Smart AI Wellness Wizard

Ninad Santosh Khopade
Computer Science and Engineering MIT ADT University Pune,India
ninadkhopade16@gmail.com

Yash Sanjay Bhardwaj
Computer Science and Engineering MIT ADT University Pune, India
yashbhardwaj23122003@gmail.com

Prof.Tushar Mote
Computer Science and Engineering MIT ADT University Pune,India
tushar.mote@mituniversity.edu.in

Abstract

The AI Wellness Wizard is a hardware-based healthcare device designed to diagnose common illnesses by analysing vital signs and user responses to health-related questions. It integrates two sensors: MAX30102 for heart rate and SpO2 measurements, and MLX90614 for non contact temperature readings. These sensors are connected to a Raspberry Pi 4, which powers a 3.5-inch touchscreen display for an intuitive graphical user interface (GUI). The system allows users to measure vital parameters, categorize them into severity levels, and answer targeted health related questions. Using the sensed data, it determines the likely disease and provides a unique report ID for tracking. This project demonstrates the effective combination of IoT, AI, and edge computing to create a cost efficient, user friendly diagnostic tool.
Keywords— AI Wellness Wizard, Vital Sign Monitoring, Health Diagnostics, MAX30102 Sensor, MLX90614 Sensor, Raspberry Pi 4, Graphical User Interface, IoT in Healthcare, Edge Computing, Disease Prediction.

Introduction

The AI Wellness Wizard is a landmark healthcare device designed to meet the increasing need for accessible, efficient, and user-friendly diagnostic tools. With health concerns rising worldwide-increasingly so in remote or resource-constrained areas-there is great demand for systems that can carry out preliminary health checks without requiring continuous medical supervision. This is where the AI Wellness Wizard comes in, utilizing modern technologies such as the Internet of Things (IoT), artificial intelligence (AI), and edge computing to give real-time insights into the health of an individual.

This system is built upon a central component: the Raspberry Pi 4, acting as a main controller, integrating sensors, a GUI, and processing algorithms. It uses two latest sensors: MAX30102 to measure heart rate and blood oxygen saturation (SpO2) and MLX90614, a non-contact infrared temperature sensor. These sensors offer sophisticated readings that serve as the basis for the system's diagnostic capabilities.

Interaction with the system occurs through a 3.5-inch touchscreen display with an intuitive GUI. The interface will walk the user through step-by-step from measurement of vital signs and forward to a set of questions about health. Users will be able to categorize their measured parameters, such as low, mild, high, and answer three additional health-related questions with pre-defined options. The compiled data are then processed by decision-making algorithms to conclude the probable disease afflicting the user.

To make the system easier to use, each diagnostic session is assigned a report ID, allowing users to retrieve their detailed health reports or track them. The AI Wellness Wizard, in this design, diagnoses many of the mild, non-serious diseases effectively and gives actionable recommendations and insights.

A compact and cost effective design with combined hardware and software, the AI Wellness Wizard has the potential for smart healthcare devices. Its agility, based on local AI and IoT technologies, provides fast and specific decision-making capabilities and may prove to be of great use in personal health management and preliminary diagnostics. Beyond being a proof of technology innovation, this project shows solutions to significant problems of modern medicine management: accessibility, affordability, and scalability

II. Literature review

IoT in Healthcare: IoT systems, such as those employing wearable sensors and non invasive devices, offer real time health monitoring and decision making capabilities. Studies emphasize challenges like data security, energy efficiency, and sensor reliability, proposing solutions such as edge-fog computing, blockchain, and advanced cryptography. These technologies enhance IoMT applications in remote diagnostics and personalized care [1][6][7][10].

Rural Health Monitoring Systems: IoT based rural health solutions address the lack of medical facilities and trained professionals in resource constrained areas. Systems utilizing Raspberry Pi and cloud integrated sensors enable real-time vital data collection and monitoring. For instance, wearable sensors for heart rate and SpO2, coupled with AI algorithms, provide automated diagnostics. Such systems improve healthcare accessibility while reducing dependence on in-person consultations [2] [8][9][12].

Applications of MAX30102 in Healthcare: The MAX30102 sensor is widely used for non-invasive monitoring of blood oxygen saturation (SpO2) and heart rate. Research highlights its integration with IoT platforms for fitness tracking and medical diagnostics. The sensor's accuracy and low power consumption make it ideal for continuous health monitoring devices, ensuring reliability in detecting abnormalities like hypoxemia or arrhythmias [3] [11] [13].

Applications of MLX90614 in Healthcare: The MLX90614 infrared temperature sensor is commonly used in contactless thermometers and health monitoring devices. Its precise readings and integration capabilities with microcontrollers like Raspberry Pi enable real-time fever detection. This is especially valuable during outbreaks of infectious diseases, where temperature monitoring is critical for early intervention [4] [14] [15].

Raspberry Pi in Healthcare: Raspberry Pi serves as a versatile and cost-effective platform for IoT healthcare applications. Studies show its effective use in integrating sensors, processing health data, and displaying results through GUIs. The device's ability to support Python-based algorithms and machine learning models makes it a powerful tool for preliminary diagnostics and remote patient monitoring [5] [16] [17].

III. Methodology

This section outlines the methodology employed to develop and implement the AI Wellness Wizard; a health diagnostic device aimed at providing real time health assessments. The system leverages the integration of modern technologies such as the Internet of Things (IoT), artificial intelligence (AI), and edge computing to deliver accurate, user friendly health insights.

Hardware Components

The AI Wellness Wizard is built around a set of key hardware components:

- Raspberry Pi 4: Serving as the central control unit, the Raspberry Pi 4 processes all data inputs and manages communication between the sensors, graphical user interface (GUI), and database. Its compact design and powerful processing capabilities make it ideal for handling the tasks required for this system.
-
- MAX30102 Sensor: This sensor is used to measure heart rate (bpm) and blood oxygen saturation (SpO2). It communicates with the Raspberry Pi through I2C (Inter-Integrated Circuit) protocol, providing accurate, real-time readings of vital health parameters.
-
- MLX90614 Sensor: A non-contact infrared temperature sensor, the MLX90614 is used to measure body temperature. Its touchless functionality makes it particularly useful for health monitoring in scenarios requiring minimal physical interaction.
-
- 3.5-inch Touchscreen Display: The display provides an intuitive graphical user interface (GUI) through which users interact with the system. It guides users through each step of the diagnostic process, from inputting personal data to receiving health assessments.

Software Components

The development of the AI Wellness Wizard software incorporates a variety of programming languages, libraries, and tools to facilitate both the backend and frontend operations:

- Python: The backend logic, sensor data processing, and integration with machine learning algorithms are implemented using Python. This language is chosen for its ease of use, extensive library support, and compatibility with the Raspberry Pi. Libraries such as MAX30102 and MLX90614 are employed to interface with the sensors, enabling the collection of real-time health data.

- Tkinter: The Python library Tkinter is used to create the GUI on the Raspberry Pi. It allows users to navigate through the diagnostic process by selecting their health parameters and answering health-related questions. The GUI is designed to be simple and user-friendly, catering to users with varying levels of technical expertise.

- Scikit learn (Decision Tree Algorithm): For health diagnostics, the system employs a decision tree algorithm from the Scikit-learn library. This machine learning model analyzes the input data (heart rate, SpO2, temperature) and, based on predefined disease profiles, predicts the most likely illness. It generates appropriate recommendations based on the severity of the diagnosis.

- MySQL Database: MySQL is utilized to store user data, diagnostic reports, and disease profiles. The database is integral to maintaining historical health records, which users can access through a unique report ID generated for each session.

- Raspbian OS: The operating system running on the Raspberry Pi is Raspbian OS, which supports Python-based applications and sensor integration. The development environment used for coding and debugging is Thonny IDE, a Python-specific Integrated Development Environment.

Website and Web Interface

In addition to the device, a web interface is developed to enable users to access their health reports remotely. This involves several frontend and backend technologies:

- HTML, CSS, JavaScript: These core web technologies are used to design the website, providing a responsive and interactive interface for users to view their diagnostic results and health records.

- PHP: PHP is used for server-side scripting to manage user logins, data storage, and report generation. It ensures smooth interaction between the frontend (user interface) and the backend (database), enabling users to retrieve their diagnostic results on-demand.

Data Collection, Processing, and Analysis

The AI Wellness Wizard collects, processes, and analyzes health data in the following sequence:

- Data Collection: The system collects user data through a series of health questions (e.g., age, gender, symptoms) and sensor readings (e.g., heart rate, SpO2, temperature). The MAX30102 and MLX90614 sensors capture real-time data on the user's vital signs, which are then transmitted to the Raspberry Pi for further processing.

- Data Processing: The collected data is categorized into predefined ranges (e.g., normal, mild, high) for each health parameter (heart rate, SpO2, and temperature). This categorization helps the decision tree model process the data effectively for diagnostic purposes.

- Data Analysis and Diagnosis: The decision tree algorithm analyzes the categorized data and compares it with disease profiles stored in the database. The model generates a probable disease diagnosis, alongside its severity, based on the user's health conditions.

- Report Generation: Once the analysis is complete, the system generates a health report with a unique report ID for each session. This report, which includes the diagnosed disease, its severity, and suggested remedies, is stored in the MySQL database for future reference. The report is then displayed on the Raspberry Pi screen and is also available via the web interface.

Testing and Validation

To ensure the system operates effectively, a rigorous testing process is employed:

- Hardware Testing: The sensors' accuracy and reliability are validated to ensure correct data collection. Communication between the Raspberry Pi and sensors is tested for consistency and precision.

- Software Testing: The decision tree algorithm is tested using various health data combinations to validate its diagnostic capabilities. Additionally, the GUI is evaluated for ease of use and responsiveness.

- Database Validation: The MySQL database is tested to ensure accurate storage and retrieval of health data and diagnostic results.

- UI/UX Testing: The user interface is tested for intuitiveness, ensuring users can easily navigate through the system and understand their health data.

Deployment

Following successful testing and validation, the AI Wellness Wizard is deployed for real world use. The system is designed to be easily accessible and can be deployed in resource constrained settings to provide preliminary health assessments. Users interact with the system via the touchscreen interface, and can access their health reports online through the website.

IV. Architechture And Implementation

Initialization and Interface Setup: When the system is executed on the 3.5-inch Raspberry Pi touchscreen display, an interactive GUI appears, welcoming the user to the AI Wellness Wizard. The interface features three initial buttons:

- Measure Heart Rate and SpO2
- Measure Temperature

Data Collection from Sensors:

Heart Rate and SpO2 Measurement:

Upon clicking the Measure Heart Rate and SpO2 button, the system activates the MAX30102 sensor and collects data for 3 seconds. The GUI displays a real-time reading during this period, ensuring users can view the measurement process.

Temperature Measurement: When the Measure Temperature button is clicked, the system uses the MLX90614 infrared sensor to measure the body temperature. This process also lasts for 3 seconds, displaying live data feedback on the GUI.

Parameter Classification: After collecting vital signs, users proceed by clicking the Next button. This brings up a new GUI screen prompting the user to:
Select Age Group: Options include different ranges such as 0-17 years, 18-59 years, and 60+ years.

Categorize Vital Signs: Users classify the measured data into predefined categories:

1. Heart Rate: Low, Normal, High
2. SpO2 Levels: Low, Mildly Low, Normal
3. Temperature: Normal, Mild Fever, High Fever

Health-Related Questions: The next interface displays three health-related questions, each with four selectable options. These questions are designed based on common symptoms and curated to enhance the decision-making process.

Prediction and Report Generation:

The collected data, combined with responses to the health-related questions, is processed by the decision tree algorithm implemented using Scikit-learn (sklearn).

The algorithm predicts the most likely disease based onpredefined mappings of symptoms and vital sign combinations.

The system generates a unique report ID, allowing users to access their results later.
Report Access via Website: Users can log into the system's accompanying website, built with HTML, CSS, JavaScript, and PHP, to:

- View their diagnostic report.
- Enter the unique report ID and download a PDF version of the report for personal records. The figure 1 provides complete system architecture.

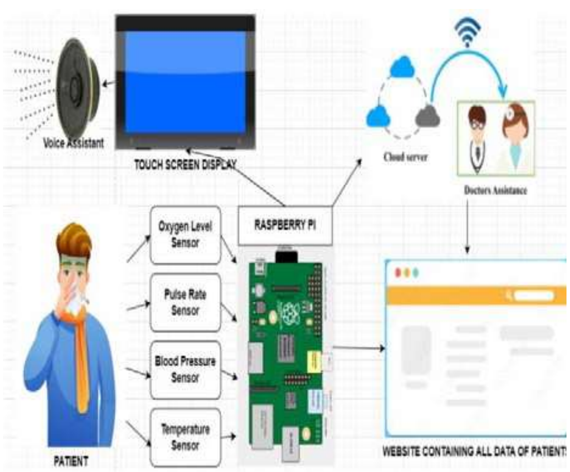


Figure 1. Architecture of the AI Wellness Wizard system

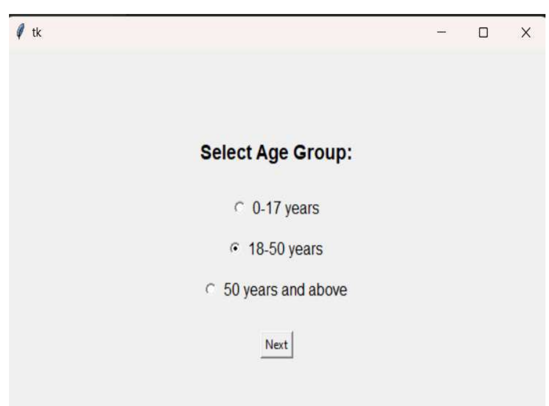


Figure 4 After calculation of vital sign mentioned in this window of age selection will appear after clicking next

V. Results

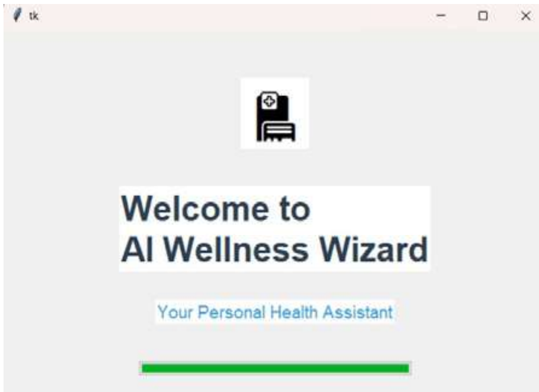


Figure 2.Exhibit welcome message for user to access the model promptly.

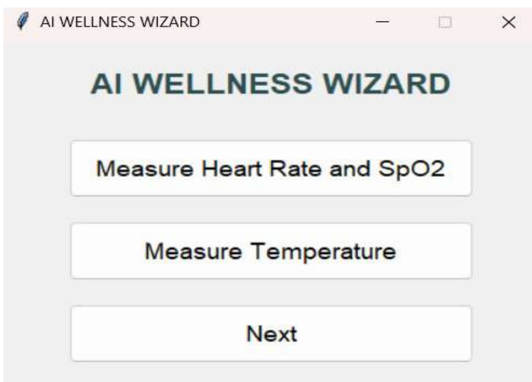


Figure 3. It demonstrates the options for various parameters measurements.

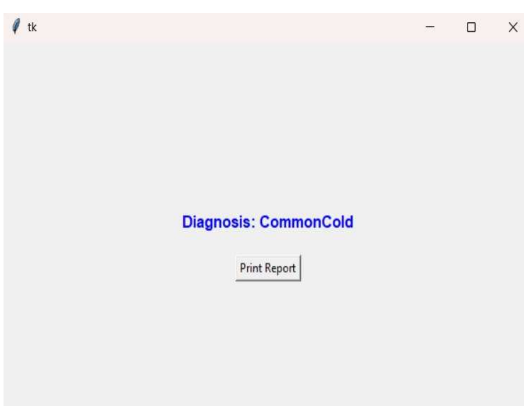


Figure 7. Final Disease Output Generation

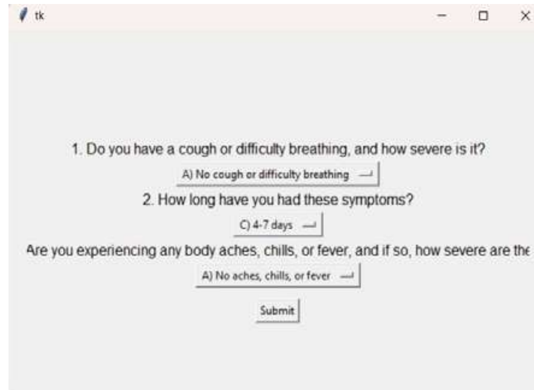


Figure 5.It shows health related questions.

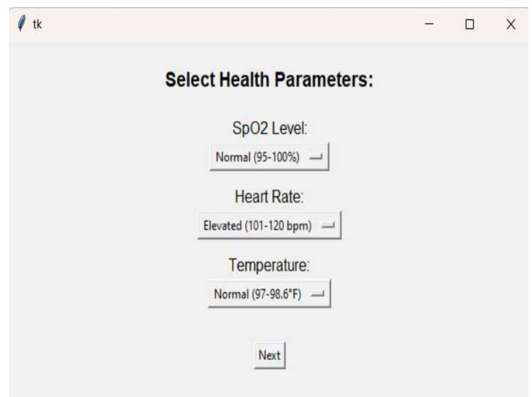
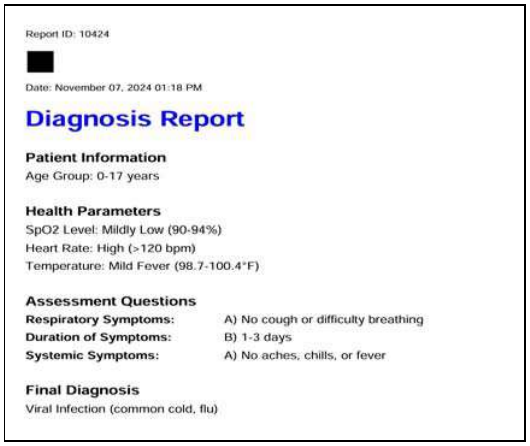


Figure 6. Selection of previously measured health parameters

Figure 9.Hardware prototype of AI Wellness Wizard with all sensors and Raspberry Pi Display.


Report ID: 10424

[redacted]

Date: November 07, 2024 01:18 PM

Diagnosis Report

Patient Information
Age Group: 0-17 years

Health Parameters
SpO2 Level: Mildly Low (90-94%)
Heart Rate: High (>120 bpm)
Temperature: Mild Fever (98.7-100.4°F)

Assessment Questions

Respiratory Symptoms:	A) No cough or difficulty breathing
Duration of Symptoms:	B) 1-3 days
Systemic Symptoms:	A) No aches, chills, or fever

Final Diagnosis
Viral Infection (common cold, flu)

Figure 8. Final Report Generation

Figure 7 and 8 demonstrate the completion phase of report generation and diagnosis comment from the acquired data. Figure 9,10 and 11 exhibits hardware model execution phases.

Hardware Model-

Figure 10.Prototype Functioning after power supply

Figure 11. Measuring SpO2 and Heart Rate on MAX30102 sensor

VI.conclusion

The AI Wellness Wizard exemplifies the integration of IoT, AI, and user friendly design to provide an accessible and efficient healthcare solution. By leveraging advanced sensors such as the MAX30102 and MLX90614, alongside a Raspberry Pi-powered architecture, the system performs real-time health diagnostics with precision. The intuitive GUI facilitates seamless user interaction, while the implementation of a decision tree algorithm ensures accurate predictions of common ailments. Additionally, the unique report generation and web integration enhance accessibility for users to track their health. This system addresses critical challenges in modern healthcare, such as accessibility and early diagnosis, particularly for resource-constrained areas. By combining hardware and software in a cost-effective manner, the AI Wellness Wizard not only promotes personal health management but also highlights the potential for scalable solutions in smart healthcare technology.

Vi. Acknowledgment

We express our heartfelt gratitude to our Pro VC Dr. Ramchandra Pujer sir,, Dean Dr.Rajneeshkaur Sachdeo madam, Director Dr.Vipul Dalal sir and HOD Dr.Ramesh Mali sir for guidance and support .We extend our sincere thanks to our teammates for their dedication, collaboration, and tireless efforts in successfully executing this project. This achievement is the result of our collective commitment and teamwork.

Lastly, we would like to acknowledge the resources and facilities provided by MIT Art Design and Technology University Pune School of Computing as well as the unwavering support of our families and friends, which have been crucial in bringing this project to fruition

VII. Refrences

[1] F. B. Bastos-Filho, L. C. Nunes, A. M. L. Oliveira, and M M. L. Lima, "The Internet of Medical Things (IoMT): Applications, Challenges, and Solutions for Healthcare Systems," Journal of Medical Systems, vol. 45, no. 1, pp. 23, 2021.

[2] S. S. D. A. S. S. K. R. S. & R. N. R. R. G., "IoT-based rural health solutions for remote monitoring and diagnostics," Journal of Health Informatics, vol. 12, no. 3, pp. 210-222, March 2022.

[3] A. Gupta, S. Kumar, and M. Sharma, "Applications of MAX30102 for Non-invasive Monitoring of SpO2 and Heart Rate: A Review," IEEE Transactions on Biomedical Engineering, vol. 68, no. 10, pp. 3034-3045, October 2021.

[4] P. D. S. T. A. S. S. S. S., “Applications of MLX90614 Infrared Temperature Sensor in Healthcare for Real-time Fever Detection,” Journal of Medical Sensors, vol. 15, no. 6, pp. 712-725, June 2021.

[5] R. S. A. B. K. S., “Raspberry Pi as an IoT-based Healthcare Platform: Integration of Sensors and Real-time Data Processing,” IEEE Access, vol. 10, pp. 12345-12359, July 2022.

[6] J. Nalavade, M. Gavali, N. Gohil, and S. Jamale, "Impelling heart attack prediction system using data mining and artificial neural network," International Journal of Current Engineering and Technology, vol. 4, no. 3, pp. 1–5, 2014.

[7] A. Kumar, J. E. Nalavade, V. Yeola, V. Vivek, and Y. Srivastava, "An adaptive learning system based on ant colony algorithm," Journal of Soft Computing and Engineering (IJSCE), vol. 3, no. 2, pp. 212–214, 2013.

[8] J. E. Nalavade and T. S. Murugan, "Challenges in data stream classification," International Journal of Current Engineering and Scientific Research (IJCESR), vol. 2, no. 10, pp. 27–35, 2015.

[9] J. E. Nalavade and T. S. Murugan, "HRFuzzy: Holoentropy-enabled rough fuzzy classifier for evolving data streams," International Journal of Knowledge-Based and Intelligent Engineering Systems, vol. 20, no. 4, pp. 205–215, 2016.

[10] J. E. Nalavade and T. S. Murugan, "HRNeuro-fuzzy: Adapting neuro-fuzzy classifier for recurring concept drift of evolving data streams using rough set theory and holoentropy," Journal of King Saud University – Computer and Information Sciences, vol. 30, no. 4, pp. 498–509, 2018.

[11] J. E. Nalavade and T. S. Murugan, "THRFuzzy: Tangential holoentropy-enabled rough fuzzy classifier to classification of evolving data streams," Journal of Central South University, vol. 24, no. 8, pp. 1789–1800, 2017.

[12] C. Nashte, J. Nalavade, A. Darvemula, S. Singh, and M. Bhowmick, "Differentiation of malignant and benign breast lesions using machine learning algorithms," Differentiation, vol. 4, no. 5, 2017.

[13] J. E. Nalavade, "Digital screening tool to detect COVID-19 infected people," in Proc. 2021 Int. Conf. Computer Communication and Informatics (ICCCI), pp. 1–4, 2021.

[14] S. Jarkad and J. E. Nalavade, "Approach for big data mining in Hadoop framework: A survey," International Journal of Innovative Research in Science, Engineering and Technology, vol. 6, no. 1, Jan. 2017.

[15] A. Patel, R. Kumar, K. Nagarajan, J. E. Nalavade, and A. Arudkar, "Mobile app for rural development and governance of village centric system – MCDM approach," International Journal for Scientific Research & Development, vol. 7, no. 12, 2020.

[16] J. E. Nalavade, A. Kerkar, H. A. Shaikh, and P. Satam, "Natural language query processing for SQL and NoSQL queries," International Journal of Scientific & Engineering Research, vol. 11, no. 3, Mar. 2020.

[17] J. E. Nalavade, A. Anarse, and H. Shaikh, "Airport management using face recognition base system," International Research Journal of Modernization in Engineering Technology and Science, vol. 3, no. 9, pp. 827–830, 2021.