Kufa Journal of Engineering
Vol. 16, No. 4, October 2025, P.P. 435-457
Article history: Received 4 January 2025, last revised 2 March 2025, accepted 2 March 2025

IOT-BASED LOW-COST WEARABLE INTERACTIVE WIRELESS EMBEDDED COMMUNICATION SYSTEM FOR HEALTH AND SPORT APPLICATIONS

Rabee M. Hagem[1], Mustafa Qassab[2], and Basman M. Hasan Alhafidh[3]

[1] Assistant Professor, Department of Computer Engineering, University of Mosul, 44002 Mosul, Iraq, Email:rabeehagem@uomosul.edu.iq

[2] Lecturer, Department of Computer Engineering, University of Mosul, 44002 Mosul, Iraq, Email:mustafa.qassab@uomosul.edu.iq

[3] Lecturer, Department of Computer Engineering, University of Mosul, 44002 Mosul, Iraq, Email:bm.alhafidh@uomosul.edu.iq

https://doi.org/10.30572/2018/KJE/160425

ABSTRACT

Internet of things is significantly important in different applications including health and sport. Size and weight in wearable sensors are very important factors the designer should consider. Designing an adaptable and scalable system using wearable wireless sensor nodes that can be used in different applications with low cost, and low energy consumption is a challenge. This paper focuses on building an affordable wearable wireless embedded system for health and sports applications based on the Internet of Things (IoT). A novel multi-purpose IoT-based node of less than 10$ with a diameter of 4.2cm and 12g of weight (25g with GPS) is presented. Fall detection using Machine Learning (ML) is considered for the health application, achieving 99% accuracy. Moreover, for sports, a real-time data gathering and analysis is performed, also simulated scenarios using OPNET showed a latency of less than 50µs at 54Mbps. Data for both applications are stored in a MySQL database and can be managed using a web server. Generally, the node operates for about 12 hours with a 240mAh battery, and methods to expand battery life are discussed. Additionally, a robust data security model is presented. When compared to alternatives, the system proves its feasibility, adaptability, scalability, cost-effectiveness, low energy consumption, and high performance.

KEYWORDS

Wireless embedded communication, Low-cost, Health and sport applications, IoT, Interactive wearable system.

1. INTRODUCTION

IoT wearable nodes are smart devices that can be divided into external and internal things. The former can be worn as accessories such as a watch or can be attached to clothing, while the latter can be smart implanted inside the body. This feature allows gathering data anytime and anywhere as it always accompanies the user, unlike ambient systems that have restrictions of indoor usage only (Fernández-Caramés and Fraga-Lamas, 2018).

The typical wearable smart sensor consists of more than one sensor known as a transducer, a microcontroller, and a transceiver, all powered by a battery (Meijer, 2008; Săcăleanu et al., 2017). The smart sensor could be an intelligent sensor (node) that can make decisions. In addition, it can provide real-time feedback to users as it has the following functions: self-testing, self-validation, self-checking, self-diagnosis, self-adaptation, self-identification, self-calibration, self-compensation (Yurish, 2010). This intelligent node can be used in different applications including health and sports applications; Therefore, different aspects should be considered such as computational capabilities, data accuracy, lifetime, cost, data transmission range, bandwidth, memory, or size. Using smart wearable devices enhances the life quality through increasing productivity and safety which are important goals for sustainability. Also, air pollution monitoring is an important application based on IoT to ensure clean air which is one of the important goals of global sustainability (AlSheikh et al., 2021).

In this paper, we are going to discuss two systems for health and sports applications. A GPS-featured fall detection system is used in the former, whereas measuring vital signs during running is an example of the latter. In health design experiments, the authors applied multiple ML algorithms using primary real-life collected data to extract the main features that should be relied on and considered.

In sports design experiments, the authors assume that 10 runners in a field with 10 TX nodes send data to a single RX node with a certain scheduling and priority to be chosen by the coach's threshold or criteria. In any wearable design, energy consumption is an important factor that should be considered as the nodes are battery-powered. In addition to the security aspect, where data is being exchanged in a real-time manner, the system network is simulated and its performance is analyzed and evaluated in terms of throughput, latency, and packet loss. The main goals of this work are divided into a design of a lightweight (Hardware and Software) node for health and sports applications, which means minimizing the software complexity as well as the size and hardware weight of the node that significantly reduces the power consumption and enhances the battery lifetime. Different techniques have been presented in this paper to maximize the lifetime of a battery such as controlling the duty cycle of the pulse, i.e. using the low-energy consumption mode in the microcontroller, with minimal interval time of operation.

As a matter of fact, the designed node can be configured and programmed to facilitate using it in various embedded wireless communication systems.

2. LITERATURE REVIEW

Smart Wearable Nodes can collect and analyze data and even make smart decisions. The wearable nodes can be classified into four major categories: health, sports and daily activity, tracking and localization, and safety (Dian et al., 2020). The focus of this paper is on sports and healthcare as case studies. Therefore, the coming sections take these research areas in more detail.

2.1. Sport Applications

Regarding sports applications, the authors in (Mencarini et al., 2019) present a review of wearable devices for sports applications. Different sports were discussed including running. The main role of wearable devices is to enable the user to do something that cannot be done without the aid of technology. Such a technology supports players to do activities in a better way, and augmenting which means doing something in a better way when compared to how it was done without the wearable devices. Most of the wearable devices that are presented are wrist-mounted. Different feedbacks for the athletes are discussed including concurrent and terminal. Similar to (Liu et al., 2016), which presents a wearable online system for running analysis with different features including compatibility, lightweight, and low power consumption. Sparse Adaptive Sensing (SAS) algorithm to reduce energy consumption is also proposed, which selectively identifies the best sampling points to maintain high accuracy while greatly reducing sensing and analysis energy overheads. Experiments were conducted with 97% accuracy. The proposed system has been used by different runners during daily training and at top level races.

In general, running is the top participatory sport (Department, 2023) as over 200 million regular runners are estimated worldwide and 50 million of them are only in the US (Galic, 2024). The annual injury rate among runners is around 50% - 70% and the main cause of that is poor running according to most physiologists (Daoud et al., 2012). Thus, studying running form and improving running activity does not only participate in the performance enhancement of runners but also has a direct impact on the reduction of injury rate. The attention to this subject was brought for over a century by sports physiologists and coaches due to its importance (Fischer and Braune, 1899).

One constraint that limits the process of gathering comprehensive running form statistics is the laboratory environment, where high-speed video cameras are equipped. Additionally, different sensors are attached to the runner's body covering various reference points. Prior to performing a test, a calibration while standing is needed before running on a treadmill then the 3D positional trajectory per reference point is created over time (Bonacci et al., 2013). Due to restrictions, long setup and post-processing time, and experimental costs, mostly elite athletes are targeted for limited-scale research studies. Furthermore, the collected data is of a short period in a controlled environment. However, this does not suit larger-scale studies that could last for weeks or months, i.e. running form effects while training, yet outdoor tracks considering weather conditions and natural outdoor terrain. Therefore, adopting a technology for motion sensing, acceleration, velocity, and position tracking of the human body such as a wearable Inertial Measurement Unit (IMU) which includes an accelerometer and gyroscope is of great need. Which is used in different applications such as a 3-axis accelerometer with optical wireless communication that has been used in swimming sport to send real time data (Hagem et al., 2012, Hagem et al., 2013). This wearable solution comes with key challenges such as real-time feedback and a limited battery life span which will be addressed in this paper. So, the wireless sensor node is designed to achieve four main requirements represented by the low-cost design, low-energy consumption, scalability, and adaptability.

2.2. Healthcare Applications (Fall Detection)

Falls represent a major public health risk worldwide for elderly people. A fall that was not assisted in time can cause functional impairment to an elder and a significant decrease in his mobility, independence, and life quality. The authors in (Yacchirema et al., 2018) suggested an IoT-based system for fall detection in elderly people in indoor environments focusing on low-power wireless sensor networks, smart devices, big data, and cloud computing. A wearable 3D-axis accelerometer embedded into a 6LowPAN device is used for collecting data. The movement collected data are processed and analyzed using a decision tree algorithm to activate an alert and send a notification to the caregivers in case of detecting a fall. The feedback could be concurrent or terminal. The experimental results show the accuracy of the proposed system. Further, different limitations of the state-of-the-art in fall detection have been discussed in (Saleh et al., 2020) including limitations related to the sensors that are used to capture the motion signals, the positions of these wearable sensors, the sampling frequency, measurement range, and the type of fall. A comprehensive data acquisition system is presented with a large dataset called FallAIID which consists of more than 26k files collected through three wearable devices that were worn on the waist, wrist and neck of an individuals. Motion signals are captured using an accelerometer, gyroscope, magnetometer, and barometer. Deep learning and classical learning algorithms have been used. In (Hussain et al., 2024) IoT based Arduino healthcare system to track vital health data such as temperature, oxygen saturation, and heart rate that are important for predicting and detecting in real-time is presented. The proposed system supports an alternating mechanism for the patient and the medical professional. Another paper (Yee et al., 2019) proposed a fall detection system to work because of limited healthcare staff in rural and urban areas. The suggested system is based on wearable devices to classify falling and non-falling activities accurately. Table 1 summarizes the previous work.

Table 1 Literature review summary.

paper	Design	Multi-purpose	Energy consumption	Security	Network analysis	Cost-effective	User interface (GUI)	Longevity of the system	Comfortability	Interoperability (compatibility)	Online (cloud usage)	Scalability	Accuracy
Liu et al., 2016	✓		✓				✓	✓	✓	✓			97%
Dian et al., 2020	✓	✓					✓		✓	✓	✓		99%
Saleh et al.,2020	✓	✓						✓	✓				90%
Yee et al., 2019	✓	✓	✓			✓	✓			✓	✓		100%
Proposed work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	99%

The remainder of this paper is organized as follows. In section III, the classification of smart wearable devices that are used in health and sports are presented. Section IV discusses the methodology of building the hardware and software of the nodes is presented. Section V presents experimental results and discussions. Section 6 discusses issues related to network simulation, performance analysis, energy consumption, link budget calculations, and security policy. Section VI discusses the challenges of the wearable nodes in IoT systems with future directions and conclusions.

3. WEARABLE IOT NODES CLASSIFICATIONS

IoT nodes could be worn on different parts of the body and measure different data for different applications such as:

A. Healthcare: The IoT wearable nodes are mainly used to monitor patients, treatment, and rehabilitation in some cases. The nodes are responsible for collecting data through the connected sensors since most of them perform essential computations before transmitting to the Internet for further analysis. The last is delegated to cloud computation to save the battery power and eventually increase the battery lifetime span.

The nodes that are used in healthcare can be classified based on the sensors used to:

1. Bio-potential Sensors: such as EEG, ECG, EMG, and PPG.
2. Motion sensors: such as accelerometer and gyroscope that are used with the current proposed node's hardware components.
3. Environmental sensors: such as ultrasound, pressure temperature, etc.
4. Biochemical sensors: such as glucose analyzer.

Fig. 1 shows the wearable healthcare sensors that have been used in IoT systems.

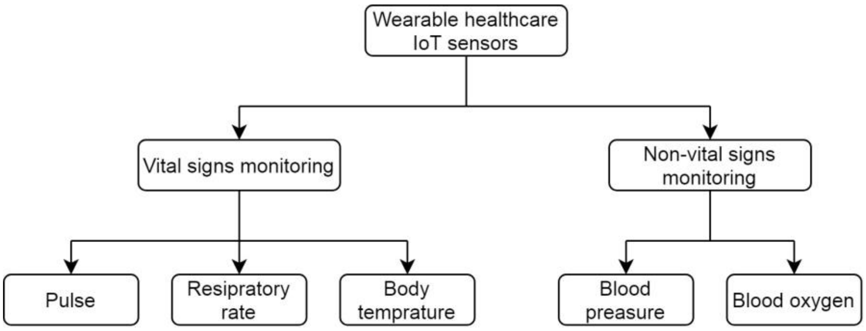


Fig. 1. Wearable healthcare IoT sensor types

B. Activity Recognition and Sports: Wearable nodes are worn during activities to record different metrics and give feedback about performance in real-time or afterward to improve the athlete's performance. Human Activity Recognition (HAR) is also important and can be done using the wireless embedded sensor that is proposed in this paper.

C. Tracking and localization: This includes finding a position of people and tracking them through using GPSM to help in analyzing the movement data.

D. Safety: This includes using wearable devices in different safety applications such as fall detection and prevention, conditions detection of the environment, and drowsiness fatigue detection.

In most cases, the node must be designed in the smallest size and with low energy consumption as possible. Fig. 2 shows different smart wearable embedded systems that are adopted for different applications.

Different aspects of wearable IoT nodes have been presented in (Seneviratne et al., 2017) which provides a survey of available wearable smart nodes in the market. Different sensors that have been used in wearable nodes are presented in (Sazonov, 2020).

4. METHODOLOGY AND SYSTEM IMPLEMENTATION

IoT wearable nodes are smart devices that can be divided into internal and external. The external is related to the devices that can be worn as accessories on the human's body such as watches and glasses or what can be embedded in clothing. On the other hand, the internals can be implanted inside the body.

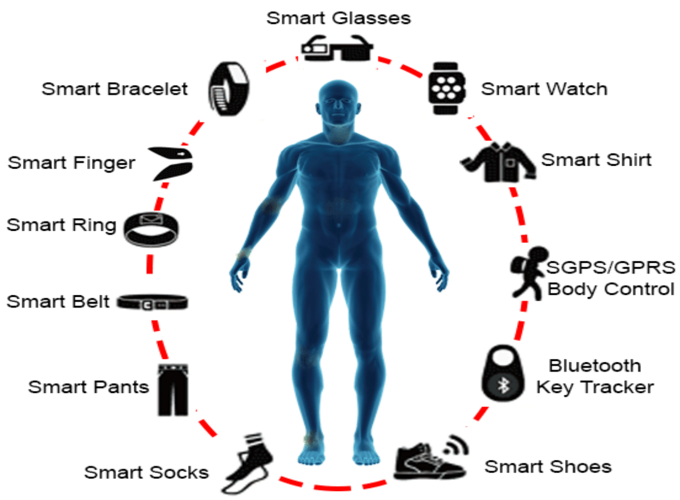
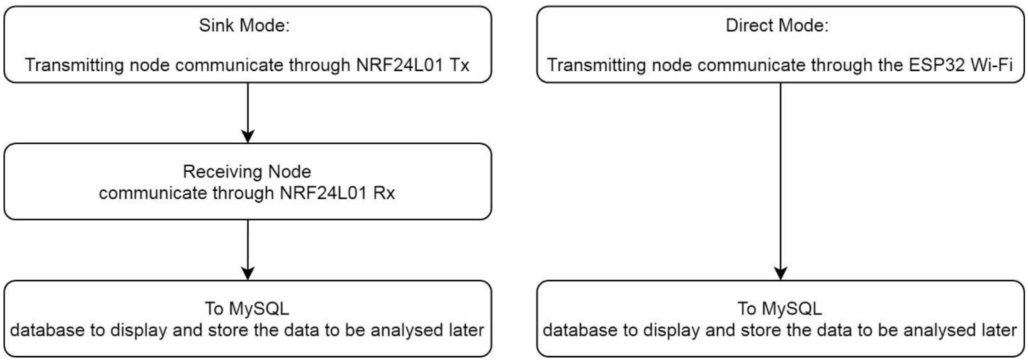


Fig. 2: Different human's wearable systems are developed for various applications [24].

4.1. Methodology

System nodes can be connected directly to the Internet (Direct Mode) to send, receive, and collect data. Another way of communication is to collect the data locally first (Sink Mode) and then send it to the cloud if needed. Therefore, we have two ways for wireless communication as clarified in Fig. 3.a. & 3.b. The first way is implemented through the use of Wi-Fi in the ESP32 or ESP12. The second uses the 2.4 GHZ NRF24L01 module which is Industrial, Scientific and Medical (ISM Band).



(a) (b)

Fig. 3. General communication modes in the IoT nodes: (a) Sink Mode. (b) Direct Mode.

Important information can be extracted from the collected data and a smart decision can be made accordingly. These wearable nodes can interact using wireless embedded communication with other devices such as smartphones and computers, as the best strategies of designing and implementing a human interaction system are discussed in (Kheder, 2023). Smart nodes are becoming increasingly important due to the mobility of the human which help in enhancing the quality of life and increasing productivity and safety. The general operation algorithm includes all stages; sending, receiving, processing, and database storage as illustrated in Fig. 4.

The two modes have been tested. In the sink mode, the transceiver which is the NRF24L01 on the Tx node collects the data and sends it to the transceiver on the Rx node, then the Rx node is responsible of uploading the data to MySQL database. This is limited with 100m range of the NRF24L01 in open area with a limitation of the message size and SNR. However, the direct mode sends the data directly to the database server through the Wi-Fi which is embedded in the ESP32 with no middle nodes (NO Rx node) and hence less energy is consumed. Both modes can be used. However, the latter is more reliable and is adopted in this paper.

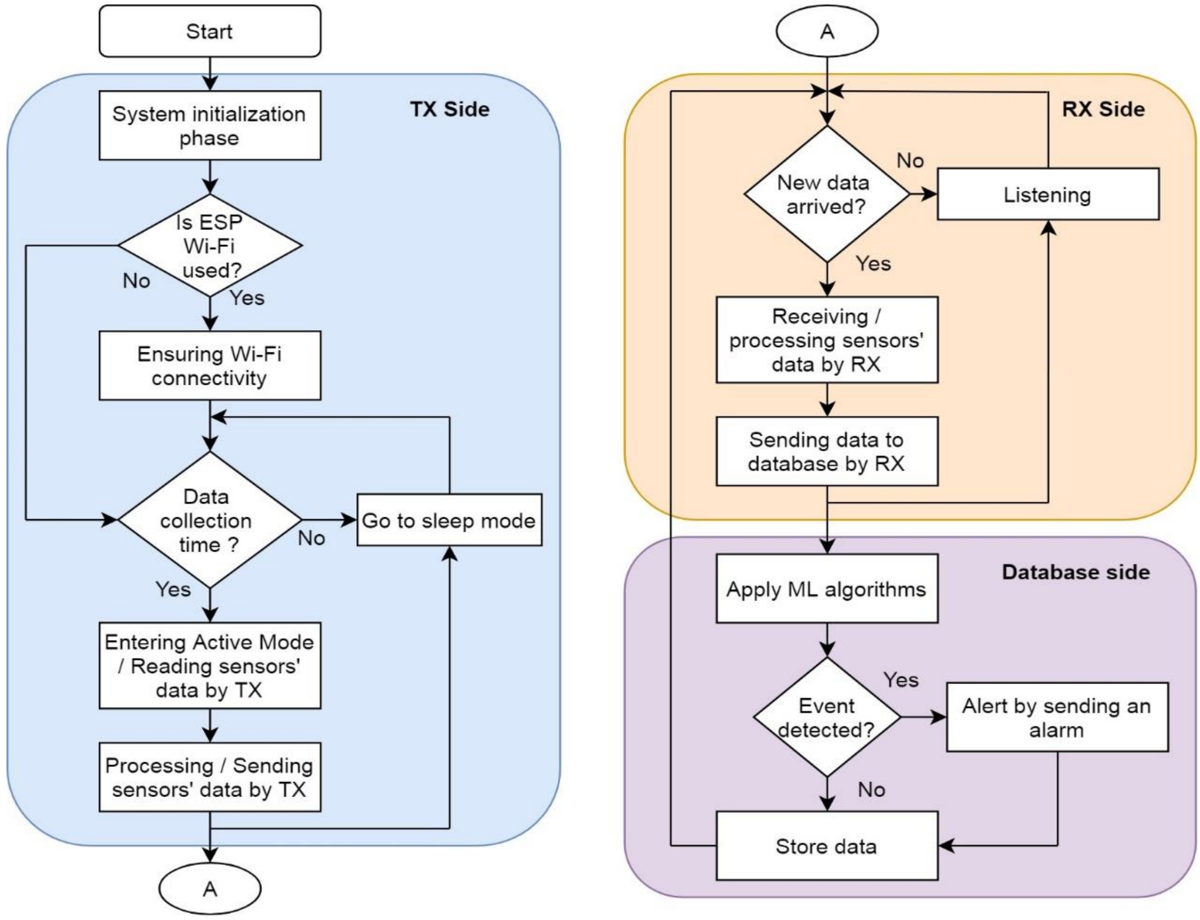


Fig. 4. The general system operation algorithm.

4.2. Node Hardware Specifications and Implementation

The design and implementation of a smart wireless embedded communication node were proposed to be used in different applications including health and sports. In general, the system comprises four main units; sensing, processing, battery, and RF (TX and RX) units. The transmitter node includes the 3-axis accelerometer and gyroscope sensors (MPU6050) which makes the movement with a 6 DOF and the GPS (Neo05) with NRF 24L01 (2.4 GHz transceiver) which are considered as the important sensors for health and sports applications and the microcontroller (ESP32) with a Wi-Fi connection. Other used components include; voltage regulator (MCP1700T), switch, LED, capacitor, resistors, PCB, rechargeable lithium battery (240mAh). The receiver node includes the (ESP32) microcontroller with the (NRF24l01) which works as a receiver to receive the motion data and the position from the transmitter node. Fig. 5, presents a detailed hardware components and clarifies the axes directions using accelerometer, and gyroscope sensors in the proposed node.

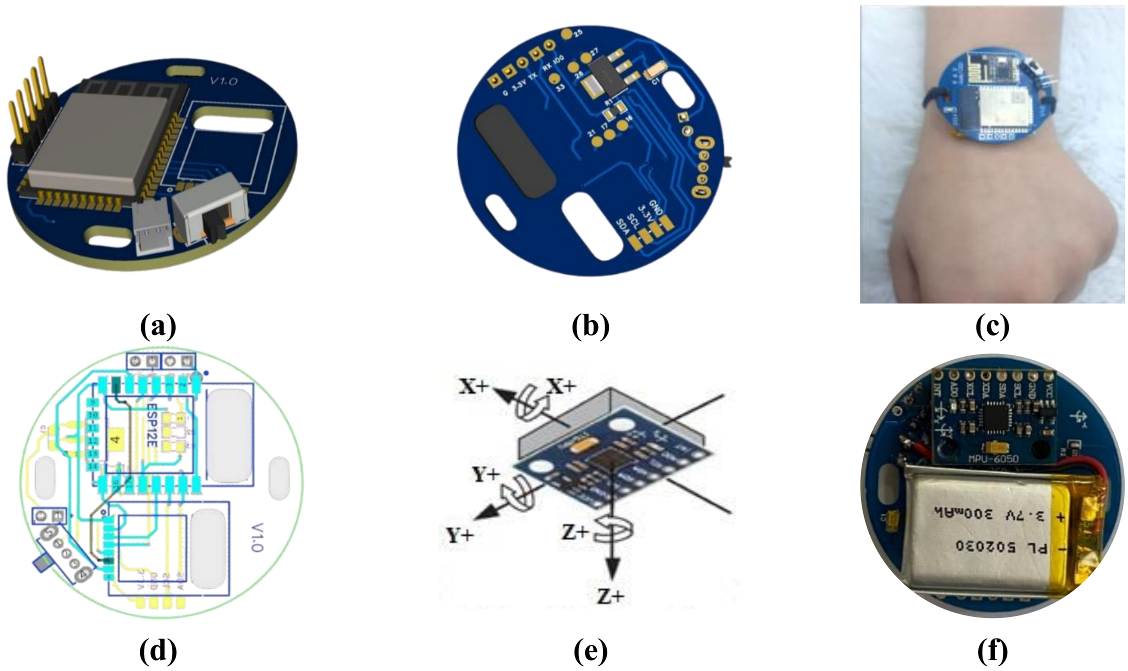


Fig. 5. (a) PCB 3D-layout: top layer. (b) PCB 3D-layout: bottom layer. (c) Actual node worn on a wrist. (d) Node layout showing the schematic diagram of the top layer. (e) Axes directions. (f) Accelerometer, gyroscope, and battery position at the bottom layer of Fig. 5-c.

The hardware node, as shown in Fig. 5-a and 5-b represents the PCB top and down layer of the proposed node. While Fig. 5-c represents the actual node that is worn on a wrist of a volunteer. Fig. 5-d shows the node layout of the top layer. Many expansion pins appear on the bottom layer of the node that can be used to attach new sensors as a means of scalability. Note that the accelerometer, Gyroscope, and battery are connected at the bottom layer of the PCB board as shown in Fig. 5-f, hence it clarifies the axes directions presented in Fig. 5-e.

All the components are soldered carefully onto the PCB using a heating source with a special soldering paste. The components are chosen with low energy consumption to maximize the battery lifetime. Choosing the best axis direction was not randomly done, instead, ML has been implemented to figure out the most effective axis when fall is occurred, results showed that the Z-axis can be relied upon to decide a true fall event. Therefore, we focused on reading only the Z-axis data.

4.3. Machine Learning Implementation on the Fall Collected Data

Machine learning techniques have been applied to the collected data to discover the most important features affecting the data of fall. Recently, artificial intelligence and machine learning have gained huge significance and popularity. At the same time, AI and ML algorithms offer practical solutions for many problems in our daily lives. Therefore, there are many fields and disciplines that have integrated AI algorithms to overcome and tackle the challenges they encounter. On the other hand, there are some fields that are developed and innovated due to the existence of AI algorithms. For example, computer vision, speech recognition, natural language processing, and biometrics are built entirely on AI and ML techniques. However, ML algorithms and techniques require real life data to capture the actual pattern and form representative distribution. Acquiring such data is not an easy task and it is very expensive process. Therefore, we designed our project in a manner that facilities the process of gathering and collecting our own dataset. Upon completing the designed hardware, we used the proposed prototype design to generate our own dataset. After some successful experiments, we were able to gather about 900 measurements.

To emphasize the data pre-processing needed, we applied data cleaning and labeling to remove noisy and redundant data. First, we annotated the collected data manually to provide proper output for each individual input. The manual annotation is a very challenging task and labor-intensive process because it requires time and effort. However, it is an essential step and inevitable in some cases. Therefore, we simply apply the gathering process by guiding our participants into a constrained environment. Ten participants collaborated to collect data in a systematic manner to record the falling/not-falling duration time. The recorded time was used to label the collected data.

After the annotation process on our dataset, which is the collected fall detection data, was completed, we divided our collected dataset into two parts. The first part of our data was used to train different types of machine learning algorithms. The second part of our dataset was used to measure the performance of the prediction (trained) models to evaluate and assess the efficiency. Cross-validation analysis was used to measure and evaluate the learned model from the training process. A 10-fold cross-validation method was selected by repeating the random splitting of the data ten times. Where each time the training segment was used throughout the training phase and the testing segment was used to evaluate the classification model. Then, we computed the average performance of the ten trails.

The complexity-accuracy tradeoff has been considered. Therefore, most of the classifiers that have been used are single and the rest are ensemble. There are numerous numbers of classification algorithms with different characteristics and properties. Therefore, selecting the best classification algorithm technique for a particular problem is a delicate task and can be quite challenging. The training phase can be considered as one of the most important steps in our project. Therefore, we used a very wide range of spectrum to implement the classification algorithms. About twenty different types of machine learning techniques were implemented to be an essential part of our system to play an important role in decision-making and the prediction process to determine if the fallen state accrues or not. The implemented classification algorithms can be categorized into linear and non-linear techniques. For instance, we trained Support Vector Machine (SVM), Linear Discriminant Analysis, and Logistic Regression algorithms as linear classifiers. Additionally, we trained Decision Tree, Quadratic Discriminant Analysis, and K Nearest Neighbors (KNN) as non-linear classifiers. Fig. 6 shows the block diagram of machine learning implementation.

On the other hand, the implemented classification algorithms can be divided into single-learning approaches and ensemble-learning approaches. All the aforementioned classification algorithms are classified as a single-classifier learning technique. An ensemble of learning techniques trains multiple classifiers and combines their results to improve the performance. For example, we trained the Extra Trees Classifier, Random Forest Classifier, and Ada Boost Classifier as the ensemble of classifier approaches.

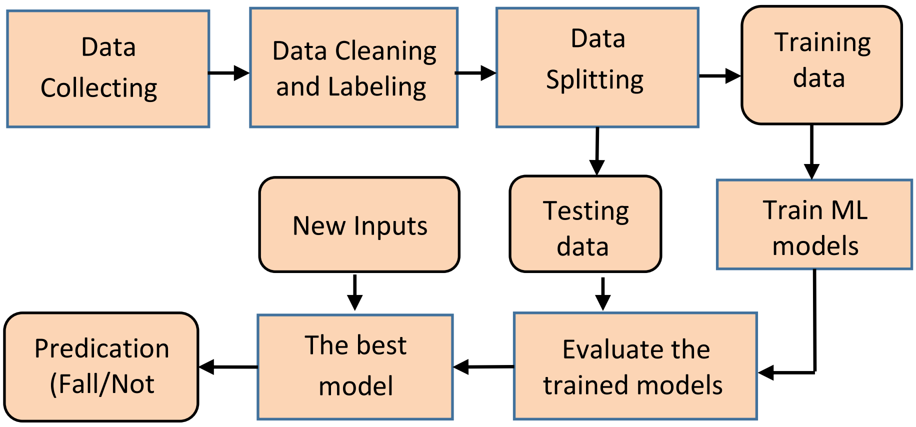


Fig. 6. Block diagram of ML implementation

4.4. Operation and Testing

The collected data are sent to a MySQL database server and analyzed using Machine Learning (ML) techniques to find out the most effective features to be extracted and considered.

The accelerometer, gyroscope, and GPS data are sent to a database that is designed to save the data every two seconds. As mentioned before in subsection 4.1, two options for sending the data wirelessly are available. The first one is through the NRF 24L01 transceiver with a range of up to 100 meters. The second option is to send the data by the ESP32 microcontroller using a Wi-Fi network to the database as shown previously in Fig. 3.

A website has been created and a PHP page for saving the data in a database. MySQL database has been used as shown in the figures below when the node was active. The link below is linking to the website: https://nodemcu.hyantalm.com/convert1.php

The resulting database of an active node can be retrieved as an MS Excel file with all the required details as shown in Fig. 7.

	A	B	C	D	E	F	G	H	I	J
1	accX	accY	accZ	gyroX	gyroY	gyroZ	lat	lon	Date	Time
2	12.34	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	20:46:24
3	12.34	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	20:47:41
4	12.34	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	20:47:45
5	15	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	20:48:46
6	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	20:48:53
7	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:15
8	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:18
9	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:20
10	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:23
11	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:26
12	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:28
13	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:31
14	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:34
15	15.44	56.78	90.12	1.23	4.56	7.89	37.7749	-122.419	29/07/2024	21:14:41

Fig. 7. The generated MS Excel file that contains the database details for the system.

5. RESULTS AND DISCUSSIONS

As the system works, the nodes create a database of the readings in a real-time manner. The collected data could be classified and only the most effective data can be considered. Data assessment and classification can utilize ML as a means to extract the most important features.

5.1. Fall detection assessments

Different experiments have been conducted to collect data. The volunteers were asked to walk normally for two minutes and then to make a fall for another two minutes. The accelerometer and the gyroscope data are recorded every 2 seconds in the web server and the database that is designed for this purpose. The saved data is analyzed later for feature extraction and the Machine Learning (ML) algorithms are applied. The 3-axis of collected acceleration, in addition to gyroscope data, are shown in Figs 8, and 9 respectively.

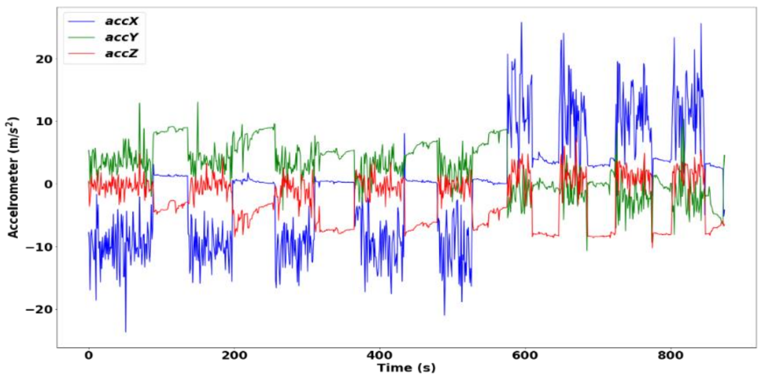


Fig. 8. The 3_axis collected acceleration data.

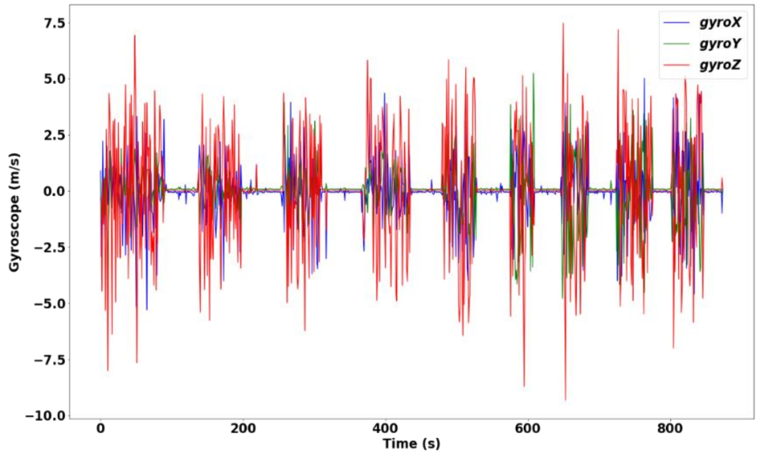


Fig. 9. The 3-axis collected gyroscope data.

5.2. Running case study

The camera can be used to get different videos then divide them into frames and extract the important features. However, for small details, it would be difficult to capture very small details that the MEMS devices can get. This paper is divided into two parts. The first part focuses on fall detection and prevention by collecting data and analyzing it using different ML algorithms. The second for the running part, when a simulation has been done using OPNET to represent the runners and their coach focusing on sending and receiving the data and studying the network performance parameters including; latency, throughput, traffic sent/received, packet loss, and jitter.

5.3. Results of Machine Learning Algorithms

In this section, we tend to cover all the conducted experiments in our work to determine the optimal classification algorithm to form the core prediction model that is responsible for falling detection. As mentioned before, the authors trained the selected ML techniques using a 10-fold validation method by splitting the data into 70% for the training and 30% for the testing. The results of the conducted experiment are shown in Table 2. In addition to the accuracy, the authors computed recall, precision, and F1-score metrics to widen the scope of the evaluation process. Furthermore, we measured the execution time of the trained classification model.

As shown in the above table, the trained classifiers were sorted from the highest accuracy to the lowest. Gaussian Process Classifier achieved about 99% of accuracy where this model learns probabilistic function based Gaussian distribution. Furthermore, we measured the confusion matrix of the trained model as shown in Fig. 10. To apply full analysis, we computed the importance of each feature in our dataset. Feature importance can provide the significant value of the features in our dataset and measure their rank. The results analysis of this step is shown in Fig. 11.

Table 2 Results of comparing different ML methods.

Model	Accuracy	Recall	Prec.	F1-score	Execution time
Gaussian Process Classifier	0.9902	0.9856	0.9930	0.9892	0.6870
Extra Trees Classifier	0.9886	0.9749	1.0000	0.9871	0.3070
Naive Bayes	0.9870	0.9713	1.0000	0.9851	0.0300
Quadratic Discriminant Analysis	0.9870	0.9713	1.0000	0.9851	0.0310
MLP Classifier	0.9869	0.9856	0.9858	0.9854	0.2790
Random Forest Classifier	0.9853	0.9713	0.9964	0.9833	0.2240
Ada Boost Classifier	0.9853	0.9820	0.9858	0.9837	0.1420
Gradient Boosting Classifier	0.9853	0.9820	0.9858	0.9837	0.2830
SVM - Radial Kernel	0.9837	0.9642	1.0000	0.9811	0.0650
K Neighbors Classifier	0.9837	0.9784	0.9858	0.9819	0.0500
Extreme Gradient Boosting	0.9837	0.9747	0.9893	0.9816	0.0730
Light Gradient Boosting Machine	0.9821	0.9747	0.9857	0.9799	0.3500
Logistic Regression	0.9820	0.9820	0.9787	0.9801	0.5550
Decision Tree Classifier	0.9788	0.9820	0.9722	0.9767	0.0310
SVM - Linear Kernel	0.9739	0.9566	0.9856	0.9698	0.0330
Ridge Classifier	0.9723	0.9749	0.9645	0.9693	0.0300
Linear Discriminant Analysis	0.9723	0.9749	0.9645	0.9693	0.0470
Dummy Classifier	0.5490	0.0000	0.0000	0.0000	0.0280

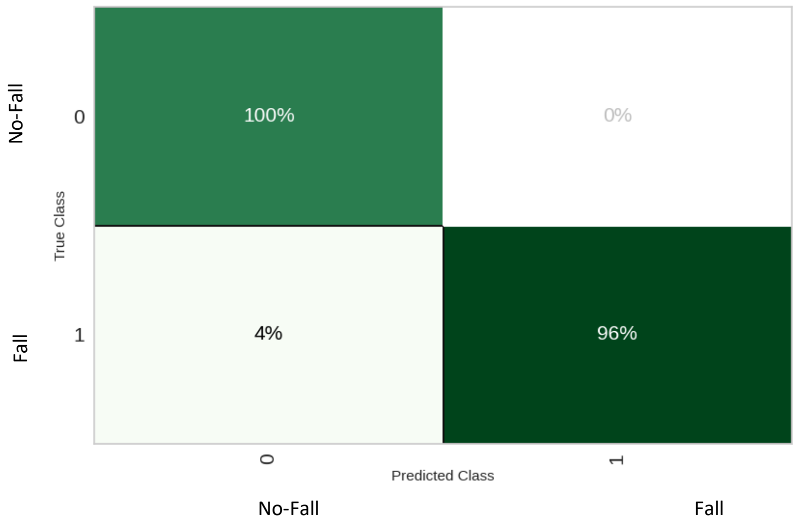


Fig. 10. Confusion matrix of the trained model.

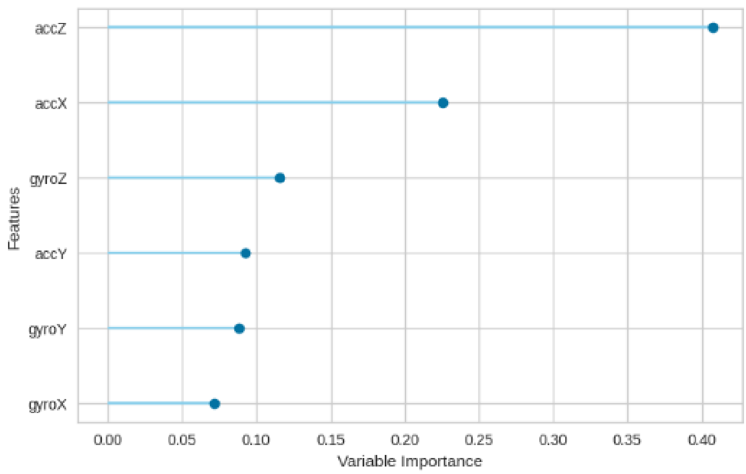


Fig. 11. Feature importance of the trained model.

6. SYSTEM ANALYSIS

6.1. Network Simulation and Performance Analysis

To estimate the expected message size in the system described, you need to consider the data being transmitted from each component. Here's how you can calculate it based on the components:

6.1.1. Components and Data Size

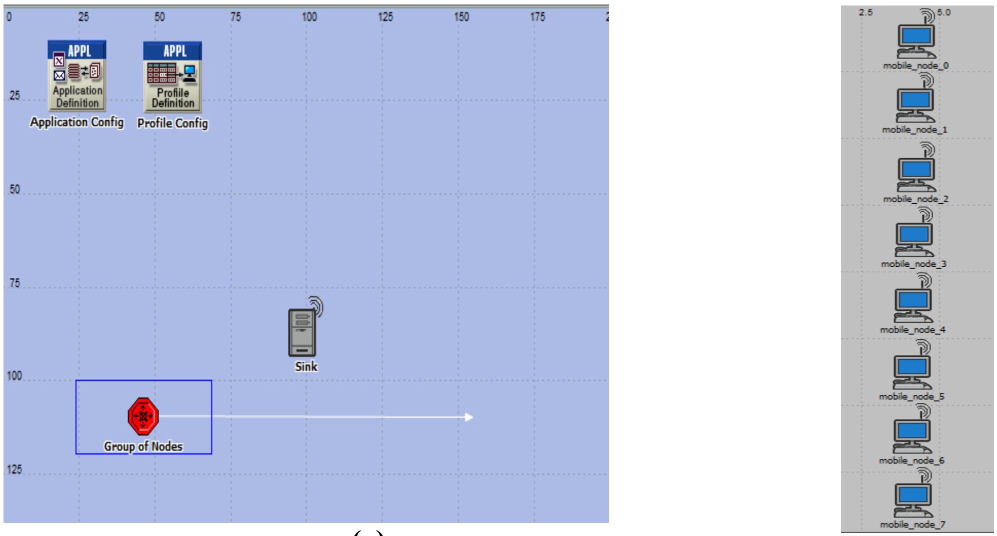
1. MPU-6050 (Accelerometer and Gyroscope)
    - Data: X, Y, and Z values for both accelerometer and gyroscope
    - Data Size:
        - Each axis value is typically a 16-bit integer.
        - Total: 3 axes (accelerometer) + 3 axes (gyroscope) =6 axes3
        - Data Size: 6 axes×2 bytes/axis=12 bytes
2. **GPS NEO-6**
    - Data: Latitude, Longitude, and possibly other values like altitude, speed, or time
    - Data Size:
        - Latitude: 4 bytes (float or 32-bit)
        - Longitude: 4 bytes (float or 32-bit)
        - Altitude: 4 bytes (float or 32-bit)
        - Total: 3 values×4 bytes/value=12 bytes
3. **nRF24L01 (Transceiver)**
    - Data Overhead:
        - Includes the data payload and additional overhead for packet headers and CRC.
        - Typically, a packet might be 16 bytes including headers and payload.
4. **ESP32**
    - Processing Data: The ESP32 processes and manages the data, but the message size mainly depends on the data payload from MPU-6050 and GPS NEO-6, which affects the overall network performance as discussed in (Hasan, 2022).

Combining the data from MPU-6050 (12 bytes) and GPS NEO-6 (12 bytes) leads to a total expected message size of 24 bytes excluding packet headers.

The nRF24L01 Packet Overhead adds additional overhead for the nRF24L01 transceiver, typically around 16 bytes, to account for packet headers, CRC, and other protocol-related data. Results in 40 bytes per message to be exchanged in the network.

6.1.2. OPNET Simulator Settings

Optimized Network Engineering Tools (OPNET) simulator is considered one of the most powerful software simulation packages for networks (Adaramola Ojo Jayeola and Olasina, 2020, Barznji and Ameen, 2021). Thus, the network diagram that mimics the proposed system is created using OPNET as shown in Fig. 12. As can be noticed, Fig. 12 (a) shows the network representing the sink node (the trainer) and a subnet which resembles a group of nodes, while Fig. 12 (b) represents the nodes within the subnet aligned vertically (runners). The sink node collects data from the mobile nodes that are mobile and follows a straight trajectory of 100 meters in length. It is better to place the sink node in the middle of the network to ensure better signal reception from all mobile nodes.



(a) **(b)**

Fig. 12. (a) The proposed system network simulated using OPNET. (b) The mobile nodes within the subnet represent the runners

OPNET 14.5 network simulator is used to simulate the proposed system. The simulation settings are shown in Table 3.

Table 3 Simulation settings.

Parameter	Value
Simulation time	15 seconds
Number of nodes	8 mobile nodes and one sink node
Network area span	200 meters x 200 meters (40 km^2)
Distance between nodes	equal
WLAN protocols and data rate	802.11b (1Mbps), 802.11b (2Mbps), 802.11b (5.5Mbps) 802.11b (11Mbps), 802.11g (54Mbps), 802.11a (54Mbps)
Nodes status	Active and follow the trajectory
Node processing rate	120M packet/sec
Node buffer size	128K Byte

The scenario assumes that there are 8 runners, as mobile nodes, wearing the proposed smart system and one coach as a single sink node. In addition, the readings are sent from each mobile node to the sink every 0.2 seconds with a data size of 40 bytes. Different wireless protocols are considered with different data rates for the simulated scenarios.

6.1.3. Simulation Results and Discussion

After running the simulator through using a different WLAN protocol for each run, the authors measure the main network performance parameters including; latency, throughput, traffic sent/received, packet loss, and jitter. The results show an acceptable value of packet loss for all different scenarios that could be ignored, which is less than 2%. Besides, the number of network packets was just above 40 packets/second, which makes traffic about 12.8 kbps of actual data being exchanged. The jitter value ranges from 0.001 μsec to 0.05 μsec depending on the WLAN protocol used and could be ignored. The throughput was the same for all cases and around 21.76 kbps for the network. However, latency was directly affected by network protocol and data rate as illustrated in Fig. 13.

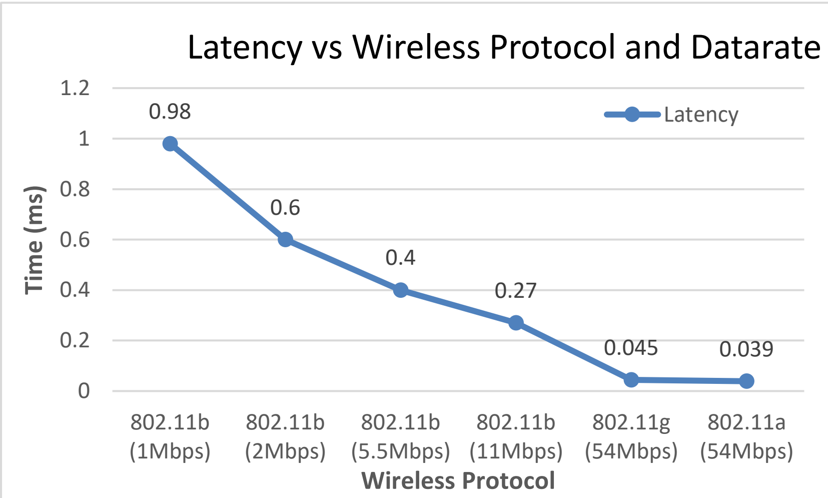


Fig. 13. WLAN protocol and data rate against network latency.

As can be seen in Fig. 13, the latency is affected by the data rate of the WLAN, and it is inversely proportional to it. This means that acting closer to a real-time manner needs a higher network data rate.

6.2. Energy Consumption Calculations

Understanding energy consumption for embedded systems in microcontroller applications, including the microcontroller and sensors in the proposed work, is crucial for optimizing performance and prolonging battery life. It involves evaluating power requirements across different operational modes such as active execution, sleep states, and transmission/reception tasks. Below is a brief definition of each operational mode:

- Active Mode: Includes data processing, code execution, peripheral control, and performing various tasks.

- Sleep Mode: Involves reducing power consumption by keeping only essential peripherals active while running the main program; the clock speed of the microcontroller is reduced to save power.
- Transmission Mode: Data is transmitted using communication interfaces like UART, Bluetooth, and Wi-Fi. This mode usually consumes more power than other modes.
- Reception Mode: The microcontroller listens for incoming data to be received.

The components used are:

- MPU-6050: 3-axis gyroscope and a 3-axis accelerometer with a power consumption of 3.9mW.
- GPS-NEO6: GPS receiver with 35mW as power consumption.
- nRF24L01: Wireless transceiver with power consumption of 11mW in transmission and 13mW in reception.
- ESP32: Microcontroller with Wi-Fi and Bluetooth, 160mW in active mode and 10mW in sleep mode.

The calculations assume the following assumptions:

Data collection rate for MPU-6050 and GPS-NEO6 = 1Hz.

Frequency of transmission and reception for nRF24L01 and ESP32 = 0.2 seconds.

Operating modes = 10% active and 90% sleep.

Battery voltage = 3.7v.

Battery capacity = 240mAh.

By considering these four modes, the consumed power can be calculated as:

$$P_{avg} = \frac{(P_{MPU-6050} \times t_{MPU-6050}) + (P_{GPS} \times t_{GPS}) + (P_{tx} \times t_{tx}) + (P_{rx} \times t_{rx})}{t_{total}} \quad (1)$$

The total operation time could be found using:

$$t_{total} = t_{active} + t_{sleep} + t_{tx} + t_{rx} \quad (2)$$

By substituting the values for each component and the required time of operation, this results in a Pavg= 262.76mWh and a total operation time of about t_{total}=12 hours.

The above calculations consider 40 bytes of data transferred for each message, including the 16-byte GPS coordinates. If the GPS feature is disabled, i.e. for the running sport application, then the message size will be reduced and hence the battery will last longer.

6.3. System Cost Breakdown

For the system to be reasonably affordable and meet the minimum requirements is important. The proposed system is composed of several nodes. In Table 4 we estimate the cost for an individual node including all its components (Electronics).

Table 4 The estimated cost for the node's components.

NO.	ITEM	QUANTITY	TOTAL COST
1	ESP32	1	$3.53
2	NRF module (NF-03)	1	$1.1
3	MPU 6050	1	$1.10
4	Neo-05	1	$1
5	Voltage regulator (MCP1700T)	1	$0.2925
6	Switch	1	$0.074
7	LED	1	$0.0109
8	Capacitor	1	$0.0849
9	Resistors (x2)	2	$0.002
10	PCB	1	$0.5
11	Battery	1	$0.85

6.4. Security Policy

For any IoT-based system, security is an important aspect to be covered to avoid intervening and manipulating the data (Mahmood and Abdul-Jabbar, 2023). Therefore, aspects such as secure boot, authentication, and data ciphering and integrity should be considered for the proposed system. This can be achieved by defining sources and types of threats and then adopting effective strategies to minimize their effect (Qaddoori and Ali, 2023, Ibrahim and Qassab, 2022). A brief definition of possible network threats is given in the following. See Table 5 for more details.

- Physical Security: Attackers may access the location of the device and tamper with it, allowing them to steal data or alter some of its functionality.
- Misconfiguration: Keeping default passwords or misconfiguring some settings could expose the system to vulnerabilities like data breaches and unauthorized access.
- Compromising IoT Devices: Gaining control of the device remotely by exploiting vulnerabilities could lead to data leakage or launching further attacks.
- Network Threats: Disrupting network communication could result in data leaks or even service interruptions; network spoofing and packet sniffing are ways to pose such threats.
- Unauthorized Access: Stolen credentials and weak passwords may lead to unauthorized admission to access data.
- Attacks on Server: This could lead to disrupting services and is often done by Distributed Denial of Service (DDoS) attacks and unauthorized attempts to sign into the server.
- Application Layer Threats: Accessing and manipulating data could be done by taking advantage of vulnerabilities within the application itself. Techniques like cross-site scripting (XSS) and SQL injection are used to serve such purposes.

Generally, considering security recommendations will improve the protection of data and user credentials. Some good practices include maintaining up-to-date protective software, monitoring and reacting to any suspicious events, and applying multi-layer authentication approaches. Adopting such practices leads to a trustworthy environment for IoT systems.

Table 5 Possible Security Threats And Countermeasures For The Proposed System.

THREAT NAME	Threat Type	Countermeasure	Justification
PHYSICAL SECURITY	Tampering	Adopting Hardware Security Modules (HSMs) and secure boot processes.	HSMs protect cryptographic keys and data, also assuring that devices run only trusted firmware by using secure boot.
	Device Theft	Keeping devices in well-locked enclosures.	Denying the intruders from physical access to devices.
MISCONFIGURATION	Default Settings	Altering vulnerable settings and default passwords.	Factory settings and passwords are widely known thereby they should be changed.
	Open Ports	Scanning for and closing unneeded open ports.	Limiting the attack surface by detecting and closing unneeded ports.
COMPROMISING IOT DEVICES	Malware	Adopting anti-malware and antivirus techniques.	Detecting and removing harmful codes and programs.
	Firmware Weaknesses	Updating device firmware regularly and applying up-to-date patches	Preventing attackers from exploiting certain security vulnerabilities.
NETWORK THREATS	Sniffing	Applying Wi-Fi encryption and VPNs.	Disallowing the capture and reading of data.
	Man-in-the-Middle Attack	Adopting data ciphering mechanisms such as SSL/TLS encryption.	Ensuring that data is received intact.
UNAUTHORIZED ACCESS	Credential Theft	Creating strong passwords and applying two-factor authentication (2FA).	Easy passwords could be guessed or found out in a short time.
ATTACKS ON SERVER	DDoS Attacks	Implementing mitigation solutions and rate limiting for DDoS.	Minimizing and managing excessive network traffic by adopting such methods.
APPLICATION LAYER THREATS	Injection Attacks	Assure and check for any user input.	Allowing only correct inputs prevents malicious code injection.

7. CONCLUSION AND FUTURE WORKS

Wearable IoT systems to be used in different applications including health and sport with low cost and low energy consumption is introduced in this paper. The proposed system is designed to be light in weight (12g) without the GPS, adaptable, and scalable with a small size with a diameter of (4.2cm). Different experiments were conducted to collect data for fall detection and prevention and save it on a database designed for this purpose. Various ML algorithms looking for the most effective features that affect the fall are tested. The results showed that the z_axis acceleration is the dominant feature and that the Gaussian process classifier has achieved an accuracy of 99%. The second application is the running, OPNET simulator has been used to get the important network performance metrics including latency, throughput, traffic sent/received, packet loss, and jitter, showing no more than 50µ seconds of delay when applying a 54Mbps data rate. Moreover, energy consumption calculations are presented in this paper proving that the node could last more than 12 hours of operation. Regarding the security threats that data can counter, different security models are presented to protect the data. The affordable, lightweight, small size, and scalable designed node is suitable to be adopted in different applications. Future work can focus on collecting data in running sports and suggest different hardware and software techniques to increase the battery lifetime.

8. REFERENCES

ADARAMOLA OJO JAYEOLA, A. O. J. & OLASINA, J. R. 2020. Network Model Analysis in Opnet Simulation. International Journal of Engineering Applied Sciences and Technology, 5, 47-51.

ALSHEIKH, R., HAGEM, R. & SALIM, O. 2021. A survey on smart monitoring system of environment based on IoT. Al-Rafidain Engineering Journal, 26, 146-158.

BARZNJI, A. & AMEEN, J. 2021. Wi-Max Network Simulation For Salahaddin University New Campus. Kufa Journal of Engineering, 12, 1-13.

BONACCI, J., SAUNDERS, P. U., HICKS, A., RANTALAINEN, T., VICENZINO, B. G. T. & SPRATFORD, W. 2013. Running in a minimalist and lightweight shoe is not the same as running barefoot: a biomechanical study. British journal of sports medicine, 47, 387-392.

DAOUD, A. I., GEISSLER, G. J., WANG, F., SARETSKY, J., DAOUD, Y. A. & LIEBERMAN, D. E. 2012. Foot strike and injury rates in endurance runners: a retrospective study. Med Sci Sports Exerc, 44, 1325-1334.

DEPARTMENT, S. R. 2023. Running & Jogging - statistics & Facts [Online]. Available: https://www.statista.com/topics/1743/running-and-jogging/ [Accessed 28 October 2024].

DIAN, F. J., VAHIDNIA, R. & RAHMATI, A. 2020. Wearables and the Internet of Things (IoT), applications, opportunities, and challenges: A Survey. IEEE access, 8, 69200-69211.

ELECTRONICS, L. Available: https://www.lcsc.com/ [Accessed 1 November 2024].

FERNÁNDEZ-CARAMÉS, T. M. & FRAGA-LAMAS, P. 2018. Towards the Internet of smart clothing: A review on IoT wearables and garments for creating intelligent connected e-textiles. Electronics, 7, 405.

FISCHER, O. & BRAUNE, C. W. 1899. Der gang des menschen, BG Teubner.

GALIC, B. 2024. 126 Running Statistics You Need to Know [Online]. Available: https://www.livestrong.com/article/13730338-running-statistics/ [Accessed 1 November 2024].

HAGEM, R. M., THIEL, D. V., O'KEEFE, S. G., DAHM, N., STAMM, A. & FICKENSCHER, T. Smart optical wireless sensor for real time swimmers feedback. SENSORS, 2012 IEEE, 2012. IEEE, 1-4.

HAGEM, R., THIEL, D., O'KEEFE, S. & FICKENSCHER, T. 2013. Real-time swimmers' feedback based on smart infrared (SSIR) optical wireless sensor. Electronics Letters, 49, 340-341.

HASAN, A. 2022. Application Based performance monitoring heavy data transmission of Local Area Network. Kufa Journal of Engineering, 13, 14-40.

HUSSAIN, I., DESHALAHRE, D. & THAKUR, P. 2024. Assessing the Effectiveness of An IoT-Based Healthcare Monitoring and Alerting System with Arduino Integration. Revue d'Intelligence Artificielle, 38, 1211.

IBRAHIM, Q. & QASSAB, M. 2022. Theory, Concepts and Future of Self Organizing Networks (SON). Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 15, 904-928.

KHEDER, H. A. 2023. Human-Computer Interaction: Enhancing User Experience in Interactive Systems. Kufa Journal of Engineering, 14, 23-41.

LIU, Q., WILLIAMSON, J., LI, K., MOHRMAN, W., LV, Q., DICK, R. P. & SHANG, L. 2016. Gazelle: Energy-efficient wearable analysis for running. IEEE Transactions on Mobile Computing, 16, 2531-2544.

MAHMOOD, M. B. & ABDUL-JABBAR, J. M. 2023. Securing Industrial Internet of Things (Industrial IoT)-A Reviewof Challenges and Solutions. Al-Rafidain Engineering Journal (AREJ), 28, 312-320.

MEIJER, G. C. 2008. Smart Sensor Systems.

MENCARINI, E., RAPP, A., TIRABENI, L. & ZANCANARO, M. 2019. Designing wearable systems for sports: a review of trends and opportunities in human–computer interaction. IEEE Transactions on Human-Machine Systems, 49, 314-325.

QADDOORI, S. L. & ALI, Q. I. 2023. An efficient security model for industrial internet of things (IIoT) system based on machine learning principles. Al-Rafidain Engineering Journal (AREJ), 28, 329-340.

SĂCĂLEANU, D., PERIŞOARĂ, L., SPATARU, E. & STOIAN, R. Low-cost wireless sensor node with application in sports. 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), 2017. IEEE, 395-398.

SALEH, M., ABBAS, M. & LE JEANNES, R. B. 2020. FallAllD: An open dataset of human falls and activities of daily living for classical and deep learning applications. IEEE Sensors Journal, 21, 1849-1858.

SAZONOV, E. 2020. Wearable Sensors: Fundamentals, implementation and applications, Academic Press.

SENEVIRATNE, S., HU, Y., NGUYEN, T., LAN, G., KHALIFA, S., THILAKARATHNA, K., HASSAN, M. & SENEVIRATNE, A. 2017. A survey of wearable devices and challenges. IEEE Communications Surveys & Tutorials, 19, 2573-2620.

YACCHIREMA, D., DE PUGA, J. S., PALAU, C. & ESTEVE, M. 2018. Fall detection system for elderly people using IoT and big data. Procedia computer science, 130, 603-610.

YEE, L. M., CHIN, L. C., FOOK, C. Y., DALI, M. B., BASAH, S. N. & CHEE, L. S. Internet of things (IoT) fall detection using wearable sensor. Journal of Physics: Conference Series, 2019. IOP Publishing, 012048.

YURISH, S. Y. 2010. Sensors: smart vs. intelligent. Sensors & transducers, 114, I.