International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Patient Healthcare Monitoring System Using Iot Based Fog Computing

ALIYANZ SHAIKH[1], K.K. SHARMA[2], VIVEK MENON[3]

[1]Research Scholar, [2]Professor, [3]Assistant Professor
Department of Information Technology
Shri G.S. Institute of Technology and Science
Indore, Madhya Pradesh 452003, India.

Abstract:
Human needs begin with health. Humanity is facing a lot of challenges. Unplanned death and high rate of diseases caused by the spectrum of diseases generated by it are due to patients not being treated at the appropriate time. The primary objective of this project is to design an efficient and functional system for keeping track of patients with the help of IoT; thus, their health professional care will be provided to their patients. Worn or not, the sensors will be implanted in the patient's body to monitor their health constantly. Information gathered in this way will be stored, analyzed, and exhaustively mined to try to meet the initial prognosis of diseases. A system in development will provide live online data via a device. on physiology. Sensors, the data acquisition system, and others constitute the patient's disease most of it coded with Arduino. The system reads, displays, and maintains the patient's heart rate, temperature, graphical awareness square measure, then sends it to the patient's and physician's mobile phone, which contains the device. Health condition is logged and monitored by the practical Health Observance System on its website. The subject discusses how to achieve the proposed model output by way of study, research methodology, training, testing, and presentation of completed work with appropriate presentation by developing the user interface application using programming languages.

Keywords: Internet of Things, Fog Computing, Arduino Uno, Esp8266, Pulse Sensor, Artificial Neural Network, LM35 Temperature Sensor, ThingSpeak.

1. INTRODUCTION

Healthcare has been transformed by the Internet of Things (IoT), which allows devices to exchange data and communicate with one another via the internet [1]. IoT- based solutions are able to deliver prompt interventions and real-time patient health monitoring. However, issues with data security and privacy come up, especially when it comes to private health information [2][3]. The sensors used by modern hospital patient monitoring systems, which essentially confine the patient to the hospital bed and provide continuous vital sign monitoring, must be connected to nearby PCs or bedside monitors. A paramedical assistant still needs to do other duties even after connecting these systems to a particular patient. You can continuously monitor and record a patient's vital parameters by keeping track of them [4][5]. Fog computing, which extends cloud computing to the network's edge, has surfaced as a solution to these issues [6]. It makes local data processing possible on edge devices, which enhances response times and performance.

Fog computing enhances data privacy and reduces latency by keeping data closer to its source. However, data Transmitted across network paths and nodes remain susceptible to losses and corruption, necessitating robust Data integrity and security mechanisms [7,8]. The majority of hospitals would not want patient data to be stored outside, which makes a basic sensor-to-Cloud architecture impractical for many healthcare applications [9]. There is also the depressing possibility of data centre or network failure, which could endanger the health of patients. Here's where fog computing Helps with medical applications. The healthcare industry has fortunately made use of smart devices. In the modern world, a variety of medical Devices that can be utilised by patients at home or even on their own clothing. Sensors are the main component Of the devices. Electrical, thermal, optical, chemical, and other signals can be detected by these sensors, which Typically include transducers [10].

In order to make the best choice in the case of a medical emergency, the patient's information is shared with the doctor and their family members. Real-time patient information dissemination to the patient's family is made possible via the Internet of Things. We can track small children and the elderly from anywhere at any Time thanks to wearables and other smart wireless technologies. Information is given in real time during an Emergency, enabling us to plan and coordinate different life-saving preventive actions. For critical medical IoT-based applications that demand more precision, real-time answers, and stable Behaviour, cloud-based architectures might have a major detrimental impact during network outages or Bandwidth delays, potentially leading to medical emergencies or even fatalities.

Edge computing and fog computing improve the overall effectiveness of the cloud system while mitigating Some of its drawbacks. The fog layer sits in between the cloud and end devices. To reduce the overall Communication latency, a little percentage of the massive input data is processed and analysed at the fog layer Before being sent to the cloud. While data that must be permanently stored is delivered to the cloud, time-Sensitive data is sent to the fog layer for rapid processing. Fog improves the functionality of cloud systems Rather than taking their place. By decreasing the number of medical visits and needless treatments, the Internet of Things enables physicians To remotely monitor patients, efficiently arrange appointments, and enhance at-home healthcare, eventually Increasing patient safety and lowering healthcare expenses [11]. Remote health monitoring, hospital procedure simplification, and ongoing environmental and bodily. Monitoring are made possible by the Internet of Things. This will support the advancement of rehabilitation And the management of chronic illnesses, as well as provide effective data connections for virtual Consultations in remote medical care [12].

One of the most important measures of human health is body temperature. The other is pulse rate, which Measures the number of heart beats per minute using standard Falls between 60 to 100 beats per minute for healthy people; adult males typically have a resting heart rate of About 70 beats per minute, while females typically have a resting heart rate of 75 beats per minute [13]. These values can vary based on factors such as age, fitness level, and overall health. Monitoring both body Temperature and pulse rate can provide essential insights into an individual's physiological state and Potential health concerns. Through the use of a context-sensitive fog computing environment, we aim to enhance the current Healthcare systems in this article. We go over the many computing jobs necessary for healthcare that will be Done at user devices and sensors, in the cloud, or in the fog layer. Information travelling via the system will be more securely protected because it won't have to move back and forth across the network, limiting data exposure. The cloud's services can be split up across multiple nodes so that the overhead time required for data to move between devices and the cloud is offset or decreased. By distributing the services across multiple nodes, some of the issues that earlier healthcare monitoring applications encountered are resolved. Monitoring a patient's vital signs closely is essential to good medical practice in the modern healthcare setting. Basic metrics such as body temperature and pulse rate serve as instantaneous gauges of an individual's health and frequently offer the first hints of underlying medical conditions. Although manually checking these vitals is a tried-and-true method, it can be time-consuming and may not always detect subtle but crucial changes in a patient's condition immediately. This is where machine learning enters the picture, providing an automated, precise, and immediate analysis of this crucial data, thus providing a potent new means to improve patient care. By using temperature and pulse rate data to differentiate between "normal" and "abnormal" states, this study attempts to automate the classification of patient health. We will accomplish this by creating and assessing two machine learning models: an intricate non-linear model (Artificial Neural Network) and a straightforward linear model

(Logistic Regression).

2. LITERATURE SURVEY

Monitoring system prototype. Low-power specialised sensor arrays for EKG, SpO2, Jimenez et al. [14] presented a wellness monitoring framework for emergency clinics, allowing family members and experts to remotely monitor patients' medical issues via the internet using an E-wellness sensor safeguard pack interface. In any case, it doesn't transmit any warnings to the specific family members and experts, such as emails or SMS.

The authors of Banerjee et al. [15] suggested a framework for heartbeat rate identification in order to avoid any discomfort. Using the plethysmography procedure, the suggested framework meticulously displayed the results, making it a continuous observation tool. The approach has proven to be reliable for the patient when compared to other intrusive techniques. The IoT-based patient health monitoring system described in this study uses an Arduino and ESP8266 to collect patients' temperature and heart rate in real time.

Smart healthcare improves patient care and raises the standard of healthcare by utilising real-time vital signs and sophisticated diagnostic tools [16]. It consists of healthcare facilities, pharmacies, home health agencies, long-term care homes, manufacturers of medical equipment, insurance companies, and governments that provide end-user services [17]. Using IoT sensors for blood pressure, temperature, pulse, and ECG, all connected to a Raspberry Pi, Acharya and Patil [18] created an Internet of Things-based healthcare monitoring system that monitors vital health parameters, such as heartbeat, ECG, body temperature, and respiration. Nevertheless, while these devices gather data and transmit it to the system, they do not have data visualisation interfaces for thorough analysis.

According to Sarmah et al. [19], this work details the steps used to design and construct a low-cost modular monitoring system prototype. Low-power specialised sensor arrays for EKG, SpO2, temperature, and movement were used in the design of this system to provide quicker and more effective medical responses in emergency scenarios. These sensors' interfaces were designed using the Internet of Things architecture: a central control unit exposes a RESTful Web interface that guarantees platform agnostic behaviour and offers an adaptable method for adding new components. The prototype's unit cost was high and deployment was challenging. By using an Android application that gathers BT, HR, and GSR data from sensors and transfers it to a single Arduino Uno platform, Hamim et al. [20] created a smart healthcare monitoring system for patients and older adults in an Internet of Things setting. This allows medical professionals to keep an eye on patients' health. And write prescriptions. A smartphone-based heart rate monitoring device was created by Gregoski et al. [21]

That wirelessly transmits pulses for heart rate measurement without the need for manual touch. It employs a Mobile light and camera to track blood flow in the fingers and compute cardiac output. The architecture isn't Appropriate for ongoing cardiac monitoring, though. The primary objective of the IoT-based healthcare system study was to enhance the quality of life for patients by giving them real-time access to information about their health through the measurement of physiological Indicators, including body temperature, pulse rate, diastolic and systolic pressure [22]. The main idea is to Treat patients by continuously checking vital signs, including body temperature, heart rate, and blood pressure, without forcing them to visit different facilities for ongoing medical monitoring. The temperature and blood Pressure data are analysed and stored in the cloud, where the patient's carers can view it from anywhere and appropriately react to any alarms. The prototype proved costly to maintain and challenging to deploy. According to Yew et al. [23], patient monitoring whether in hospitals or at home is a crucial part of the modern Healthcare system. This study presents an intelligent patient monitoring system that automatically checks the Patient's health state using a variety of sensors.

3. PROPOSED METHODOLOGY

3.1 *Hardware Description*

The basic block diagram for the ESP8266 and Arduino- based Internet of Things-based patient health monitoring system is shown in Figure 1.

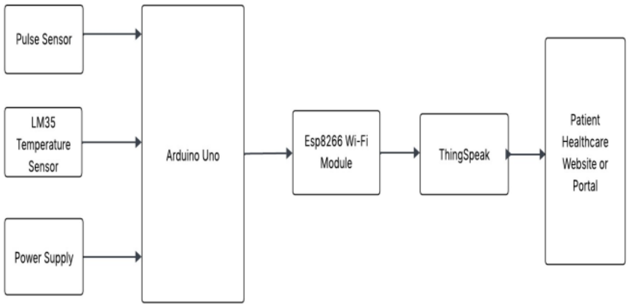


Figure 1. IoT Based Patient Health Monitoring System

The Arduino Uno, temperature sensor, pulse sensor, ThingSpeak IoT platform, Wi-Fi module, and power supply are all used in our suggested setup. In addition to helping patients stay healthy, a smart patient health monitoring system will also save doctors' workloads and patients' time.
Hardware Components:

- Arduino Uno
- ESP8266 Wi-Fi Module
- Pulse Sensor
- LM35 Temperature Sensor
- 16 × 2 LCD Display

3.1.1 Arduino Uno

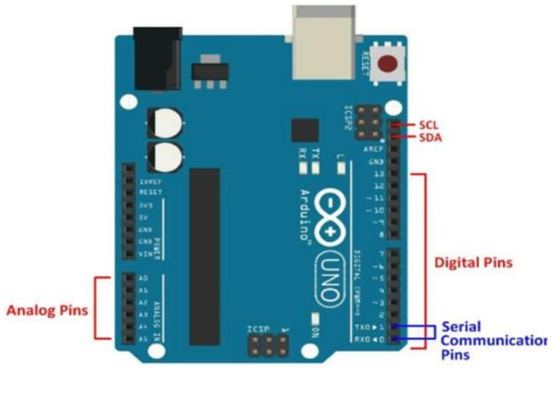


Figure 2. Arduino Uno Diagram

The proposed patient monitoring system uses an Arduino Uno to track the patient's health parameters. Once it has been connected to the Internet, the Arduino Uno is connected to a cloud database system which can act as a server. The server then automatically sends the data to the recipient machine and allows the doctor to continuously monitor the patient's health data. Any sudden variations in these parameter values can be discovered at the earliest opportunity so the doctor can administer the necessary treatment quickly.

The Arduino Uno is an open-source microcontroller board created by Arduino.cc based on the Microchip ATmega328P microprocessor. It is possible to connect expansion boards (shields), and other circuits to the board via its digital and analogue input/output (I/O) pins [24]. The Arduino IDE (Integrated Development Environment) can be used to program the board. It has 14 digital I/O pins (6 can be used for PWM), 6 analogue I/O pins, and a Type B USB cable [25]. It is powered via 7 to 20-volt external power supply (9-volt battery) or by the USB cable and is similar in stage to Leonardo microcontrollers and the Arduino Nano.

3.1.2 ESP8266 Wi-Fi Module

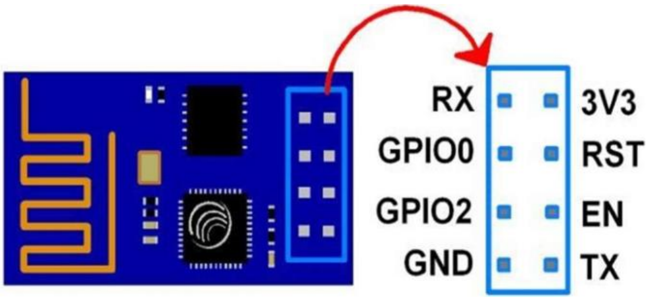


Figure 3. ESP8266 Module

The ESP8266 is a 32-bit RISC microcontroller that has Wi-Fi capability and operates at 80 MHz or 160 MHz.

Its RAM data is equal to 96 KB, and its instruction RAM is equal to 64 KB [26]. The ESP8266 was designed for wearable electronics, Internet of Things, and mobile applications, using a unique set of techniques and solutions to achieve the lowest power consumption [25]. ESP8266 is publishing the patient sensor information to the servers that use ThingSpeak. *Pin Configuration:*

3.1.2.1 3V3 – 3.3 V Power Pin
3.1.2.2 GND – Ground Pin
3.1.2.3 RST – Active Low Reset Pin
3.1.2.4 EN – Active High Enable Pin
3.1.2.5 TX – Serial Transmit Pin of UART
3.1.2.6 RX – Serial Receive Pin of UART
3.1.2.7 GPIO0 and GPIO2 – General Purpose I/O pins that determine if the module boots in boot or normal mode and if TX/RX pins are available for serial I/O or programming the module.

3.1.3 Pulse Sensor

Figure 4. Pulse Sensor Diagram

The Pulse Sensor is a heart rate monitor that works with Arduino. Anyone who wants to utilize live heart-rate data in their work, whether you're an athlete, artist, student, game developer, or smartphone developer, can do so. The system consists of a noise-reducing circuit sensor with an integrated optical amplification circuit. You can connect the Pulse Sensor to your microcontroller and clip it to your fingertip or earlobe for heart rate measurements. The Pulse Sensor has three pins: the analogue pin, VCC pin, and GND pin.

3.1.4 LM35 Temperature Sensor

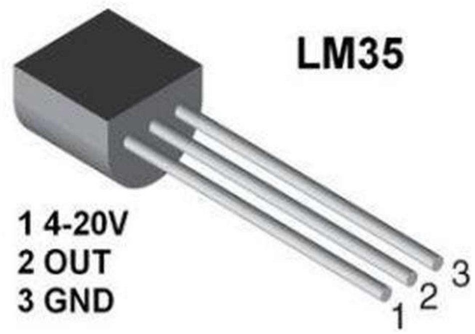


Figure 5. Temperature Sensor

The LM35 is an accurate analog temperature sensor that is known for its ease of use and reliability. The LM35's main advantage is its linear output voltage that is proportional to temperature in Celsius at a scale factor of 10mV per degree Celsius. The LM35 also has beneficial features over other types of temperature sensors. The LM35 is more effective than a standard thermistor, requires no external calibration, and has a greater output voltage than thermocouples, which often disallows the need for signal amplification. Furthermore, the low power consumption (approx. 60µA) and low self-heating mean that the measurement by the LM35 is accurate and stable. The sensor itself is encapsulated circuitry that has protection from oxidation and other environmental effects.

3.1.5 LCD Display

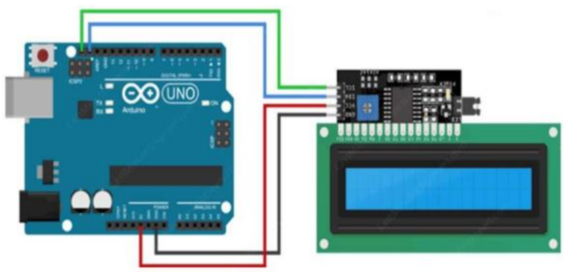


Figure 6. 16 × 2 LCD Display

A liquid-crystal display (LCD) is a video display that uses the light-modulating properties of liquid crystals to display text or images on a screen [23]. An LCD works with an Arduino microcontroller. By using an LCD display connected to an Arduino's pins, an Arduino microcontroller can be programmed to display a certain text string or image on a display.

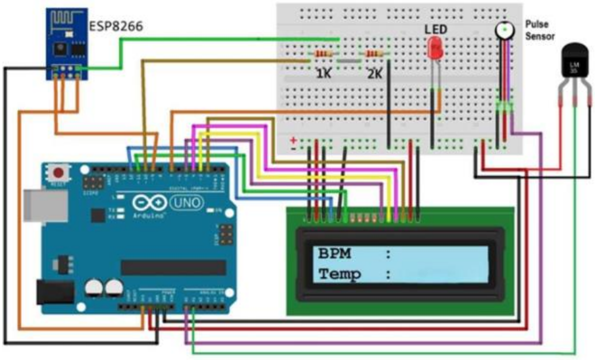


Figure 7. Circuit Diagram

3.2 Software Description

3.2.1 Arduino IDE

The open-source Arduino IDE is the program used to generate and upload sketches to any of the Arduino-compatible boards. As it functions in multiple operating systems, it is cross-platform [25]. Arduino sketches are uploaded into the prototype using the Arduino IDE. Industrial temperature sensor data is retrieved through the serial monitor.

Figure 8. Arduino IDE Platform

3.2.2 Thing Speak

ThingSpeak is a cloud-based IoT analytics platform that enables users to collect, view, and analyze real-time data streams [24, 27]. The data generated by your devices are sent to ThingSpeak and displayed in real time. ThingSpeak is used to present the physiological status of the patient. Then we can supervise and manage our prototype system on the cloud taking advantage of the channels and web pages supplied by ThingSpeak [27–30]. Now, there is no need for ThingSpeak users to get a MATLAB license from MathWorks to analyze and visualize the uploaded data using MATLAB because ThingSpeak offers incorporated support for the numerical computation software MATLAB. For users, a partnership has been formed between ThingSpeak and MathWorks, Inc. In fact, registered MathWorks user identities can also be valid login credentials on the ThingSpeak website, and all the documentation provided by ThingSpeak has been merged under MathWorks MATLAB documentation site.

3.3 Model Implementation

The methodology for this research study was designed to allow us to engage in a systematic and reproducible process in the following components of the analysis: data preparation, model implementation, and strong assessment process.

3.3.1 Dataset and Features

The study utilized a private dataset of 452 anonymized patient records or entries. Each entry had a number of attributes; from those attributes, 2 features were selected to be analyzed: Pulse Rate (beats per minute) and Temperature (degrees Celsius). The outcome variable was the state of the patient, meaning the categorical label of either ‘normal’ or ‘abnormal’.

3.3.2 Data Preprocessing

A multi-step preprocessing pipeline was followed to prepare the data for machine learning:

3.3.2.1 Data Cleaning: Non-predictive attributes (timestamps and patient identifiers) were removed to allow the analysis to focus solely on the physiologic data.

3.3.2.2 Label Encoding: The categorical target variable, status, was converted to a numerical variable. The ‘normal’ class was encoded to be equal to 0, and the ‘abnormal’ class was encoded to be equal to 1 to allow for binary classification.

Feature Rescaling: A Standard Scaler was used to prevent features with larger numerical ranges from unduly influencing the model. The Standard Scaler standardized both Pulse Rate and Temperature, so they both had the same mean (mean = 0) with a standard deviation of 1. This allowed both features to drive the model equally during model training. The following is the formula calculated during standardization:

$$z = \frac{x - \mu}{\sigma}$$

Where x is the original value of the feature, μ is the mean of the feature, and σ is the standard deviation.

3.3.3 Implementation

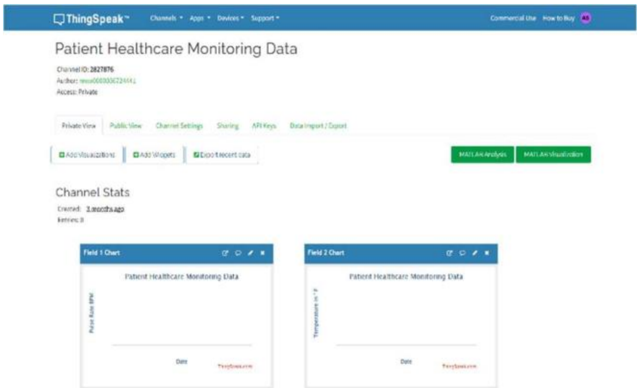


Figure 9. ThingSpeak Platform

Two supervised learning models were implemented:

3.3.3.1 Logistic Regression: A Logistic Regression model was implemented as a strong baseline. This model predicts the probabilities of a binary output by utilizing the sigmoid function.

3.3.3.2 Artificial Neural Network (ANN): A feedforward Artificial Neural Network model was implemented to explore a more complex, non-linear function. The architecture consisted of an input layer with two neurons (the two features), followed by two hidden layers with 16 and 8 neurons, respectively, both employing the Rectified Linear Unit (ReLU) activation function.

There are a number of popular activation functions; ReLU is often used because it is proven to fight against the vanishing gradient issue and it introduces complexity into the model as it is non-linear. The ReLU activation function is simplified to:

$$ReL(x) = max(0, x)$$

The output layer contained a single neuron, using the Sigmoid activation function (which is useful for binary classification, since it can return a probability between 0 and 1).

The model was compiled with the Adam optimizer and binary cross-entropy loss function, which defines the gap between the predicted probability p and actual label y:

$$L = -[y \cdot \log(p) + (1 - y) \cdot \log(1 - p)]$$

The model was trained over 50 epochs.

3.4 Evaluation Framework

The performance of both models was evaluated using standard classification metrics derived from the four components of the confusion matrix: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

- Accuracy: The proportion of total predictions that were correct.

Accuracy = $\frac{TP+T}{TP+TN+FP+FN}$

- Precision: The proportion of positive predictions that were actually correct.

Precision = $\frac{TP}{TP+F}$

- Recall (Sensitivity): The proportion of actual positives that were identified correctly.

Recall = $\frac{TP}{TP+F}$

F1 Score: The harmonic mean of Precision and Recall.

F1- Score = $\frac{2.(Precision.Recall)}{(Precision+Rec\quad)}$

Additionally, Receiver Operating Characteristic (ROC) curves and the corresponding Area Under the Curve (AUC) were computed to provide an aggregate measure of model performance across all classification thresholds.

4 RESULTS AND ANALYSIS

Using affordable and frequently available sensors and components, the prototype of a patient monitoring system based on the Internet of Things (IoT) was developed. The device measures a patient's temperature and pulse using sensors and in real time. Data is gathered and processed by an Arduino board, and the results are uploaded to the ThingSpeak data servers through an ESP8266 Wi-Fi module, which transmits the vital data through the internet. Finally, it is visualized in real-time on the dashboard to be monitored from remote locations.

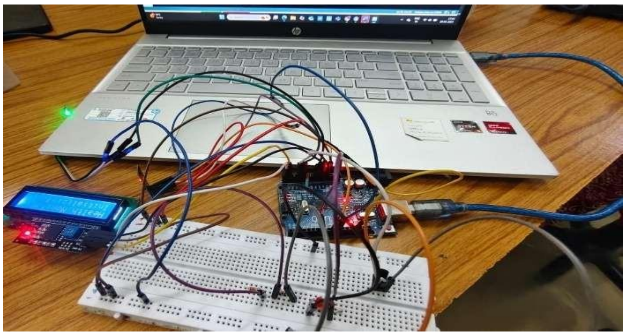


Figure 10. Patient Healthcare Monitoring System Using Components

Once the Arduino board is connected to a laptop, via a common USB cable, the entire circuit is powered, converting 5 volts from the USB to provide steady power to the Arduino. The Arduino then powers all the components, based on their voltage requirements: As the temperature sensor requires 5V, it is powered directly from one of the 5V pins on the Arduino board.

The pulse sensor and the ESP8266 Wi-Fi module are sensitive components and require 3.3V; we carefully powered these directly from the 3.3V pin from the Arduino board. After the circuit has been powered and the code successfully uploaded to the board, we can, right away, see sensor readings and the system status in the serial monitor.

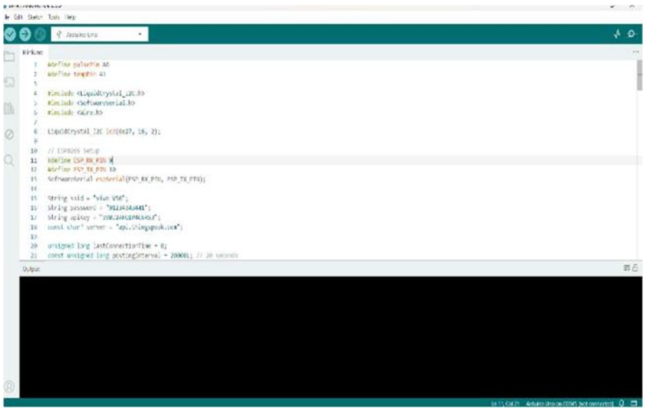


Figure 11. Arduino Uno Code

When the code is uploaded, check the connectivity of the Wi-Fi and the ThingSpeak and check whether they both are working or not. If we connected to the Wi-Fi, then the data sent on ThingSpeak easily.

Figure 12. Wi-Fi Connectivity Status

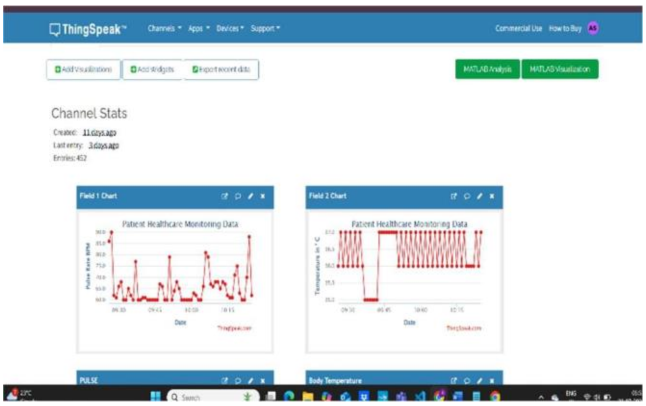
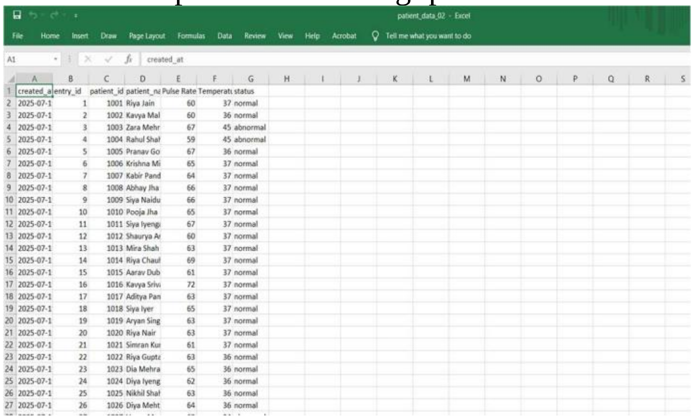


Figure 13. Data observed on ThingSpeak Channel

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

These Patient data CSV file which are exported from ThingSpeak Platform:



created_a	entry_id	patient_id	patient_na	Pulse Rate	Temperatu	status
2025-07-1	1	1001	Riya Jain	60	37	normal
2025-07-1	2	1002	Kavya Mal	60	36	normal
2025-07-1	3	1003	Zara Mehr	67	45	abnormal
2025-07-1	4	1004	Rahul Shal	59	45	abnormal
2025-07-1	5	1005	Pranav Go	67	36	normal
2025-07-1	6	1006	Krishna Mi	65	37	normal
2025-07-1	7	1007	Kabir Pand	64	37	normal
2025-07-1	8	1008	Abhay Jha	66	37	normal
2025-07-1	9	1009	Siya Naidu	66	37	normal
2025-07-1	10	1010	Pooja Jha	65	37	normal
2025-07-1	11	1011	Siya Iyeng	67	37	normal
2025-07-1	12	1012	Shaurya Ar	60	37	normal
2025-07-1	13	1013	Mira Shah	63	37	normal
2025-07-1	14	1014	Riya Chaul	69	37	normal
2025-07-1	15	1015	Aarav Dub	61	37	normal
2025-07-1	16	1016	Kavya Srivi	72	37	normal
2025-07-1	17	1017	Aditya Pan	63	37	normal
2025-07-1	18	1018	Siya Iyer	65	37	normal
2025-07-1	19	1019	Aryan Sing	63	37	normal
2025-07-1	20	1020	Riya Nair	63	37	normal
2025-07-1	21	1021	Simran Kur	61	37	normal
2025-07-1	22	1022	Riya Gupta	63	36	normal
2025-07-1	23	1023	Dia Mehra	65	36	normal
2025-07-1	24	1024	Diya Iyeng	62	36	normal
2025-07-1	25	1025	Nikhil Shal	63	36	normal
2025-07-1	26	1026	Diya Meht	64	36	normal

Figure 14. Patient Data CSV

After processing the dataset with the ANN algorithm, we obtained some resulting of its performance depicted in graphical form.

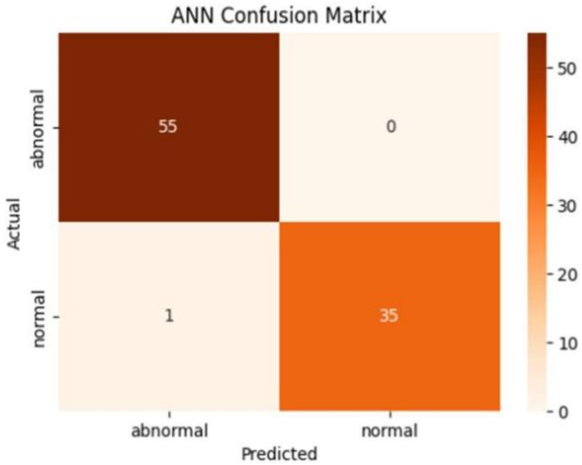


Figure 15. ANN Confusion Matrix

The confusion matrix for the ANN model generated the following results:

- True Negatives (Top-left): The model correctly identified 55 of the cases as ‘normal’.
- False Positives (Top-right): The model identified ‘abnormal’ instead of the actual state of ‘normal’ for 0 cases.
- False Negatives (Bottom-left): The model identified the state of ‘normal’ instead of the actual state of ‘abnormal’ for 1 case.
- True Positives (Bottom-right): The model identified 35 of the cases as ‘abnormal’.

With these values, we can deduce that the ANN model performed fairly well. When considering the identification with respect to ‘normal’, the model did not execute a false positive, despite it making an error by wrongly identifying 1 ‘abnormal’ instance as “normal”; all others were correct.

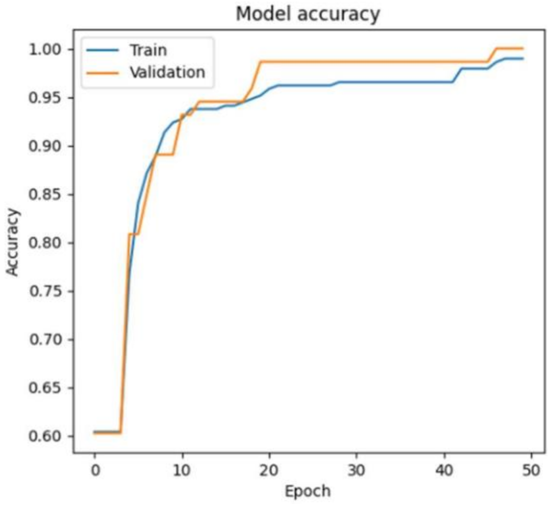


Figure 16. ANN Model Accuracy

The figure shows the training accuracy and validation accuracy of the Artificial Neural Network (ANN) model across epochs.

- The blue line is training accuracy: how well the model is performing on the data the model was trained on.
- The orange line is the validation accuracy: how well the model is performing from a reserved proportion of the training data not used to train the model.

It is preferable to see both lines increase and stay close together in value. If the training accuracy is significantly better than the validation accuracy, then it may indicate that the model is overfitting to the training data. In this figure, we can see both accuracies increase and appear to stay close, suggesting that the model is learning properly without a lot of overfittings.

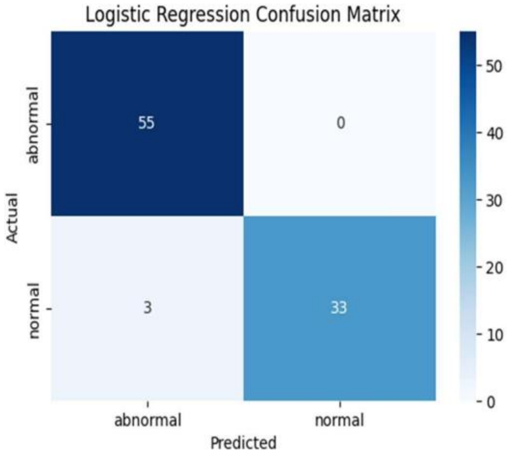


Figure 17. Logistic Regression Confusion Matrix

This data presents a confusion matrix for the Logistic Regression model. Likewise, the ANN confusion matrix, it displays the model's predictions compared against the actual classes.

In this matrix, we can see:

- True Negatives (Top-left): The model got 55 instances right as predicted 'normal'.
- False Positives (Top-right): The model got 0 instances wrong as predicted 'abnormal' as compared to 'normal'.
- False Negatives (Bottom-left): The model got 3 instances wrong as predicted 'normal' as compared to 'abnormal'.
- True Positives (Bottom-right): The model got 33 instances as 'abnormal'.

In looking at the ANN confusion matrix, the Logistic Regression model got similar results with no false positives but had a bit more false negatives with 3 compared to 1. This means that the Logistic Regression misclassified more ‘abnormal’ cases as ‘normal’.

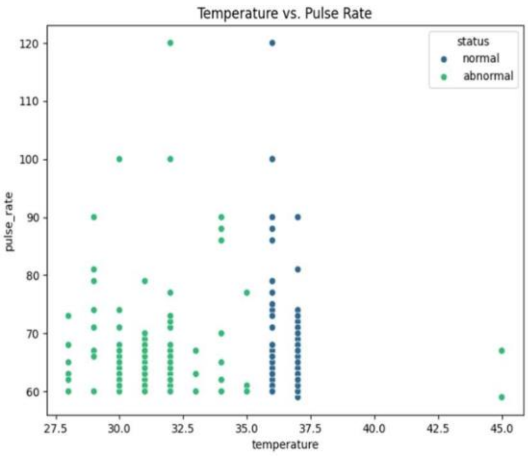


Figure 18. Temperature vs Pulse Rate

The scatterplot below provides a visual representation of the relationship between ‘Temperature’ and ‘Pulse Rate’; the points are colored according to the ‘status’ (‘normal’ or ‘abnormal’). Each point on the plot represents a single patient entry; the position of the temperature and pulse rate determined its placement on the plot. The color of the point indicates whether that patient’s status was ‘normal’ or ‘abnormal’. This type of plot will allow you to visualize whether there are clusters or patterns in temperature and pulse rate that correspond to the patients’ status. You should be able to view whether patients with ‘abnormal’ status tend to have different temperatures or pulse rates as compared to those with ‘normal’ status.

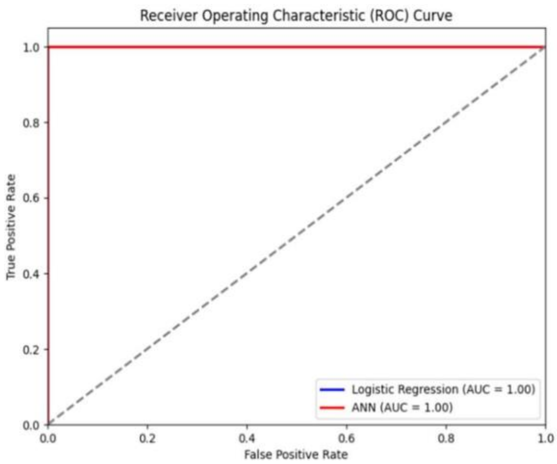


Figure 19. ROC Curve

This plot demonstrates the plots of the Receiver Operating Characteristic curves (ROC) for the Logistic Regression and ANN models so that you can visually compare their performance.
The ROC curve implicates a plot of True Positive Rate (sensitivity) versus False Positive Rate (1-specificity) at different threshold settings; it demonstrates how accurately identifying positive cases must be balanced with incorrectly classifying negative cases as positive cases.
One metric that can summarize a binary classification model’s overall performance is the area under the curve (AUC). Ideally, a model is perfect if it has an AUC of 1 or worse than random if it has an AUC of 0.5.
In this case:

- The blue line is the ROC curve for Logistic Regression with an AUC of 0.96.
- The red line is the ROC curve for the ANN with an AUC of 0.99.

Higher AUC will indicate better model performance than lower AUC. Both models clearly performed well as you saw in the plot, but the ANN appears to have a slight performance advantage compared to LR because of its marginally higher AUC and improved discrimination between the "normal" and "abnormal" classes.

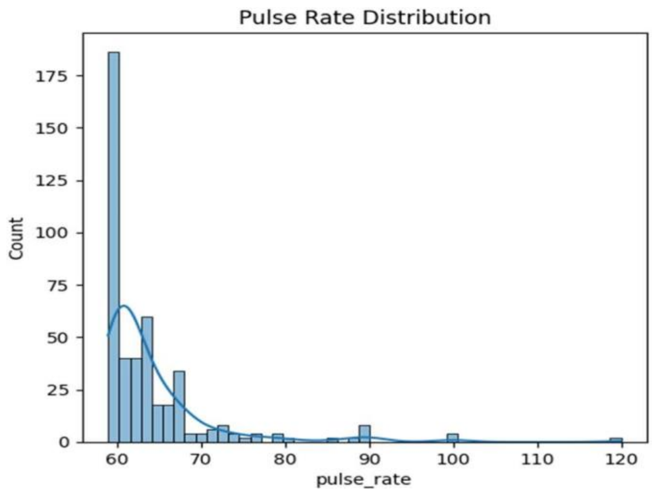


Figure 20. Pulse Rate Distribution

This image represents a histogram of the dataset and its 'Pulse Rate' values. A histogram is a way of showing the frequency or count of data points falling into certain ranges, or "bins". The x-axis shows the range of pulse rates, and the y-axis shows how many patients fall into each range.
From this histogram, you can identify:

- The shape of the distribution (e.g., symmetrical, skewed).
- The most frequent pulse rate values (the tallest bars).
- The range of pulse rates covered in this dataset.

This is useful to understand what are the typical pulse rates of the patients in this dataset and to see if there are any anomalies.

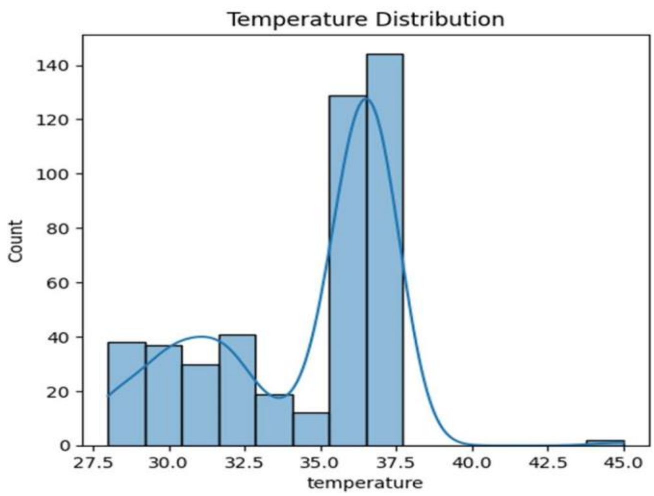


Figure 21. Temperature Distribution

Consider this histogram a visual means of counting patients with different temperatures.

- The line on the bottom of the histogram (the horizontal axis) is like a temperature scale that outlines the varying temperatures.
- The line on the left side of the histogram (the vertical axis) is the count of patients that fit into each temperature category.
- The bars are the counts themselves – larger bars mean more patients had temperatures in that category.

Ultimately, this picture provides you a quick overview of the more common temperatures among your patients and the variability in temperatures among your patients. You can easily see the clustering of patient temperatures, as well as any spikes or valleys in the counts. In summary, this gives you a sense of the variability of temperature within this dataset as well as a sense of the typical temperature.

4.1 Patient Healthcare Portal

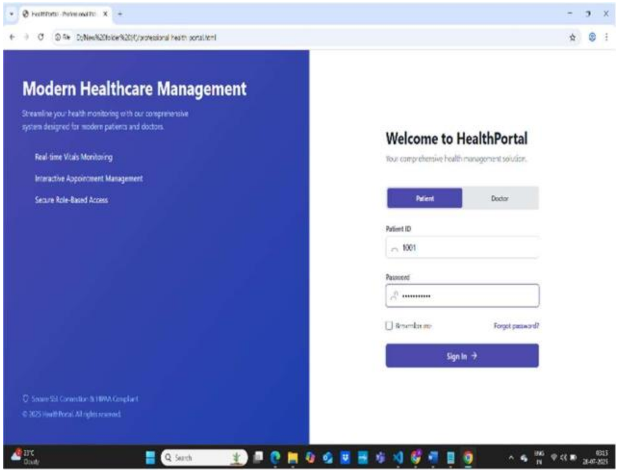


Figure 22. Patient Healthcare Dashboard

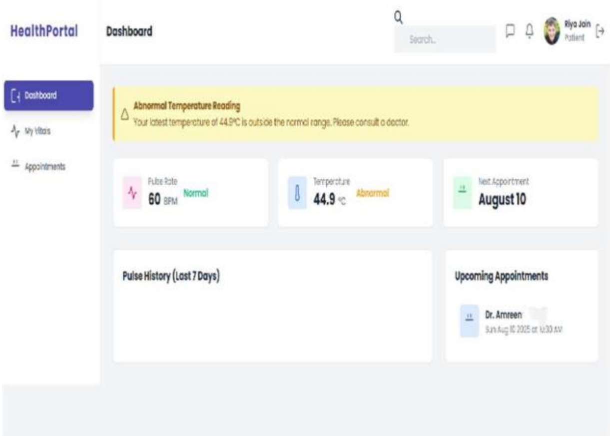


Figure 23. Abnormal Condition of Patient

This image shows the patient's personal health dashboard, which acts as the patient's base command center once logged in. The interface is designed to be clear and straightforward so that the patient, Riya Jain, gains insight into her current health status immediately. The most impactful information appears first. The banner at the top, nearly impossible to miss, signals an "Abnormal Pulse Reading" which is important to address promptly for matters of the patient's health. Just below this vastly noticeable banner is a group of cards containing Riya's key vitals, Pulse Rate, and Temperature, both identified as "abnormal," as well as where Riya's next appointment is located.

The dashboard provides context for Pulse Rate in the form of a "Pulse History" chart that visualizes Riya's rate over the past week. This, coupled with the list of "Upcoming appointments," provides the dashboard with the quality of addition, converting it from a visualization of the patient's health data into an interactive device that helps the patient self-manage aspects of health.

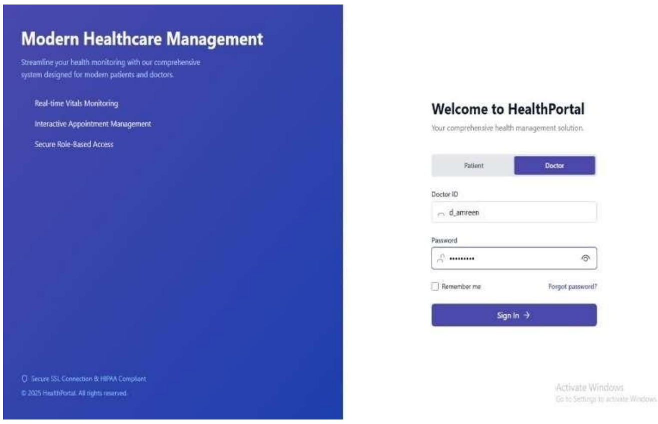


Figure 24. Doctor Login Dashboard

This image shows the instant where Dr. Amreen pulls a specific patient file in greater detail. After selecting Riya Jain from her main list of patients, she is shown an extensive individual dashboard. An important design principle at work here: the doctor is seeing the exact same interface as the patient sees. It is important that information is replicated for effective remote healthcare practice; creating a mutual context ensures that both the doctor and patient are relying on the same information (and data) in the meeting.

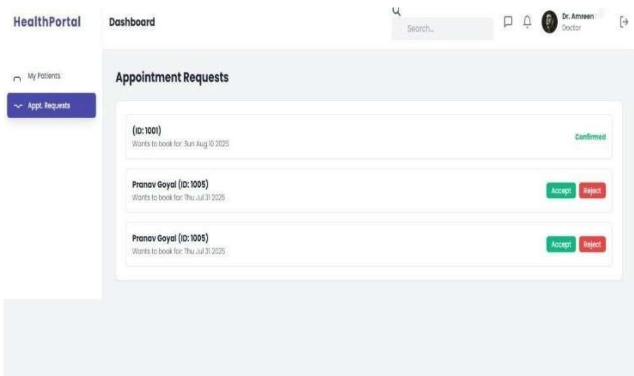


Figure 25. Doctor Appointment Dashboard

Dr. Amreen can see Riya's vital alerts at the moment, her health statistics over time, and her historical data charts. And, for convenience, there is a "Back to Patient List" button at the top which allows the doctor to zoom out of one case and back to her own complete view of past patients to maintain an efficient clinical workflow.

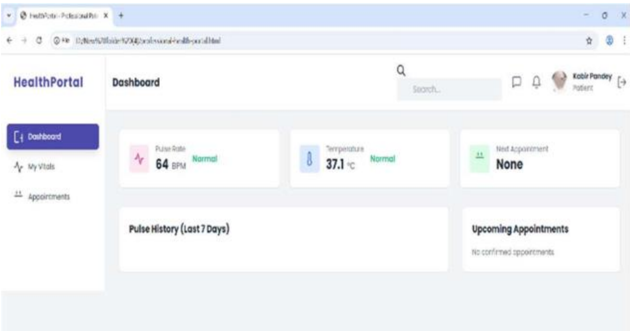


Figure 26. Normal Patient Dashboard

This figure highlights the Health Portal's dashboard as it exists for a patient called "Kabir Pandey," who exhibits stable health data with normal parameters. The interface provides an easy-to-understand summary that comforts and educates the patient equally during wellness and during an alert. Importantly, the screen has no warning banners. It goes to show one of the basic system design tenets: communication of contextual information.

The portal spares the patient from going into unnecessary anxiety by reserving alerts for instances of vital signs' appearance outside the range of normal. When an alert appears, the second it comes should be just the right moment for presentation, neither more urgent nor less.Immediate positive confirmation is provided by the primary-statistical cards:

- The Pulse Rate reads steadily at 64 BPM.
- The Temperature reads standard at 37.1°C.

Both readings are explicitly labeled as "Normal," which is reassuring. In addition, the dashboard accurately reflects the user's agenda, indicating "None" for the next appointment. The UI, as a result, is minimal and calming, comforting the patient. This view highlights the system's balanced approach, showing the value in not only suggesting possible diseases but also affirming and validating the health status of a patient.

5. CONCLUSION AND FUTURE SCOPE

In this study, one of the basic challenges at the center of modern healthcare is how one can check up on a patient's health When they are out of the doctor's office. For others, loopholes in checking up can transform simple issues of Health into full-blown crises. Fortunately, we are in the midst of a technological revolution, and the Internet Of Things (IoT) promises a new dawn. Smart devices are constructing a link between patients and doctors, Connecting and proactively enabling healthcare. Our project takes this promise a step further with an accessible Health Monitoring System. It employs basic Sensors to monitor an individual's heart rate and temperature, sending this critical data to a Portal or on their Phone and a safe online site. What's most important, though, is that if the readings indicate an emergency, the System does not wish to install fear within it immediately notifies a doctor so they can respond quickly and resolutely. Health Portal also keeps patients in control of their well-being by tracking Appointments, monitoring their progress in simple-to-understand graphs, and having their health history all in one location. We think this system is more than technology; it's a move towards a future in which health care is more Personal, responsive, and efficient for all. Future developments in this system can be divided into two broad areas: response time and increasing Diagnostic capability.

First, to maximize the clinical response process, a voice notification through sound can be incorporated into the physician's smartphone app. This would be in addition to the visual alert, ensuring that significant status Changes in a patient are promptly and explicitly conveyed, thus lessening the risk of delay. Second, to add more of an overall health picture, the sensor array on the device could be greatly extended. Adding sensors for blood pressure and SpO2 (blood oxygen saturation) would give a greater picture of a Patient's cardiovascular and respiratory system condition. For sophisticated cardiac monitoring, the addcardia Of an ECG sensor would be a significant addition, allowing for the detection and analysis of cardiac Arrhythmias. Furthermore, an environmental humidity sensor might provide useful contextual information, especially for patients with respiratory diseases that are sensitive to changes in the atmosphere.

REFERENCES:

[1] Iqbal W, Abbas H, Daneshmand M, Rauf B and Bangash Y A 2020 An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security IEEE Internet of Things Journal 7(10) 10250– 10276

[2] Khattak H A, Shah M A, Khan S, Ali I and Imran M 2019 Perception layer security in the Internet of Things Future Generation Computer Systems 100 144–164

[3] Javed B, Iqbal M W and Abbas H 2017 Internet of things (IoT) design considerations for developers

and manufacturers Proc. 2017 IEEE International Conference on Communications Workshops (ICC Workshops) pp 834–839
[4] Archip A, Nicolae B, Elena Ș, Herghelegiu P C and Zală A 2016 An IoT based system for remote patient monitoring Proc. 17th International Carpathian Control Conference (ICCC) (Slovakia: IEEE)
[5] Kumar R and Pallikonda Rajasekaran M 2016 An IoT based patient monitoring system using raspberry Pi Proc. International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16) (India: IEEE)
[6] Stojmenovic I, Wen S, Huang X and Luan H 2016 An overview of fog computing and its security issues Concurrency and Computation: Practice and Experience 28(10) 2991–3005
[7] Khan S, Iqbal W, Waheed A, Mehmood G, Khan S, Zareei M and Biswal R R 2022 An efficient and secure revocation-enabled attribute-based access control for e-health in smart society Sensors 22(1) 336
[8] Anawar M R, Wang S, Zia M A, Jadoon A K, Akram U and Raza S 2018 Fog computing: An overview of big IoT data analytics Wireless Communications and Mobile Computing 2018
[9] Kraemer F A, Braten A E, Tamkittikhun N and Palma D 2017 Fog computing in healthcare–a review and discussion IEEE Access 5 9206–9222
[10] Stantchev V, Barnawi A, Ghulam S, Schubert J and Tamm G 2015 Smart items, fog and cloud computing as enablers of servitization in healthcare Sensors & Transducers 185(2) 121–128
[11] Meliá S et al 2021 MoSIoT: Modeling and Simulating IoT Healthcare-Monitoring Systems for People with Disabilities International Journal of Environmental Research and Public Health 18(12)
[12] Philip N Y et al 2021 Internet of Things for In- Home Health Monitoring Systems: Current Advances, Challenges and Future Directions IEEE Journal on Selected Areas in Communications 39(2) 300-310
[13] Gupta, P., Shukla, M., Arya, N. and Singh, U., 2023. 5 Internet Smart and of Intelligent Things in Healthcare Systems. IoT in Healthcare Systems: Applications, Benefits, Challenges, and Case Studies, p.77.
[14] Jimenez F and Romina T 2015 Building an IoT- aware healthcare monitoring system Proc. 34th International Conference of the Chilean Computer Science Society (SCCC) (Chile: IEEE)
[15] Banerjee S and Subhajit R 2016 Design of A Photo Plethysmography Based Pulse Rate Detector International Journal of Recent Technology and Engineering 2 302-306
[16] da Costa C A et al 2018 Internet of Health Things: Toward Intelligent Vital Signs Monitoring in Hospital Wards Artificial Intelligence in Medicine 89 61-69
[17] Abdulmalek S et al 2022 IoT-Based Healthcare- Monitoring System towards Improving Quality of Life: A Review Healthcare 10(10) 1993
[18] Acharya A D and Patil S N 2020 IoT Based Health Care Monitoring Kit Proc. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) (Erode, India) pp 363-368
[19] Sarmah S S 2020 An efficient IoT-based patient monitoring and heart disease prediction system using deep learning modified neural network IEEE Access 8 135784-135797
[20] Hamim M et al 2019 IoT Based Remote Health Monitoring System for Patients and Elderly People Proc. 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (Dhaka, Bangladesh) pp 533-538
[21] Gregoski M J et al 2012 Development and Validation of a Smartphone Heart Rate Acquisition Application for Health Promotion and Wellness Telehealth Applications International Journal of Telemedicine and Applications 2012
[22] Rahman A, Rahman T, Ghani N H, Hossain S and Ud-din J 2019 IoT Based Patient Monitoring System Using ECG Sensor Proc. International Conference on Robotics, Electrical and Processing Techniques (ICREST)
[23] Yew H T, Ng M F, Ping S Z, Chung S K, Chekima A and Dargham J A 2020 IoT Based Real-Time Remote Patient Monitoring System Proc. 16th IEEE International Colloquium on Signal Processing and Its Applications (CSPA) (Malaysia: IEEE)

[24] Gupta, P., Varma, S., Arya, N. and Singh, U., 2023. IoT-Based Smart Chair for Healthcare Supporting System. In Intelligent Sensor Node-Based Systems (pp. 39-52). Apple Academic Press
[25] Pragaspathy S, Saravanan S, Kalaiyarasi M, Karunanithi K, Karthi S and Kadali K S 2022 IoT- based Patient Monitoring Healthcare System IoT and Analytics for Sensor Networks (Singapore: Springer)
[26] Gabriel M M and Kamweru P K 2020 Arduino uno, ultrasonic sensor HC-SR04 motion detector with display of distance in the LCD International Journal of Engineering Research and Technical Research 9(5) 936-942
[27] Gupta, P., Varma, S., Arya, N. and Bhagel, R., 2023. Intelligent Security System Based on the Internet of Things (IoT). In Intelligent Sensor Node-Based Systems (pp. 177-191). Apple Academic Press
[28] Al Dahoud A and Fezari M 2018 Arduino, WSN applications, Integrated Development Environment (or “IDE”)
[29] Pasha S 2016 Sensor and monitoring system for the Internet of Things based on Thing Speak and Matlab Analysis International Journal of New Technology and Research (IJNTR) 2(6) 19–23
[30] Badamasi A Y 2014 Arduino Working Principal Proc. 11th International Conference on Electronics, Computers, and Computers (ICECCO) (Nigeria: IEEE)