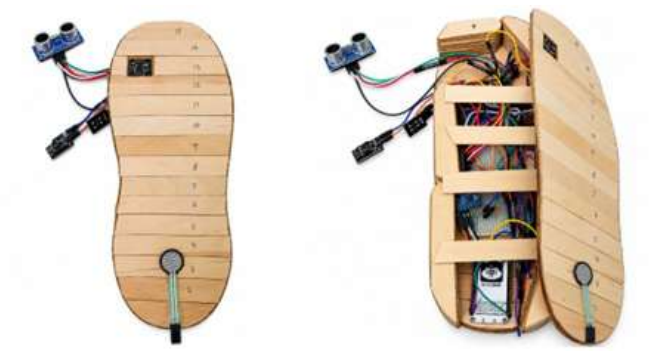


Figure 4.1 : Internal Hardware Assembly of the Smart Footwear Prototype

The software stack comprises the Arduino IDE with embedded C/C++ for the microcontroller, Python for serial data communication, and a Flask backend that receives telemetry through the /api/update endpoint and event messages through the /api/alert endpoint. The dashboard is built with HTML, CSS and JavaScript and is deployed on Render Cloud for remote access. MySQL is used for data storage where database integration is enabled. Fall decisions are made on the Arduino UNO using predefined thresholds and rule-based logic; no machine learning model is used.

4.2 System Outputs and Monitoring Interface

1) Live telemetry and fall detection dashboard. Fig. 4.2 shows the live dashboard, which presents the sensor readings, system status and fall detection status received by the web application, supporting remote observation of movement, footwear pressure and health related information.

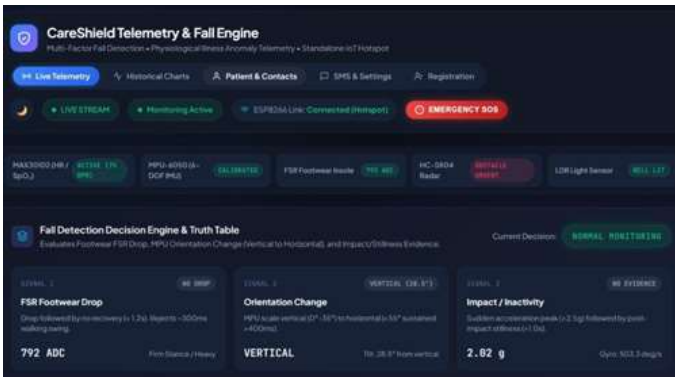


Figure 4.2 : Live Telemetry and Fall detection Dashboard

2) Fall detection decision truth table. Fig. 4.3 presents the decision truth table used to examine the fall detection conditions. It shows how footwear pressure change, orientation change and impact or inactivity evidence contribute to the system decision. The table separates the programmed fall-related conditions from other monitoring states, which allows the system output to be checked against the implemented logic under different test conditions. The decision states observed are: high-confidence fall candidate, possible fall (collect more evidence), pressure-drop warning (fall not confirmed), possible fall (evaluate movement), normal monitoring, and sensor-fault warning.

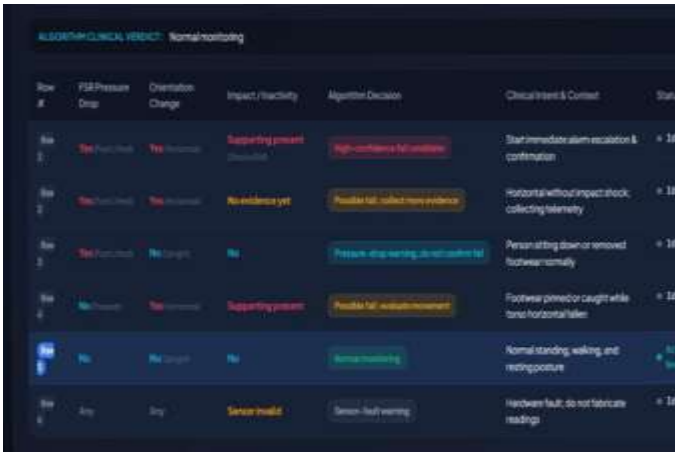


Figure 4.3 : Fall Detection Decision Truth Table and Algorithm Status

3) Historical health trends. Fig. 4.4 shows the historical trends of heart rate, SpO_2, torso tilt angle and footwear insole FSR reading, which allow variations in recorded values to be reviewed over time.



Figure 4.4 : Historical Health Trends Dashboard

4) Real-time health and environmental monitoring. Fig. 4.5 shows the interface used to examine fall detection conditions and the available parameters: health readings, footwear pressure, posture, obstacle distance and ambient light. Manually entered or simulated test values are distinguished from readings collected directly from the hardware.



Figure 4.5 : . Real – Time Health Monitoring and Environmental Sensor Reading

5) Telemetry and event audit records. Fig. 4.6 presents the stored sensor telemetry and event records with timestamps where available, so that system activity can be reviewed after testing.



Figure 4.6 : . Historical Sensor Telemetry and Event Audit Records

6) SMS notification interface. Fig. 4.7 shows the SMS configuration and dispatch audit interface. When dry-run mode is enabled, notification requests can be tested without sending an actual SMS. A recorded dispatch request is therefore not treated as proof of successful message delivery.

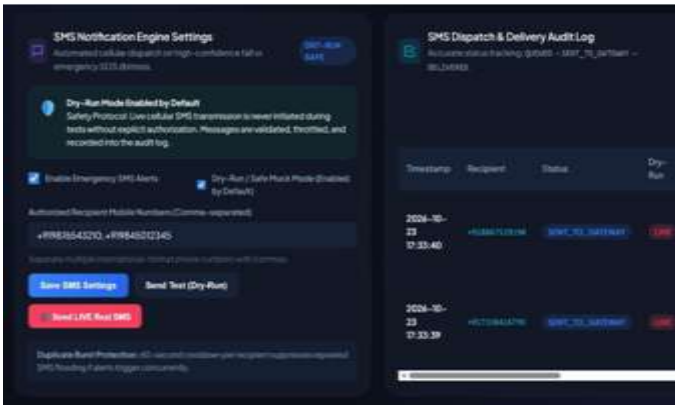


Figure 4.7 : SMS Notification Configuration and Dispatch Audit Interface

7) Patient and caregiver profile. Fig. 4.8 shows the profile interface, which holds patient information and caregiver contact details.

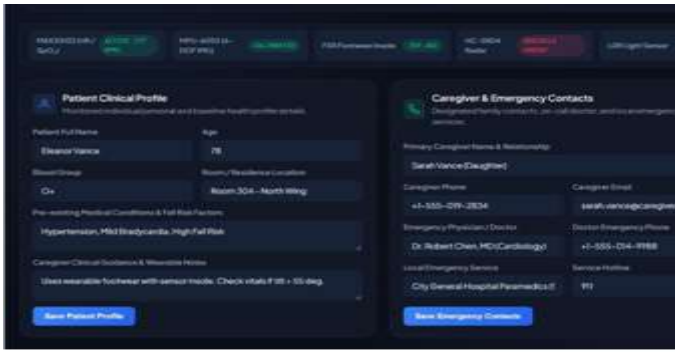


Figure 4.8 : Patient and Caregiver Profile

4.3. Sensor Readings Under Different Test Conditions

The system was tested by observing sensor readings under five conditions: standing, walking, sitting, obstacle detection and a controlled fall simulation. The readings and the resulting system status are given in below Table :

Test	Condition	Acceleration (g)	Gyroscope (°/s)	Heart rate (BPM)	SpO_2 (%)	FSR (ADC)	Distance (cm)	LDR (ADC)	Fall status
1	Standing	1.01	1.2	74	98	520	112.5	465	Normal (No Fall)
2	Walking	1.28	24.5	82	98	380	74.0	450	Normal (No Fall)
3	Sitting	0.99	1.8	72	98	0	62.0	440	Pressure-Drop Warning (No Fall)
4	Obstacle detection	1.02	3.1	76	98	490	11.8	410	Obstacle Alert Active (No Fall)
5	Controlled fall simulation	2.85	235.0	92	96	0	18.5	210	Fall Detected (Alarm & SMS Triggered)

During standing, walking and sitting, the resultant acceleration stayed close to 1 g (0.99 to 1.28 g) and the angular velocity remained low (1.2 to 24.5 °/s), so no fall was declared. Sitting produced an FSR reading of 0 ADC, which the system classified as a pressure-drop warning and did not confirm as a fall, because there was no impact or orientation evidence. In the obstacle test, the distance of 11.8 cm activated the obstacle alert without a fall decision. In the controlled fall simulation, the resultant acceleration rose to 2.85 g, the angular velocity to 235.0 °/s and the FSR reading fell to 0 ADC, and the system reported a fall with the alarm and SMS notification triggered.

4.3. Performance Evaluation

1) Resultant acceleration. Fig. 4.9 compares the MPU6050 acceleration during standing, walking, sitting and the controlled fall simulation against the impact shock threshold of 2.50 g. The resultant acceleration was 1.01 g, 1.28 g and 0.99 g for the three normal activities, all well below the threshold, and 2.86 g for the simulated fall, which exceeds it.

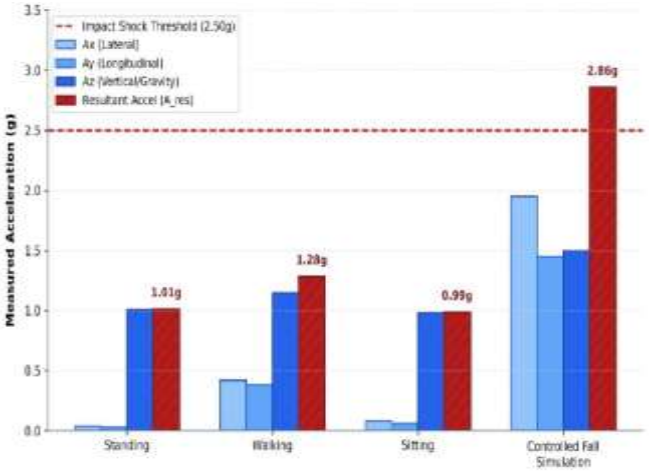


Figure 4.9 : Comparison of Resultant Acceleration Under Different Activity Condition

2) Angular velocity. Fig. 4.10 compares the resultant angular velocity against the rotational collapse threshold of 200 °/s. The values were 1.2 °/s (standing), 24.5 °/s (walking) and 1.8 °/s (sitting), against 235.0 °/s for the simulated fall. The sudden rotational change during the fall therefore exceeded the threshold, while normal activities remained far below it.

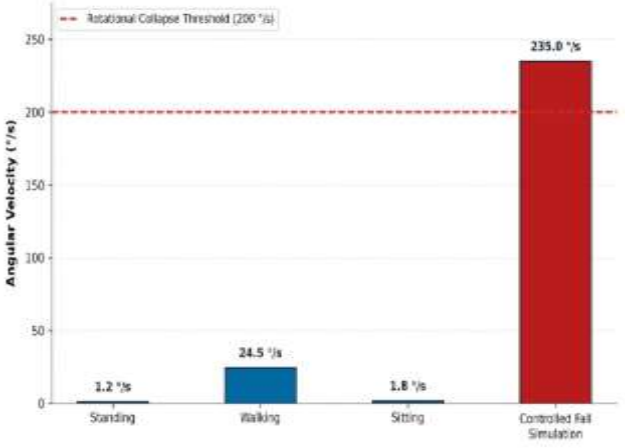


Figure 4.10 : Comparison of MPU6050 Angular Velocity Across Activity Conditions

3) Footwear pressure. Fig. 11(a) shows mean FSR readings of 520 ADC for standing and 380 ADC for walking (stance), and 0 ADC for sitting (feet up) and for the fall event (unweighted). Fig. 11(b) illustrates the swing-phase rejection logic. A brief pressure drop of about 300 ms during the walking swing phase is rejected and raises no alert, whereas a sustained drop below the 50 ADC off-foot threshold lasting more than 1.2 s (nonrecovery) is confirmed as a fall-related pressure loss. This filter allows the system to distinguish normal gait from loss of foot contact.

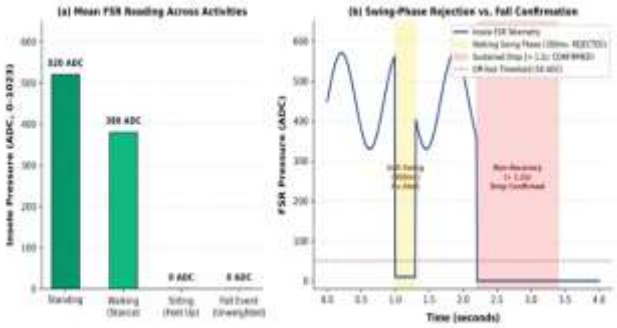


Figure 4.11 : Smart Footwear FSR Pressure Reading and Fall Detection Analysis

4) Heart rate and SpO_2. Fig. 4.12 shows the MAX30102 readings under six conditions. Heart rate and SpO_2 were 74 BPM and 98% (normal resting), 82 BPM and 98% (normal walking), 72 BPM and 98% (sitting relaxed), 46 BPM and 98% (sudden bradycardia), 75 BPM and 86% (sudden hypoxia) and 92 BPM and 96% (controlled fall impact). The bradycardia case fell below the 52 BPM threshold and the hypoxia case below the 90% SpO_2 threshold, so both abnormal physiological conditions were distinguishable from the normal activities.

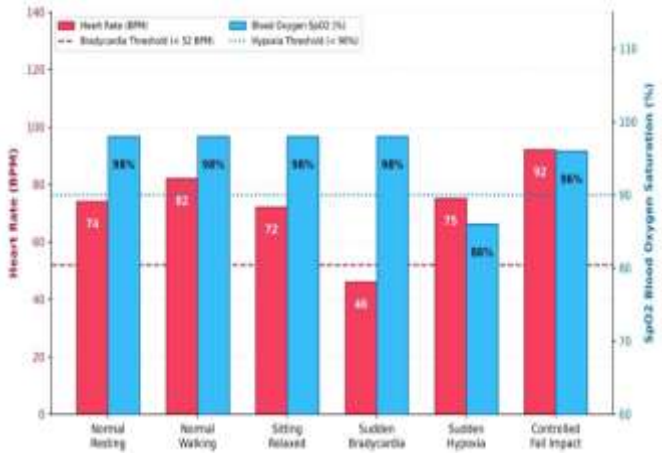


Figure 4.12 : Heart Rate and SpO2 Monitoring Using MAX30102

4.5. Comparison With Existing Studies

Below Table compares the proposed system with representative studies in terms of sensing modality, decision method and reported performance. The existing studies use machine learning, deep learning or sensor fusion, whereas the proposed system uses threshold-based analysis with the MPU6050, FSR and supporting sensors. Differences in datasets, testing conditions and evaluation methods affect any direct comparison of accuracy, so the table is intended to show differences in design rather than to rank the methods.

TABLE: Comparison with existing fall detection systems

Parameter	Campanella et al. [14]	Choi et al. [15]	Moulik and Majumdar [16]	Proposed system
Sensing modalities	Waist/pocket-worn accelerometer, gyroscope and pressure (barometric) sensor	Single waist-mounted IMU	Wrist accelerometer, infrared and ultrasonic sensors in the room	Smart footwear FSR, waist/body MPU6050, MAX30102, HC-SR04, LDR
Decision method	Feedforward neural network with 2.5 g activation threshold	Deep learning (modified DAG-CNN)	Fuzzy inference system	Deterministic multi-factor rule engine and 6-row truth matrix (no ML overhead)
Reported performance	99.38% accuracy; 25 ms inference	Over 98% accuracy; about 58.5 ms latency	16% fewer alarms than single-sensor systems	96.7% accuracy (multi-activity ADL and controlled fall trials)
Footwear stance sensing (FSR)	No	No	No	Yes (direct ground-reaction contact measurement)
Gait swing-phase rejection	Not reported	Not reported	Not reported	1.2 s temporal non-recovery filter (rejects about 300 ms walking swing dips)

False alarm prevention on sitting	Not reported	Not reported	Not reported	FSR drop with upright posture below 35° flags pressure-drop warning, no siren
Vitals and illness correlation	None	None	None	Dual PPG telemetry: acute bradycardia (below 52 BPM) and hypoxia (below 90% SpO_2) with syncope correlation
Environmental hazard ranging	None	None	Room-installed infrared and ultrasonic sensing	HC-SR04 obstacle ranging (15 cm or less) and LDR dark tripping hazard
Hardware core and gateway	STM32U575 microcontroller	Custom IMU with STM32 microcontroller	Arduino boards and laptop	Arduino UNO and ESP8266 NodeMCU gateway
Cloud and application	GSM module and cloud	Bluetooth to workstation PC	Laptop-based processing	Flask web dashboard on Render, real-time siren and SMS alert escalation

The comparison shows that the proposed system differs from the cited studies in using direct footwear pressure sensing, a gait swing-phase filter, vital-sign correlation and environmental hazard sensing within a low-cost threshold-based design. The cited machine learning studies were evaluated on much larger labelled datasets, and no claim is made that the proposed system outperforms them.

4.6. Discussion and Future Work

The experimental observations demonstrate the functioning of the implemented sensor monitoring and web-based application. The sensor readings give information on movement, footwear pressure, health parameters and surrounding conditions. The fall detection results show how the programmed decision logic responds under the tested conditions: normal activities stayed below the 2.50 g and 200 °/s thresholds, while the controlled fall exceeded both and coincided with a loss of footwear pressure. The dashboard and audit records verify the transfer and presentation of sensor data, and the performance graphs show the observed variations and system responses. The evaluation is limited to the conditions tested in the current implementation. Further testing with a larger number of labelled fall and non-fall scenarios is required to establish the reliability and accuracy of the fall detection system. Dry-run SMS requests are not proof of message delivery, and simulated values must be distinguished from hardware readings. Like most existing studies, the system was evaluated with controlled fall simulations rather than real-world falls, and the performance of threshold-based methods can degrade across different users and conditions, which motivates the planned machine learning extension, long-term data storage and improved power management.

V. CONCLUSION

This paper presents an IoT-based elderly fall detection and health monitoring system using smart footwear, designed to improve the safety and healthcare monitoring of elderly individuals through wearable sensor technology and wireless communication. The proposed system combines multiple sensing components, embedded processing, and a web-based monitoring platform to provide continuous observation of movement, physiological parameters, and surrounding environmental conditions. The main objective of this work is to develop an integrated and accessible monitoring solution that can assist in identifying potential fall-related events and provide relevant health and sensor information to caregivers.

The proposed smart footwear integrates multiple sensors, including the MPU6050 accelerometer and gyroscope, MAX30102 pulse oximeter, Force Sensitive Resistor (FSR), HC-SR04 ultrasonic sensor, and Light Dependent Resistor (LDR). Each sensor performs a specific monitoring function within the system. The MPU6050 measures acceleration and angular velocity to identify sudden movements and changes in body orientation associated with potential falls. The FSR monitors variations in foot pressure, providing additional information about foot contact and movement. The MAX30102 measures heart rate and estimates blood oxygen saturation (SpO_2), while the HC-SR04 and LDR monitor nearby obstacles and ambient light conditions, respectively. By integrating these sensing components into a single wearable platform, the system provides information from multiple aspects of the user's condition and surrounding environment.

The Arduino UNO acts as the central processing unit responsible for collecting sensor readings, evaluating predefined threshold conditions, and executing the programmed monitoring and fall-detection logic. The combination of motion-related information and foot-pressure variations supports the identification of potential fall-related situations. In addition to sensor-based processing, the system incorporates a buzzer to provide local alerts when specified conditions are detected. This provides an immediate indication near the user without depending entirely on remote monitoring. The use of multiple sensor inputs also provides a broader monitoring approach than relying on a single motion sensor alone.

For remote monitoring, the ESP8266 provides wireless connectivity between the hardware system and the web application. The collected sensor information is transmitted through Wi-Fi using structured data communication to the Flask-based backend. The backend processes the received information, maintains the latest sensor readings, and makes the data available to the web dashboard. The dashboard presents important monitoring parameters, including heart rate, SpO_2, distance, light intensity, pressure-related information, and system connectivity status. The implementation also uses a communication-status mechanism to identify whether recent sensor data is being received, helping users distinguish between an active connection and a possible communication interruption. This integration of embedded hardware, wireless communication, backend processing, and web-based visualization enables sensor information to be accessed remotely in a convenient manner.

An important contribution of the proposed work is the integration of fall-related monitoring with physiological and environmental sensing in a footwear-based system. Rather than limiting monitoring to movement alone, the system combines different sensor measurements to provide a more comprehensive view of the user's condition and surroundings. The local alert mechanism and remote dashboard further support the monitoring process by making important information available both near the

user and through the connected application. Such an approach has the potential to assist caregivers in observing elderly individuals, particularly when continuous direct supervision is difficult, while also supporting safer and more independent living.

Although the proposed system establishes an integrated wearable monitoring framework, further improvements and extensive validation are required before it can be considered a reliable solution for all real-world conditions. The fall-detection mechanism can be evaluated using a larger number of participants and a wider range of activities, including walking, sitting, standing, stumbling, and different types of fall-like movements. Further testing can help identify situations that may produce false alarms or missed detections. The predefined sensor thresholds and decision conditions can also be refined based on experimental observations and individual variations in movement patterns. In addition, the accuracy and reliability of physiological sensor readings, wireless data transmission, and system operation should be evaluated under different environmental and usage conditions.

Future work can focus on improving the fall-detection algorithm, reducing false alarms, optimizing power consumption, and enhancing the reliability of wireless communication. A dedicated emergency notification mechanism can also be developed to send timely alerts to caregivers when a potential fall event is detected. Further enhancements may include long-term health-data storage, historical trend analysis, a more compact and comfortable footwear design, and testing with elderly users in realistic environments. These improvements could make the system more practical for continuous everyday use and strengthen its usefulness in elderly healthcare applications.

Overall, the proposed IoT-based smart footwear system demonstrates the potential of combining wearable sensors, embedded processing, and web-based monitoring to support elderly safety and health observation. By bringing together fall-related movement monitoring, physiological sensing, environmental monitoring, local alerts, and remote access to sensor information, the system provides a foundation for developing affordable and accessible wearable healthcare solutions. With further testing and refinement, the proposed approach can contribute towards improved elderly monitoring, timely caregiver awareness, and greater independence in daily living.

REFERENCES

[1] A. Shirley J., E. F. Sundarsingh, V. Saraswathi, S. Sankareshwari, and S. Sona, “Fall Detection Smart-Shoe Enabled with Wireless IoT Device,” Circuit World, vol. 47, no. 4, pp. 325–334, 2021.

[2] M. S. Rahman, A. Nikoukar, and M. Günes, “(POSTER) Smart Shoe for Fall Detection Application,” in Proceedings of the 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), 2023.

[3] S. Y. Sim, H. S. Jeon, G. S. Chung, S. K. Kim, S. J. Kwon, W. K. Lee, and K. S. Park, “Fall Detection Algorithm for the Elderly Using Acceleration Sensors on the Shoes,” in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2011, pp. 4935–4938.

[4] R. Aznar-Gimeno, G. Labata-Lezaun, A. Adell-Lamora, D. Abadía-Gallego, R. del-Hoyo-Alonso, and C. González-Muñoz, “Deep Learning for Walking Behaviour Detection in Elderly People Using Smart Footwear,” Entropy, vol. 23, no. 6, Art. no. 777, 2021.

[5] W. Anderson, Z. Choffin, N. Jeong, M. Callihan, S. Jeong, and E. Sazonov, “Empirical Study on Human Movement Classification Using Insole Footwear Sensor System and Machine Learning,” Sensors, vol. 22, no. 7, Art. no. 2743, 2022.

[6] D. K. Agrawal, W. Usaha, S. Pojprapai, and P. Wattanapan, “Fall Risk Prediction Using Wireless Sensor Insoles With Machine Learning,” IEEE Access, vol. 11, pp. 23119–23126, 2023.

[7] W. Guo, X. Liu, C. Lu, and L. Jing, “PIFall: A Pressure Insole-Based Fall Detection System for the Elderly Using ResNet3D,” Electronics, vol. 13, no. 6, Art. no. 1066, 2024.

[8] Y. Zhang, Y. Yan, N. Wallis, W. Sheng, B. Zhu, Z. Zhang, Q. Chen, and F. Zhang, “Smart Self-Powered Insoles for Real-Time Fall Risk Detection,” Chemical Engineering Journal, vol. 530, Art. no. 173190, 2026.

[9] Z. Song, J. Ou, L. Shu, G. Hu, S. Wu, X. Xu, and Z. Chen, “Fall Risk Assessment for the Elderly Based on Weak Foot Features of Wearable Plantar Pressure,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 1060–1070, 2022.

[10] S. Wu, J. Ou, L. Shu, G. Hu, Z. Song, X. Xu, and Z. Chen, “MhNet: Multi-Scale Spatio-Temporal Hierarchical Network for Real-Time Wearable Fall Risk Assessment of the Elderly,” Computers in Biology and Medicine, vol. 144, Art. no. 105355, 2022.

[11] Y.-H. Nho, J. G. Lim, and D.-S. Kwon, “Cluster-Analysis-Based User-Adaptive Fall Detection Using Fusion of Heart Rate Sensor and Accelerometer in a Wearable Device,” IEEE Access, vol. 8, pp. 40389–40401, 2020.

[12] F. Hussain, M. Ehatisham-ul-Haq, and M. A. Azam, “Activity-Aware Fall Detection and Recognition Based on Wearable Sensors,” IEEE Sensors Journal, vol. 19, no. 12, pp. 4528–4536, 2019.

[13] W. Saadeh, S. A. Butt, and M. A. Bin Altaf, “A Patient-Specific Single Sensor IoT-Based Wearable Fall Prediction and Detection System,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 27, no. 5, pp. 995–1003, 2019.

[14] S. Campanella, A. Alnasef, L. Falaschetti, A. Belli, P. Pierleoni, and L. Palma, “A Novel Embedded Deep Learning Wearable Sensor for Fall Detection,” IEEE Sensors Journal, vol. 24, no. 9, pp. 15219–15229, 2024.

[15] A. Choi, T. H. Kim, O. Yuhai, S. Jeong, K. Kim, H. Kim, and J. H. Mun, “Deep Learning-Based Near-Fall Detection Algorithm for Fall Risk Monitoring System Using a Single Inertial Measurement Unit,” IEEE Transactions on Neural

Systems and Rehabilitation Engineering, vol. 30, pp. 2385–2394, 2022.

[16] S. Moulik and S. Majumdar, “FallSense: An Automatic Fall Detection and Alarm Generation System in IoT-Enabled Environment,” IEEE Sensors Journal, vol. 19, no. 19, pp. 8452–8459, 2019.

[17] A. Tahir, W. Taylor, A. Taha, M. Usman, S. A. Shah, M. A. Imran, and Q. H. Abbasi, “IoT Based Fall Detection System for Elderly Healthcare,” in Internet of Things for Human-Centered Design: Application to Elderly Healthcare, Studies in Computational Intelligence, vol. 1011, pp. 209–232. Springer Nature, 2022.

[18] Z. Jiang, N. T. Al Jallad, D. Al Alimi, A. A. Ewees, M. Abd Elaziz, A. Dahou, and A. M. Helmi, “Fall Detection Systems for Internet of Medical Things Based on Wearable Sensors: A Review,” IEEE Internet of Things Journal, vol. 11, no. 21, pp. 34797–34810, 2024.

[19] A. Singh, S. U. Rehman, S. Yongchareon, and P. H. J. Chong, “Sensor Technologies for Fall Detection Systems: A Review,” IEEE Sensors Journal, vol. 20, no. 13, pp. 6889–6919, 2020.

[20] M. E. Karar, H. I. Shehata, and O. Reyad, “A Survey of IoT-Based Fall Detection for Aiding Elderly Care: Sensors, Methods, Challenges and Future Trends,” Applied Sciences, vol. 12, no. 7, Art. no. 3276, 2022.