[1]Qassim A. Al-Jarwany

Ahmed Rifaat Hamad[2]

Bayan Mahdi Sabbar[3]

Mohannad Jabbar Mnati[2,]

Adnan Hussein Ali[2]

Alex Van Den Bossche[4]

System Design and Implementation for Machine Learning and Internet of Things Based Anomaly Detection in Patient Movement

Abstract: - Emerging trends in healthcare have seen an increased need for improved monitoring, and as a result, machine learning (ML) and the Internet of Things have been integrated to develop smart systems that can provide rapid and accurate health assessments. With an inclusive view on mobility anomaly detection in patients, this paper aims to improve hospital patient safety. Patients wear or place these things, and the IoT sensor network then gathers real-time motion data. The information is then moved to the central server. The system was taught the machine learning part using a large number of normal and abnormal patient movement data. It also shows a way to differentiate between common activities and those that might indicate ill health. It comprises an IoT sensor network, a data processing unit, and an ML anomaly discovery module. There are sensors on the Internet of Things that have a wide range of movement data. Then, these are run through an ML model that captures the spatial-temporal dependencies between different sets based upon more sophisticated techniques like the Naive Bayes Classifier. Where there are abnormalities discovered after analyzing them, an alert is sent out; this will help in quick emergency response by saving lives even before diseases become last-stage acute conditions

Keywords: IoT, ML, Naive Bayes, ESP32, MPU6050, Healthcare, and Patient Movement Detection

I. INTRODUCTION

An unprecedented issue has arisen due to global population aging, which has necessitated constant observation of clients within various care settings. This procedure has been made easier by the introduction of cutting-edge technology like wearables, ambient sensors, and integrated communication networks. These tools provide vital information on the health of patients, allowing for prompt interventions and maybe lower expenditures. However, since constant observation raises concerns regarding permission, data security, and striking a balance between oversight and autonomy, ethical and privacy issues must be taken into account. [1-5].

The latter system was suggested as the best method of finding patient mobility irregularities by integrating machine learning algorithms with an Internet of Things sensor network. It includes wearables, motion sensors and other surveillance devices that allow for gathering data concerning the patient's environment without causing any suspicion. The accumulated information served as grounds for inference. These systems use various machine learning strategies to establish what every single patient believes is a regular movement pattern such that it can easily note uncommon events. Such inconsistencies might imply that there are underlying health conditions that need immediate attention, ranging from increased restless periods to potentially fatal falls.

Machine learning (ML) techniques like Naïve Bayes algorithm are integrated seamlessly with an Internet of Things (IoT) sensor network in TP4056 Figure 2c to form the basis of PMAD recommended system illustrated at figure one. The module is designed to operate covertly within a patient's surrounding using motion sensors to gather information. Upon collection, the data underwent analysis to yield valuable insights. Using a range of machine learning techniques, the system learns what each person perceives as normal movement, allowing it to

[1] [1] University of Babylon, Material Engineering College, Department of Polymers and Petrochemicals Industrials Engineering, Babylon, Iraq.

[2] Middle Technical University, Institute of Technology – Baghdad, Department of Electrical Technology, Al-Za'franiya, 10074, Baghdad, Iraq.

[3] Al-Mustaqbal University, College of engineering and engineering techniques, medical instrumentation engineering techniques department, Babylon, Iraq.

[4] Ghent University; Faculty of Engineering and Architecture; Laboratory of Electrical Energy-EELAB; 9000 Ghent; Belgium;

mohannad.mnati@mtu.edu.iq, m.j.mnati@gmail.com

accurately recognize anomalous behavior. These abnormalities might indicate underlying health problems that have to be treated right away. They might involve anything from an increase in the frequency of restlessness to the potentially deadly occurrence of a fall. [6-10].

The implementation of such systems is expected to provide a variety of consequences. It promises better patient safety through recognizing emergencies early, better use of healthcare resources by letting careers attend to real needs instead of just routine check-ins, and possibly lower overall healthcare costs by averting events that might result in readmissions to the hospital. This research examined many case studies and pilot projects where the system was implemented and provided factual proof of its effectiveness. It also describes the difficulties that arose during deployment, including managing false positives in anomaly detection, scaling, and interaction with current healthcare systems. In conclusion, the suggested system is a major advancement at the nexus of IoT, ML, and healthcare. This is an example of how advances in technology if utilized carefully, may transform patient care and support the more general goals of personalized care plans and predictive health monitoring. This essay's remaining sections are organized as follows: Section Two displays the relevant publications; Section Four discusses the final software and hardware system findings; Section Five presents the conclusions; and Section Three provides a detailed explanation of the hardware and software system design parameters.

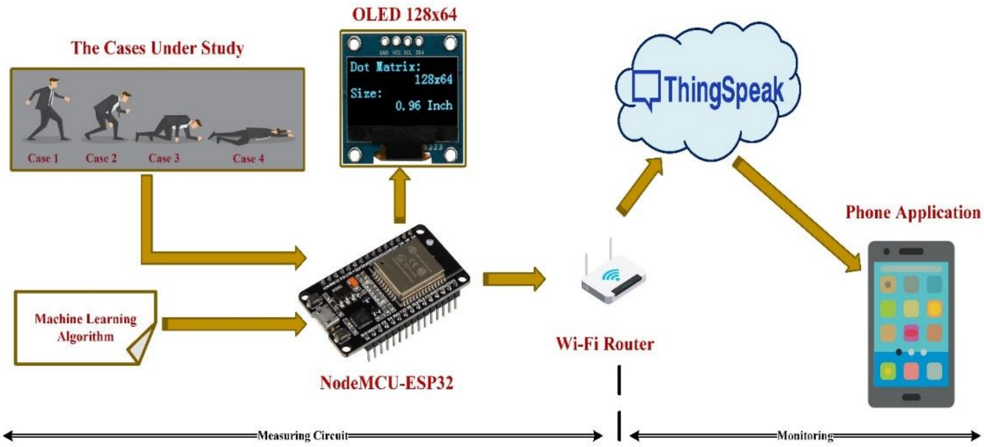


Figure 1: The block diagram of the proposed system

II. RELATED WORK

This section will showcase the planned system in addition to highlighting other similar efforts in the same area, as shown in Table 1. Similar studies using various technologies and methodologies are included in these works; for example, studies employing various types of sensors for Anomaly Detection in Patient Movement using Machine Learning and IoT.

III. SYSTEM REQUIREMENTS

This part can be divided into two main parts: the hardware requirements for a practical system. All the electronic parts used to design the proposed system are discussed in this section. The second section discusses the most important software programs used to program the system and obtain the results.

A. Hardware Requirements

All electronic parts will be discussed in this part of the paper.

1. NodeMCU-ESP32

ESP32 in Figure 2a is a semiconductor device that combines Wi-Fi and Bluetooth capabilities and operates at a frequency of 2.4 GHz. It has been specifically engineered using SMC low-power 40 nm technology. The design of this system aims to optimize the power and radio frequency (RF) performance, demonstrating resilience, adaptability, and dependability over a diverse range of applications and power conditions. ESP32 can also work like a whole standalone system or it can operate as a slave device to a host MCU; this way, the communication stack overhead on the main application processor is reduced. Using its SPI, SDIO, or I2C/UART interfaces, the ESP32 can be interfaced with other systems to provide Wi-Fi and Bluetooth capabilities. Table 2 [19] shows the major characteristics of ESP32.

Table 1: List of Related Work Papers

Related Work [Reference] Year	Type of Sensor	Type of Microcontroller	Type of Communication	IoT	Monitoring	Cloud
A. M. Said et al. [11], 2021	Oxygen, Heart rate, Temperature	Beagle Bone Black	Wi-Fi	YES	Terminal	YES
M. R. Islam et al. [12], 2023	ECG, SP02, Temperature	ESP8266	Wi-Fi	YES	Mobil APP	YES
S. Ayouni et al. [13], 2023	Heart rate, Blood pressure, Temperature	ESP8266	Wi-Fi	YES	GUI	YES
M. Karmakar et al. [14], 2023	X-rays direction	PC	NO	NO	PC	NO
B. G. Mohammed et al. [15], 2023	Heart rate, Blood pressure, Temperature	Raspberry Pi 4B	4G GSM	YES	Mobil APP	YES
J. Heaney et al. [16],2022	ECG, Heart rate, Blood pressure, Temperature	MKA1010	Wi-Fi	YES	OLED	YES
S. Saleh et al. [17],2023	Heart rate, Blood pressure, Temperature , CO2	ESP32	Wi-Fi	YES	Terminal	YES
M. M. Islam et al. [18], 2020	Heart rate, Blood pressure, Temperature , CO2	ESP32	Wi-Fi	YES	Terminal	YES

Table 2: Main properties of the ESP32[19]

Property	Description
Microcontroller	ESP32
Operating Voltage	3.3V
Digital I/O Pins	22-36 (depends on the board variant)
Analog Input Pins	6-18 (depends on the board variant)
Clock Speed	Up to 240 MHz
CPU	Xtensa® dual-core 32-bit LX6 microprocessor
Wi-Fi	802.11 b/g/n (802.11n up to 150 Mbps)
Bluetooth	Bluetooth v4.2 BR/EDR and BLE standards
Flash Memory	4 MB (variants may offer more)
SRAM	520 KB
SPI /I2C / UARTs	4 / 2 / Typically 3
LED PWM	Up to 16 channels

2. MPU6050 Gyroscope.

The MPU6050 in Figure 2b is a microelectromechanical system (MEMS) that includes a triaxial accelerometer and triaxle gyroscope integrated into its structure. This facilitates the quantification of the acceleration, velocity, direction, displacement, and several other properties associated with the motion of a system or object. Robotics, cell phones, gaming consoles, and other motion-tracking gadgets have extensively used MPU6050. This chip is a common option for three-dimensional space measurement applications because it offers six degrees of freedom (3-axis gyroscope and 3-axis accelerometer data). By combining a gyroscope and accelerometer, MPU-6050 can measure the rotational rate as well as the linear acceleration. Table 3 lists the primary characteristics of the MPU6050 Gyroscope used in this study [20].

Table 3: Main properties of the MPU6050 Gyroscope [20]

Property	Description
Gyroscope Sensor Range	±250, ±500, ±1000, ±2000 degrees per second (dps)
Accelerometer Range	±2g, ±4g, ±8g, ±16g
Communication	I2C (up to 400 kHz)
Supply Voltage	2.375V - 3.46V (VDD)
Digital Output	16-bit ADC
Operating Temperature	-40°C to 85°C
Output Data Rate (ODR)	Up to 1 kHz
Gyroscope Sensor Range	±250, ±500, ±1000, ±2000 degrees per second (dps)
Accelerometer Range	±2g, ±4g, ±8g, ±16g
Communication	I2C (up to 400 kHz)

3. The TP4056 Li-Ion Battery Charging Module.

TP4056 in Figure 2c is a popularly used and highly respected integrated circuit (IC) specifically built for charging of single-cell lithium-ion (Li-ion) batteries. It has become increasingly popular among amateurs and professionals because it is tough, cheap, and easy to handle. Since Li-ion cells can get severely discharged and overcharged, this module often has built-in protection and charging functions to ensure safe battery charging. Being able to charge constant-voltage/constant-current (CC/CV) profile batteries makes it more suitable for modern devices requiring effective power management. Because of this, as well as its onboard LED charging indications, low necessity of extra components, and adjustable charging current, the TP4056 is the recommended choice for portable power management. The main features of the TP4056 Li-ion battery charging module are listed in Table 4. [21].

Table 4: Principal characteristics of the Li-Ion battery charging module TP4056 [21]

Property	Description
Battery Type	Single Cell Lithium-Ion
Charging Mode	Constant Current (CC) / Constant Voltage (CV)
Charge Current	up to 1A
Charge Precision	1.50%
Input Voltage	4.5V - 5.5V
Operating Ambient Temperature	-40°C to +85°C
Package	SOP-8
Standby Current	<2uA when no battery is connected

4. OLED 128x64

A popular small and versatile screen for many electrical applications is the Figure 2d organic light-emitting Diodes 128 × 64 display. This type of display contains 128×64 individual OLED pixels that can be controlled separately for display graphics, text, and animations. Because every pixel produces its light, deep black depths and high contrast ratios are possible without the need for a backlight. Their good visibility, large viewing angles, and low power consumption make this type of display popular. The main properties of OLED 128x64 are shown in Table 5 [22].

Table 5: Main properties of the OLED 128x64 [22].

Property	Description
Resolution	128x64 pixels
Interface	Commonly I2C, SPI, or 8-bit parallel
Active Area	Approximately 0.96 inches diagonally
Driver IC	Commonly SSD1306 or similar controller chip
Operating Voltage	3.3V or 5V (varies by model)
Viewing Angle	Up to 160 degrees
Operating Temperature	-40°C to 70°C
Communication Speed	Depending on the interface (400kHz for I2C, higher for SPI)

5. Micro SD Card Reader Module.

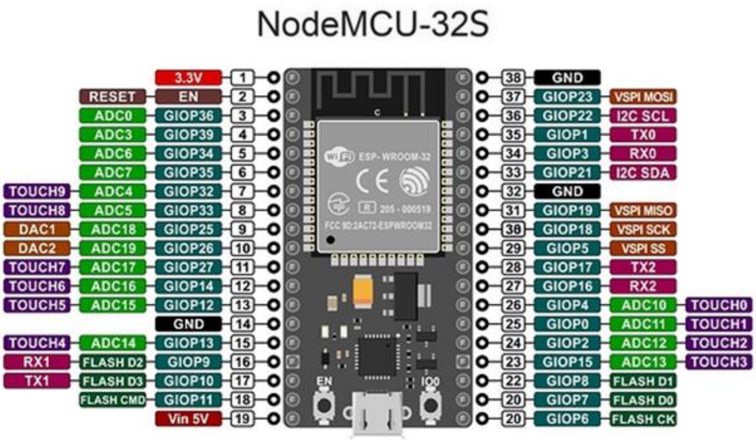
An easy way to move data to and from a normal SD card is to use a microSD Card Module. The pinout may be used with various microcontrollers, although it is directly compatible with the Arduino. This allowed us to include data logging and bulk storage in any project. The module typically provides a slot for inserting a Micro SD card and has a set of pins or contacts that connect to the host system, as shown in Figure 2e. Communication between the module and the microcontroller is generally achieved through a serial peripheral interface (SPI), although some models may support other protocols, such as I2C or parallel interfaces, for faster data transfer rates. It has an SPI interface that is compatible with any SD card and uses either a 3.3V or 5V power supply that is compatible with Arduino UNO/Mega, ESP32, etc. Data loggers, audio, video, and graphics make up some of these applications for SD modules. This module therefore increases the ability of Arduino because it has limited memory [23].

B. Software Requirements

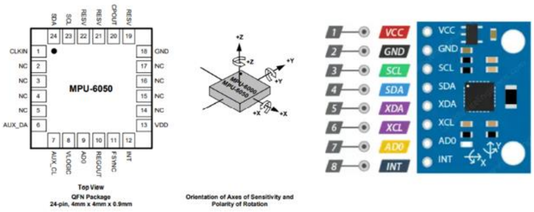
All the software that will be used will be discussed in this section.

1. Python 3.10 software.

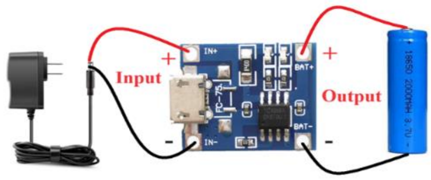
Python 3.12 in Figure 3, released in October 2021, introduces several noteworthy features that further refine and improve the Python programming language. Among the most significant additions is structural pattern matching, which allows complex data patterns to be expressed and matched with more readable and concise syntax and precise types, thus enhancing support for type annotations [24]. Other enhancements include new syntax features, such as the match statement, which is akin to a switch or case statement found in other languages but more powerful because of its ability to destroy and match data structures. Python 3.12 also improves error messages, providing more context to help developers debug their codes more efficiently. The release continued the tradition of optimizing and streamlining the language, maintaining Python's reputation for its readability and ease of use.



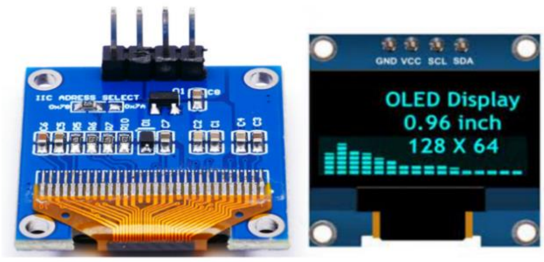
(a)



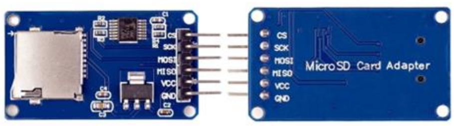
(b)



(c)



(d)



(e)

Figure 2: Hardware Required: a) NodeMCU-ESP32 pinout [19], b) MPU6050 Gyroscope sensor [20], c) TP4056 Li-Ion battery charging module [21], d) OLED 128x64 module [22], and e) Micro SD card reader module [23].

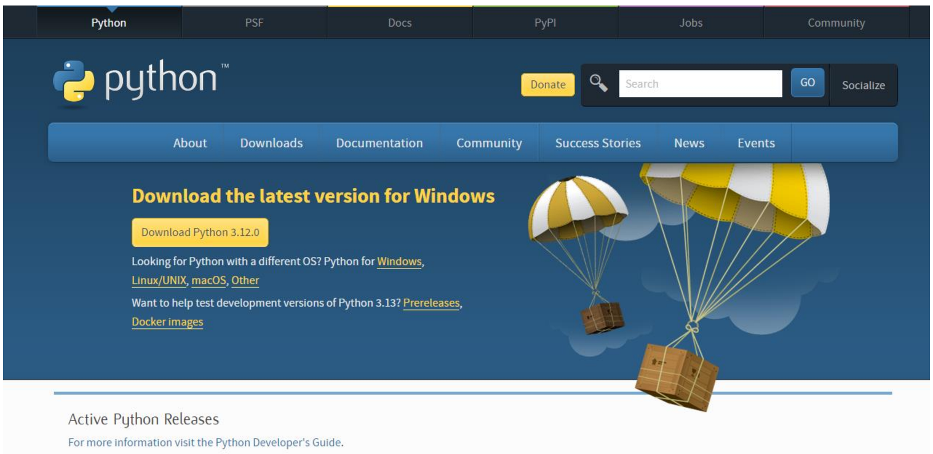


Figure 3: Main website window of Python [24].

2. Arduino IDE Software Platform

The Arduino Integrated Development Environment in Figure 4 is also known as Arduino Software (IDE). It consists of a text editor where one can write code, a message box, a text terminal, and a toolbar that contains buttons for frequently used tasks, among other menu items. It establishes a connection to Arduino hardware to upload programs and interact with them [25]. Sketches are programs created using Arduino software (IDE). The drawings were created in a text editor and stored as files with suffixes. Texts can be replaced, cut, or pasted using editor functions. Moreover, the message box shows feedback about exporting and saving instead of displaying errors alone. All error warnings and other texts outputted from Arduino Software (IDE) will be shown in the console Complete error messages, along with other textual results from Arduino Software (IDE), appear on the console board. At the bottom right corner, you can see what board is configured and what serial port is used. Selectable toolbar buttons allow you to create new programs by opening them, saving them, and verifying them. A serial monitor can also be opened which creates, opens, verifies sketches

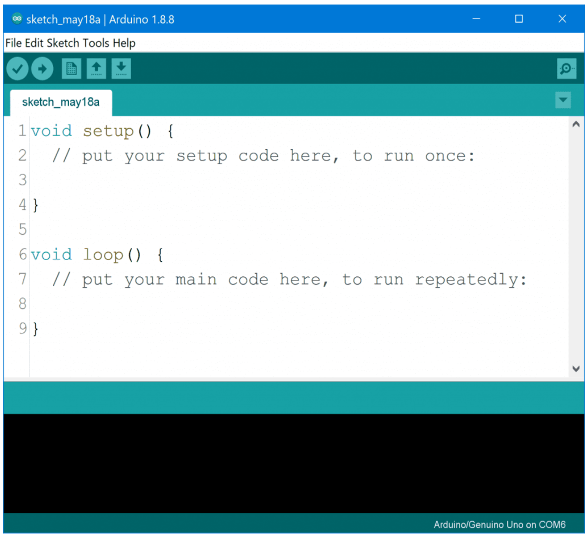


Figure 4: Arduino IDE platform main window [25].

3. ThingSpeak Cloud.

ThingSpeak in Figure 5 is a cloud-based IoT analytics platform that enables the collection, display, and analysis of real-time data. It is possible to send alerts, send data from devices to ThingSpeak, and instantly visualize the live data [26]. Using common IoT protocols, such as HTTP and MQTT, users can quickly configure their devices to submit data to ThingSpeak. For each stream, we use individual API keys for security purposes only accessible by authorized users. Its MATLAB analytics engine enables extensive data analysis efforts without external applications, thereby enabling ThingSpeak's built-in capability according to visualization tools for complex data structures. Furthermore, Reacts, Time Control as well as Talk Back come pre-installed with ThingSpeak which enables action automation against schedules or certain thresholds on data thus furthering the autonomy and interactivity of IoT devices. This makes ThingSpeak a suitable platform for research projects involving IOT data use in industry monitoring or even home automation.

Figure 5: ThingSpeak cloud website window [26]

4. MIT APP Inventor.

The open-source software MIT App Inventor was created by Google and is now overseen by the Massachusetts Institute of Technology, as shown in Figure 6. With its intuitive drag-and-drop interface, anybody can design Android applications, regardless of their experience level. This abstracts the complexity of coding and enables users to construct useful applications. The MIT App Inventor is a user-friendly framework that allows users to create apps without prior programming skills. This serves as a teaching aid by introducing computational thinking and software development ideas. The blocks in the framework help to avoid syntax mistakes, which is a common challenge for new users [27]. MIT App Inventor provides instructional materials, tutorials, and forums for app development in a vibrant community. It is used by enthusiasts and business owners in commercial applications and classrooms to teach programming. Their ease of use and smartphone capabilities enable the development of apps that communicate with hardware and sensors, thereby creating opportunities for mobile computing.

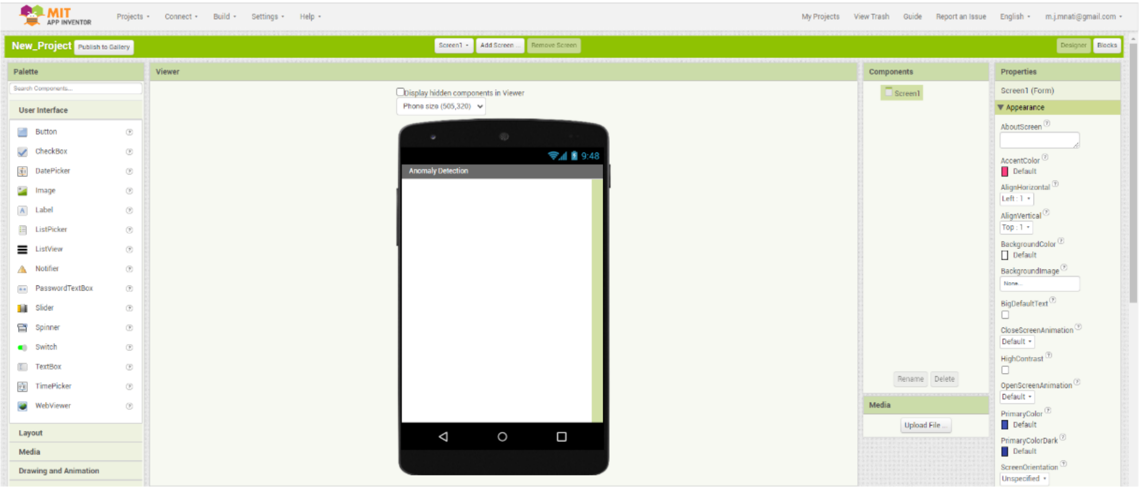


Figure 6: MIT App Inventor window [27].

C. Naïve Bayes Classifier Algorithm.

Machine learning is a part of artificial intelligence that gives self-programming abilities learned from experience with data. Its ability to be integrated into systems in many areas has proven essential for improving functionality, automating hard jobs, and offering predictive capabilities. The Naive Bayes classifier is an essential machine-learning algorithm.

Based on Bayes' theorem, naïve Bayes is a probabilistic machine-learning approach that can be used for a range of categorization problems. It functions based on the independence of the predictors. In particular, Naive Bayes is well known for being straightforward and effective. By incorporating these algorithms into systems, data analysis can be expanded to include previously unattainable or impractical insights. In this section, we cover the Naïve Bayes technique and associated issues in detail to ensure that there are no knowledge gaps [28, 29]

The conditional probabilities were calculated using Bayes' theorem, which is a straightforward mathematical process. As in Equation (1), the conditional probability is the likelihood that an event will occur if another event (by assumption, guess, assertion, or fact) occurs.

$$P(X|\mathrm{Y}) = \frac{P(Y|\mathrm{X}) \square P(X)}{P(Y)} \tag{1}$$

Where: P(X|Y): How often does X occur when Y occurs? P(Y|X):
how often does Y happen given that A happens?

P(X): how likely X is on its own;
P(Y): how likely Y is on its own

Probability P(X|Y) can be broken down using the chain rule: Equation (2)

$$P(X|Y) = P(X_1, X_2, \dots \dots, X_n|Y) \\ = P(X_1|X_2, \dots . X_n, Y) * P(X_2|X_3, \dots . X_n, Y) \square\square\square\square P(X_n|Y) \tag{2}$$

Equation (3) illustrates how the naive conditional independence principle ensures that conditional probabilities are independent of each other.

$$PP(X|Y) = P(X_1|Y) * P(X_2|Y) \square\square\square\square P(X_n|Y) \tag{3}$$

Consequently, Equation (4) provides the findings we get from conditional independence.

$$(Y|\mathrm{X}) = \frac{P(X_1|Y) * P(X_2|Y) \square\square\square P(X_n|Y) * P(Y)}{P(X_1) * P(X_2) \square\square\square\square P(X_n)} \tag{4}$$

Furthermore, the posterior probability may resemble Equation (5), because the denominator is constant for all values.

$$P(X_1, X_2, \dots \dots, X_n|Y) \; \alpha \\ P(Y) \prod_{i=1}^{n} P(X_1|Y) \tag{5}$$

This model was used with a decision rule based on a Naive Bayes classifier. The maximum a posteriori decision rule (MAP rule) is a widely used guideline that involves choosing the hypothesis with the highest probability, as shown in Equation (6).

$$Y = argmax_B P(Y) \prod_{i=1}^{n} P(A_1|Y) \tag{6}$$

IV. FINAL HARDWARE SETUP

two distinct sections, each of which plays a crucial role in overall functionality. The first section focuses on collecting essential training data for the Naive Bayes classifier. At this point, the system collected a vast dataset of different movement examples and captured the complex patterns required for strong training. This dataset serves as the foundation for the machine learning model's comprehension, allowing it to accurately identify and classify motions. A portion of the practical system concentrates on setting up of the last circuit for testing and collecting the findings once the first training data is acquired. This stage shows how the learned Naive Bayes classifier is used in the real world. Using patterns, it has learned, the circuit analyses group actions, guesses, and real-time data. It is probably made up of the MPU6050 gyroscope and the ESP32 board. Important details about the classifier's accuracy, effectiveness, and general performance under real-world conditions were gleaned from the testing phase. It was also important to divide the actual system into two main sections to develop a comprehensive and organized approach, which includes learning and system use: final circuit construction for testing and an initial set of training data collection

A. The System for Gathering the Crucial Instructional Data

The circuit in Figure 7 is a good example of advanced components used for specific purposes. It is the ESP32 board, which serves as the central processing unit, that has already been programmed to perform its tasks on wireless communication. Also, when ESP32 is being used, it can easily process data, make decisions, or link with networks or other external devices. The MPU6050 accelerometer sensor in the companion ESP32 is vital for three-dimensional acceleration data acquisition (X,Y, and Z). This accelerometer sensor helps determine how items are moving or oriented. The combination of ESP32 and MPU6050 allows real-time processing and collection of motion-related data

The circuit's performance is further enhanced by the SD card module as an additional component. One of the useful features provided by this module is the ability to log and preserve data which helps in tracking or analyzing acceleration data over time. The addition of the SD card module provides a circuit with the ability to maintain a record of past movements, which makes it easier to analyze or confirm the system's performance. The synergy between these components signifies a well-rounded architecture. The ESP32 controls the flow of data, the MPU6050 records the important motion-related data, and the SD card module ensures that the data are saved so that they can be used later or analyzed. Figure 7 shows how this fully integrated system works to provide a complete answer for the uses that need to detect motion and store data.

Recognizing the limitations of the raw data, a crucial step involves the accumulation of N instances, as shown in Figure 8 (an example of one of the data collected). Notably, the choice of N as a power of two holds significance for subsequent data-processing techniques. This strategic selection is likely to aid in optimizing the computational efficiency during the subsequent steps. The MPU6050 accelerometer sensor data collection process was done over and over again to make sure that the dataset for analysis in the Naïve Bayes algorithm was diverse and representative. Each repetition involved subjecting the sensor to various scenarios and capturing acceleration data in three dimensions. This iterative approach aims to cover a spectrum of movements and enhance the ability of the algorithm to generalize and accurately classify different activities. The final dataset is a good base for training the Naïve Bayes algorithm. This shows the importance of collecting many different kinds of data to create a good model for analyzing and predicting movements.

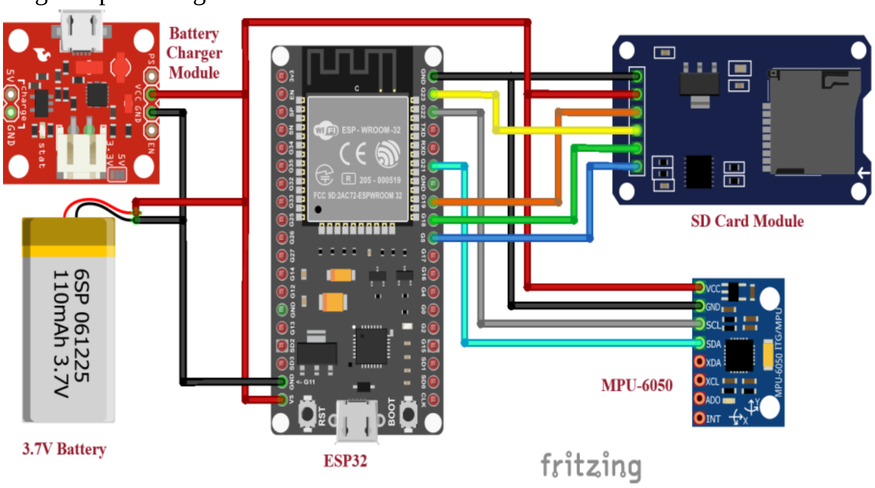


Figure 7: The block diagram architecture circuit for collecting data.

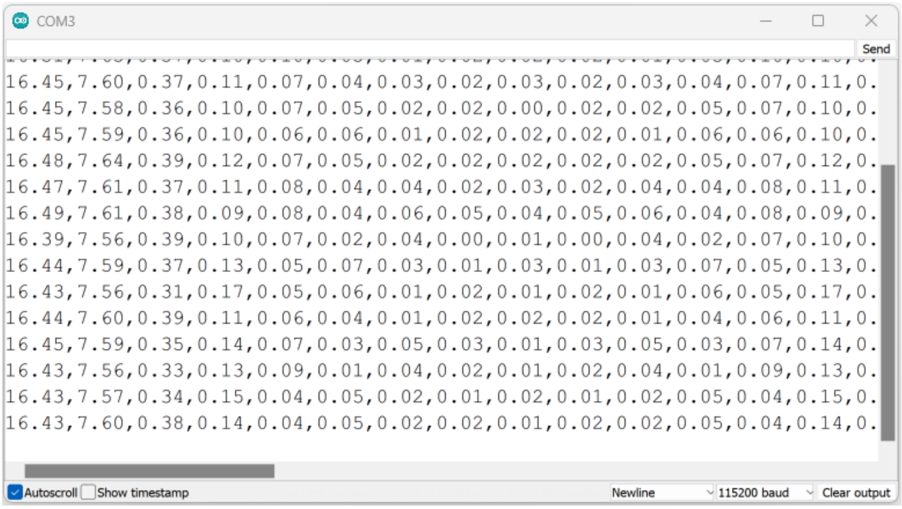


Figure 8: An example of one of the data collected.

B. ***The Proposed circuit***

After the successful gathering of data from the circuit (Figure 7) outlined in the previous section, the next pivotal step involved training the Naive Bayes algorithm in Python. A specialized program developed specifically for this purpose facilitates training. During training, the algorithm processes the accumulated dataset and learns and extracts the most crucial patterns and features associated with different movements. The resulting trained model was then prepared to distill the acquired knowledge into a format suitable for installation on the ESP32 microcontroller. The significance lies in transforming the learned insights into a compact and efficient representation that can be loaded onto the ESP32, thereby forming the intelligent core of the proposed circuit. Python-based training and the subsequent transfer of relevant data to the ESP32 work together without problems. This shows that machine learning and embedded systems can work together well, building a strong base for the circuit's ability to predict and label movement.

The final circuit that has been recommended by this discourse is illustrated in Figure 9. It's a simple assembly made up of the key components needed again for complete functioning. In its basic form ESP32 stands out as a versatile microprocessor having high processing capabilities while supporting wireless communication. Additionally, the circuit contains MPU6050 accelerometer sensor enabling momentary capturing of three-dimensional acceleration data which provides a better capability for detecting diverse physical movements by the system. Moreover, there exists some other interfaces such as OLED 128×64 display which either show the data at once or give visual recommendations on how things work out at present. It is also possible to have the three modules ESP32, MPU6050 and OLED 128×64 connected together at once, which demonstrate a link between motion sensing, human interaction and microcontroller capability in the direction of our study.

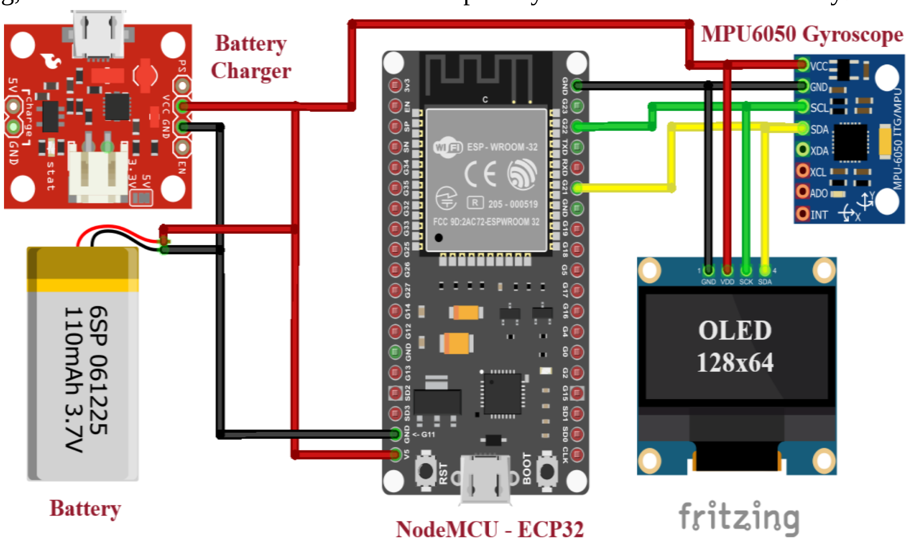


Figure 9: The final suggested circuit schematic architecture.

Each circuit in this part was created using Fritzing software. Fritzing being an open-source tool serves as an easy mean to draw and visualize electrical circuits. This software allows for specification of a particular circuit by its vast library of components and user friendly interface designed for assembling it rightly. The demonstration of functionality for the recommended circuits employed Fritzing because it supports open-source software ideals such as accessibility and collaboration. Therefore, it offers an approachable explanation along with illustrations for electrical design [30].

*C. **The Program for the Final Proposed Circuit***

Figure 10 shows the operation of the programmed circuit and includes:

a) Start Microcontroller: This process starts with the initiation of microcontroller, which creates a program execution environment.

b) Upload Initial Data: Upon starting, microcontroller uploads initial data that is necessary for running a program. These data may include configurations, libraries among other relevant information.

c) Read Real Data from MPU6050 Accelerometer Sensor: The program reads real-time data from the MPU6050 accelerometer sensor. It is essential for capturing information on orientation, acceleration or any other movement related to the device under test through this sensor.

d) Machine Learning (Naïve Bayes Classification): After acquiring the sensor data, machine learning was introduced into our system using Naive Bayes classification. This step entails training or utilizing an already trained model to analyze and classify accelerometer data. Naïve Bayes classification is used widely for pattern recognition as it is probabilistic in nature.

e) Present Results on OLED 128 × 64: Following completion of the machine learning process, the outcomes were visualized on an Organic Light Emitting Diodes (OLEDs)-based display screen having 128 × 64 pixels capability resolution. This display might be employed for exposing classified information or any other output which can result out of a machine-learning algorithm.

f) Check Wi-Fi availability: At this stage, there is a check if Wi-Fi connection exist. This will determine whether external servers or cloud platforms are reachable by this device in next step.

g) Send Data to ThingSpeak Cloud (if Wi-Fi is available): The microcontroller starts the process of transmitting data to the ThingSpeak Cloud when it detects a Wi-Fi connection. An Internet of Things (IoT) platform called ThingSpeak enables data collection and visualization.

h) End:

The flowchart concludes after completing all the specified steps. The system is now actively reading, processing, classifying, and potentially transmitting data based on the sensor readings.

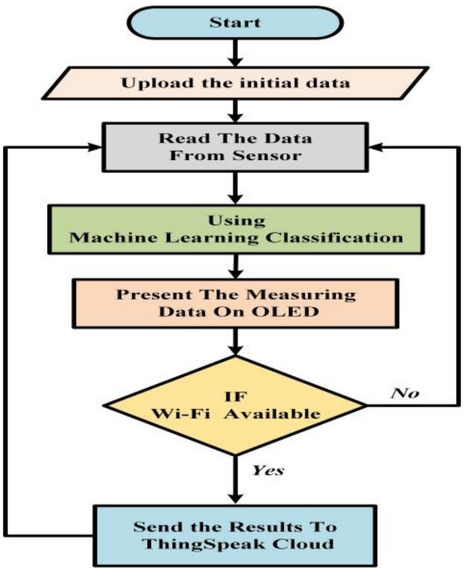


Figure 10: The flowchart of the final proposed circuit

V. RESULTS AND DISCUSSION

After carefully reviewing all the electronic parts and programs that are important to the circuit in this study, along with the first sketches, the focus now shifts to showing the final design. At this point, we have a full understanding of the electronic components and turned the theoretical framework into real-world diagrams. This is the step between conceptualization and realization. The final practical design encompasses the synthesis of hardware and software components, ensuring that the circuit operates as intended. Figure 11 shows the physical combination of the electronic parts. This is an important step in which the microcontroller or processing unit is programmed, and connections are made based on preliminary diagrams. Through this meticulous process, the envisioned circuit takes shape and can be tested and evaluated for functionality[31,33].

The study has made significant progress by transforming Figure 11a to Figure 11b, which shows the final practical circuit. Figure 11a shows the initial data-collection process circuit involving the assembly and integration of electronic components. Figure 11b shows a refined circuit specifically designed for older people, ensuring its effectiveness in real-world scenarios. This represents a significant step forward for this study.

(a)

(b)

Figure 11: The Final hardware circuit: a) Initial data collection process circuit, and b) Proposed circuit.

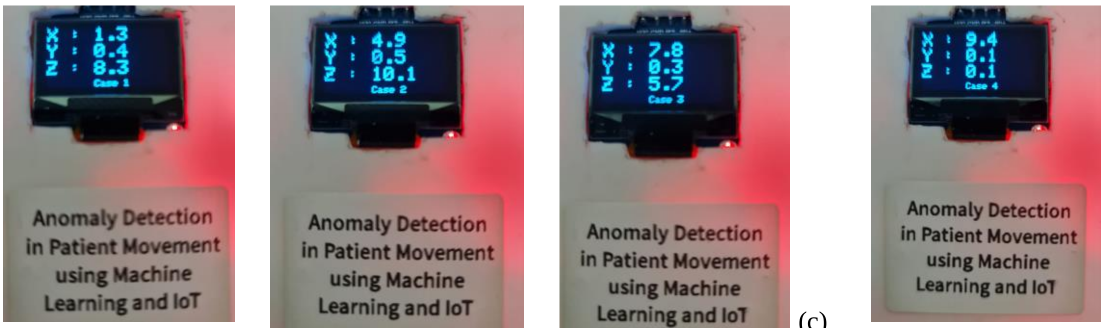
Figure 12 serves as an illustrative representation of various practical situations involving a single individual, facilitating the collection of the initial data, as outlined in Figure 1. By showing different real-life situations, the figure emphasizes the system's ability to work in a variety of settings, making it reliable and useful in many situations. Each depicted situation likely corresponds to a specific movement, activity, or condition relevant to the objectives of the study. The comprehensive portrayal in Figure 12 provides a nuanced understanding of how the designed system operates under different circumstances, offering insights into its flexibility and potential for capturing a comprehensive dataset representative of real-life scenarios. This visual representation connects the ideas shown in Figure 1 to the little things that happen in real life, which makes the study's approach to gathering and analyzing data more complete.

Figure 13a provides an insightful snapshot of the monitoring system at the initiation of its operation, capturing a set of initial readings immediately following its activation. This representation acts as a fundamental point of reference for understanding the initial behavior of the system. The MPU6050 accelerometer sensor data play a crucial role in identifying the subject's motion, as further elucidated in Figures 13b–13c. Simultaneously, the outcomes of the Naive Bayes algorithm-based machine-learning error analysis are displayed. This analysis stage is essential to evaluating the system's precision and effectiveness in categorizing the collected sensor data. Notably, Figure 13d draws attention to a troubling detail namely, that a person's situation is not ideal. This instantaneous insight into a person's posture or position highlights the system's ability to gather data as well as provide quick analysis and input on pertinent factors. These numbers are presented sequentially to give a thorough overview of the monitoring system's performance and to provide important insights for future improvement and optimization.

(a) (b)

(c) (d)

Figure 12: There are many locations for gathering main data: case 1, case 2, case 3, and case 4.



(a) (b) (c) (d)

Figure 13: The final hardware monitoring circuit that produced various outcomes Cases 1, 2, 3, and 4 are listed in alphabetical order.

In the context of Internet of Things (IoT) systems, Thing Speak is a useful tool for proof-of-concept and prototype development, particularly where analytics are an essential component. The cloud-based IoT analytics platform ThingSpeak was very important for the study because it allowed us to see real-time data streams and judge the overall health of the finished system, as shown in Figure 12. Figure 14 shows the outcomes of this evaluation, showing the insights and analytics derived from the system. The capability of ThingSpeak to display data in real-time directly from the main circuit provides a dynamic and immediate overview of the system's performance.
Moreover, a specialized software application designed explicitly for this purpose extends the accessibility of the findings to Android smartphones. As shown in Figure 15, this software allows displaying similar results on Android devices, so users can easily access the system’s data and results. This mobile phone connectivity provides for its usage “on the fly” to track changes within an application or for making some improvements while users are on the go. Using ThingSpeak together with other applications not only validates whether such a completed system may work but also illustrates how easily it could be incorporated into the daily lives of people and applied practically.

Figure 14: System results screen for the ThingSpeak website: a) Values of the three axes, b) Case number

Figure 15: Smartphone windows results: a) Three Axes values, and b) Case No.

The paper showed how senior-level circuit design went from the simplest path of gathering information about something to a fully operating device. This program is used as a tool for IoT development due to its ability to provide real-time streams for detailed analysis. The customized Android smartphone software enhances the accessibility of this system since real-time tracking and analysis can be done when users are outside the office. This combination emphasizes its robustness as well as flexibility while hinting at its potential use in real-world settings.

VI. CONCLUSIONS

As a result, we have witnessed a tremendous breakthrough in medical technology; thanks to an integrated approach involving Machine Learning (ML), as well as Internet of Things (IoT), that has led to development and implementation of a system for anomaly identification in patient movements. To ensure proper and timely detection of abnormalities associated with patients' movements during their illness periods, mobile sensors are utilized by this platform for gathering patients' real-time motion data and accurate machine learning algorithms (such as Naïve Bayes classifier). This means that any inability of the patient to move freely will be quickly identified and correctly diagnosed.

From a software perspective all the way to hardware integration, implementation is quite simple yet still effective making it friendly in terms of healthcare system adoption. The architecture of this system was designed such that it linked the sensors via which all information was exchanged as well as oversaw activity on the ThingSpeak cloud to ensure the security of data. An Android smartphone application was specifically designed for this study to monitor the movements of older people.

The efficacy and dependability of this system have been proven through extensive testing and validation by medical specialists. Ultimately, this creative approach improves patient safety, offers helpful assistance to medical staff, and increases the standard of care in healthcare institutions. This system raises the bar for patient monitoring by using the power of the IoT and ML, opening the door for future developments in patient-centered and more sophisticated healthcare solutions.

REFERENCES

[1] Petre, Ion, Flavia Barna, Daniela Gurgus, Laurentiu Cezar Tomescu, Adrian Apostol, Izabella Petre, Cristian Furau, Miruna Lucia Năchescu, and Anca Bordianu. "Analysis of the Healthcare System in Romania: A Brief Review." In Healthcare, vol. 11, no. 14, p. 2069. MDPI, 2023.

[2] Dahleez, Khalid Abed, Imad Bader, and Mohammed Aboramadan. "E-health system characteristics, medical performance and healthcare quality at UNRWA-Palestine health centers." Journal of Enterprise Information Management 34, no. 4 (2021): 1004-1036.

[3] Valentin, Gitte, Claus Vinther Nielsen, Anne-Sofie Meldgaard Nielsen, Merete Tonnesen, Kristina Louise Bliksted, Katrine Tranberg Jensen, Karen Ingerslev, Thomas Maribo, and Lisa Gregersen Oestergaard. "Bridging Inequity Gaps in Healthcare Systems While Educating Future Healthcare Professionals—The Social Health Bridge-Building Programme." International Journal of Environmental Research and Public Health 20, no. 19 (2023): 6837.

[4] Pramesh, C. S., D. V. R. Seshadri, Evita Fernandez, Gullapalli N. Rao, Manisha Dutta, Pavitra Mohan, and Equitable Healthcare Access Consortium. "Healthcare in post-COVID India: a call for a decentralized healthcare system." Journal of Family Medicine and Primary Care 10, no. 12 (2021): 4337.

[5] Khan, Umair Ali, and Ari Alamäki. "Designing an Ethical and Secure Pain Estimation System Using AI Sandbox for Contactless Healthcare." International Journal of Online & Biomedical Engineering 19, no. 15 (2023).

[6] Kotenko, Igor, Igor Saenko, and Alexander Branitskiy. "Framework for mobile Internet of Things security monitoring based on big data processing and machine learning." IEEE Access 6 (2018): 72714-72723.

[7] Kim, Sungwon, Meysam Alizamir, Youngmin Seo, Salim Heddam, Il-Moon Chung, Young-Oh Kim, Ozgur Kisi, and Vijay P. Singh. "Estimating the incubated river water quality indicator based on machine learning and deep learning paradigms: BOD5 Prediction." Mathematical Biosciences and Engineering 19, no. 12 (2022): 12744-12773.

[8] Ali, Aitizaz, Hashim Ali, Aamir Saeed, Aftab Ahmed Khan, Ting Tin Tin, Muhammad Assam, Yazeed Yasin Ghadi, and Heba G. Mohamed. "Blockchain-Powered Healthcare Systems: Enhancing Scalability and Security with Hybrid Deep Learning." Sensors 23, no. 18 (2023): 7740.

[9] Jaffery, Mujtaba Hussain, Muhammad Adil Ashraf, Ahmad Almogren, Hafiza Mahnoor Asim, Jehangir Arshad, Javed Khan, Ateeq Ur Rehman, and Seada Hussen. "FSR-based smart system for detection of wheelchair sitting postures using machine learning algorithms and Techniques." Journal of Sensors 2022 (2022): 1-10.

[10] Yadav, Sangeeta, Preeti Gulia, Nasib Singh Gill, and Jyotir Moy Chatterjee. "A real-time crowd monitoring and management system for social distance classification and healthcare using deep learning." Journal of Healthcare Engineering 2022 (2022).

[11] Said, Abdel Mlak, Aymen Yahyaoui, and Takoua Abdellatif. "Efficient anomaly detection for smart hospital IoT systems." Sensors 21, no. 4 (2021): 1026.

[12] M. R. Islam, M. M. Kabir, M. F. Mridha, S. Alfarhood, M. Safran, and D. Che, "Deep Learning-Based IoT System for Remote Monitoring and Early Detection of Health Issues in Real-Time," Sensors, vol. 23, no. 11, pp. 1–18, 2023, doi: 10.3390/s23115204.

[13] S. Ayouni, M. Maddeh, S. Al-Otaibi, M. B. Alazzam, N. Alturki, and F. Hajjej, "Development of a Smart Hospital Bed Based on Deep Learning to Monitor Patient Conditions," J. Disabil. Res., vol. 2, no. 2, pp. 25–36, 2023, doi: 10.57197/jdr-2023-0017.

[14] M. Karmakar, B. Choudhury, R. Patowary, and A. Nag, "An IoT-Based Deep Learning Framework for Real-Time Detection of COVID-19 through Chest X-ray Images," Computers, vol. 12, no. 1, 2023, doi: 10.3390/computers12010008.

[15] Mohammed, B.G. and Hasan, D.S., 2023. Smart Healthcare Monitoring System Using IoT. International Journal of Interactive Mobile Technologies (iJIM), 17(01), pp.141-152.

[16] J. Heaney, J. Buick, M. U. Hadi, and N. Soin, "Internet of Things-Based ECG and Vitals Healthcare Monitoring System," Micromachines, vol. 13, no. 12, 2022, doi: 10.3390/mi13122153.

[17] Saleh, S., Cherradi, B., El Gannour, O., Gouiza, N. and Bouattane, O., 2023. Healthcare monitoring system for automatic database management using mobile application in IoT environment. Bulletin of Electrical Engineering and Informatics, 12(2), pp.1055-1068.

[18] Islam, Md Milon, Ashikur Rahaman, and Md Rashedul Islam. "Development of smart healthcare monitoring system in IoT environment." SN computer science 1 (2020): 1-11.

[19] Espressif Systems. (2020). ESP32 Datasheet. Available online: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf . (Accessed on February 2024).

[20] MPU6050 Gyroscope. Available online: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf . (Accessed on February 2024).

[21] The TP4056 Li-Ion Battery Charging Module. Available online: https://www.handsontec.com/dataspecs/module/18650-Lithium%20charger.pdf. (Accessed on February 2024).

[22] OLED-128O064D-BPP3N00000. Available online: https://www.vishay.com/docs/37902/oled128o064dbpp3n00000.pdf. (Accessed on February 2024).

[23] Micro SD Card Reader Module. Available online: http://www.supertalent.com/datasheets/5_112.pdf . (Accessed on February 2024).

[24] Python 3.10 software. Available online: https://www.python.org/downloads/release/python-3100/ . (Accessed on February 2024).

[25] Arduino IDE. Available online: https://www.arduino.cc/en/software. (Accessed on February 2024).

[26] ThingSpeak for IoT Projects. Available online: https://thingspeak.com/. (Accessed on February 2024).

[27] MIT App Inventor. Available online: http://appinventor.mit.edu/ . (Accessed on February 2024).

[28] Saritas, Mucahid Mustafa, and Ali Yasar,2019, "Performance analysis of ANN and Naive Bayes classification algorithm for data classification." International journal of intelligent systems and applications in engineering 7, no. 2: 88-91.

[29] Ren, Jiangtao, Sau Dan Lee, Xianlu Chen, Ben Kao, Reynold Cheng, and David Cheung, 2009, "Naive bayes classification of uncertain data." In 2009 Ninth IEEE international conference on data mining, pp. 944-949. IEEE.

[30] Fritzing is Available online: https://fritzing.org/ . (Accessed on February 2024).

[31] Ismaeel, N.Q., Mohammed, H.J., Chaloob, I.Z. et al. Application of Healthcare Management Technologies for COVID-19 Pandemic Using Internet of Things and Machine Learning Algorithms. Wireless Pers Commun (2023). https://doi.org/10.1007/s11277-023-10663-2

[32] Alhayani, B.A., AlKawak, O.A., Mahajan, H.B. et al. Design of Quantum Communication Protocols in Quantum Cryptography. Wireless Pers Commun (2023). https://doi.org/10.1007/s11277-023-10587-x

[33] Omar A. AlKawak, Bilal A. Ozturk, Zinah S. Jabbar, Husam Jasim Mohammed,Quantum optics in visual sensors and adaptive optics by quantum vacillations of laser beams wave propagation apply in data mining,Optik,Vol,273,2023,170396,https://doi.org/10.1016/j.ijleo.2022.170396.