Smart Health Monitoring System for Home Applications

H H Sayed[1], B István[2]

[1] Mechatronics Engineering BSc, University of Debrecen, Debrecen, Hungary

[2] Associate Professor, Department of Mechatronics, Faculty of Engineering, University of Debrecen, Debrecen, Hungary

Abstract. This paper presents the development of a health monitoring system prototype that integrates five essential sensors—heart rate, pulse oximetry, temperature, galvanic skin response (GSR), and blood glucose—using a Raspberry Pi 5 microcontroller, a mobile application, and machine learning algorithms. The system is designed to address the rising demand for accessible, real-time health monitoring, particularly for individuals with chronic conditions. The prevalence of chronic diseases and the high cost of frequent hospital visits underscore the need for affordable alternatives, making this system an efficient and user-friendly solution for continuous health monitoring. The Raspberry Pi 5, with its powerful quad-core processing capabilities and seamless integration with multiple sensors, acts as the core of the system, facilitating real-time data collection and transmission. Sensor data is visualized and analyzed through a mobile application, offering users immediate insights into their health status. The system employs MATLAB's machine learning algorithms to detect abnormal sensor readings, providing accurate feedback on potential health risks and allowing users to take proactive steps in managing their conditions. Stress management is a key feature of this system, as the GSR sensor measures skin conductivity to assess stress levels, contributing to a deeper analysis of the relationship between stress and health. Although the project faced challenges in automating blood glucose monitoring, manual input is currently used until automation is fully implemented. Future improvements will focus on refining machine learning models and expanding the mobile application's interactive features to enhance user experience. This project demonstrates the viability of developing a low-cost, real-time health monitoring solution that integrates cutting-edge technologies, with the potential to significantly improve chronic disease management and overall healthcare accessibility for users around the world.

1. Introduction

The rapid advancement of technology has paved the way for innovative health monitoring solutions. According to the Australian Institute of Health and Welfare, chronic conditions have become more common [1]. The increase in this phenomenon necessitates affordable and user-friendly health monitoring systems. Although traditional health systems have proven to be more reliable, frequent visits to a clinic or hospital are both time-consuming and costly. Considering the study conducted by the World Health Organization in 2021, it has been proven that over 4.5 billion people are not fully covered by insurance. Other solutions are required to improve human health [2]. A promising solution is the integration of sensors with mobile applications. Mobile applications provide real-time health data monitoring and analysis. This project is particularly relevant to mechatronics engineering, which combines mechanical, electronic, computer, and control engineering to create intelligent systems.

1.1. What: This paper presents a prototype health monitoring system. Four sensors—heart rate, pulse oximetry, temperature, galvanic skin response (GSR), and blood glucose— were chosen due to the various correlations between their parameters. The sensor data is integrated with a Raspberry Pi 5 controller and a mobile application.

1.2. Why: With the rise of chronic diseases and the overpriced insurance system, there is a growing need for affordable and user-friendly health monitoring systems. Frequent hospital visits are time-consuming and costly. A real-time monitoring system can provide continuous health data, improving the quality of life for individuals with chronic conditions. This system also stresses the importance of stress management by integrating a stress sensor. This aims to enhance the analysis of the relationship between stress and conditions.

1.3. How: The system collects sensor data through a Raspberry Pi 5 controller and transmits it to a mobile application. The app visualizes and analyzes the data, offering insights and alerts based on predefined health parameters.

2. Methods

The health monitoring system is designed to collect real-time data using a Raspberry Pi 5 microcontroller, which connects various sensors to measure heart rate, oxygen levels, temperature, skin conductance, and blood sugar. Data analysis and anomaly detection are conducted through MATLAB's machine learning capabilities, enhancing the system's ability to provide meaningful insights. Below, each component of the system is detailed, including hardware integration and data processing methods.

2.1. System Structure

In this system design, the central processing unit (CPU) acts as the core component responsible for gathering data from various sensors, including those for heart rate, temperature, and other vital signs. The sensors are connected to the CPU, which processes the incoming data in real-time as displayed in figure 1. After initial processing, the data is transmitted to a machine learning model implemented in MATLAB, where it undergoes further analysis to detect anomalies or abnormal values across the sensors. The results of the machine learning model are fed back into the system, providing real-time insights and user feedback via the Human-Machine Interface (HMI). Additionally, the CPU is connected to the Internet of Things (IoT) network, enabling remote access, data logging, and further analysis through cloud-based services. This connection between components, machine learning, and interfaces ensures efficient data collection, processing, and intelligent monitoring capabilities.

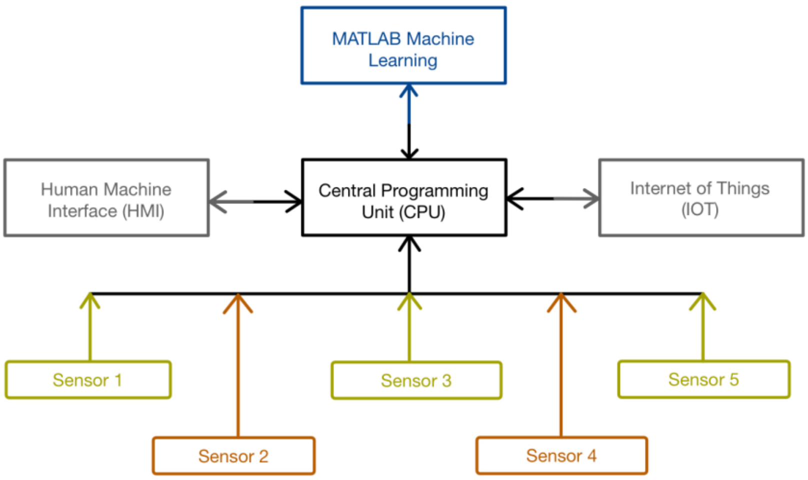

Figure 1: System Structure

2.2. Raspberry Pi 5 and Sensors Used

The Raspberry Pi 5 was chosen to serve as the core of the health monitoring system. It features a quad-core ARM Cortex-A76 CPU that runs 2.4 GHz and LPDDR4X RAM (4GB or 8GB) [3]. It also features various connectivity options, including Wi-Fi 6 and Bluetooth 5.2, making it ideal for real-time health monitoring and IoT applications. Its GPIO pins enable easy integration with external sensors. The Raspberry Pi 5 is also highly capable of handling data from multiple sensors, performing machine learning tasks, and running real-time analytics [4].

2.2.1. Heart Rate Sensor

The sensor chosen to measure heart rate is the Pulse Sensor Amped. Pulse Sensor Amped utilizes photoplethysmography (PPG) measuring heart rate by detecting blood volume changes in the skin [5]. It does so by using an LED that shines light onto the skin and a photodetector that measures the reflected light. This optical signal varies with the pulsing of blood, enabling heart rate detection [6]. The analog signal from the sensor is converted to digital using an MCP3008 ADC for processing by the Raspberry Pi.

2.2.2. Pulse Oximetry

The sensor chosen to measure the blood oxygen saturation (SpO2) is MAX30102. It provides real-time monitoring of oxygen levels, which is critical for detecting respiratory issues. The MAX30102 also measures heart rate offering a second method to recheck the heart rate detected by the Pulse Sensor Amped. It also uses PPG technology, employing both red and infrared LEDs to measure the oxygen levels in the blood [7].

2.2.3. Temperature Sensor

The sensor chosen to measure and monitor the body's temperature is the DS18B20 digital temperature sensor. It measures human temperature using a digital thermometer that operates based on the 1-Wire protocol [8]. It has an internal sensor that detects temperature changes through its metal surface, converting this data into a digital signal. The sensor measures temperature by detecting the resistance change in its semiconductor components, which varies with heat [9]. It provides highly accurate readings, typically within ±0.5°C and can be easily integrated with the Raspberry Pi through its 1-Wire protocol.

2.2.4. Galvanic Skin Response (GSR)

The Galvanic Skin Response (GSR) sensor is used to measure changes in skin conductivity, which correlates with emotional arousal and stress levels [10]. This sensor helps analyze stress and anxiety by detecting variations in sweat gland activity, a physiological response to stress, making it an important addition to the system's overall health monitoring capabilities.

2.2.5. Blood Glucose Sensor

Due to the rise of diabetic patients around the world and across all ages, monitoring the blood glucose levels is important. Although the project aims to incorporate a blood glucose sensor, it is currently experiencing challenges. The system is being designed to automate this process in the future, but for now, manual input is required to track glucose levels, limiting its functionality compared to the other sensors.

2.3. MATLAB Machine Learning Model

In my project, I developed a machine learning model using MATLAB to analyze sensor data from a health monitoring system. The goal was to detect abnormal readings across several health metrics, including heart rate, SpO2 (blood oxygen saturation), body temperature, galvanic skin response (GSR), and blood sugar levels. The process began by loading the sensor data from a CSV file and defining normal ranges for each metric. I then initialized a new column in the dataset to label readings as either normal or abnormal based on these predefined thresholds. After labeling the data, I split it into features (the sensor readings) and labels (normal/abnormal) for analysis. For model training, I divided the dataset into training and testing sets, allocating 80% of the data for training and 20% for testing. I selected a decision tree model for this task due to its interpretability and suitability for structured data. The decision tree provides clear decision rules, making it easy to understand how different sensor readings contribute to the model's predictions as displayed in figure 2. Once trained, the model was evaluated using the test data, achieving a high accuracy in classifying the readings. Additionally, I implemented a feature to predict labels for new sensor readings, which allows for real-time monitoring. The model not only indicates whether the readings are normal or abnormal but also identifies which specific sensors might be causing issues, facilitating targeted interventions as shown in figure 3.

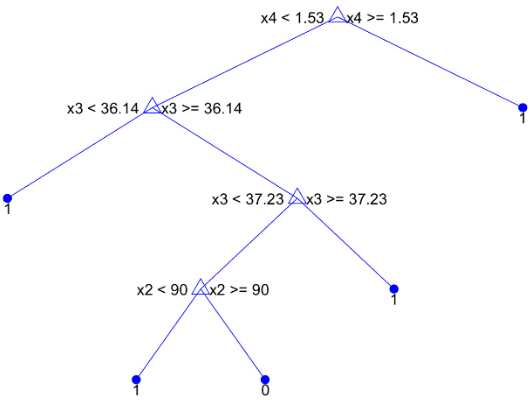

Figure 2: Decision Tree View

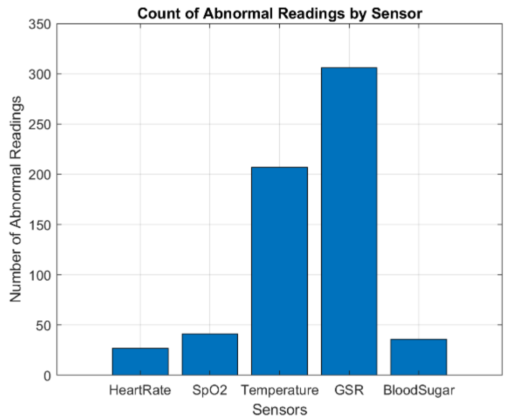

Figure 3: Abnormal Sensor Readings Bar Chart

2.4. Challenges

During the development of the machine learning model, I faced several challenges, especially while experimenting with different algorithms. One notable model I tested was the Support Vector Machine (SVM). Despite its theoretical advantages, the SVM consistently produced low accuracy, hovering around 50%, which felt more like a guessing game than a reliable predictive system. Consequently, I opted to transition to a decision tree model, which offered better interpretability and performance. I soon identified a significant factor contributing to the model's initial challenges: limitations within the dataset. Many sensor readings were either unlabelled or incorrectly categorized, undermining the model's training. To remedy this, I established predefined thresholds for each health metric to define a baseline for normal and abnormal readings. This initial labelling was crucial for teaching the model the acceptable ranges for each sensor, ultimately providing clearer distinctions for future predictions.

3. Results

The decision tree model demonstrated significant improvements over earlier attempts using the Support Vector Machine (SVM), which struggled with low accuracy. Notably, the decision tree achieved an accuracy of over 99% on the test dataset and approximately 97% on new data. The effectiveness of the model's predictions was validated through a confusion matrix, clearly illustrating its capability to classify normal and abnormal readings. Additionally, the model's ability to identify specific sensors contributing to abnormal readings provides actionable insights, allowing healthcare providers to target their interventions more effectively. To ensure the model's reliability, data analysis was conducted to prevent overfitting. The decision tree not only excels in classifying readings but also pinpointing the sensors responsible for abnormal values, thereby enabling targeted healthcare interventions. The strong performance of the model underscores its potential for real-time health monitoring applications. Future enhancements may include exploring ensemble methods and expanding the dataset to further improve accuracy and robustness.

4. Discussion and Conclusion

This study successfully developed a machine learning model using a decision tree algorithm to analyze sensor data from a health monitoring system. By addressing the challenges posed by data quality and establishing clear thresholds for normal and abnormal readings, the model demonstrates a significant improvement in predictive accuracy over initial attempts with the SVM.

Furthermore, the implementation of the decision tree algorithm allowed for greater interpretability of the results. This transparency is critical in healthcare applications, where understanding the rationale behind predictions can enhance trust among users and healthcare providers. This work contributes to the growing field of health monitoring technology, offering valuable insights into the application of machine learning in detecting potential health issues early. The integration of multiple sensors in the monitoring system provides a comprehensive overview of an individual's health status. This multi-faceted approach is vital for chronic disease management, where monitoring various parameters can lead to timely interventions and better patient outcomes. Continued research in this domain is essential for refining these systems and enhancing their utility in real-world healthcare settings.

In future iterations of this project, I plan to investigate the incorporation of additional machine learning techniques, such as ensemble methods, which could further improve prediction accuracy. Moreover, I aim to explore the integration of user feedback mechanisms within the mobile application, allowing users to report their symptoms and experiences directly. This could provide valuable context for the machine learning model, ultimately improving its predictive capabilities. Moreover, exploring the user interface of the mobile application is crucial for ensuring that users can easily understand their health data and the implications of any abnormal readings. The usability of such applications can significantly impact user engagement and adherence to health monitoring recommendations. A user-friendly design will empower users to take proactive steps toward managing their health.

References:

[1] Australian Institute of Health and Welfare, "Chronic Disease Overview," *Australian Institute of Health and Welfare,* Aug. 15, 2023. Available: https://www.aihw.gov.au/reports-data/health-conditions-disability-deaths/chronic-disease/overview#:~:text=Chronic%20diseases%20are%20long%20lasting. [Accessed: Jul. 22, 2024]

[2] World Health Organization, "Universal Health Coverage (UHC)," www.who.int, Oct. 05, 2023. Available: https://www.who.int/news-room/fact-sheets/detail/universal-health-coverage-(uhc)#:~:text=This%20indicates%20that%20in%202021. [Accessed: Aug. 01, 2024]

[3] E. Upton, "Introducing: Raspberry Pi 5!," Raspberry Pi, Sep. 28, 2023. Available: https://www.raspberrypi.com/news/introducing-raspberry-pi-5/. [Accessed: Sep. 27, 2024]

[4] S. Collins, "The Life of Pi: Ten Years of Raspberry Pi," University of Cambridge, Feb. 25, 2022. Available: https://www.cam.ac.uk/stories/raspberrypi. [Accessed: Sep. 27, 2024]

[5] PulseSensor, "PulseSensor Playground Toolbox," World Famous Electronics llc., 2024. Available: https://pulsesensor.com/pages/pulsesensor-playground-toolbox#:~:text=The%20depiction%20of%20the%20pulse. [Accessed: Sep. 16, 2024]

[6] Curious Cyborg, "What is Photoplethysmography PPG? - Curious Cyborg," Curious Cyborg -, Apr. 27, 2022. Available: https://curiouscyborg.com/photoplethysmography-ppg/#:~:text=Photoplethysmography%20PPG%2C%20pronounced%20%E2%80%9CFOTO%2D. [Accessed: Sep. 16, 2024]

[7] A. Babalola and Ubochi, "The Performance of the STM32 Microcontroller and MAX30102 for Remote Health Monitoring Device Design," *ResearchGate,* **Vol. 10**, No. 03, Aug. 2022, doi: https://doi.org/10.22624/AIMS/DIGITAL/V10N1P4. Available: https://www.researchgate.net/publication/366464801_The_Performance_of_the_STM32_Microcontroller_and_MAX30102_for_Remote_Health_Monitoring_Device_Design. [Accessed: Oct. 10, 2024]

[8] M. Fezari and A. Al Dahoud, "Exploring One-wire Temperature sensor 'DS18B20' with Microcontrollers," *ResearchGate,* Feb. 04, 2019. Available:

https://www.researchgate.net/publication/330854061_Exploring_One-wire_Temperature_sensor_DS18B20_with_Microcontrollers. [Accessed: Oct. 10, 2024]
[9] Cecelia, “DS18B20 Temperature Sensor: Circuit, Pinout, and Datasheet | Easybom,” Easybom.com, Feb. 25, 2022. Available: https://www.easybom.com/blog/a/ds18b20-temperature-sensor-circuit-pinout-and-datasheet. [Accessed: Oct. 10, 2024]
[10] Seraphina, “Grove - GSR Sensor - Seeed Wiki,” wiki.seeedstudio.com, Jan. 06, 2023. Available: https://wiki.seeedstudio.com/Grove-GSR_Sensor/. [Accessed: Aug. 05, 2024]