WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Patient Vital Signs Monitoring System Using Socket Communication

*M. Esakkiammal[a], K. Merriliance[b]**

[a] *Student, Reg. no: 24081205300112007, Department of Computer Application and Research Centre, Sarah Tucker College (Autonomous), Tirunelveli.*

[b] *Associate Professor, Department of Computer Application and Research Centre, Sarah Tucker College (Autonomous), Tirunelveli.*

A B S T R A C T

Real-time tracking of patient vital signs is essential in emergency care and clinical decision-making. Conventional systems have hitherto been based on manual monitoring, which is workload-demanding for healthcare professionals and may result in delayed detection of abnormal conditions. To overcome these shortcomings, the present paper introduces a socket-based Patient Vital Signs Monitoring System with the aim of providing real-time healthcare surveillance. The system is a client-server system with every patient node being a client and sending the critical parameters like heart rate, body temperature, and oxygen saturation (SpO_2) to a central server through socket communication. The server calculates and keeps the readings in a relational database as well as updating Streamlit-based [9] dashboard that can show twenty patients at a time. It has an auto alert feature that alerts serious patients on the dashboard and in real-time through WhatsApp notification and voice alert to enable hospital staff to respond in time. It minimizes human observation, prevents human error, and maximizes patient safety overall. Its modularity and scalability make it the ideal solution for installation in hospital wards, ICU wards, and off-grid medical practices.

Keywords: Socket Communication; Patient Monitoring; Real-Time Healthcare; Dashboard, Alerts; Telemedicine.

1. Introduction

In contemporary healthcare, the capacity to monitor in-real-time patient vital signs is most important in determining safety, detection of emergencies at an early stage, and timely interventions. Dangerously life-threatening vital signs like heart rate, body temperature, and oxygen saturation (SpO_2) are key signs of a patient's state of physiology. Abnormalities in such parameters are the first signs of deteriorating health, especially in intensive care conditions like ICUs, post-operative recovery units, and ERs. Conventional patient monitoring systems tend to rely on manual nurses' observations or specialized monitoring devices that are extremely expensive. While bedside monitors do exist, they typically restrict to a single patient and do not offer centralized overview of several wards. Manual observation is time-consuming as well as prone to delays, especially where a single nurse is responsible for monitoring several patients at a time. These constraints can lead to premature diagnosis of life-altering diseases, premature interventions, and avoidable deaths with the progression of digital healthcare technologies, a growing demand for affordable, scalable, and automated monitoring systems that reduce the workload on healthcare personnel and enhance patient safety is underway. In this paper, a real-time patient monitoring system based on socket communication is proposed that intends to bridge the gaps of conventional practices. By utilizing a client–server architecture [6], database integration, and an easy-to-use visualization panel, the system allows for real-time monitoring of greater than one patient. In addition, its alert mechanism ensures critical cases are alerted in real-time to their relevant attending medical personnel.

2. Related Work

Monitoring patient vital[10] signs has been an area of research for decades, and most technology solutions have proved to bring the real-time and compliant healthcare monitoring due to the demand. Traditional bedside monitors, commonly utilized in hospitals and intensive care units, measure patient parameters like heart rate, blood pressure, temperature, and oxygen saturation with accuracy. Yet, these types of systems tend to be patient-specific, costly, and not offered with central visibility across several wards. This renders them unsuitable for settings where large numbers of patients need to be overseen at the same time.

In recent years, growth in Internet of Things (IoT)[7] devices have opened up new potential for remote health monitoring. A number of research studies have investigated wearable sensors that are able to send health information to cloud platforms for storage and analysis. These types of systems enable physicians and caregivers to remotely monitor patients, especially in rural and underserved communities. Although such IoT[1] technologies are revolutionary, they are plagued with issues of cost, network stability, and privacy of data. Additionally, the dependence on internet connectivity lowers their use in cases where it cannot be ascertained that the networks will remain stable.

Cloud-based[5] monitoring systems also gained popularity, providing scalability and analytics on data to healthcare facilities. These systems are usually interfaced with machine learning algorithms to forecast patient health patterns and issue early alarms of impending complications. Their installation, however, calls for strong infrastructure, technical expertise, and huge operating expenses, which could be out of reach for small clinics and hospitals.

In contrast to all the available solutions, this socket-based monitoring system offered in this paper is cost-effective, scalable, and offers real-time responsiveness. In contrast to the conventional bedside monitors, it enables several patients to be centrally monitored. In contrast to most IoT and cloud-based solutions, it does not depend significantly on high-end sensors or sophisticated infrastructure. The system instead exploits socket communication to attain low-latency data transfer, coupled with a dashboard and automated alert to ensure optimal response time.

3. System Architecture

The structure of the suggested patient monitoring system is illustrated in Figure 1. It has four primary elements: clients, server, database, and dashboard that communicate among themselves to facilitate real-time monitoring of patient health.

The Clients (Patient Simulators) create uninterrupted streams of life vital data like heart rate, body temperature, and oxygen saturation (SpO_2). The data are sent to the Server (Nurse Station) via socket communication. The server is the system's central processing unit where all the vitals are checked against threshold values to identify if the patient's status is normal or critical.

After processing the data, the server updates the Database (MySQL) with the latest patient records so that historical data is saved to be analyzed. In abnormal or critical cases, the server triggers alerts in two ways: a beep signal in the local system and an instant WhatsApp message to alert healthcare providers. This guarantees that critical cases get attention immediately even if medical staff are off the monitoring screen.

The data stored can be fetched and displayed with the Dashboard (Doctor's Screen), developed in Streamlit. The dashboard makes queries to the database and presents up-to-date patient records in an interactive and user-friendly manner. Doctors and healthcare professionals are able to see normal patients, filter critical cases, and monitor recent updates in real-time. The architecture, therefore, provides unhindered data flow from patients to caregivers with both on-site and remote alerting mechanisms.

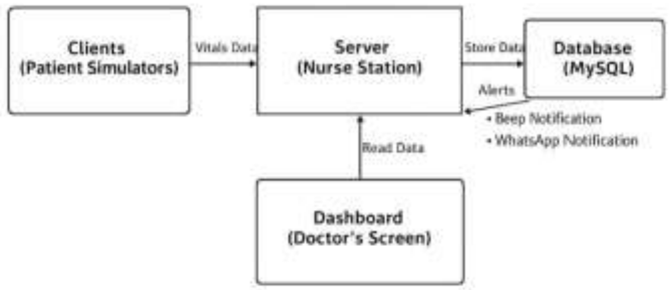

Figure 1. System Architecture of the Patient Monitoring System

4. Implementation

The Patient Vital Signs Monitoring System is implemented using Python, incorporating socket programming, multithreaded programming, database management, and a visualization framework. The implementation is modular, with each module for the client, server, database, and dashboard separated. Each module has been implemented with well-defined responsibilities, making it reliable, scalable, and easy to maintain.

Client

The client module is used to simulate individual patients, each sending their vital signs to the central server. This module was coded with Python's multiprocessing and socket libraries, which allow for parallel runtimes of several patient processes. Each process will simulate a distinct patient by periodically producing random values for heart rate, body temperature, and oxygen saturation. The information is encapsulated into a JSON format for efficient structured communication, making it compatible with the server's data-manipulation mechanism.

The client is configured to report updates at fixed intervals, with jitter in timing to mimic real-world patient monitoring [4] scenarios. For instance, heart rate is modeled to be within the normal range for most of the time and abnormal readings to check the system's capability to identify critical situations. The data structure in JSON format contains patient ID, bed number, and vital parameters, thus being easily human-readable as well as machine-parseable. The multiprocessing usage enables the system to increase to twenty patients at a time, thus simulating the hospital ward scenario.

Server

The server module is the central component of the system, where it accepts and processes data from various clients simultaneously. Written using Python's socket and thread libraries, the server accepts incoming connections and creates a separate thread for every patient. This avoids any single client from getting blocked or overlooked and allows real-time monitoring continuously.

Upon receiving data, the server parses the JSON string, extracts the patient information, and classifies the health status based on predefined thresholds. The classification algorithm evaluates whether the parameters fall within the acceptable physiological ranges. If a patient's condition is critical, the server not only updates the database but also activates the alert mechanisms, including an audio beep and a WhatsApp notification. The server module design focuses on low latency and reliability to guarantee that no time passes between receiving data and creating an alert.

Database Integration

The database is implemented as a MySQL component to store and manage patient records. Every record in the database contains fields like patient ID, bed number, heart rate, body temperature, oxygen saturation, health status, and timestamp. The server updates the records dynamically whenever new data are received so that the database always presents the latest condition of every patient.

The utilization of a relational database offers some benefits, such as persistence of data, structured storage, and effective querying. It also accommodates scalability, permitting the system to be scaled out from the initial twenty patients without major architectural changes. In addition, having a history of patient records is useful in doing retrospective analysis and auditing, which can be useful for clinical research and quality assurance in healthcare [8] facilities.

Dashboard Module

The display of patient data is done through a dashboard constructed with Streamlit, an open-source Python library for data-driven web applications. The dashboard directly interfaces with the MySQL database, fetching and showing patient data in real time. An automatic refresh feature maintains the refreshed data on screen without intervention by healthcare staff.

The dashboard's layout emphasizes usability and simplicity. Patients are presented in tabular form with corresponding parameters, bed numbers, and health status. To enhance efficiency, critical patients are emphasized in bold, so that staff members can easily spot them and act accordingly. Interactive controls like buttons for filtering or clearing data views are also provided on the dashboard, allowing flexibility for various monitoring requirements. By blending automation with human-centric visualization, the dashboard improves situational awareness and facilitates quick decision-making in a clinical environment.

Alerting Mechanism

The system has a dual alerting mechanism to provide assurance that life-critical conditions will not be missed. The first is a local sound beep issued on the server machine when a patient reaches a critical state. This allows for instant attention from personnel working close to the monitoring station. The second mechanism is providing instant WhatsApp messages to healthcare [2] professionals via the pywhatkit library. These are automatically sent notifications that include precise patient details such as ID, bed number, and vital sign values so that remote personnel remain updated as well.

In order to avoid unnecessary notifications, the system has an anti-spam mechanism that restricts notifications to one per patient per minute. Through this, healthcare professionals are not bombarded by multiple notifications while still being notified of all critical events. With both local and remote integration of alerts, the system achieves a balance between urgency and realizability, minimizing the chances of delayed interventions.

5. Results and Discussion

The proposed Patient Vital Signs Monitoring System was evaluated through a series of tests simulating a hospital environment with up to twenty patients monitored simultaneously. The results demonstrated that the system effectively achieved real-time data transmission, visualization, and alerting. Each client representing a patient continuously generated and transmitted vital signs to the central server without noticeable delay. The server processed this data concurrently, ensuring that no client connection was ignored even under high-load conditions. These observations validated the scalability and responsiveness of the socket-based architecture. One of the most significant outcomes of the system is its ability to detect abnormal health parameters in real time. For instance, when a patient's oxygen saturation dropped below 95% or the body temperature exceeded 38 °C, the system immediately classified the case as critical. This classification triggered both local and remote alerts: a beep sound was generated at the server terminal,

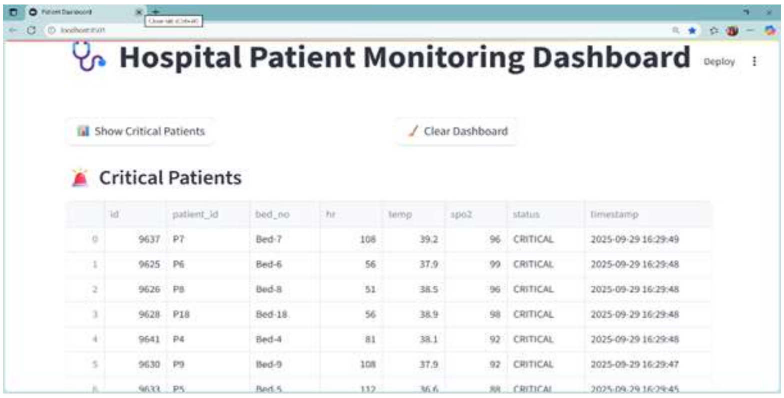

Figure 2: Dashboard Display of Critical Patients.

Normal patient records were displayed clearly and updated automatically.

Figure 4: Normal Patient Records Displayed in the Dashboard.

The automatic refresh feature, which updated the display every five seconds, ensured that the information presented was always current. This eliminated the need for manual refreshes and allowed staff to focus on decision-making rather than data management.

In comparison with traditional manual monitoring methods, the proposed system offers several advantages. Manual approaches require periodic observation by nurses or doctors, which introduces a risk of missing sudden changes in patient conditions. In contrast, the automated system continuously tracks vital signs and provides immediate notifications, significantly reducing response time. Furthermore, by centralizing patient data on a single dashboard, the system reduces the workload of healthcare staff who would otherwise need to visit each patient individually.

Another important consideration is the cost-effectiveness and adaptability of the system. Unlike advanced IoT- or cloud [3]-based healthcare solutions, which often require expensive hardware and extensive infrastructure, the proposed system can be implemented using standard computers and existing hospital networks. This makes it accessible to smaller hospitals and clinics with limited budgets. Additionally, the modular design ensures that the system can be extended to accommodate more patients or integrated with wearable devices in future implementations.

Overall, the results highlight that the Patient Vital Signs Monitoring System is not only functional but also practical for real-world healthcare settings. By reducing manual workload, minimizing human errors, and ensuring rapid responses to critical conditions, the system contributes to improved patient safety and more efficient healthcare delivery.

6. Application

The Patient Vital Signs Monitoring System has a wide range of applications in both traditional hospital environments and emerging healthcare scenarios. Its flexibility, scalability, and ability to deliver real-time alerts make it suitable for diverse contexts where continuous patient monitoring is critical.

One of the most direct applications of the system is within hospital wards and intensive care units (ICUs). In these settings, the ability to observe multiple patients simultaneously is invaluable. Instead of relying solely on bedside monitors or manual checks by nursing staff, the centralized dashboard enables doctors and nurses to supervise up to twenty patients in real time. Critical cases are highlighted instantly, ensuring that urgent conditions are addressed without delay. This centralized approach not only reduces the physical workload on medical staff but also enhances situational awareness in fast-paced clinical environments.

The system is equally relevant in remote healthcare monitoring and telemedicine. Patients located in rural or underserved areas often lack access to constant medical supervision. By deploying this system, vital signs can be transmitted to a central server located at a hospital or health center, while WhatsApp alerts notify doctors of critical conditions even if they are physically distant. This makes the system a valuable tool for bridging the gap between patients and healthcare providers, especially in regions where medical infrastructure is limited.

Another important application lies in elderly care and home-based monitoring. Elderly patients or those with chronic health conditions often require continuous observation but may prefer staying at home rather than being admitted to a hospital. With this system, caregivers and healthcare professionals can monitor their vital signs remotely and respond quickly if abnormalities arise. The WhatsApp alert mechanism further ensures that family members or caregivers are also informed, creating a multi-layered support system.

Finally, the system can be adapted for emergency medical services such as ambulances. During transportation of critically ill patients, continuous monitoring is essential to prepare receiving hospitals for immediate intervention upon arrival. By integrating this system into ambulances, paramedics can transmit real-time data to hospital servers, allowing medical teams to be ready with the necessary equipment and treatment protocols before the patient reaches the facility.

In summary, the applications of the proposed system extend beyond hospital wards, making it suitable for ICUs, remote healthcare, elderly care, and emergency services. Its adaptability to different environments underscores its potential as a versatile solution for modern healthcare challenges.

7. Conclusion

The socket communication-based Patient Vital Signs Monitoring System offers an efficient, scalable, and robust solution to real-time health monitoring. Through the combination of a client–server architecture with socket programming, the system allows for multi-patient observation in real-time and ensures timely detection of abnormal conditions. The use of a relational database guarantees persistence of patient records, while the dashboard, based on Streamlit, offers an interactive platform for the healthcare staff to view normal and critical cases. In addition, the use of the alerting mechanism, which is in the form of sound signals accompanied by WhatsApp messages, greatly improves the responsiveness of healthcare workers, making delayed interventions unlikely.

The outcome of system testing proves that the solution has a positive impact in lessening the workload of manual monitoring and avoiding human errors. Through ongoing monitoring of patient parameters like heart rate, temperature, and oxygen saturation, and tagging them as NORMAL or CRITICAL, the system prioritizes patients at risk. In contrast to conventional monitoring techniques, the system proposed here is not only more efficient but also offers the additional domain of healthcare delivery for remote and underserved regions. Its modular nature allows it to be suitable for all kinds of environments such as hospitals, intensive care units, elderly home care, and emergency medical services.

8. Future Enhancements

While the system succeeds in meeting its key goals, there are some avenues of future development. One such area is incorporating wearable IoT devices, through which direct measurement of vital signs would be possible in place of simulated values. Another area with high potential is implementation of cloud-based architecture to facilitate mass deployment across multiple hospitals and health centers. Moreover, the integration of artificial intelligence and machine learning patterns may make predictive healthcare possible, in which patterns within patient information are used to anticipate potential health risks prior to them becoming severe. Lastly, creation of a mobile application would offer physicians, caregivers, and family members instant access to patient status, further expanding the accessibility and usability of the system.

References

1. Alaa, M. Mohammed, and M. Al-Rawi, "Real-Time Monitoring System for Patient Vital Signs Using IoT and Cloud Computing," *Journal of Medical Systems*, vol. 44, no. 12, pp. 1–12, 2020.

2. Ghosh, T. Chakraborty, and S. Chatterjee, "Socket Programming Based Real-Time Data Transmission in Distributed Healthcare Systems," *International Journal of Computer Applications*, vol. 182, no. 35, pp. 12–18, 2019.

3. Sood and R. Sandhu, "Real-Time Data Analytics for Patient Monitoring Using Cloud and IoT," *Future Generation Computer Systems*, vol. 98, pp. 654–665, 2019.

4. D. G. Lowe, "Real-Time Patient Monitoring Systems: A Review of Current Technology and Future Directions," *Health Informatics Journal*, vol. 25, no. 4, pp. 1478–1492, 2019.

5. M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, "Wearable 2.0: Enabling Human-Cloud Integration in Next Generation Healthcare Systems," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 54–61, 2017.

6. P. P. Ray, "A Survey on Internet of Things Architectures," *Journal of King Saud University – Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

7. R. Gupta and A. K. Sharma, “IoT-Based Patient Health Monitoring System Using Wireless Sensors,” *International Journal of Engineering Research and Technology*, vol. 9, no. 6, pp. 245–250, 2020.
8. S. M. Riazul Islam, D. Kwak, M. Humaun Kabir, M. Hossain, and K. Kwak, “The Internet of Things for Health Care: A Comprehensive Survey,” *IEEE Access*, vol. 3, pp. 678–708, 2015.
9. S. R. Bharathi and V. K. Jha, “A Streamlit-Based Dashboard for Healthcare Data Visualization,” *International Journal of Research and Analytical Reviews*, vol. 8, no. 4, pp. 102–109, 2021.
10. World Health Organization, *Digital Health Systems and Patient Monitoring in Developing Countries*. WHO Report, 2020.