Internet of Things Based Person Health Monitoring and Detection Using Machine Learning

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Internet of Thing and Sensor System

by

Prakash P

21MTS0017

Under the guidance of

Dr. Jaffino G

Assistant Professor (Sr)

Department of Sensor and Biomedical Engineering

School of Electronics Engineering,

Vellore Institute of Technology, Vellore.

April 2023

DECLARATION

I hereby declare that the thesis entitled "Internet of Things based Person Health Monitoring and Detection using Machine Learning" submitted by me, for the award of the degree of *Master of Technology in Internet of Things and Sensor System* to VIT University is a record of bonafide work carried out by me under the supervision of Dr. Jaffino G.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: VELLORE
Date: 28-04-2023

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Internet of Things Based Person Health Monitoring and Detection using Machine Learning" submitted by **Prakash P**, School of Electronics Engineering, VIT University, for the award of the degree of *Master of Technology in Internet of Things and Sensor System*, is a record of bonafide work carried out by him under my supervision, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date : 28-04-2023

Signature of the Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

Head of the Department

ABSTRACT

With increasing growth of new healthcare technology IoT is rapidly revolutionizing the healthcare industry. Nowadays there are lot of Internet of Things devices which is used to monitor the health over internet. By using these smart devices doctors and caretakers also taking advantages to keep an eye on the health conditions of the patients. In this work the proposed health monitoring of a patient using IoT based sensors which records the patient heartrate, blood oxygen using MAX30102 sensor, body temperature using LM35 sensor, room temperature and room humidity using DHT11 sensor and quality of air using MQ135. ESP32 Microcontroller board which controls these sensors sends these sensed values to the cloud and these sensed values can be viewed through our created website using various devices. The sensed values are also displayed in 16x2 i2c LCD display module.

If the Heath condition is normal/abnormal it sends emergency message to the doctors/caretakers. The timely detection of abnormal health conditions is crucial for effective medical intervention and improved patient outcomes. This research proposes a machine learning-based approach to differentiate between normal and abnormal health conditions using six distinct algorithms, including Support Vector Machines, Random Forest, Gradient Boosting, Decision Tree, Naïve Bayes, and Multilayer Perceptron neural networks. Real-time health monitoring data is collected from MAX 30102 and LM35 sensors, which record various vital signs, including heart rate, blood oxygen and body temperature. The sensed values from these sensors are exported to ThingSpeak, a cloud-based platform for the Internet of Things, and used in the analysis. Furthermore, publicly available datasets that contain various health conditions and their corresponding features are used in this research.

Preprocessing techniques such as removing missing values and outliers are used to prepare the datasets. To ensure consistency, the features are standardized to the same scale. The dataset is then divided into training and testing sets as 70:30, 30:70, 50:50, and 90:10 ratios. Each of the six algorithms is trained and tested on both training and testing sets using cross-validation techniques.

The hyperparameters for each algorithm are optimized to obtain optimal performance, and various metrics such as accuracy, precision, recall, F1 score, and Receiver Operating Characteristics curves are employed to evaluate algorithm performance. The Gradient Boosting outperforms other algorithms, achieving an accuracy of 100% on the 70:30 dataset ratio, 100% on the 50:50 dataset ratio, 96% on the 30:70 dataset ratio, and 100% on the 90:10 dataset ratio. Moreover, this algorithm shows significant potential for real-time health condition classification using sensor data. The proposed approach has potential to assist medical professionals in early detection of abnormal health conditions using real-time sensor data. The findings of this research can contribute to the development of automated systems for health condition classification and early detection of abnormal health conditions such as in real-time monitoring scenarios.

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to our beloved **Chancellor Dr. G. Viswanathan** for proving necessary facilities to carry out and finish the project successfully. I am grateful to all our **Vice Presidents** and our **Vice Chancellor** for their support and encouragement.

My special thank goes to **Dr. Sivanandham, Dean, SENSE** and **Dr. Vidhya S, HOD, Department of Sensor and Biomedical Technology, SENSE for** providing an environment that encouraged me in working towards my goal and supported in completion of my project. This project gave me an opportunity to apply creative and critical thinking skills.

I would like to present my deep sense of gratitude to my project guide, **Dr. Jaffino G, Assistant Professor (Sr), SENSE** who has always been a source of motivation and firm support for carrying out the project. The supervision and support that he gave truly helped in the progression of the project. His cooperation is much indeed appreciated. I am highly obliged to him for his valuable advices and moral support during research period.

Finally, my greatest and special gratitude goes to my family for their love and support.

Place: VELLORE
Date: 28-04-2023

Prakash P

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

List of Abbreviations

IoT	Internet of Things
PHMS	Person Health Monitoring System
ESP	Express if Systems Processor
ML	Machine Learning
MAX	Maxim Integrated Model
DHT	Digital Humidity Temperature
MQ	Module Quality
Wi-Fi	Wireless Fidelity
IFTTT	If this then then that
AI	Artificial Intelligence
GB	Gradient Boosting
SVM	Support Vector Machine
DT	Decision Tree
NB	Naïve Bayes
MLP	Multilayer Perceptron
RF	Random Forest

Chapter 1

1. Introduction

1.1 Motivation

The significance of the internet of things (IoT) system depends on the correct analysis and interpretation of the data generated. The real time data and analyzing the data to uncover patterns is some of the primary ways to help improving the patient outcomes by IoT based patient health monitoring. Therefore, transmitting the data to the cloud for analytics using appropriate tools and applications is vital. To explore wireless and wearable sensor - based monitoring systems and categorize the different types of sensors used in health monitoring and also to highlight the challenges and open issues related to healthcare security, privacy, and quality of service (QoS) [13]. The IoT works in tandem with other technologies such as Wireless Sensor Network (WSN), REST, and other protocols, including smart mobile devices, radio frequency data and these technologies work together to improve healthcare outcomes and provide more efficient and effective care for patients [14].

One major advantage of incorporating IoT into healthcare procedures is the significant reduction in the time taken to access a patient's medical records. With IoT- based mobile health care systems, clinicians can access a patient's data and medical history remotely, even before the patient reaches the hospital. This is especially useful in situations where medical institutions are overwhelmed with patients or when emergency medical care is required and there is a shortage of doctor health condition of a person or patient can be monitored by doctors and medical staff and provide timely medical care, even if they are not physically present. This technology helps to improve the efficiency of medical care and enables patients to receive timely and appropriate treatment, regardless of their location [12]. In the health care industry, the timely detection of abnormal health conditions is of utmost importance, as it can significantly impact patient outcomes and healthcare costs.

The integration of IoT technology has revolutionized the healthcare industry, providing new tools and perspectives to improve the delivery of health care solutions. These applications range from aiding in disease detection, such as diabetes and cancer detection, to improving pathology classification, including radiology scans and electrocardiogram interpretations for cardiac analysis, and forecasting diseases using Machine Learning and Deep Learning algorithms, which have been developed to combat the COVID-19 pandemic. However, despite substantial investments made in technological advancements by the healthcare industry, AI deployment and integration in healthcare remain in the nascent stages. The pressing concerns of workforce scarcity and exhaustion, as well as the shift towards long-term disease management, underscore the urgent need for AI to significantly augment healthcare infrastructure through its broad applicability [6].

1.2 Scope and Objective

This approach generates valuable data that can facilitate informed decisions. Multiple benefits are expected from this device. Consequently, the healthcare industry is exploring such systems to enhance patient care. Monitoring the health conditions of the patients safely outside the hospitals by the doctors is possible by the IoT based devices. With rapid growth in technology industry like wireless technology, sensors, cybersecurity protocols that made possible for sensing the patients real time monitoring of the body conditions. Healthcare providers can access and share their patient data at the exact moment they need to do by using the cloud computing. By these improved medical technologies this innovation helps to reduce the cost, time spent in hospital, increasing the comfort and also manage to improve their own health.

With the advent of sensor technology, health monitoring applications have become more sophisticated and effective, as sensors are capable of detecting and measuring various physical quantities, such as heart rate, blood oxygen levels, and body temperature, which are essential indicators of an individual's health status. Moreover, the integration of machine learning algorithms with sensor technology has paved the way for a more efficient and accurate approach towards detecting abnormal health conditions. Machine learning algorithms can analyze large datasets of sensor data and generate insights that would be difficult to obtain using traditional statistical methods. They can learn patterns and trends from the data and provide predictions with a high degree of accuracy. The primary dataset was used to conduct an experiment utilizing various machine learning techniques on selected symptoms. The performance of these algorithms was then evaluated by calculating metrics such as accuracy, precision, recall, F1 score, root-mean-square error, and area under the curve score [26].

The health care providers to collect more data from the patient than before effectively by using the IoT based health devices. To develop software for uncovering the patterns and trends of the patient data can improve the treatment and diagnosis. Microcontroller that senses the real time data of a patient using sensors and will be sent to the cloud. After sending the sensor values to the cloud it will be send it to the webserver. Doctors, caretakers can view these sensed values in their created website and mobile app. By using the IFTTT Platform it can send SMS to the doctors and caretakers with emergency alert if the patient body condition is abnormal or critical. With IoT-enabled devices and applications, we can now automate various healthcare procedures that were once time-consuming and prone to human error. For instance, networked devices can now control temperature and airflow in operating rooms, resulting in a more efficient and controlled environment. and enabled an integrated healthcare network [4]. Early detection of abnormal health conditions is critical for effective medical intervention and improved patient outcomes. To address this need, a study proposes a machine learning based approach using six distinct algorithms, including Support Vector Machines, Random Forest, Gradient Boosting, Decision Tree, Naïve Bayes, and Multilayer Perceptron neural networks. Real-time health monitoring data is collected from MAX 30102 and LM35 sensors, recording vital signs such as heartrate, blood oxygen, and body temperature.

This data is exported to ThingSpeak, an IoT cloud-based platform, and publicly available datasets are utilized to differentiate between normal and abnormal health conditions. The datasets undergo preprocessing techniques, such as missing values and outliers, and standardized to ensure consistency. These advancements have greatly enhanced the quality of healthcare early detection of abnormal health conditions is critical for effective medical intervention and improved patient outcomes. To address this need, a study proposes a machine learning based approach using six distinct algorithms, including Support Vector Machines, Random Forest, Gradient Boosting, Decision Tree, Naïve Bayes, and Multilayer Perceptron neural networks. Real-time health monitoring data is collected from MAX 30102 and LM35 sensors, recording vital signs such as heartrate, blood oxygen, and body temperature. This data is exported to ThingSpeak, an IoT cloud-based platform, and publicly available datasets are utilized to differentiate between normal and abnormal health conditions. The datasets undergo preprocessing techniques, such as missing values and outliers, and standardized to ensure consistency. The global increase in health care that utilizes Internet of Things of its market size and forecast yearly is shown in the Fig 1.1

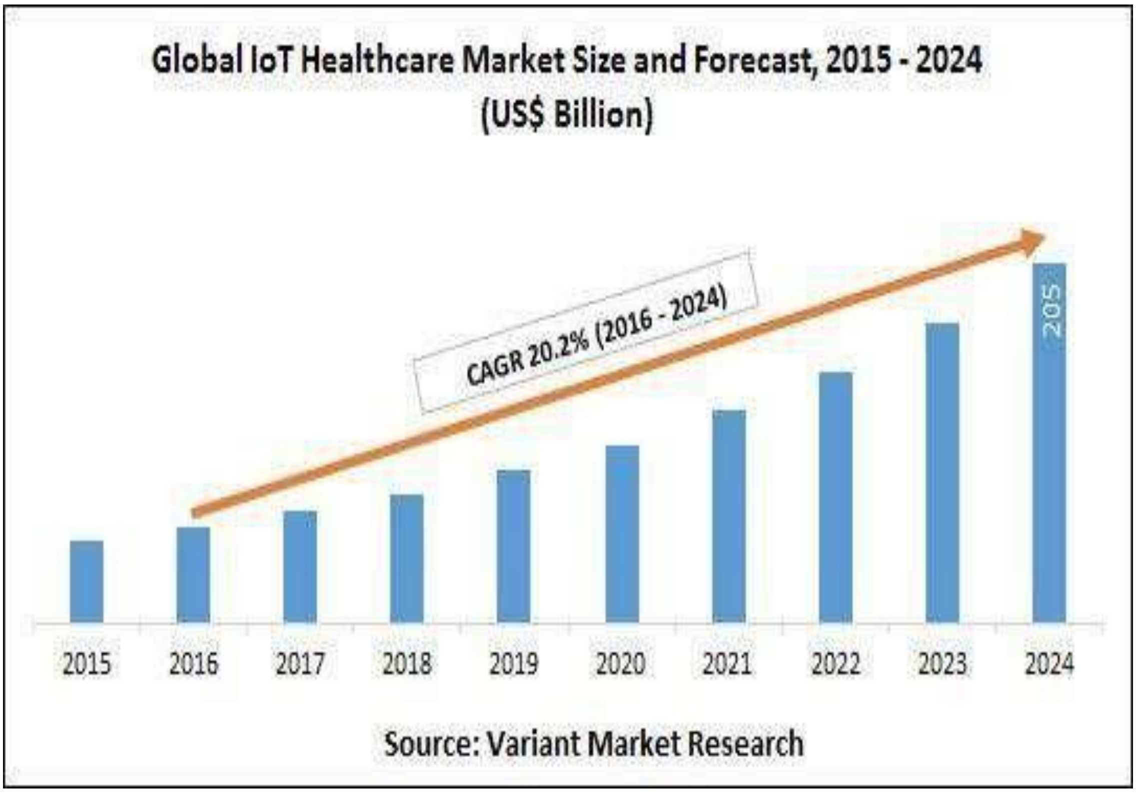


Fig 1.1 Global IoT in Health care Market growth

Objectives:

1. Development of Person Health Monitoring System using Internet of Things.

2. Development of Person Health Condition Detection using Machine Learning

1.3 Problem Definition

The current global health crisis has highlighted the importance of leveraging technology to its fullest extent. Among the most promising technologies are the Internet of Things (IoT) and machine learning (ML), which have shown significant potential in the fight against the coronavirus pandemic. The IoT is a network of devices that can sense and transmit valuable data about the environment, and machine learning can analyze this data to provide insights and solutions [27]. Maintaining good health is crucial for individuals to lead a productive life. However, due to various factors, such as sedentary lifestyle, poor eating habits, and genetic predisposition, individuals may develop various health conditions. Early detection of these conditions is important to prevent their progression and improve the quality of life of individuals [28].

However, there are several barriers to the effective use of these technologies in healthcare. For instance, there may be concerns about data privacy, and security, as well as issues related to the interoperability and standardization of devices and data formats. Moreover, there may be challenges related to the training and deployment of ML models, as well as the need for skilled personnel who can interpret and act upon the insights generated by these models. To address these challenges, it is essential to build a robust and integrated healthcare ecosystem that can leverage the full potential of IoT and ML technologies. This ecosystem should include a variety of stakeholders, including healthcare providers, patients, technology companies, and regulators, who can collaborate to develop and implement innovative solutions that can improve health outcomes and reduce healthcare costs. The problem at hand is the need for a innovative and integrated healthcare ecosystem that can leverage the full potential of IoT and ML technologies to address the challenges we face, particularly in the context of COVID-19 pandemic. By working together to develop and implement effective solutions, we can improve health outcomes, reduce healthcare costs, and ultimately enhance the quality of life for individuals and communities around the world.

Artificial Intelligence made its debut in the medical sector in 1976, when a computer algorithm was employed to determine the underlying causes of severe abdominal pain. Since, then a multitude of AI applications have been introduced in healthcare, aimed at augmenting the strengths and addressing the limitations of the existing medical infrastructure. These applications range from aiding in disease detection, such as diabetes and cancer detection, to improving pathology classification, including radiology scans and electrocardiogram interpretations for cardiac analysis, and forecasting diseases using Machine Learning and Deep Learning algorithms, which have been developed to combat the COVID-19 pandemic. However, despite substantial investments made in technological advancements by the healthcare industry, AI deployment and integration in healthcare remain in the nascent stages. The pressing concerns of workforce scarcity and exhaustion, as well as the shift towards long-term disease management, underscore the urgent need for AI to significantly augment healthcare infrastructure through its broad applicability [29]. It is essential to build a robust and integrated ecosystem that can support their deployment and integration. However, the deployment and integration of these technologies in healthcare remain in the nascent stages, and it is essential to build a robust and integrated ecosystem that can support their adaptation and implementation [30].

1.4 Existing Technologies

Monitoring or observance of the patient health using Internet of Things Temperature, Heart rate, Blood oxygen of a patient and parameters like Room Humidity, Room Temperature and toxicity level of environment control are additionally implemented. This method will generate emergency alert once when they arise an abnormal condition in patient based on sensing of the real time different sensors and different modules for acting a function of unique variety [1].

Asthma health monitoring of patient is designed for monitoring the health condition of asthma patient where MAX 30102 is utilized to sense heart rate and oxygen saturation, while the DHT11 is used to monitor room humidity and temperature, air quality using MQ130, nostril temperature using LM35 controlled by an ESP8266 microcontroller. All these sensed data are sent to Firebase through Wi-Fi and doctors can view the sensed values using the created website and mobile application through any devices. Using this method patients can consult with the doctor and get a prescription through video calling feature. This system is designed in such a way that patients are freed from visiting the doctor by going to hospitals over and over again [2].

To visualize the patient health as to issue the major concerns of the health such as cardiovascular diseases and heart attacks. A Wi-Fi module enabled microcontroller ESP8266 and sensors that are wearable are measured and heart rate, blood oxygen, body temperature and ECG with reliable data of different health parameters are sensed and are uploaded to the cloud where application created will be more user friendly and helpful for the medical professionals where they can visualize or remotely access these data anywhere at any time from any location [3]. To monitor the health using Arduino uno microcontroller board along with MAX30102 heart rate sensor for sensing the heart rate, blood oxygen, 16x2 LCD Display module that displays the sensed values in the LCD Display, ESP8266 Wi-Fi module for sending these sensed values to the cloud. This Paper portrays predicting based on the patient sensed values in the real time data streams in the cloud for accurate predictions using various machine learning algorithms [4].

They address the three key factors for monitoring each individual health as the adaptation of technology acceptance in health care providers for the patients. First key factor that discuss about the adoption of Internet of Things based in-home remote monitoring is a major factor driving its growth. Second key factor that present the latest advances and key building blocks of IoT-based in-home monitoring are significant factors driving its growth. Third discuss about analyzing the potential of in -home health monitoring with IoT and offering recommendations for future development [5]. In this work proposes real time monitoring of an automated pain assessment system that utilizes facial expressions for analysis and bio potential monitoring of scalable Internet of Things system developed. Wireless sensor node is integrated into Internet of Things that can be utilized as wearable devices. A wearable device featuring a bio-sensing facial mask is proposed for monitoring a patient pain intensity along with facial surface electromyogram (SEMG) and up to 8 channels is sampled in the sensor node. The frequency range of the SEMG signal is sampled and transmitted to a cloud server. The created website and mobile application process and visualize real-time pain data collected from the wearable device used for monitoring [6].

By utilizing IoT technology, the researchers developed a system to monitor air quality and used LTE to transmit the collected data to a webserver. The various parameters such as VOC, CO, CO2, and humidity-temperature of air quality can be monitored. The webserver is integrated along with the cloud computing for analyzing the data and air quality can be monitored. The cloud that stores all the data from the webserver and further quality of air can be analyzed. [7]. In this work they proposed a system with Internet of Things and Machine Learning based system for managing and monitoring the physical and healthcare activities using smart wearable technology. This IoT- based framework is wireless, smart, and wearable, and utilizes fog computing to analyze health-related data as well as bodily movement data in real-time. Heart rate, electrocardiogram, breath rate can be determined using a 3D-acceleration. A module for gym activity recognition (GAR) has been developed to track the body's vital signs and movements during exercise in real-time. Health hazard alarming and identification is responsible by health zone module [8].

They designed the real time monitoring of human health due to the diseases suffered by the patients. By using wireless communication module and set of medical sensors is connected with the embedded ARM microcontroller. ARM microcontroller checks the health condition of the patient and to save the life of a patient by analyzing the scan. The proteus 8 professional software and through embedded C software [9]. PPG signals are used by pulse oximeters to measure the variation in blood volume in tissues. This is a non-invasive technique that involves using a light source and a detector. There are two types of PPG: transmittance and reflectance. In transmittance PPG, the light source is placed on one side of the tissue and the detector is placed on the other side to measure the variation in the tissue. However, this type of PPG can only be used in small volume organs such as ear lobes and fingers. On the other hand, reflectance PPG uses both the light source and detector on the same side of the tissue. The light source emits light into the tissue, and the detector measures the variation in the reflected light. Reflectance PPG can be used on any part of the body, making it a versatile option for pulse oximetry measurements [18].

Wireless healthcare monitoring devices have been utilized with the intension of integrating them with artificial intelligence technology. The monitoring process involves the use of neural networks and a fuzzy system, which is supported by an increased number of sensor nodes for collecting diverse health information in a secure manner. The collected patient data is then transmitted through a GSM module to Azure IoT, where the raw data is transformed into a linguistic representation. A fuzzy-based inference system (FBIS), which has been trained with the help of a logic-based algorithm, is responsible for monitoring the patient's health condition in a more reliable and accurate manner. This system is designed to securely send the patient health status to medical experts, thereby enhancing the security and reliability of the patient's health data. [10].

Active research has been conducted in recent years on the utilization of artificial intelligence and machine learning techniques in healthcare, presenting promising opportunities in the monitoring of human activities and vital signs through wearable devices. This technology has great potential for assisting in the diagnosis of diseases and can greatly aid in elderly care and patient health monitoring and diagnostics. Recent advancements in medical sensors and the miniaturization of electronic chips have led to the development of more wearable device applications [32].

The advent of Industry 5.0 and 5G technology has paved the way for the development of intelligent sensorsthat are bothefficient and affordable, enabling real-time health monitoring for individuals. This has revolutionized the provision of healthcare services, making it possible to provide fast, reliable, and cost-effective health monitoring from remote locations, which was not possible before. Additionally, the integration of blockchain frameworks has enhanced the security and privacy of patients' confidential data, preventing any misuse of the data. The use of deep learning and machine learning techniques has also played a significant role in analyzing health data, leading to the achievement of various targets and promoting preventive healthcare and fatality management in patients. Moreover, cloud computing and storage have been integrated to make the services more efficient and real-time. This review comprehensively examines the advancements and challenges in structural health monitoring [33].

The emergence ofthe Internet of Things (IoT) has paved the wayfor the development of a plethora of technologies aimed at preventing chronic diseases, especially those that involve continuous and real-time monitoring systems. Wearable medicaldevices equipped withsensors, health clouds, and mobile applications have generated an enormous amount of streaming big data. However, the rapid pace at which this data is generated poses a challenge for its real-time collection, processing, and analysis, which can impede prompt action in emergency situations and make it difficult to extract valuable insights through conventional methods that are often time-consuming. Therefore, there is a pressing need for an efficient and scalable solution that can process real-time big data streams. This study proposesa novelarchitecture for a real-time healthstatuspredictionand analytics systemthat leverages big datatechnologies to addressthis challenge [34].

A thorough investigation was conducted to explore the available technologies for monitoring heart-related diseases. The analysis revealed that the collected raw data is contaminated with irrelevant and erroneous contents, commonly known as noise. This noise and significant variation in the data adversely affect the accuracy, sensitivity and precision of the classification process, rendering the data useless for diagnosis. To overcome this challenge, a novelpre-processing technique is proposed in this paper that effectively eliminates noise and irrelevant data from electrocardiogram signals [35]. Technologyhas made significant advancements in the field ofmedicine, providing doctorswitha range of high-tech tools to aid them in diagnosis and treatment. It has greatly reduced the time and effort required for doctors to analyze diseased cells within the human body. As a result, disease diagnosis equipment based on machine learning has become increasingly sophisticated, efficient, and intelligent in their design and manufacture [36].This has generated interest in using a data-driven approach to structural health monitoring, particularly due to the availability of historical data. However, a lack of sensor datafor different damage scenarios remains a significant challenge, leading to limited robustness and generalizability of most supervised machine-learning / deep-learning techniques. To address this issue, Physics-informed learning is proposed, which integrates domain knowledge into the learning process. In this regard, preliminary results from dynamic modelling of beam structures using physics-informed artificial neural networks are presented, demonstrating the superiority of this approach over purely data-driven techniques in automated damage detection. Several methods of incorporating domain knowledge into the machine learning pipeline through case studies, such as visual inspection and impact diagnosis in Non-destructive Inspection/Structural Health Monitoring [37]. To achieve precision medicine that positively impacts patient outcomes and offers real-time decision support, it is crucial to integrate disparate data sources and identify patient-specific patterns of disease progression using electronic health records. This requires analyticaltools, technologies, and databases that improve networking and interoperabilityofclinical, laboratory, and public healthsystems, while also addressing ethicaland social issues related to data privacyand protection.

By developing multifunctional machine learning platforms for clinical data extraction, aggregation, management, and analysis, clinicians can efficiently stratify subjects and optimize decision-making. The implementation of artificial intelligence in healthcare presents a compelling vision that can lead to significant improvements, providing real-time personalized and population medicine at a lower cost [38]. To timely identify health sensor data, a rapid detection strategy that utilizes machine learning - based approaches to detect code patterns. Our approach leverages to analyze the code extracted from health sensor data, which can be fed into these models as either sequences of bytes/tokens or single bytes/tokens. Some bytes of program data have been collected both normal and abnormal, that have been labelled for this purpose. The mainchallenges ofthistask include feature selection, modifying the three models for training and testing the health sensor data set, and evaluating the models and features [39]. The primary goal of this review is to assess the current literature on smart health monitoring, emphasizing evolving Machine learning based techniques and presenting readers with an overview of smart health monitoring's applications. Various sensor network can be used to gather data from a structure in various states and use guided waves, hierarchical non-linear primary component analysis and machine learning to detect and classify damage and also covers the implementation of vibration-based and vision-based surveillance, as well as recent developments in smart health monitoring [40]. The proposed method of our patient health monitoring and detection is described in flow chart as shown in Fig 1.2

1.5 Person Health Monitoring and its Detection Types

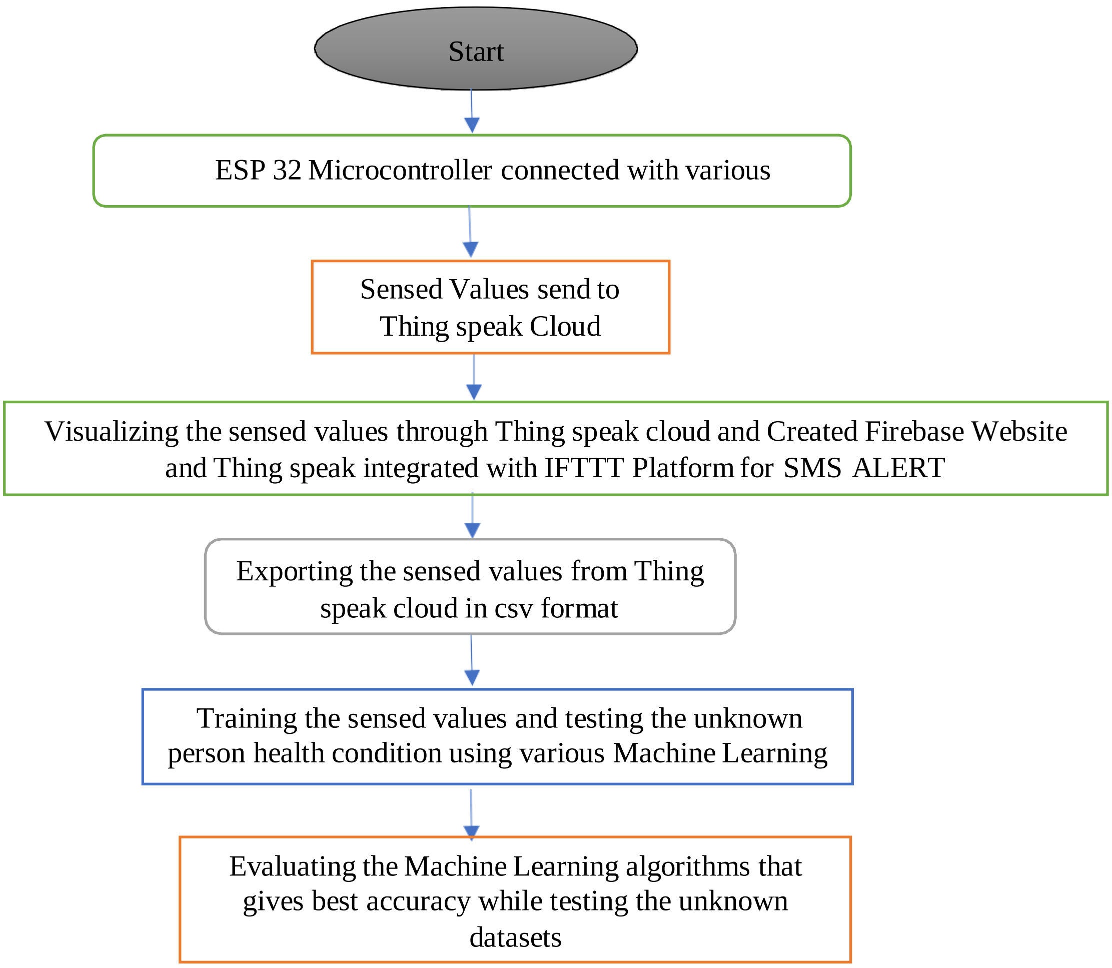


Fig 1.2 Flow Chart of our Proposed System

1.5.1. What is IoT?

The Internet of Things (IoT) refers to the network of physical devices, vehicles, buildings, and other items embedded with electronics, software, sensors, and connectivity that enables these objects to collect and exchange data. This connected network of devices has the ability to share data and make intelligent decisions based on that data, without the need for human intervention. IoT devices use a variety of communication protocols such as Wi-Fi, Bluetooth, ZigBee, and others to connect to the internet, which allows them to share data with other devices and systems. This data then be analyzed and used to make informed decisions, such as adjusting the temperature in a room based occupancy patterns, or alerting maintenance personnel to issues with a machine before it breaks down.

The applications of IoT are vast and varied, from human automation and smart cities to industrial automation and healthcare. For example, in the healthcare industry, IoT devices can monitor patients remotely and transmit vital signs to medical professionals in real-time. In the manufacturing industry, IoT devices can be used to track inventory, monitor production lines, and identify maintenance needs. One of the biggest challenges with IoT is the need for data security and privacy. With so many connected devices collecting and sharing data, there is a risk of sensitive information being compromised. This has led to the development of new security protocols and standard to ensure the production of data in IoT systems. Overall, the Internet of Things has the potential to transform many aspects of our lives, making them more efficient, productive, and enjoyable. However, as with any new technology, there are challenges and risks that must be addressed to ensure its successful implementation.

1.5.2. What is a Sensor?

Sensors are devices that are designed to detect changes in the physical environment and convert them into measurable signals that can be processed and analyzed by electronic systems. These signals can then be used to control, monitor, or analyze various aspects of the environment or a system. Sensors work by various physical phenomena, such as light, sound, temperature, pressure, and motion, to detect changes in the environment. They then convert these changes into an electrical signal that can be processed and analyzed by electronic systems, such as microprocessors, computers, or other control systems. There are many types of sensors, each designed to detect a specific type of physical phenomenon. For example, cameras and microphones are used to detect light and sound, respectively, while thermometers are used to detect changes in temperature. Other types of sensors include accelerometers, which detect changes in motion and acceleration, and pressure sensors, which detect changes in pressure.

Sensors are used in a wide range of applications, from consumer electronics, such as smartphones and fitness trackers, to industrial control systems and medical equipment. For example, sensors are used in automotive systems to monitor engine performance, detect changes in road conditions, and provide safety features such as anti-lock brakes and airbags. They are also used in medical equipment to monitor patient vital signs and provide feedback to medical professionals. Sensors are essential components of modern technology and are used to detect and measure changes in the physical environment. They convert these changes into electrical signals that can be processed and analyzed by electronic systems to provide real-time data and control various aspects of the system.

1.5.3. What is PTM?

Patient or person health monitoring involves tracking and evaluating a person's health over time using a variety of techniques and technologies to gather information on their symptoms, vital signs, and other important health indicators. Both passive and active monitoring techniques can be used to conduct this monitoring, which can take place in a range of locations such as residences, clinics, and hospitals. Health information is gathered by passive monitoring, which doesn't require the patient or person being observed to actively participate. Wearable gadgets, like smartwatches or fitness trackers, which can monitor a person's heart rate, activity level, sleep patterns, and other parameters, can be used for this type of monitoring. In some circumstances, passive monitoring could also make use of sensors or other devices used to monitor a person's movements or other behaviors and put in the environment, such as a room.

Contrarily, active monitoring calls for the subject to take action, such as taking a reading of their blood pressure, blood sugar, or other vital indicators and documenting it. Using specialized tools or gadgets, this form of monitoring is frequently carried either by medical experts or patients themselves. The purpose of patient or person health monitoring is to give patients and healthcare professionals a better awareness of the individual's health condition, allowing for earlier diagnosis and treatment of health issues as well as assisting people in making more informed healthcare decisions. Healthcare providers can spot potential health issues by routinely checking a person's health.

1.5.4. Description of Hardware Components

The various hardware components that we used is explained briefly as follows. Sure, here's a rephrased passage that conveys the same meaning but avoids plagiarism. The ESP32 as shown in Fig 1.3 is a microcontroller created by Espress if Systems, designed to be versatile, low-cost, and low-power. It is an improvement over the ESP8266 microcontroller, featuring faster processing power, connectivity, and input/output interfaces. One of the key features of the ESP32 is its dual-core processor, running at up to 240 MHz. This allows for efficient processing of data, making it a suitable option for applications that require real-time processing or complex computations.

Fig 1.3 ESP 32 Microcontroller

Additionally, the ESP32 is equipped with Wi-Fi and Bluetooth connectivity, which enables it to connect to the internet, other devices, and form local networks. As a result, it is an excellent choice for various applications such as industrial monitoring, smart farming, and home automation. The ESP32 also boasts a range of input/output interfaces, including SPI, I2C, UART, and PWM, which allow it to connect to sensors, actuators, and other electronic devices, making it a highly adaptable option for different applications. Furthermore, the ESP32 includes an integrated analog-to-digital converter (ADC), a digital-to-analog converter (DAC), and a low-noise amplifier (LNA) for radio-frequency (RF) applications. This makes it ideal for wireless data transfer, as well as applications that require accurate measurement of analog signals. Finally, the ESP32 is popular due to its affordability, versatility, and ease of use. It is well-supported by various development tools and software frameworks such as Micro Python and the Arduino IDE, making it a popular choice among hobbyists, makers, and professional developers alike. The specification of ESP 32 Microcontroller is described in table 1.1

Table 1.1: Specifications of ESP-32 WiFi Microcontroller Module

Specification	Explanation
Microcontroller	The ESP32 devkit module is powered by a dual-core Tensilica LX6 microcontroller with clock speeds of up to 240 MHz.
Connectivity	It comes with integrated Wi-Fi and Bluetooth connectivity supporting 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth 4.2 BR/EDR and BLE
Memory	The module has 520 KB SRAM and 4 MB flash memory
GPIO	There are 36 GPIO pins available for digital input/output, PWM, I2C, SPI, UART, and other purposes.
Analog Input	The module has 18 analog input pins, with a resolution of up to 12 bits for reading analog signals.

Power	The devkit module can be powered using a USB cable or an external power supply with a voltage range of 5-12V. It also includes a built-in voltage regulator and battery charging circuit.
Programming	The module can be programmed using the Arduino IDE or the ESP-IDF (ESP32 IoT Development Framework).
Other Features	The devkit module includes a built-in OLED display, microSD card slot, and support for OTA (Over-The-Air) updates.

MAX 30102 Sensor as shown in Fig. 1.4 of internal LEDs, photodetectors, optical elements, and low-noise electronics with ambient light rejection, as a biosensor module it combines pulse oximetry and heart rate monitoring functionalities. This sensor is designed to measure pulse oximetry (SpO2) and heart rate (BPM) signals using two LEDs (one infrared and one red), optimized optics, a photodetector, and low-noise analog signal processing.

Figure 1.4 MAX30102 Sensor

By detecting the amount of light reflected back to the photodetector using a single LED at a time, the device accurately calculates blood oxygen levels and heart rate. The MAX 30102 module has 8 pins for connectivity, including a Vin power pin that can connect to a microcontroller's 3.3V or 5V output, and I2C clock and data pins (SCL and SDA). The INT pin generates an interrupt for each pulse and can be programmed accordingly, while the IRD pin has an LED driver for SpO2 and heart rate measurements. The RD pin controls the red LED, and the GND pin serves as the ground connection. The MAX30102 can be fully customized through software registers, and its digital output data can be stored in a 32-deep FIFO buffer within the integrated circuit. This FIFO feature allows the device to be connected to a microcontroller or processor on a shared bus, where data can be saved and retrieved intermittently, without requiring continuous communication with the MAX30102's registers and this device compact size and superior performance make it an excellent choice for use in wearable devices and table 1 describes the technical specifications of MAX 30102 Sensor.

This time interval is then converted to beats per minute (BPM) using a simple formula.

Heart rate (BPM) = 60 / time interval between peaks in seconds

Most pulse oximeters will also display the waveform of the infrared signal, which can be used to visually confirm the accuracy of the heart rate measurement. Its crucial to note that pulse oximeters may not be reliable in all situations and that some medical conditions may affect the accuracy of the measurement. As with any medical device, its crucial to follow the manufacturer's instructions and consult with a healthcare professional if you have any questions or concerns. Furthermore, SpO2 is a non-invasive measurement of the percentage of hemoglobin in the blood that is saturated with oxygen, and it can be calculated using a pulse oximeter that shines two different wavelengths of light through a thin part of the body, such as a finger or earlobe. The absorption of light at each wavelength corresponds to the levels of oxygenated and deoxygenated hemoglobin present in the blood. The SpO2 formula is based on the ratio of red and infrared light absorbed by the blood and is calculated using a specific formula that incorporates the values of RED, RED average, IR, and IR average. By deriving the ratio R from these values, the SpO2 can be determined using an empirical formula that is based on the saturation of hemoglobin with oxygen.

R = (square root of (RED / RED average)) / (square root of (IR / IR average))

SpO2 = -23.3 * (R - 0.4) + 100

This SpO2 formula is used to calculate the percentage of hemoglobin that is saturated with oxygen based on the value of R, which is the ratio of red and infrared light absorbed by the blood. The constant value of 0.4 in the formula represents the baseline of R for fully oxygenated blood, while in the value of -23.3 is derived from empirical measurements. The formula has been shown to provide accurate estimates of SpO2 in clinical settings and technical specification of MAX 30102 Sensor is provided in table 1.2.

TABLE 1.2: Technical Specifications of MAX 30102 Sensor

Electrical Supply	3.3v to 5.5v
Red light wavelength range	660nm
Infrared light wavelength range	880nm
Operating Temperature	-40^0C to 85^0C
Current draw	600µA (during measurements) 0.7µA (during standby mode)

DHT 11 Sensor as shown in Fig. 1.5 is a digital temperature and humidity sensor that employs a capacitive humidity sensor and a thermistor to measure the surrounding air, providing a digital signal output for further use. The DHT11 sensor is frequently used to measure temperature and humidity. It's simple to connect with other microcontrollers and includes a dedicated 8-bit microcontroller for transmitting temperature and human readings as serial data.

The sensor is capable of measuring temperature from 0 to 50 degree Celsius and humidity levels from 20% to 90% with an accuracy of plus or minus 1% and plus or minus 10 Degree Celsius, respectively and technical specifications of DHT 11 sensor is explained in table 1.3 for further analysis of the sensor.

Fig 1.5 DHT 11 Sensor

$$RH = (P_w/P_s) \times 100\%$$

TABLE 1.3: Technical Specifications of DHT 11 Sensor

Operating temperature	0 to 50 Degree Celsius
Humidity parameters	A humidity range of 20% to 90%
Supply voltage	3.5 volt to 5.5 volt
Working current	0.3 mA (measuring) 60µA (standby)
Outcome	Data transmission
Precision	The temperature and humidity measurements are each encoded as 16-bit values

LM 35 Sensor as shown in Fig. 1.6 is an integrated precision circuity that delivers a voltage output directly proportional to the Celsius temperature. It doesn't require external calibration and has a sensitivity of 10 millivolts per degree Celsius

Fig 1.6 LM-35 Sensor

As the temperature rises, the output voltage of the LM-35 also increases. The LM35 sensor outputs the voltage in linearly proportion to the Celsius temperature. To calculate Temperature of a body we can measure the voltage on the LM32's OUT pin and the output scale factor of the LM35 is 10 mV/^{0}C. LM35 temperature sensor has three pins. VCC pin is connected to the 5V of VCC. The GND pin is connected to the GND (0V). The OUT pin outputs the voltage in proportion to the temperature value. The formula to calculate the sensor values to be implemented can be determined as follows:

To measure the temperature sensor's ADC output

ADC Val = analog Read (PIN_LM35)

To calculate the millivolt voltage from the ADC value of the temperature sensor

ADC VREF mV = (Millivolts * ADC RESOLUTION) / ADC Val

To determine the temperature in Celsius based on a voltage reading

Celsius temp = ADC VREF mV / 10

To change a temperature from Celsius to Fahrenheit, a conversion formula is needed

Fahrenheit temp = Celsius temp * 9 / 5 + 32

MQ 135 Sensor as shown in Fig 1.7 is Capable of detecting harmful substances such as smoke and benzene, making it suitable for a variety of applications including air quality monitoring, detecting noxious fumes, and identifying pollution in homes, industries, and portable settings. By measuring the concentration of NH3, NOx, alcohol, benzene, smoke and CO2 in the air, this versatile sensor can assist in identifying potential dangers and promoting a safer environment.

Fig 1.7 MQ-135 Sensor

Breadboard as shown in Fig 1.8 is an essential tool for creating temporary circuits, as it allows designers to add or remove components. It is a platform for prototyping or building circuits without soldering. The breadboard features numerous small sockets that allow for the placement of components and connections on the board. Additionally, the sockets are electrically connected in groups, enabling designers to construct circuits with ease.

Fig 1.8 Breadboard

Jumper Wires as shown in Fig 1.9 are electrical cables designed to connect distant circuits on printed circuit boards. These wires can be used to create a shortcut or disrupt the electric circuit by attaching them to the circuit. Jumper wires are available in three types: male-to-male, male-to-female and female-to-female.

Fig 1.9 Jumper Wires

AC Power Source as shown in Fig 1.10 deliver and distribute electricity over long distances to power microcontrollers like the Arduino, ESP Wi-Fi Module, and Raspberry Pi through electrical grid systems. The main AC power supply operates at around 230 volts for the UK mains supply, and its frequency is set at 50 Hz. This means the current changes direction 50 times per second, allowing for the safe and efficient distribution of electrical energy.

Fig 1.10 Power Source

The 16x2 i2c LCD display as shown in Fig 1.11 equipped with an I2C Interface and can show 16x2 characters on 2 lines. The display consists of a 5x7 pixel matrix for each character, allowing for a total of 224 different characters and symbols to be displayed. The LCD also features two separate registers, one for commands and one for data, to allow for easy control of the display output.

Fig 1.11 16x2 i2c LCD Display

Creating custom characters on LCD is a straightforward process that involves understanding the CG-RAM of the LCD and the LCD controller chip. Once you have designed your custom character and stored it in the CG-RAM, it can be displaced on the LCD screen just like any other character. The process may require some programming knowledge, but it is not overly complex and can be accomplished with the help of documentation and online resources.

1.1.5 What is Machine Learning?

Machine Learning is a subfield of artificial intelligence that employs statistical algorithms and models to identify patterns and make predictions from data without explicit instructions. This is achieved by training a model on a labeled dataset, which allows the model to learn the relationships between input and output variables. Once trained, the model can then be used to make predictions on new, previously unseen data. As shown in Fig 1.12 Machine learning algorithms can be supervised, unsupervised, or semi-supervised, depending on the availability of labeled data during training. In the context of health condition detection, machine learning algorithms can be trained on sensor data collected from patients to detect patterns and classify abnormal health conditions in real-time, potentially enabling earlier diagnosis and intervention.

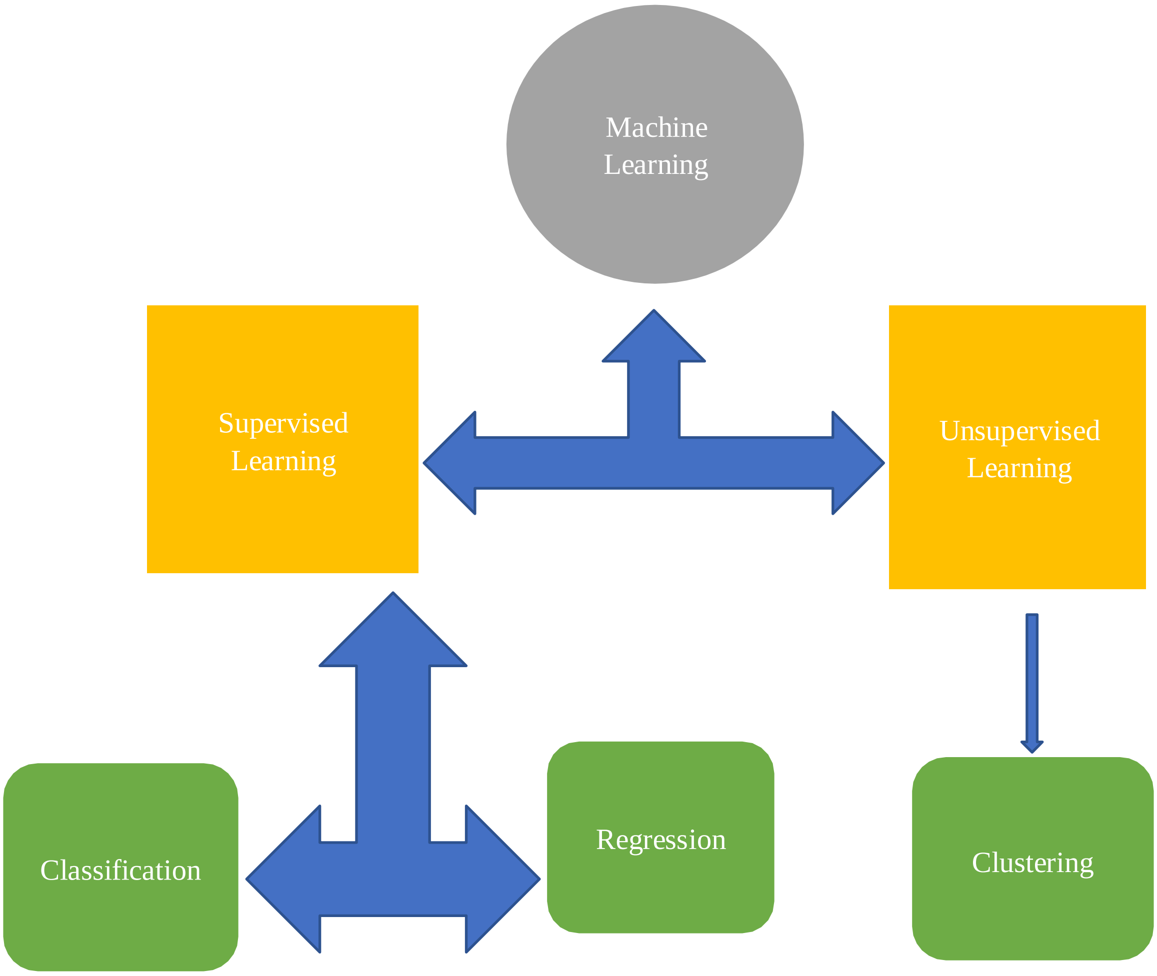


Figure 1.12 16x2 i2c LCD Display

1.1.6 Machine Learning Algorithms

Machine learning algorithms are computational methods that allow machines to learn from data without explicit programming. These algorithms enable computers to recognize patterns in data, make predictions or decisions based on new data, and improve their performance over time. There are three primary types of machine learning algorithms:

1. Supervised Learning: Supervised learning algorithms learn from labeled data, where the input features are mapped to a known output. These algorithms attempt to establish a correlation between input and output variables and use this correlation to predict outcomes for new, unseen data. Linear regression, Logistic regression, Support Vector Machine, Decision Trees, and Random Forests are examples of supervised learning algorithms.

2. Unsupervised Learning: Unsupervised learning algorithms learn from unlabeled data and aim to discover patterns or structures in the data without any prior knowledge of the output. These algorithms attempt to group similar data points together, identify underlying structures in the data, or reduce the dimensionality of the data. Examples of unsupervised learning algorithms include k-means clustering, Principal component analysis, and association rule learning.

3. Reinforcement Learning: Reinforcement learning algorithms learn from interactions with an environment, where they receive feedback in the form of rewards or penalties for each action taken. These algorithms aim to learn a policy that maximizes cumulative reward over time. Reinforcement learning algorithms are commonly used in robotics, gaming, and autonomous vehicles.
   Each machine learning algorithm has its own strengths and limitations, and the choice of algorithm depends on the specific problem and the available data. Some modern machine learning algorithms combine techniques from multiple categories, such as deep learning, which uses supervised and unsupervised learning to learn complex and hierarchical representations of input data. Some of the machine learning algorithms used in this paper includes support vector machine, decision tree, gradient boosting, random forest, naïve bayes and multilayer perceptron .

1. Support Vector Machines (SVM): SVM is a powerful machine learning algorithm used for both classification and regression tasks. SVM works by identifying the hyperplane that best separates the data points into different classes. In SVM, the goal is to find a hyperplane that maximizes the margin between two classes. The margin is the distance between the hyperplane and the closest data points from both classes. The idea is that the larger the margin, the better the generalization performance of the model. To explain SVM, let's consider a binary classification problem where we have two classes (positive and negative) and we want to separate them using a hyperplane. Let's assume that our input data consists of two features x1 and x2. First, we need to represent our data in a higher dimensional space. This is done by applying a kernel function to map the data from the input space to a higher-dimensional feature space. There are different types of kernel functions, such as linear, polynomial, and radial basis function (RBF). The choice of kernel depends on the data and the problem we are trying to solve. Once we have mapped the data into a higher-dimensional space, we can find the hyperplane that best separates the data. The hyperplane is defined as:

$$W*x + b =0$$

where w is the weight vector, x is the input vector, and b is the bias. The goal is to find the values of w and b that satisfy the following constraints:

$$y_i*(w*x_i+b)>=1 \text{ for all } i$$

where y_i is the class label (+1 or -1) of the i-th data point. This constraint ensures that all data points are correctly classified and that they are on the correct side of the hyperplane.

$$\text{Margin} = 2 / ||w||$$

Where ||w|| is the norm of the weight vector. The goal is to maximize the margin, subject to the constraint above. To solve this optimization problem, we can use lagrange multipliers to find the optimal values of the weight vector and bias. The lagrange multipliers are used to convert the optimization problem into a dual problem, which can be solved more efficiently. Once we have found the optimal values of the weight vector and bias, we can use them to classify new data points. We simply compute the dot product between the weight vector and the input vector, and add the bias. If the result is greater than or equal to zero, we classify the data point as positive, otherwise, we classify it as negative. In summary, SVM is a powerful machine learning that works by finding the hyperplane that maximizes the margin between two classes. It is based on the idea of mapping the data to a higher-dimensional space and finding the values of the weight vector and bias. SVM is widely used in a variety of applications, including image classification, text classification and bioinformatics.

2. Random Forest (RF): Random Forest is a machine learning algorithm that combines the predictions of multiple decision trees to make accurate predictions for classifications and regression tasks. To construct each decision tree, the algorithm randomly selects a subset of the training data and a subset of the features. This helps to reduce overfitting and improve the algorithm's generalization performance. The algorithm evaluates the importance of each feature using the mean decrease impurity metric, which quantifies the reduction in accuracy of the model when a particular feature is randomly permuted. For classification tasks, the final prediction is based on the majority vote of the individual tree predictions, while for regression tasks, the final predication is the average of the individual tree predictions. The following formula are used to evaluate feature importance and make predictions:

   Mean decrease impurity metric for feature importance:

   mean decrease impurity = accuracy without feature – accuracy with feature

   This formula quantifies the importance of each feature by comparing the model's accuracy and without that feature.

   Predicted class probabilities for classification:

   predicted probability = (probability from tree 1 + probability from tree 2 + … + probability from tree n) / n

This formula calculates the predicted probability of each class for a given input by averaging the predicted probabilities from each individual tree in the ensemble.

Predicted value for regression:

predicted value = (predicted value from tree 1 + predicted value from tree 2 + …. + predicted value from tree n) / n

This formula calculates the predicted value for a given input by averaging the predicted values from each individual tree in the ensemble.

3. Naïve Bayes (NB): Naïve Bayes is a classification algorithm commonly used in machine learning due to its effectiveness in various applications. The algorithm is based on Bayes theorem, which is a fundamental theorem in probability theory. Naïve bayes works by calculating the conditional probability of each class given the feature values of a new data point, and then selecting the class with highest probability as the predicted class. The algorithm makes the simplifying assumption that all features are independent of each class. There are three main types of Naïve Bayes algorithms. Gaussian Naïve Bayes, Multinomial Naïve bayes, and Bernoulli Naïve bayes. Each of these algorithms makes different assumptions about the distributions of the feature values and uses different formulas to calculate the probabilities. For example, Gaussian Naïve bayes assumes that the feature values are normally distributed, and uses the Gaussian probability density function to calculate the conditional probabilities. The formula for calculating the conditional probability of class c given feature vector x is given by:

$$P (c \mid x) = (1 / (\mathrm{sqrt}(2\pi) * \sigma c)) * \exp (-(x - \mu c) ^\wedge 2 / (2 * \sigma c^\wedge 2))$$

where μc and σc are the mean and standardized deviation of the feature values for class c.

Multinomial Naïve Bayes, on the other hand, assumes that the feature values are discrete and follow a multinomial distribution, and uses a different formula to calculate the probabilities. The formula for calculating the conditional probability of class c given feature vector x is given by:

$$P (c \mid x) = (\Pi i \, P (xi \mid c) ^\wedge xi) * P(c)$$

where (xi | c) is the probability of observing feature value xi given class c, and P(c) is the prior probability of class c.

Bernoulli Naïve Bayes assumes that the feature values are binary and uses the Bernoulli distribution to calculate the probabilities. The formula for calculating the conditional probability of class c given feature vector x is given by:

$$P (c \mid x) = (\Pi i \, P(xi \mid c) ^\wedge xi * (1- P(xi|c)) ^\wedge (1-xi)) * P(c)$$

where P (xi | c) is the probability of observing feature value xi given class c, and P(c) is the prior probability of class c.

To classify a new data point, the algorithm first calculates the prior probabilities of each class based on the training data. It then calculates the conditional probabilities of each class given the feature values of the new data point, using the appropriate formula for the chosen type of naïve bayes algorithm. Finally, it selects the class with the highest probability as the predicted class for the new data point.

4. Gradient Boosting (GB): Gradient Boosting is a machine learning algorithm that combines multiple week models to create a strong learner. The algorithm works by fitting models in a sequential manner to the residual errors of the previous model, with each subsequent model focusing on the errors of the previous model.
   This approach allows the algorithm to gradually improve its predictions by learning from the mistakes of the previous models. To apply Gradient Boosting, the model is first initialized with a constant value, usually the mean of the target variable. In each iteration, a weak learner is fitted to the negative gradient of the loss function with respect to the current model's prediction. This negative gradient indicates the direction in which the loss function is decreasing most rapidly. For regression problems, the mean squared error (MSE) loss function is commonly used:

$$MSE = (1 / n) * \sum (y\_ i - \hat{y}\_ i)^2$$

   where y_i is the true target value for the i-th example, ŷ_ i is the predicted value, and n is the number of examples. For classification problems, the cross-entropy (CE) loss function is used:

$$CE = -(1 /n) * \sum (y\_ i * \log (\hat{y}\_ i) + (1 – y\_ i) * \log (1- \hat{y}\_ i)]$$

   where y_i is the true class label for the i-th example (either 0 or 1), ŷ_ i is the predicted probability of the positive class, and n is the number of examples. The weak learner used in Gradient Boosting is typically a decision tree, but other models can also be used. Each tree is fitted into the negative gradient of the loss function and trained to predict the residual errors of the previous model. The learning rate is a hyperparameter that influences the contribution of each tree to the final model. A lower learning rate results in a slower but more accurate learning process, while a higher learning rate leads to a faster but less accurate learning process. In practice, Gradient Boosting is often implemented using optimized and efficient libraries such as XG Boost or Light GBM.

5. Multilayer Perceptron (MLP): Multilayer perceptron is a form of neural network architecture that comprises numerous interconnected neurons or nodes, organized into multiple layers. MLPs are commonly employed in supervised learning scenarios, including regression and classification. Each neuron's output is influenced by a non-linear activation function that computes the weighted sum of inputs to the neuron. This output is then forwarded to the following layer of neurons, and so on, until the output layer is reached, which produces the networks prediction.
   Mathematically, the neuron's output in an MLP can be expressed as:

$$A_j = f \left(\sum\nolimits^{n}_{i=1} w_{ij}x_i + b_j\right)$$

   where a_j refers to the jth neuron's output, w_{ij} represents the weight of the connection between the ith input and jth neuron, x_i denotes the ith input, b_j represents the bias term for the jth neuron, and f is the activation function. Common activation functions in MLPs include the sigmoid function and the rectified linear unit (ReLU) function.

The sigmoid function is given by:

$$f(z) = 1 / 1 + e^{-z}$$

whereas the ReLU function is defined as: $f(z) = \max(0, z)$

The MLP's weights and biases are learned through backpropagation, a process that adjusts the weights and biases based on the difference between the network's prediction and the true output. This is accomplished by computing the gradient of the error with respect to the weights and biases and using it to update their values using an optimization algorithm such as stochastic gradient descent. To summarize, MLP is a neural network architecture consisting of interconnected neurons arranged into layers. Each neuron applies a non-linear activation function to generate an output. The MLP's weights and biases are updated via backpropagation, enabling the network to make accurate predictions on new data.

6. Decision Tree (DT): A decision tree is a type of hierarchical model used in supervised learning, which can handle both classification and regression tasks. It partitions the data recursively into subsets based on a decision criterion or split rule. Starting from the root node, the algorithm works its way down to the leaves using a decision criterion at each internal node to split the data into child nodes. The split rule is selected to maximize information gain or minimize impurity. The process continues until a stopping criterion is met, such as reaching a maximum depth or a minimum number of instances in a node. At each leaf node, a prediction is made based on the majority class or the average of the target variable. The algorithm can be represented as a binary tree where each internal node represents a split and each leaf node represents a prediction. To select the best split rule, various measures of impurity can be used, such as the commonly used Gini impurity or information gain. For a binary classification problem, the Gini impurity of a node can be expressed as:

$$I_G(p) = 1 - p_1^2 - p_0^2$$

where p_1 and p_0 are the probabilities of instances belonging to class 1 and 0, respectively. Information gain can be calculated as:

$$IG(D_p, f) = I(D_p) - \text{sum}(N_j / N_p * I(D_j))$$

where $IG(D_p, f)$ is the information gain of feature f on data D_p, $I(D_p)$ is the impurity of data D_p, N_j is the number of instances in the jth child node, N_p is the number of instances in the parent node, and $I(D_j)$ is the impurity of the jth child node. However, decision tree can easily overfit if they are too complex or if the splitting rules are not chosen carefully. To address this issue, regularization techniques such as pruning or setting a minimum number of instances per leaf can be applied.

Chapter 2

2. LITERATURE SURVEY

2.1 Literature survey for Person Health Monitoring

Sr. No.	Title	Author	Publication Details	Focus Area
1	IoT based E-health Monitoring and Room Environment Controlling System	N.D. Gedam, Pranav Rajurkar, Nayun Hatwar, Atharva Sawai	International Research Journal of Engineering and Technology (IRJET), 2021.	Emergency SMS alert when abnormal health condition for patient arises
2	Internet of Things Based Real-Time Vital Physiological Parameter Monitoring System for Remote Asthma Patients	Khairul Islam , Farabi Alam , Abid Ibna Zahid , Mohammad Monirujjaman Khan , Muhammad Inam Abbasi	Research Article of Hindawi Wireless Communications and Mobile Computing (WILEY), 2022	A Person can consult with doctor and get a prescription through video calling feature
3	Remote Health Monitoring System using Internet of Things	Prasun Biswas , Shreyashi Haldar	International Research Journal of Engineering and Technology, 2020.	Medical professionals where they can visualize or remotely access the date anywhere at any time from any location
4	Health Monitoring and Prediction using Internet of Things and Machine Learning	Riyazulla Rahman .J, Shridhar Sanshiand N. Nasurudeen Ahamed	International Conference on Advanced Computing and Communication Systems, 2021.	Persons sensed values are displayed in 16x2 i2c display
5	IoT for In-Home Health Monitoring Systems	Nada Y. Philip and Honggang Wang	IEEE Journal on Selected Areas in Communications, Vol.39, No.2, 2021.	Potential of in-home health monitoring with IoT

Table 2.1 Person Health Monitoring System

This literature review as described in table 2.1 presents an analysis of studies that have investigated the development and implementation of IoT-based systems for health monitoring and management. Each of the five studies reviewed has focused on a unique aspect of IoT-based health monitoring. One study focused on using IoT for e-health monitoring and room environment control, with a particular emphasis on the development of a system that can detect abnormal health conditions and alert medical professionals [1]. Another study developed an IoT-based system for real-time monitoring of vital physiological parameters in remote asthma patients, which also includes a video calling feature for remote consultations with doctors and prescription issuance [2]. A third study explored the development of a remote health monitoring system using IoT, with the goal of enabling medical professionals to access patient data from any location and at any time [3]. In a fourth study, an IoT-based health monitoring system was designed to predict health conditions using machine learning and to display the sensed values on a 16x2 i2c display [4]. Lastly, one study investigated the potential of IoT-based in-home health monitoring systems to improve healthcare outcomes and reduce healthcare costs [5]. These studies collectively demonstrate the potential for IoT-based systems to revolutionize health monitoring and management. By enabling real-time monitoring of vital parameters and remote access to patient data, such systems can lead to improved healthcare outcomes and reduced costs. Nonetheless, further research is required to optimize these systems for widespread adoption, and to address privacy and security concerns.

2.2 Literature survey for Person Health Condition Detection

Sr.No.	Title	Author	Publication Details	Focus Area
1	IoT-Enabled Framework for Early Detection and Prediction of COVID-19 Suspects by Leveraging Machine Learning in Cloud	Mahmood Hussain Mir, Sanjay Jamwal, Abolfazl Ehbodniya, Tanya Garg, Ummer Iqbal, Issah Abubakari Samori,	*Journal of Healthcare Engineering,* vol.2022, Article ID 7713939, 2022	The integration of machine learning algorithms with sensor technology has paved the way for a more efficient and accurate approach towards detecting abnormal health conditions

2	A Survey on Machine Learning and Internet of Medical Things-Based Approaches for Handling COVID-19: Meta-Analysis	Shahab S.Band, Sina Ardabili, Atefeh Yarahmadi, Bahareh Pahlevanzadeh, Adiqa Kausar Kiani, Amin Beheshti, Hamid Alinejad-Rokny, Iman Dehzangi, Arthur Chang, Amir Mosavi, and Massoud Moslehpour	*Frontiers in Public Health,* PMID 35812486, 2022	Machine learning that analyzes the sensed data to provide insights and solutions
3	A Real Time IoT Based Patient Health Monitoring System Using Machine Learning Algorithms	Yedla Vineetha, Yogesh Misra, K.Krishna Kishore	*European Journal of Molecular and Clinical Medicine,*vol.7, Issue 4, 2020.	Early detection of health condition improve the quality of life of individuals
4	IoT-Based Healthcare-Monitoring System towards Improving Quality of Life	Suliman Abdulmalek, Abdul Nasir, Waheb A.Jabbar, Mukkarram A.M. Almuhaya, Anupam Kumar Bairagi, Md. Al-Masrur Khan, Seong-Hoon Kee	*Healthcare MDPI,* 2022	Automate various healthcare procedures that were time consuming and prone to human error
5	Internet of things-enabled real-time health monitoring system using deep learning	Xingdong Wu, Chao Liu, Lijun Wang , Muhammad Bilal	Neural Computing and Applications, 2021.	To objectively assess postural control in challenging and addresses the limitations of existing medical infrastructure

Table 2.2 Person Health Condition Detection

This literature review examines different studies that center on the development and implementation of IoT-based systems to monitor and manage healthcare, with a particular focus on COVID-19. The first study aims to develop an IoT-enabled framework that utilizes machine learning algorithms to detect abnormal health conditions and enable early detection and prediction of COVID-19 suspects [1].

The second study is a survey of machine learning and IoMT-based approaches to handling COVID-19, providing insight and solutions using machine learning algorithms to analyze sensed data [2]. The third study aims to develop a real-time IoT-based system for patient health monitoring using machine learning algorithms, focusing on early detection to improve individual quality of life [3]. Another study presents an IoT-based healthcare monitoring system designed to automate various procedures and reduce the likelihood of human error [4]. Lastly, the fifth study focuses on developing an IoT-enabled real-time health monitoring system using deep learning to assess postural control in challenging situations and address limitations in current medical infrastructure [5]. These studies highlight the potential of IoT-based systems to transform healthcare monitoring and management during the COVID-19 pandemic, with real-time vital parameter monitoring and remote access to patient data leading to improved healthcare outcomes and cost reductions. Nonetheless, further research is necessary to optimize these systems for widespread use and address concerns surrounding privacy and security.

Chapter 3

3. METHODOLOY

The main purpose of this project is to monitor the real time health condition using IoT based devices. The various parameters such as heart rate, blood oxygen, body temperature, room humidity, room temperature and air quality is sensed by various sensors and these sensed values are controlled by microcontroller and by using the Wi-Fi module these sensed values is send to the cloud and doctors/caretakers can view these sensed values through our created website and mobile application and an emergency alert SMS is sent to doctors and caretakers if the health condition of the patient is abnormal/critical. The Proposed flow chart diagram of the Internet of Things based health monitoring system is explained in Fig 3.1. The various sensors include MAX 30102, LM 35, DHT 11, MQ 135 Sensors. The real time database is created in the firebase and it is integrated with the created website. The ThingSpeak is integrated along with IFTTT by creating Thing HTTP and react in ThingSpeak and creating different applets in If this then that platform if the sensed values is set to a particular condition and if it satisfies then the react triggers the applet in IFTTT and the SMS messages is sent to the registered mobile number. Figure 1 shows the flow diagram of our proposed model. During the prototype phase, each sensor node is individually connected to the cloud to ensure continuous and uninterrupted data flow. The sensor data is sent directly to the cloud, where Smart Ambient Behavior System (SABOS) employs ThingSpeak, an IoT analytics platform, and MATLAB for data analysis. SABOS also integrates the if this then that (IFTTT) service, which enables it to respond to unexpected behavior data and adjust environmental conditions accordingly. IFTTT as a free web service that uses a REST-based API server-style architecture to facilitate application programming interface using HTTP requests to access and utilize data. With IFTTT, SABOS can create conditional statements called applets that are triggered by changes that occur within other web services, such as ThingSpeak. By leveraging this technology, SABOS can effectively monitor and analyze behavior and proactively optimize environmental conditions to achieve optimal outcomes [16]. The proposed method for detecting abnormal health conditions is a machine learning based approach that employs six distinct algorithms, including Support Vector Machines, Random Forest, Gradient Boosting, Decision Tree, Naïve Bayes, and Multilayer Perceptron neural networks. As shown in Figure 1. real-time health monitoring is acquired from MAX 30102 and LM35 sensors, which capture vital signs such as heart rate, blood oxygen, and body temperature. The collected sensor is transferred to ThingSpeak, a cloud-based platform for the Internet of Things, and utilized in the analysis. The proposed method for detecting abnormal health conditions is a machine learning based approach that employs six distinct algorithms, including Support Vector Machines, Random Forest, Gradient Boosting, Decision Tree, Naïve Bayes, and Multilayer Perceptron neural networks. As shown in Figure 3.1 real-time health monitoring is acquired from MAX 30102 and LM35 sensors, which capture vital signs such as heart rate, blood oxygen, and body temperature. The collected sensor is transferred to ThingSpeak, a cloud-based platform for the Internet of Things, and utilized in the analysis.

Additionally, publicly available datasets containing various health conditions and their corresponding features are preprocessed to remove missing values and outliers are standardized to a common scale. The preprocessed dataset is divided into training and testing sets using varying ratios, and each of the six algorithms is trained and tested on both sets using cross-validation techniques. To assess the method's efficacy, diverse evaluation metrics are utilized, including accuracy, precision, recall, F1 score, and Receiver Operating Characteristics curves. The Gradient Boosting algorithm outperforms the other algorithms, demonstrating the potential for real-time health condition using sensor data. The proposed approach has the potential to support medical practitioners in the early detection of abnormal health conditions using real-time sensor data and may contribute to the development of automated systems for health condition classification and early detection in real-time monitoring scenarios. The various machine learning techniques and libraries for performing classification on medical data of different techniques.

3.1. Real Time Person Health Monitoring

3.1.1. Flow Diagram of PHM

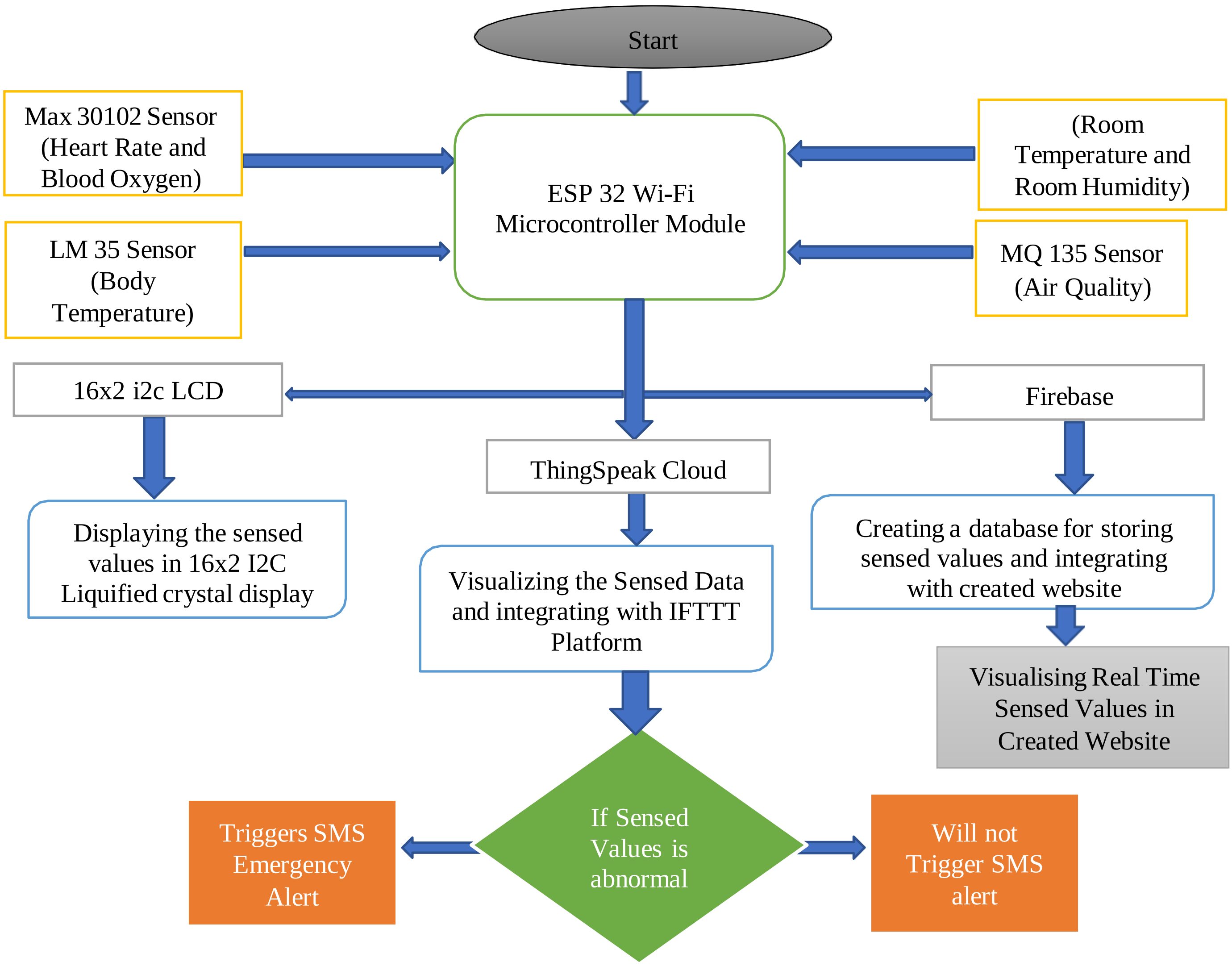


Figure 3.1 16x2 i2c LCD Display

3.1.2. Hardware Design

Component used

- ESP 32 Microcontroller
- MAX 30102 Sensor
- DHT 11 Sensor
- LM 35 Sensor
- MQ 135 Sensor
- 16x2 i2c LCD Display

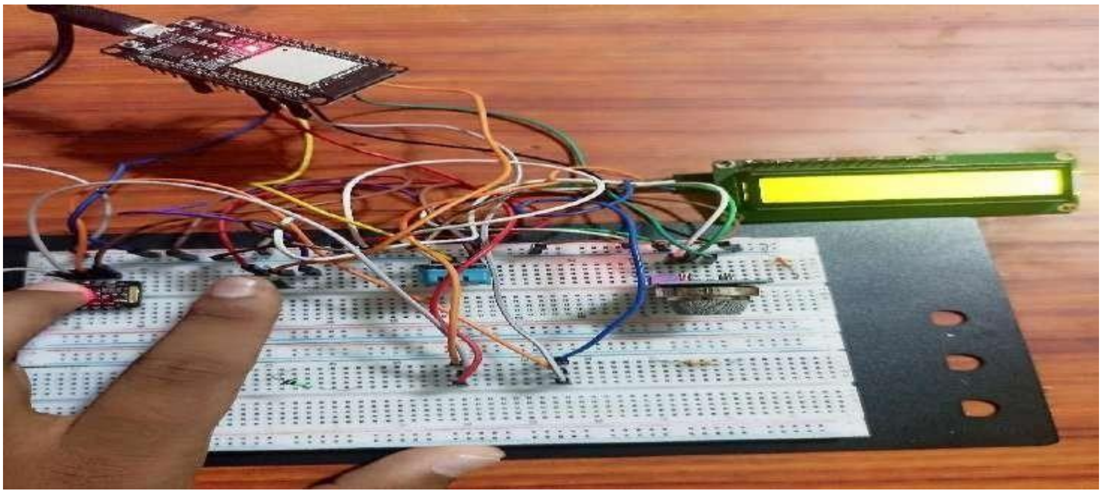


Figure *3.2* Hardware design of our proposed IoT Health Monitoring System

The hardware design as shown in Fig 3.2 consists of a breadboard, microcontroller and various sensors and a computer. To measure heart rate and SpO2 MAX 30102 is used. LM-35 is used to sense body temperature. DHT11 is used to sense the room temperature and room humidity of surrounding environment. MQ-135 is for air quality sensing. Each sensor is connected with a microcontroller ESP32 Wi-Fi module which controls these sensors and making a connection with IP. The microcontroller is powered by USB and it is connected to various sensor using jumper wires. AC Power supply is used by the connecting components for supplying electrical energy. Sensors that sense values from the human body and surrounding environment and these sensed values are send to the cloud using microcontroller and these sensed values are also displayed in the 16x2 LCD display. The microcontroller that sends the measured values to the firebase and ThingSpeak cloud.

3.1.3. Software Design

Real-time data from a database can be received by the Firebase system. The Firebase system can send all the sensed values to a mobile application and a website. These values can be viewed on the website by doctors, patients, and caretakers.

The ThingSpeak cloud enables aggregation, visualization, and analysis of live data streams, as well as display of all sensed values. This cloud can be integrated with the IFTTT Platform. The website is designed using HTML, CSS, JavaScript and features a user sign-in option. Once verified by the admin, users can access the application and their information is stored in the cloud. A detailed description of the website design is provided. The Website was constructed using an assortment of web technologies, such as HTML, CSS, Bootstrap, jQuery, and JavaScript. HTML, also referred to as hypertext markup language, is utilized to format and display online content as web pages, which often contain hypertext links to other pages. CSS, or Cascading Style Sheets, is utilized to design and organize web pages, allowing users to personalize design features such as font, color, size and spacing. Interactive and dynamic content for web applications and browsers is produced using JavaScript, a scripting language.

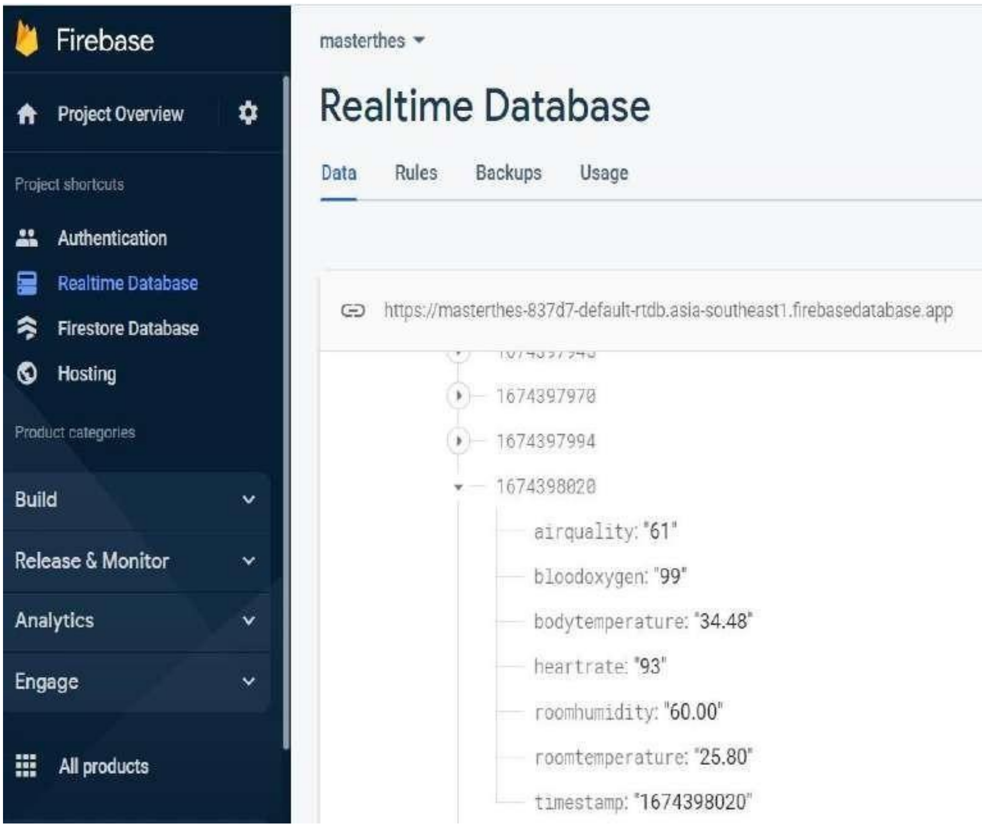


Figure 3.3 Real Time Database created in Firebase

By leveraging Firebase hosting, the web application is served over a global CDN and equipped with a secure SSL certificate. This allows access to our web application from anywhere via the Firebase-generated domain name as shown in Figure 3.3. Real-time sensor readings are stored in Firebase's database and the database is protected with a set of rules. As shown in below Fig 3.4 is the login page of our created website application for securing the patient or person's personal health data information from unauthorized users.

Figure 3.4 Displaying Login page of our created website

IFTTT, an abbreviation for If This Then That, is a free online tool that allows users to automate web -based activities and improve productivity. IFTTT integrates with various developer devices, services, and applications to create customized applets for automation purposes. As shown in Fig 3.5 Applets are created with Triggering event. If a person's physical condition becomes abnormal or critical, IFTTT triggers the applet that automatically sends emergency alarm SMS messages to doctors and caretakers, enabling them to respond promptly to the patient.

This project discusses about the Design and Development of IoT based health monitoring system that utilizes various sensors to measure a person or patient's heart rate, blood oxygen, body temperature, room humidity, room temperature, and air quality. Though testing, we found that our proposed system allows for real-time monitoring of a person's body health. The Heart rate and Blood Oxygen sensed by MAX30102 sensor, DHT-11 that senses Room Temperature and Room Humidity, LM35 that senses the body temperature along with MQ135 that senses the surrounding air quality level and all these data are processed in ESP 32 Wi-Fi module and it is also interfaced along with 16X2 LCD module which displays our sensed values in the Liquid Crystal Display.

Figure 3.5 Created Applet in IFTTT for Triggering the SMS

The Arduino ide serial monitor displays with a baud rate of 115200 describes the measured values of our body's health condition. Baud rate is a measure of how many signal changes per second occur when data is transmitted through a medium. A higher baud rate indicates faster data transfer during transmission and reception. After the various sensors senses the heart rate, blood oxygen, body temperature, room humidity, room temperature and air quality and microcontroller that controls the sensors can display the sensed values in the 16x2 i2c LCD Display. When a person is sensed by the sensors there sensed values are displayed in the real time as shows the sensed value of a person's heart rate as 93 BPM. Similarly, all the sensed value are displayed in LCD Display module withbody temperature as 35.02 celsius, heart rate as 93 beats per minute (bpm), blood oxygen as 95%, room humidity as 54%, room temperature as 27.10 degree celsius and air quality as 55 parts per million (ppm). The sensed data are sent to the ThingSpeak cloud by ESP module which visualizes our sensed data and it integrated along with IFTTT platform that sends an emergency SMS alert when the patient health condition is critical to ourselves/doctors/caretakers so that necessary precautions can be taken for our health as soon as possible. These sensed values are also sent to the created website by firebase database that stores our real time sensed values. We can able to further monitor the real time sensed values automatically for every time when the sensor senses the values. We can further able to delay the displaying sensed values according to our wish by changing it in the arduino ide code.

Table 3.1: Average range of Sensor Values

Measurement	***Average normal range***
Heart Rate	*60-100 BPM (18 and over)* *70-100 BPM (6-15)*
Blood Oxygen	*95 – 100 %*
Body Temperature	*30-40^0 C*
Room Humidity	*30-70%*
Room Temperature	*15-30*
Air Quality	*0-50: This Air Quality range indicates a high level of purity and is not hazardous to health.* *51-100: This Air Quality range suggests a moderate level of pollution, which is acceptable but may cause some discomfort for certain individuals.* *101-150: This Air Quality range may be detrimental to individuals with sensitivity to airborne pollutants* *151-200: This Air Quality range can cause difficulty breathing and other negative health effects in some individuals.* *201-300: This Air Quality range is considered to be of emergency status and may pose a health risk to the general population.* *301-500: This Air Quality range is highly hazardous and can cause serious health problems, including respiratory diseases and other respiratory issues.*

The Normal or Average range of sensed values is discussed briefly in table 3.1. If the sensed values is above or below this mentioned average range it is considered as abnormal condition and necessary steps should be taken by the doctors, caretakers or ourselves by maintaining with a normal sensed values accordingly. This would be helpful for those who needs to maintain good health condition with their sensed values and helps to improve the health condition of an individual person who are connected with these sensors.

3.2. Person Health Condition Detection using ML

The person health condition is normal or abnormal is detected by training the sensed values from real time health monitoring and then we test the unknown datasets using the trained existing datasets so that detecting the health condition of person becomes more easy in health care by implementing various machine learning algorithms. The machine learning can be done by various processes as mentioned in the flow diagram.

3.2.1. Flow Diagram of ML Process

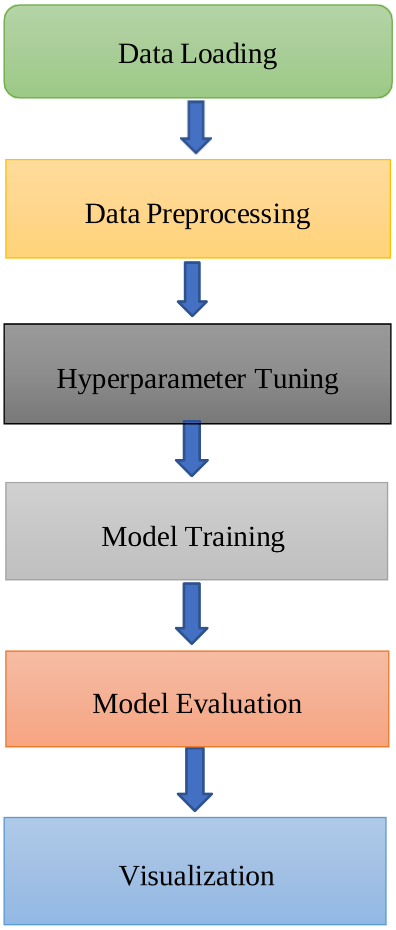


Figure 3.6 Flow chart of Machine Learning Processing Steps

Developing a machine learning model involves a series of steps as shown in Fig 3.6 including data loading, data preprocessing, hyperparameter tuning, model training, model evaluation, and visualization. First, the raw data is imported into the model during the data loading phase. Next, data preprocessing is performed to prepare the data for use in the model, which may involve tasks such as data cleaning, normalization, and encoding categorical variables.

The hyperparameter tuning phase involves selecting the optimal values for various model parameters to optimize its performance. Once the model parameters are tuned, the model is trained using an appropriate algorithm such as gradient descent or random forest. Following training, the model is evaluated on a separate test set to assess its performance, typically using metrics such as accuracy, precision, recall, and F1 score.

Finally, visualization techniques are used to analyze the performance of the model, including methods such as confusion matrices and ROC curves. To build an effective machine learning model, all of these steps must be executed effectively, with attention paid to the quality of the data, the selection of appropriate algorithms and parameters, and the thorough evaluation of the model's performance.

3.2.2. Diagram Representation of Health Condition Detection Method

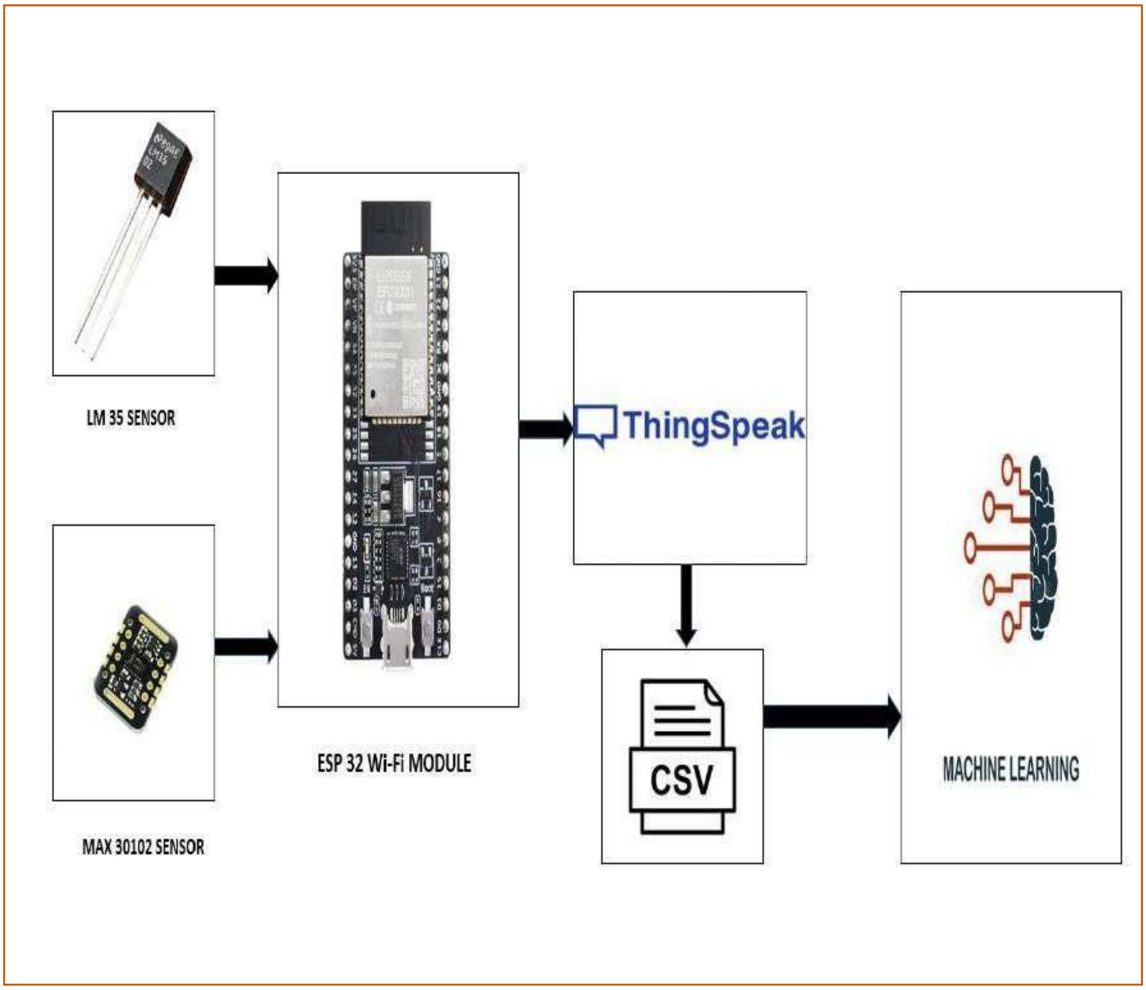


Figure 3.7 Representation of Health Condition Detection Model

The health condition detection of our proposed method as shown in Fig 3.7 describes that system utilizes the real time sensed values of heart rate, blood oxygen from MAX 30102 Sensor, and Body Temperature from LM 35 Temperature Sensor values are sent to thing speak cloud by ESP-32 Microcontroller board and we are exporting the sensed values in csv format.

Our proposed approach for detecting the health status of patients involves a series of interconnected steps that include data loading, preprocessing, hyperparameter tuning, model training, evaluation, and visualization. Initially, our program utilizes the Pandas library to preprocess and load data from CSV files. This library is a popular tool for data manipulation and analysis and is used to load both the training and testing datasets from CSV files. Following this, the data undergoes preprocessing to separate the feature variables and target variable from the training data. This step guarantees that the model trains only on relevant features, and the target variable is separated for training purposes. Afterward, the program proceeds with hyperparameter tuning, where we define the hyperparameter search space for the various algorithms, including learning rate, maximum depth, and the number of estimators. Our Program then utilizes Randomized Search CV to perform hyperparameter tuning on the model.

This technique selects random combinations of hyperparameters from the defined search space and assesses their performance using cross-validation. This helps in identifying the optimal hyperparameters for the model to achieve the best possible performance. Once the hyperparameters are tuned, the program builds a different classifier model with the chosen hyperparameters and trains it on the preprocessed training data. During the model training, the program uses the preprocessed data to train the model on how to make predictions. After the model training, the program evaluates its performance on unknown data. It prints the predicted health status of unknown patients using the trained model, along with the best hyperparameters found by Randomized Search CV. Also, the program generates a classification report that provides a detailed analysis of the model's performance on each class, displaying the precision, recall, f1-score, and support for each class. This analysis allows users to pinpoint areas that require improvements.

Lastly, our program generates a bar chart that visually represents the precision, recall, and f1-score for each class in the classification report and ROC (Receiver Operating Characteristic) curve is a visual tool used to evaluate the performance of a binary classifier by plotting the true positive rate (TPR) against the false positive rate (FPR) at different threshold settings. The AUC-ROC (Area Under the ROC Curve) is a commonly used metric for measuring the overall performance of classification models. A higher AUC-ROC value indicates better performance. The ROC curve and AUC-ROC are often included in the evaluation section of a journal paper to demonstrate the effectiveness of the proposed classification model.

Chapter 4

4. COMPONENT AND TECHNOLOGIES

To develop this system various electronic component as well as various technologies are used which are explain in details as follows-

4.1. Common technologies & components for IoT based health monitoring

- ESP 32 Microcontroller
- Proetus PCB design simulation software
- Python Programming Language.
- C Programming Language
- Arduino IDE
- IFTTT Platform

4.1.1 ESP-32 Microcontroller Module

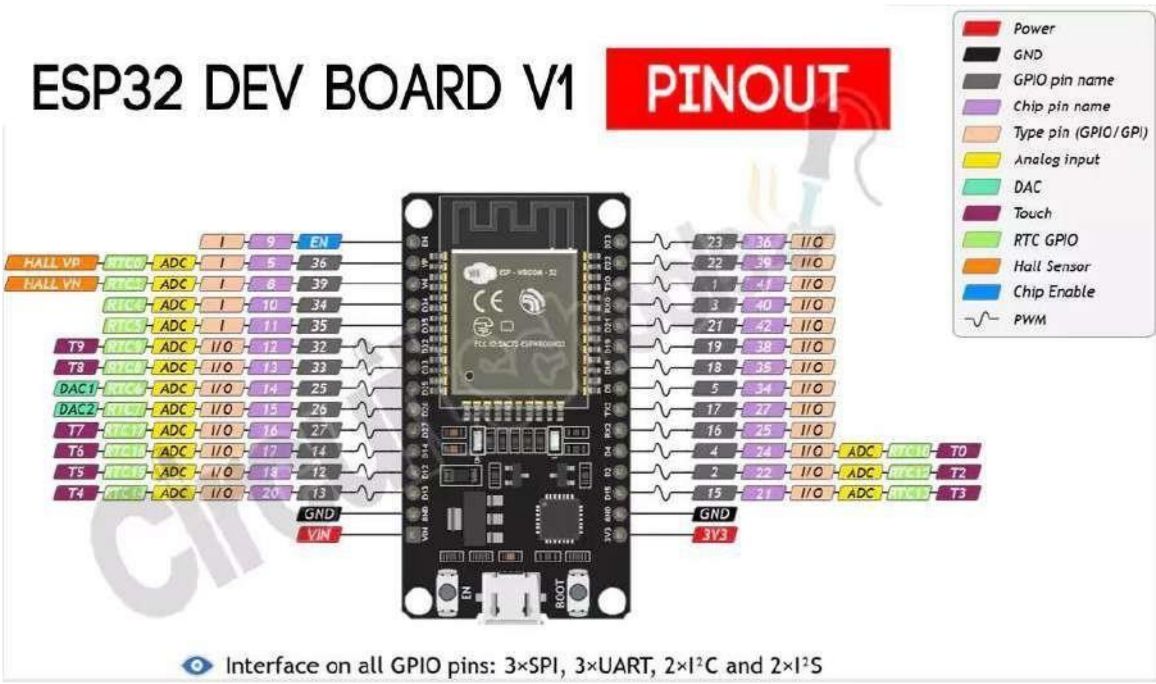


Figure 4.1 Representation of ESP-32 Devkit Board

The ESP32 Dev Kit development board is equipped with 36 GPIO pins, labeled GPIO0 to GPIO35, which can be configured for various functions such as digital input/output, PWM output, interrupt sources, I2C communication, SPI communication, UART, analog input, touch input, and more. GPIO0 and GPIO2 are boot-mode selection pins during the boot process and GPIO0 is also used as a programming mode selection pin for firmware uploading. GPIO4 to GPIO11 can be used for digital input/output, PWM output, interrupt sources, and I2C communication. GPIO12 to GPIO15 can be used for digital input/output, PWM output, interrupt sources, and SPI communication. GPIO16 can be used for digital input/output, interrupt sources, and as a wake-up pin from deep sleep mode. GPIO17 to GPIO19 can be used for digital input/output, PWM output, interrupt sources, and SPI communication. GPIO21 to GPIO23 can be used for digital input/output, PWM output, interrupt sources, and I2C communication. GPIO25 to GPIO27 can be used for digital input/output, PWM output, and interrupt sources. GPIO32 to GPIO39 can be used for digital input/output, PWM output, interrupt sources, and analog input. It's important to note that some of the GPIO pins have specific functions and limitations. GPIO0, GPIO2, and GPIO15 should be left floating or pulled high during normal operation, GPIO34 to GPIO39 are input-only pins, and some pins have specific functions during the boot process. Therefore, it's essential to carefully consider the functions and limitations of each pin before using them in a project.

4.1.2 Proetus PCB design simulation software

Figure 4.2 Proteus Software Logo

Proteus is a simulation software developed by Lab center Electronics. It is primarily used for simulating and designing electronic circuits. The software offers a range of tools that enable the user to create, test, and troubleshoot electronic circuits before actually building them. Proteus consists of two main components: ISIS and ARES. ISIS is the schematic capture tool, which allows the user to create and edit circuit schematics. ARES is the layout editor tool, which enables the user to design the physical layout of the circuit board. One of the key features of Proteus is its extensive library of pre-built components, which includes a wide range of microcontrollers, sensors, and other electronic components. The library also includes models for popular microcontrollers, such as the PIC and AVR series, which allows users to simulate their firmware code and debug it within the Proteus environment. Another useful feature of Proteus is its ability to simulate real-world environmental factors, such as temperature and humidity. This is particularly useful for testing and validating sensors and other components that may be affected by changes in environmental conditions. Proteus also allows the user to simulate and test complex digital circuits, such as microprocessors and FPGAs. This is accomplished through the use of virtual instruments, which include oscilloscopes, logic analyzers, and function generators. In addition to its simulation capabilities, Proteus also includes a range of design verification tools, such as design rule checks and netlist comparison, which help to ensure that the final circuit design is error-free and ready for production. Overall, Proteus is a powerful and versatile tool for designing and simulating electronic circuits. Its extensive library of components, simulation capabilities, and design verification tools make it a valuable tool for both hobbyists and professionals alike.

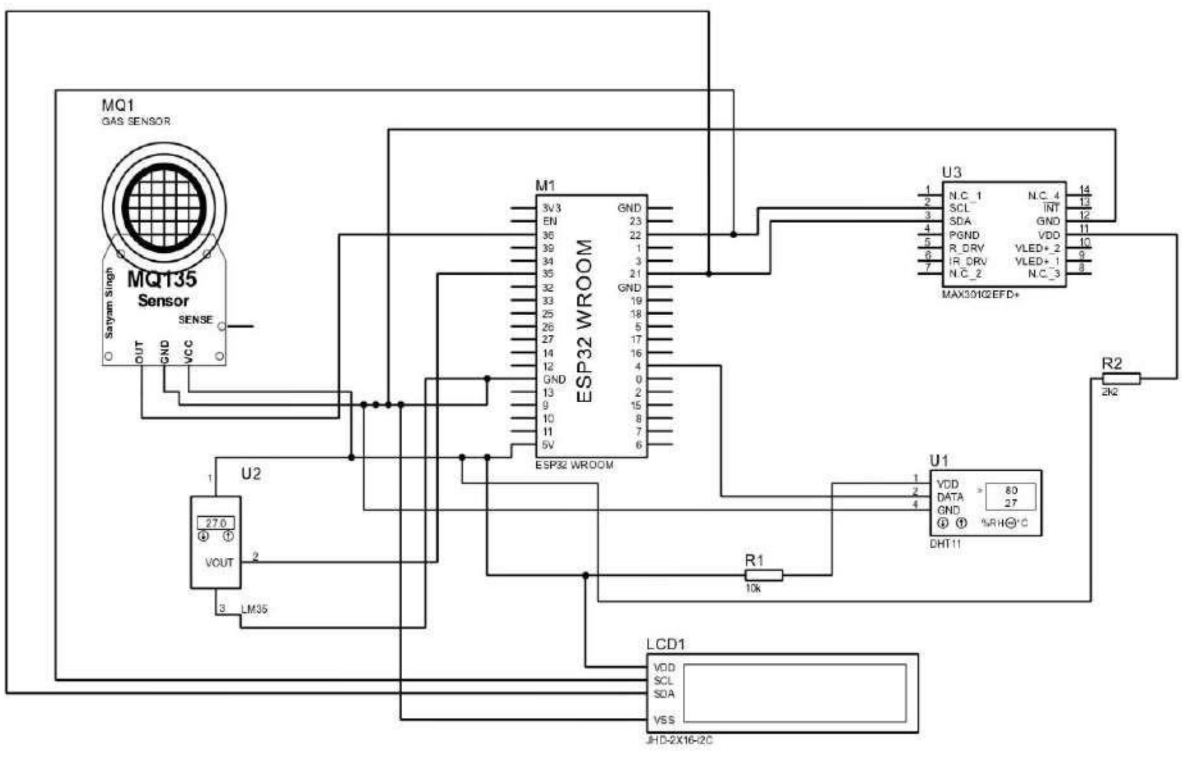


Figure 4.3 Proteus Circuit diagram

The circuit diagram presented in Figure 4.3 showcases the hardware design implemented in Proteus software. The design consists of a microcontroller and several sensors and interconnected to perform a particular function. This software enables the designer to create a virtual prototype of the hardware design, simulate its behavior, and test its performance before producing a physical prototype. By doing so, the designer can detect and rectify any potential issues in the design, thus minimizing the risk of failure during implementation [25]. The circuit presented provides a concise overview of the pin configurations required for the interfacing several sensors and modules with the ESP 32 microcontroller board. The MAX 30102 intended for pulse oximetry and heart-rate monitoring, necessitates a 5v power supply connected to its VCC pin. The SCL and SDA pins are employed for I2C communication and must be connected to GPIO22 and GPIO21, respectively. The GND pin must be connected to ground to complete the circuit. For Temperature measurements, the LM35 temperature sensor requires its VOUT pin to be connected to GPIO35, while its VCC pin must be connected to a Vin pin for power input. Ground must be connected to the GND pin. The MQ-135, sensor used to measure air quality, necessitates its AOUT pin connected to GPIO32 to receive an analog signal. Its VCC and GND pins should be connected to VIN and ground, respectively. The DHT11 sensor, used for humidity and temperature measurements, requires its VCC pin to be connected to VIN for power input, with its data pin connected to D4 (a GPIO pin), and the GND pin connected to ground. Finally, for display purposes, the I2C LCD module requires the GND pin to be connected to ground, with the VCC pin connected to Vin for power input. The SDA and SCL pins must be connected to GPIO21 and GPIO22, respectively, for I2C communication. Each sensor is connected with a microcontroller ESP32 Wi-Fi module which controls these sensors and making a connection with IP. AC Power supply is used by the connecting components for supplying electrical energy. Sensors that sense values from the human body and surrounding environment and these sensed values are send to the cloud using microcontroller and these sensed values are also displayed in the 16x2 LCD display. The microcontroller that sends the measured values to the firebase and ThingSpeak cloud. Proper configuration of these sensors and modules to the ESP32 microcontroller board provides an excellent platform to design and implement Internet of Things based various applications like IoT based Real Time Health Monitoring with precision and efficiency.

4.1.3 Arduino IDE

Figure 4.4 Arduino IDE Software Logo

The Arduino Integrated Development Environment (IDE) is a software application that enables users to write, compile, and upload code to Arduino boards. Arduino is an open-source platform that provides microcontrollers for creating electronic devices. The IDE offers an intuitive and user-friendly interface for developing Arduino-based applications. The IDE is coded in Java and can be used on various operating systems such as Windows, Mac OS, and Linux. It features a wide range of libraries and examples to simplify the development process for Arduino projects. Additionally, it includes a serial monitor to assist in debugging and communication between the computer and the Arduino board. Users can write code in C/C++ language using the built-in text editor of the Arduino IDE, which supports syntax highlighting, code folding, and auto-completion features.

The Arduino IDE supports a wide range of Arduino boards, including popular models such as Arduino Uno, Mega, Leonardo, and Nano, as well as other boards like ESP8266, ESP32, and STM32. Typically, developing an Arduino project using the IDE involves writing the code, verifying the code's syntax and functionality, and uploading it to the board using the built-in compiler and uploader. The Arduino community is extensive and offers an abundance of support and resources for users. Users can access a wealth of libraries, examples, add-ons, forums, tutorials, and online resources to extend the functionality of the IDE and help new users learn and create projects using the Arduino platform. In summary, the Arduino IDE is a powerful tool for developing and programming Arduino-based projects. Its user-friendly interface, extensive libraries, and compatibility with a wide range of boards make it a popular choice for both beginners and advanced users. The vast community of users and developers provides an abundance of resources and support to assist users in creating innovative projects with the Arduino platform.

4.1.4 ThingSpeak

ThingSpeak

Figure 4.5 ThingSpeak Software Logo

ThingSpeak offers a range of features, including the ability to schedule events, use RESTful and MQTT APIs, collect data in private channels, and share data in public channels. In addition, the platform provides powerful MATLAB analytics and visualizations to help you make sense of your data. ThingSpeak collects data from sensors or actuators, processes and displays it, and can trigger actions based on the data. It has been utilized in various IoT projects. To gain access to the website, a new account was created and a channel was set up with six fields: Displaying Body Temperature, Heart rate, Blood oxygen, Room temperature, Room humidity and Air Quality. An API key unique to the channel was provided, which was then incorporated into the code using the Arduino IDE and uploaded to the Node MCU microcontroller board.

4.1.5 IFTTT

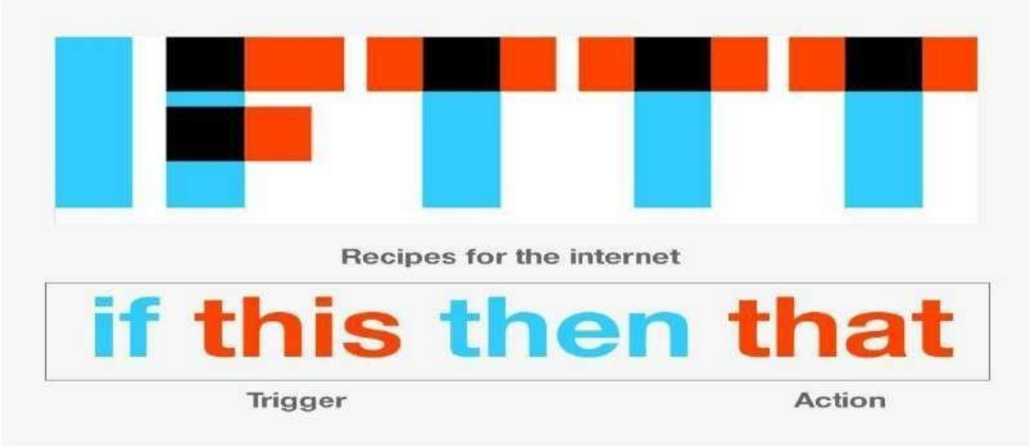


Figure 4.6 Diagram Representation of IFTTT

IFTTT is an online platform that enables users to automate tasks across different internet-connected devices and services. The platform's name stands for "If This Then That," which reflects its fundamental principle that users can create customized "recipes" that connect different services and devices based on a specific trigger and action. To start using IFTTT, users first need to create an account on the platform's website. Once they have an account, they can create recipes that consist of a trigger and an action. The trigger is an event that happens on one service or device, such as receiving a new email or a new tweet. The action is the task that IFTTT performs on another service or device, such as sending a text message or creating a calendar event. IFTTT supports a broad range of services and devices, including popular options such as Gmail, Dropbox, Twitter, and Philips Hue lights. The platform also integrates with many IoT devices, such as smart thermostats, home security systems, and fitness trackers. For example, a user can create a recipe that automatically saves all their Gmail attachments to a specific Dropbox folder. The trigger for this recipe would be receiving a new email with an attachment, and the action would be to save that attachment to a predetermined folder in the user's Dropbox account. In addition to the web-based service, IFTTT provides a mobile app for iOS and Android devices, allowing users to create and manage their recipes on the go. The app also provides access to "Do" recipes, which enable users to perform simple actions with a single tap, such as sending a text message or turning off the lights. Furthermore, IFTTT offers a range of "channels" that enable developers to integrate their own services and devices with IFTTT. Developers can create custom triggers and actions that allow their services and devices to work with other services and devices in new and innovative ways. Overall, IFTTT offers users a flexible and easy-to-use platform for automating tasks across different internet-connected devices and services. The platform's popularity is due to its integration capabilities and versatility, making it a popular choice for individuals and developers looking to optimize their workflows and enhance their productivity.

4.2 Components costs for our proposed project

TABLE 4.1: Cost of Components

Component	***Model***	***Quantity***	***Price / unit (BDTK)***	***Price (BDTK)***
Microcontroller	*ESP32*	*1*	*380*	*380*
Heart Rate Sensor	*MAX-30102*	*1*	*140*	*140*
Temperature Sensor	*DHT-11*	*1*	*90*	*90*
Air Quality Sensor	*MQ-135*	*1*	*105*	*105*
Jumper Wires	*Generic*	*3(20 per unit)*	*20*	*60*
Breadboard	*Large*	*1*	*250*	*250*
Total Cost				*1025*

The main motive of our project is to build a low cost IoT based real time health monitoring so that it can be useful for many people who could not able to spend more cost for health care and to avoid huge charges in hospitals. Our proposed system cost of components as shown in table 4.1 and details of costs for each hardware component purchased is described.

Chapter 5

5 Result and discussion

5.1 Patient Health Monitoring using IoT

This paper discusses about the Design and Development of IoT based health monitoring system that utilizes various sensors to measure a person or patient's heart rate, blood oxygen, body temperature, room humidity, room temperature, and air quality. Though testing, we found that our proposed system allows for real-time monitoring of a person's body health.

5.1.1 Visualizing the Sensed Values of PHM

The Heart rate and Blood Oxygen sensed by MAX30102 sensor, DHT-11 that senses Room Temperature and Room Humidity, LM35 that senses the body temperature along with MQ135 that senses the surrounding air quality level and all these data are processed in ESP 32 Wi-Fi module and it is also interfaced along with 16X2 LCD module which displays our sensed values in the Liquid Crystal Display.

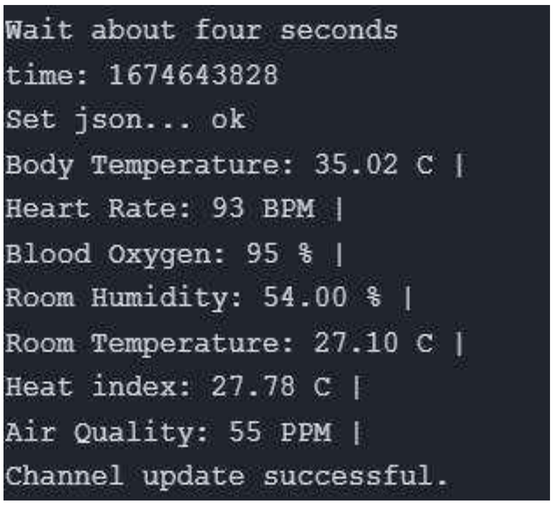


Figure 5.1 Displaying sensed values in the Arduino ide serial monitor

The Arduino ide serial monitor display with a baud rate of 115200 as shown in Fig 5.1 describes the measured values of our body's health condition. Baud rate is a measure of how many signal changes per second occur when data is transmitted through a medium. A higher baud rate indicates faster data transfer during transmission and reception.

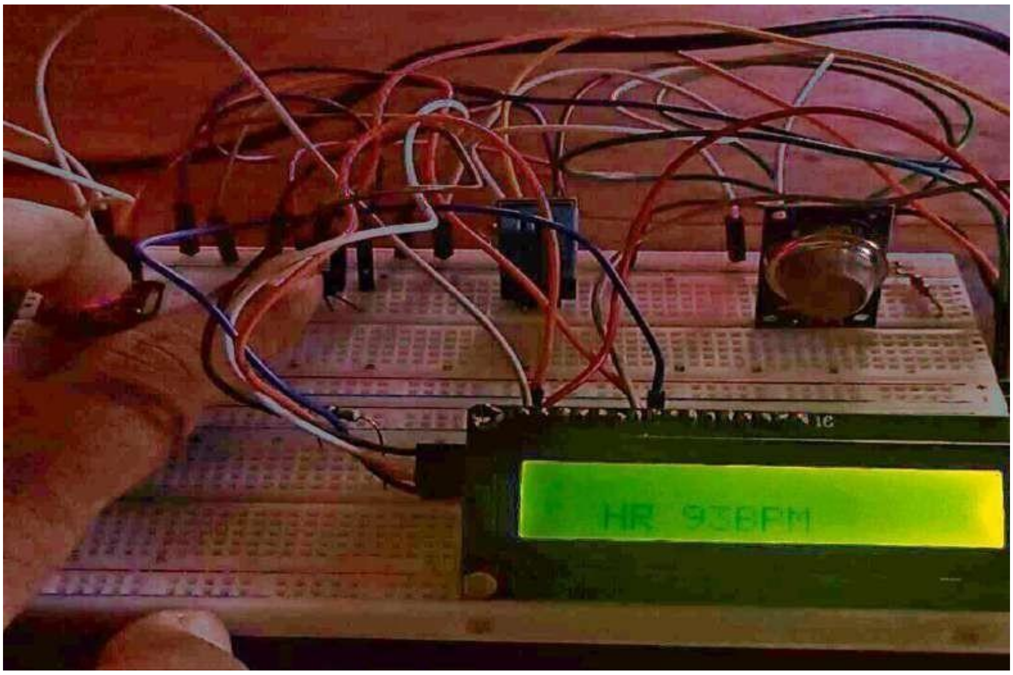


Figure 5.2 Heart Rate Sensed as 93 BPM in LCD Display

After the various sensors senses the heart rate, blood oxygen, body temperature, room humidity, room temperature and air quality and microcontroller that controls the sensors can display the sensed values in the 16x2 i2c LCD Display. When a person is sensed by the sensors there sensed values are displayed in the real time as Fig 5.2. shows the sensed value of a person's heart rate as 93 BPM. Similarly, as shown in Fig 5.3. all the sensed value are displayed in LCD Display module withbody temperature as 35.02 celsius, heart rate as 93 beats per minute (bpm), blood oxygen as 95%, room humidity as 54%, room temperature as 27.10 degree celsius and air quality as 55 parts per million (ppm).

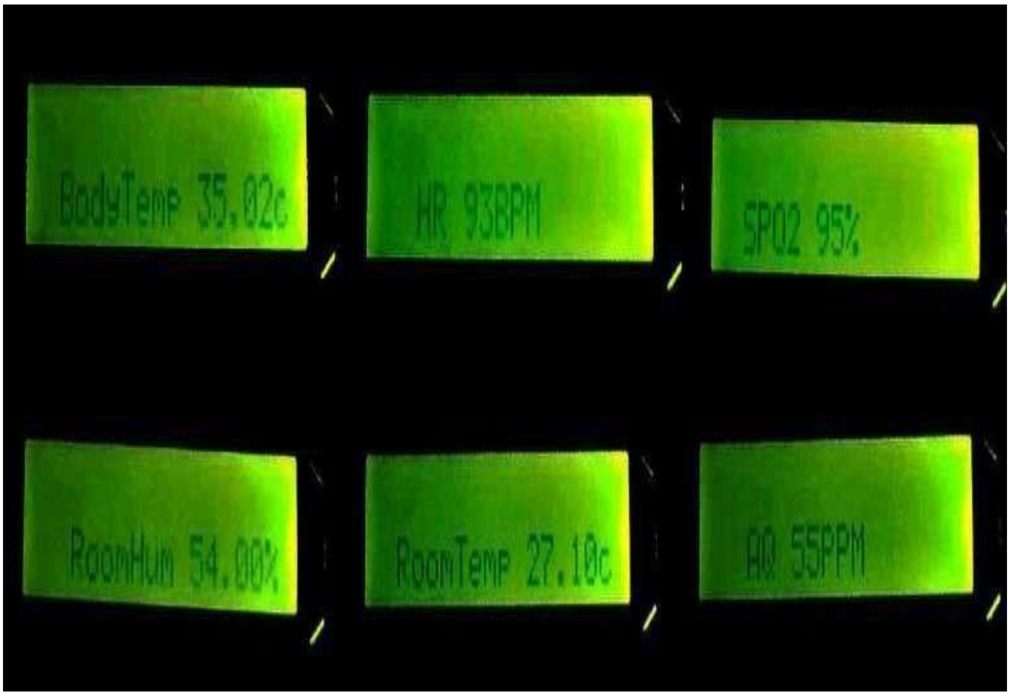


Figure 5.3 All Sensed Values Displayed in LCD Display

The sensed data are sent to the ThingSpeak cloud by ESP module which visualizes our sensed data and it integrated along with IFTTT platform that sends an emergency SMS alert when the patient health condition is critical to ourselves/doctors/caretakers so that necessary precautions can be taken for our health as soon as possible. These sensed values are also sent to the created website by firebase database that stores our real time sensed values. We can able to further monitor the real time sensed values automatically for every time when the sensor senses the values. We can further able to delay the displaying sensed values according to our wish by changing it in the arduino ide code. The visual representation of the Body Temperature and Heart Rate values as shown in Fig 5.4. that were sensed by our system. This allows for easy monitoring and analysis of the data, providing a visual representation of the changes in these health indicators over time. In the above graph we can able to visualize that body temperature of one particular point shows us that it sensed 33.94 degree Celsius and at the same we can able to visualize the heart rate is sensed as 88 beats per minute with respect to date and time.

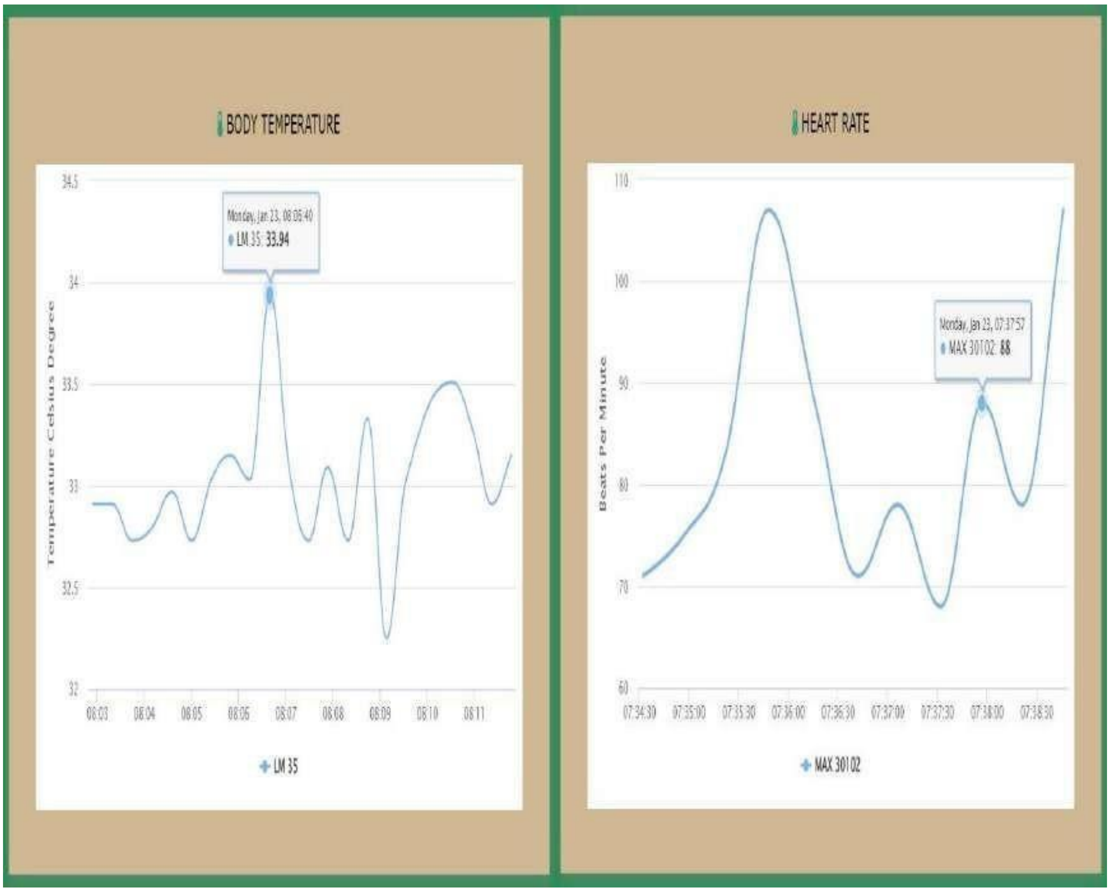


Figure 5.4 Graphical representation of the sensed values through our created website

The real time health monitoring of our sensed values in the created website in a numerical format representation is shown as in Fig 5.5. It describes the sensed values of the person’s body temperature as 33.70 degree celsius, heart rate as 75 beats per minute, blood oxygen(SpO2) as 99%, Room temperature as 26.20 degree celsius, Room humidity as 68% and Air quality as 65 PPM.

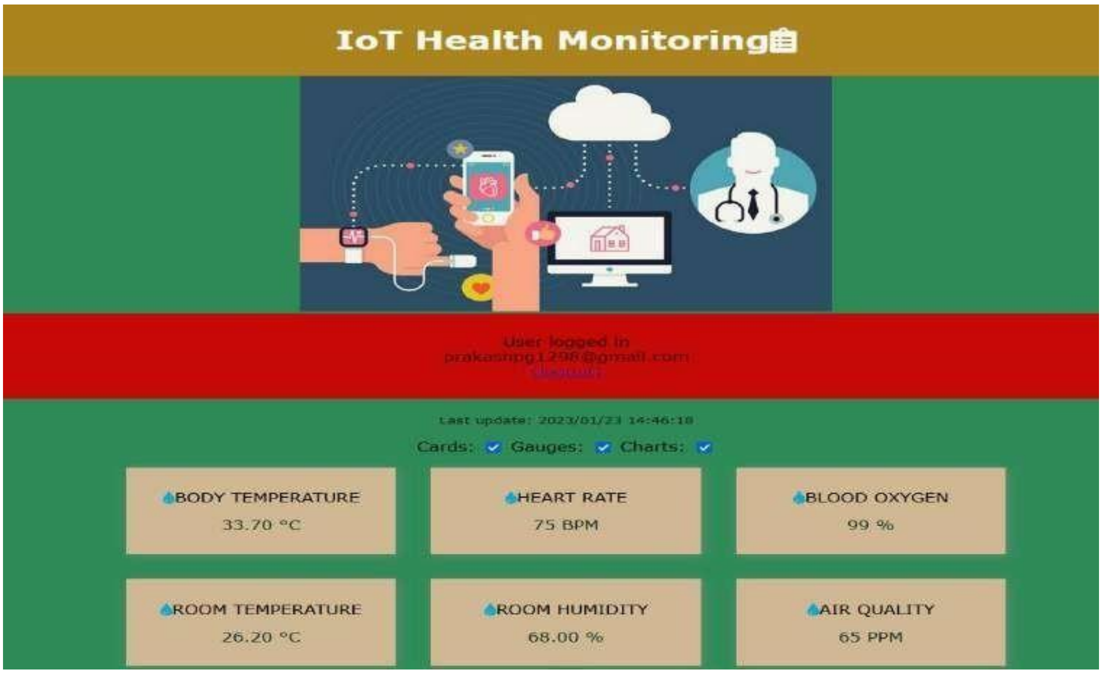


Figure 5.5 Displaying the real time sensed values in the created website

The tabular representation of some of the sensed values in the created website as shown in table 5.1 consits of updated time with date, month and year, Body Temperature, Heart Rate, Blood Oxygen, Room Humidity , Room Temperature and Air Quality. For every 24 seconds gap the sensed values is updated in the tabular column in the website. It will be more useful and easy for the users to monitor the real time data of the persons or patients in more efficient way

TABLE 5.1: Tabular Representation of the sensed values in the created website

Time	Body Temp (°C)	HR (BPM)	SPO2 (%)	Room Temp (°C)	Room Hum (%)	Air Quality (PPM)
2023/01/22 20:09:49	34.72	71	98	25.80	60.00	60
2023/01/22 20:09:24	34.78	88	98	25.80	60.00	61
2023/01/22 20:09:00	34.60	78	97	25.80	60.00	59
2023/01/22 20:08:35	34.36	83	100	25.80	60.00	60
2023/01/22 20:08:11	34.66	65	98	25.80	60.00	61
2023/01/22 20:07:46	34.66	107	100	25.80	60.00	60
2023/01/22 20:07:21	34.84	83	100	25.80	60.00	61
2023/01/22 20:02:01	34.18	83	87	25.80	60.00	61
2023/01/22 20:01:36	34.24	78	92	25.80	60.00	61
2023/01/22 20:01:12	34.60	75	99	25.80	60.00	63
2023/01/22 19:18:30	32.73	81	99	26.70	60.00	56
2023/01/22 19:18:05	33.82	82	99	26.70	66.00	62
2023/01/22 19:17:41	33.76	82	99	26.20	76.00	62
2023/01/22 19:17:16	33.94	78	100	25.80	59.00	62
2023/01/22 19:16:50	33.88	78	99	25.80	59.00	60
2023/01/22 19:16:25	33.94	78	99	25.80	60.00	60
2023/01/22 16:23:34	32.67	100	100	26.70	70.00	56
2023/01/22 16:23:10	32.61	88	99	26.70	71.00	57
2023/01/22 16:22:45	32.61	83	100	26.70	71.00	55

ThingSpeak provides a login forum for security purposes, with the username and password only known to the patient, medical professional or caretakers administering their care, which enables them to view the patient's data (readings) while safeguarding their medical information against hacking and leaks [17]. Figure 10 represents the displaying of sensed values in ThingSpeak and we can able to visualize the sensed values in graphical representation as body temperature as 35.68 degree Celsius, heart rate as 78 beats per minute, blood oxygen as 99%, room humidity as 71%, roomtemperature as 25.8 degree Celsius, and air quality as 59 PPM.

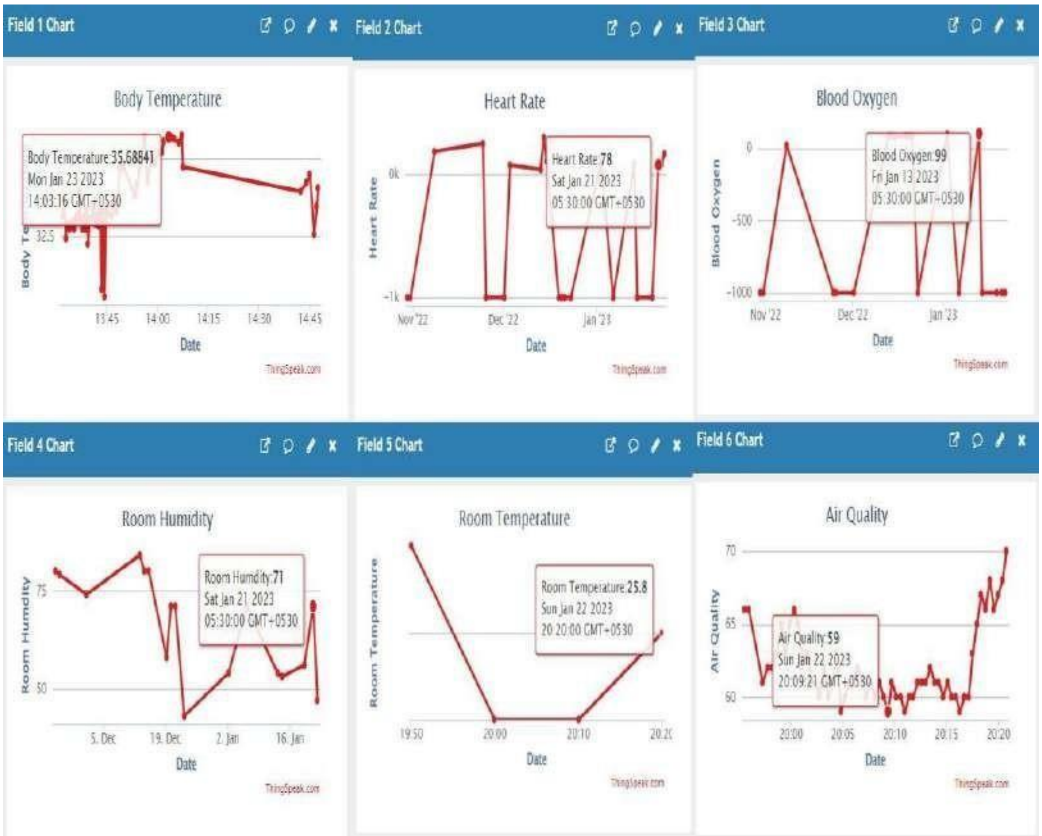


Figure 5.6 Displaying the sensed values in ThingSpeak Platform

We can able to visualize the sensed values changes in the graph according to changes in values with x axis as date and y axis as different values sensed with respect to time and date as shown in Fig 5.6. As we Discussed Earlier with ThingSpeak, we have access to an IoT analytics platform that enables us to consolidate, display, and examine real-time data streams in the cloud. Our devices can transmit data to ThingSpeak, and we can immediately create visualizations of this data and even set up alerts. ThingSpeak is a time-series database. Each channel can include up to eight data fields

With ThingSpeak, we have the capability to:

a. Gather - Securely transfer sensor data to the cloud.
b. Analyze - Use MATLAB to interpret and visualize the data.
c. React - Trigger a response based on the data collected.

TABLE 5.2: The Real time data collected from ThingSpeak cloud and exported to an Excel Sheet for further analysis

SL	Heart Rate in BPM	Blood Oxygen (SPO2) %	Body Temperature In degree C	Room Humidity	Room Temperaturein degree C	Air Quality (CO2)
1	80	99	36.32	58	29.6	58
2	78	96	38.15	57	29.4	57
3	76	98	33.65	56	29.6	56
4	82	96	39.26	58	29.8	58
5	84	98	35.08	59	28.4	59
6	92	95	32.22	59	29.8	59
7	76	97	39.43	59	29.6	59
8	86	99	37.65	59	29.8	59
9	94	96	34.22	59	29.4	59
10	73	99	32.13	59	28.3	59
11	84	96	30.12	60	28.4	60
12	89	99	38.18	68	28.6	68
13	78	96	36.12	60	28.6	60
14	94	92	35.16	60	28.4	60
15	81	97	34.15	61	28.8	61
16	76	100	32.18	61	28.4	61
17	82	100	32.88	61	28.6	61
18	93	97	32.65	69	28.4	69
19	81	96	39.12	61	28.2	61
20	107	99	38.08	61	27.4	61
21	85	96	37.14	67	27.2	67
22	89	100	36.07	61	26.6	61
23	82	99	36.28	69	27.2	64
24	103	96	35.35	72	26.0	62
25	96	99	35.98	76	25.8	62
26	81	100	33.15	78	26.0	62
27	84	98	32.22	82	26.2	66
28	71	99	32.36	76	25.4	62
29	92	96	31.58	72	24.2	63
30	88	99	32.45	68	25.6	65

We can able to export the data from our Thing speak channel to a CSV file by selecting My channels under Channels and select the data that needs to be exported / imported and the above table 5.2. describes the sensed values that is derived from the ThingSpeak platform and we can able to download the exported csv file for various purposes such as data analytics and machine learning process if needed.

5.1.2 SMS Alert Messages Sent Via IFTTT

The SMS alert messages as shown in Fig 5.7 represents body temperature as 39.01 degree Celsius, Heart rate as 107 beats per minute, blood oxygen as 92%, room humidity as 76%, abnormal surrounding temperature, air quality as 67 ppm.

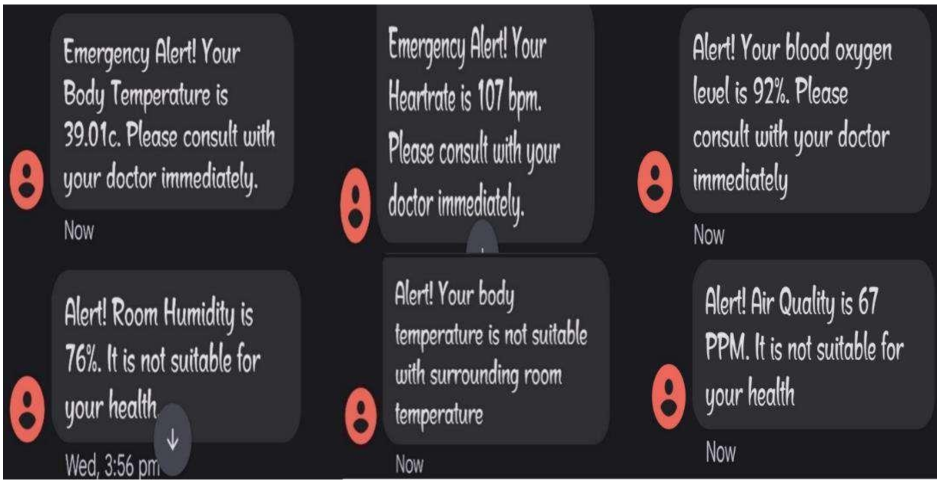


Figure 5.7 SMS Alert Messages Triggered by IFTTT

When the sensed values is set to a particular value for Triggering in ThingSpeak React and Things HTTP using IFTTT URL and when it reaches the trigger value then an applet created using IFTTT will send alert messages via SMS to the registered mobile number. This will be useful for doctors, caretakers as well as individuals for those monitoring the real time health condition.

5.2 Person Health Condition Detection using ML

In this study, as we discussed before we aimed to predict the health condition of unknown patients using four different machine learning models. The performance of these models was evaluated by comparing their accuracy, precision, recall, and F1-score. To begin-with, we determined the best hyperparameters for each model using a grid search algorithm. These hyperparameters were then used to detect the health condition of the patients in our dataset. We can visualize the results through the Heatmap and ROC Curve. A Heat map is a graphical representation that utilizes colored cells to display the values of a two-dimensional data matrix. Heatmaps can be created using programming languages like Python or R, or visualization software like Tableau. Heatmaps are applicable to a wide range of data types, including geographical, financial, and scientific data, and can be generated using various software tools. Additionally, it is important to make the heatmap easily comprehensible and visually attractive, with well-labelled axes and color choices. The ROC (Receiver Operating Characteristic) curve is a visual representation of the performance of a binary classifier system when its discrimination threshold is altered. By plotting the true positive rate (TPR) against the false Positive rate (FPR) at different threshold values, the ROC curve provides a way to evaluate a classification model’s performance. The AUC (Area Under the Curve) score is a measure of the classification model’s performance at all classification thresholds.

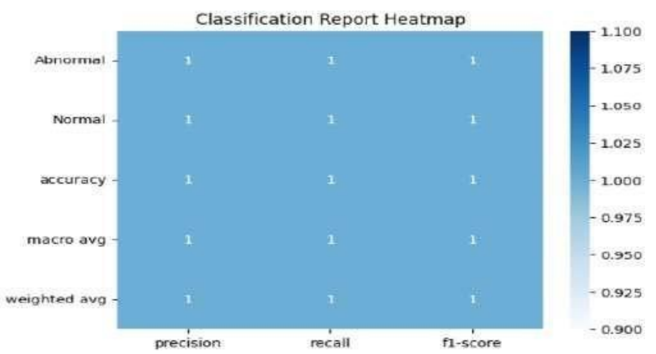
5.2.1 Training 70 and testing 30 datasets:

The results as shown in Table 5.3 revealed that Gradient Boosting achieved a perfect accuracy of 100%, indicating that they correctly classified all instances in the dataset. Similarly, their precision, recall, and F1 scores are also 1.00, indicating that they achieved perfect performance in all these metrics. The Random Forest and Decision Tree achieved an accuracy of 0.97, indicates that it correctly classified 97% of the instances in the dataset for its precision, recall with F1 score is 0.96 and 0.98 precision and recall score in random forest indicates it achieved a good balance between precision and recall and MLP model achieved an accuracy of 0.98, indicating that it correctly classified 98% of the instances in the dataset. Its precision score is 0.99, indicating that it predicted positive instances with 99% accuracy. Its recall score is 0.99, indicating that it correctly identified all positive instances in the dataset. Its F1 score is 0.98, indicating that it achieved a good balance between precision and recall.

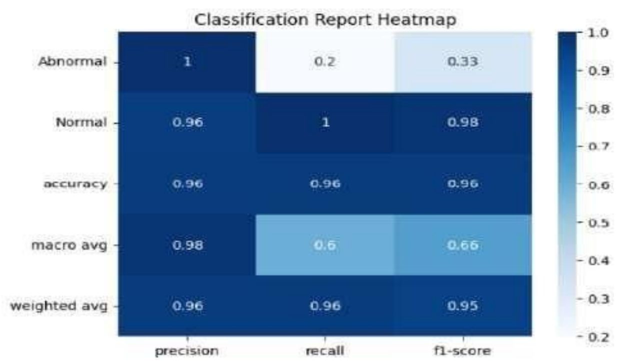
TABLE 5.3: Accuracy Score for Training 70 and Testing 30 Datasets

Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Gradient Boosting	1.00	1.00	1.00	1.00
Random Forest	0.97	0.98	0.98	0.97
MLP	0.98	0.99	0.99	0.98
Naïve Bayes	0.96	0.97	0.97	0.96
SVM	0.95	0.96	0.96	0.95
Decision Tree	0.97	0.97	0.97	0.96

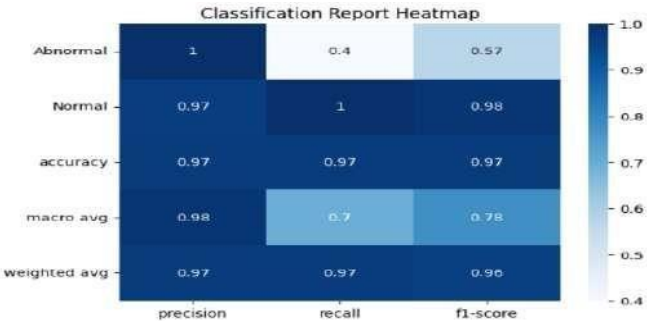
The naïve bayes model achieved an accuracy of 0.96, indicating that it correctly classified 96% of the instances in the dataset. Its precision score is 0.97, indicating that it predicted positive instances with 97% accuracy. Its recall score is 0.97, indicating that it predicted positive instances with 97% accuracy. Its recall score is 0.97, indicating that it correctly identified positive instances in the dataset. Its F1 score is 0.96, indicating that it achieved a good balance between precision and recall.



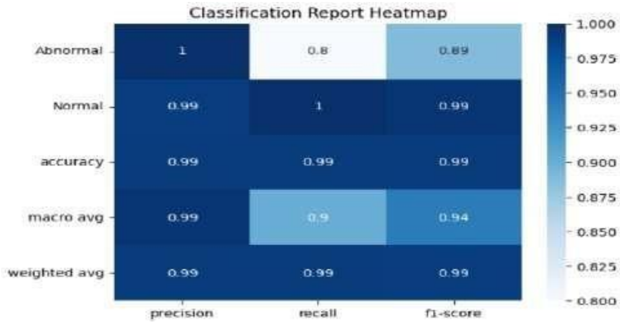
(a) GB



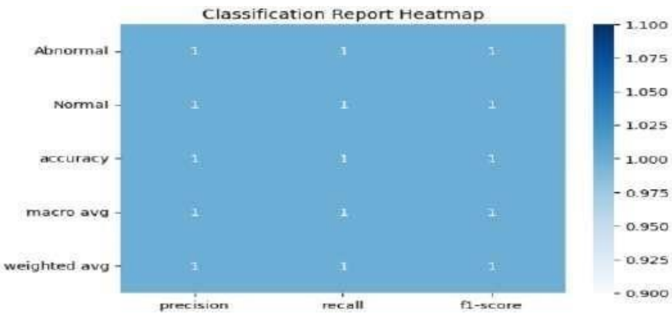
(b) SVM



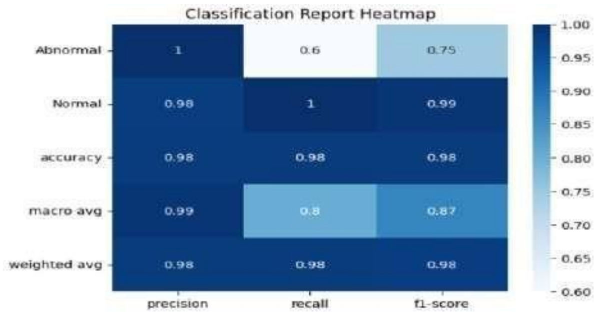
(c) DT



(d) NB



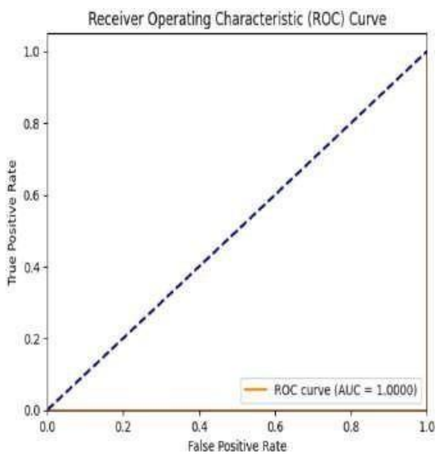
(e) MLP



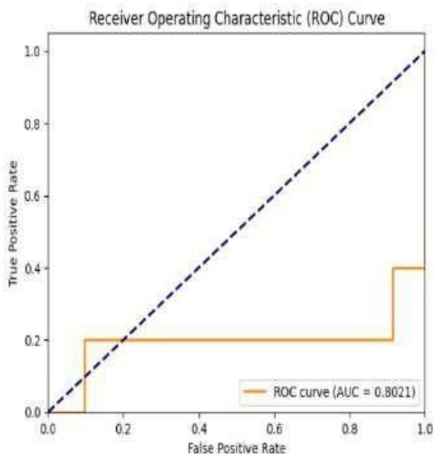
(f) RF

Figure 5.8 Visualization of Heat Map for Training 70 and Testing 30 Datasets

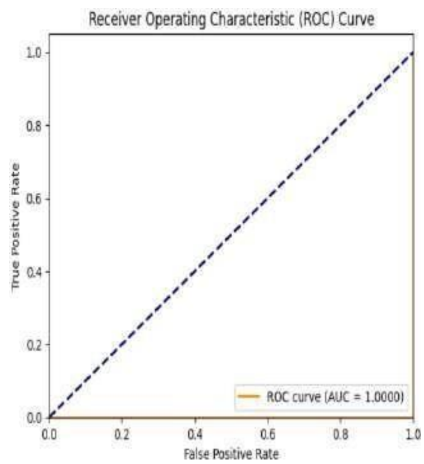
Finally, the SVM model achieved an accuracy of 0.95, indicating that it correctly classified 95% of the instances in the dataset. Its precision score is 0.96, indicating that it predicted positive instances with 96% accuracy. Its recall score is 0.96, indicating that it correctly identified 96% of the actual positive instances in the dataset. Its F1 score is 0.95, indicating that it achieved a good balance between precision and recall, but was slightly lower than the other models in this metric and we can visualize through heatmap as shown in Fig 5.8.



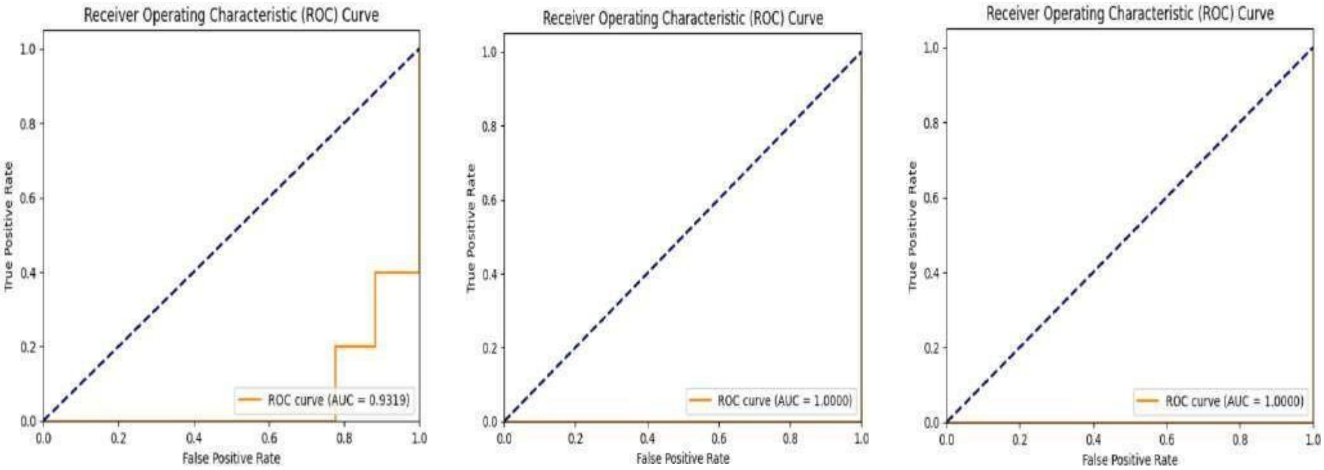
(a) GB



(b) SVM



(c) DT



(d) NB (e) MLP (f) RF

Figure 5.9 Visualization of ROC Curve for Training 70 and Testing 30 Datasets

In this study, the performance of six models in classifying the data was assessed by plotting their receiver operating characteristics (ROC) curves and calculating their area under the curve (AUC) values as shown in Fig 5.9. All the models have an AUC score of 1.000, except for the Support Vector Machine (SVM) and Naïve Bayes models, which have AUC scores of 0.8021 and 0.9319, respectively. The ROC curves for Gradient Boosting, Decision Tree, MLP, and Random Forest models suggests that these models can perfectly suggest that these models can perfectly distinguish between positive and negative classes, with an AUC score of 1.000. This indicates that these models are ideal for binary classification tasks. On the other hand, the ROC curve for SVM model has decent accuracy in classifying the positive and negative classes, but it still has room for improvement. Similarly, the ROC curve for Naïve Bayes shows that the model can distinguish between positive and negative classes with good accuracy, but not as good as other models with AUC scores of 1.000. The AUC score of 0.9319 suggests that naïve bayes can still be optimized to improve its classification performance. Overall, the ROC curves and AUC score provides a comprehensive understanding of the model's classification performance. A model with an AUC score of 1.0 is ideal for classification tasks, while a lower AUC score may indicate the need for further optimization to improve the model's performance.

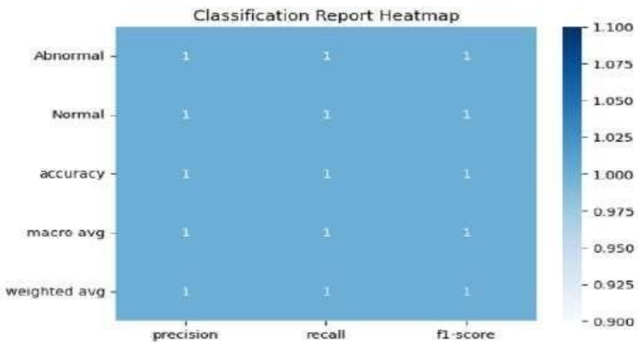
5.2.2 Training 90 and testing 10 datasets:

The results as shown in Table 5.4 based on the classification report, we can see that the Gradient Boosting model has the highest accuracy, precision, recall, and f1-score of 1.0, meaning that it has correctly classified all instances. This means that the model has a high level of accuracy and ability to identify positive instances. The Random Forest model achieved an accuracy of 0.97 and MLP achieved an accuracy of 0.98 across all metrics means that the model has a high level of accuracy and ability to identify positive instances, but may have misclassified some instances.

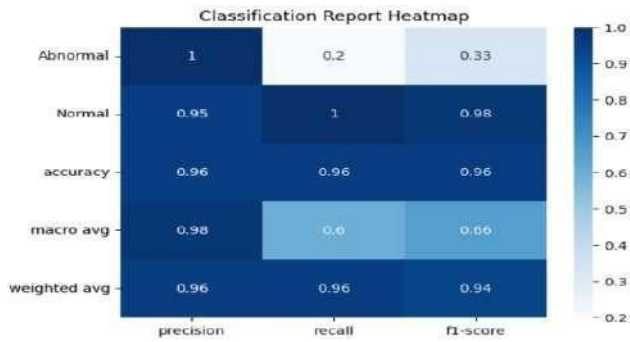
TABLE 5.4: Accuracy Score for Training 90 and Testing 10 Datasets

Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Gradient Boostir	1.00	1.00	1.00	1.00
Random Forest	0.97	0.97	0.97	0.97
MLP	0.98	0.98	0.98	0.98
Naïve Bayes	0.96	0.96	0.96	0.95
SVM	0.95	0.95	0.95	0.94
Decision Tree	0.98	0.98	0.98	0.98

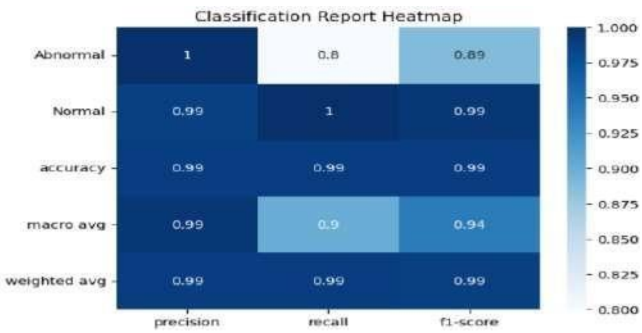
The naïve bayes model achieved an accuracy of 0.96 with same precision and recall score, and 0.95 for f1-score with SVM model achieving an accuracy of 0.95 with good precision, recall and 0.94 for f1-score and decision tree model has good precision, recall, and f1-scores of 0.98 across all metrics means that the model has a high level of accuracy and ability to identify positive instances, but may have misclassified some instances. Finally, the SVM model achieved an accuracy of 0.95, indicating that it correctly classified 95% of the instances in the dataset. Its precision score is 0.96, indicating that it predicted positive instances with 96% accuracy. Its recall score is 0.96, indicating that it correctly identified 96% of the actual positive instances in the dataset. Its F1 score is 0.95, indicating that it achieved a good balance between precision and recall, but was slightly lower than the other models in this metric and by heatmap visualization we can clearly visualize as shown in Fig 5.10.



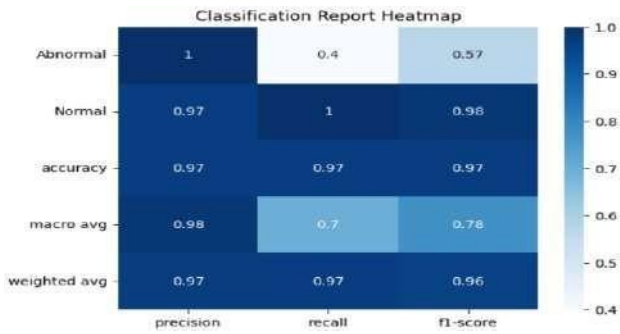
(a) GB



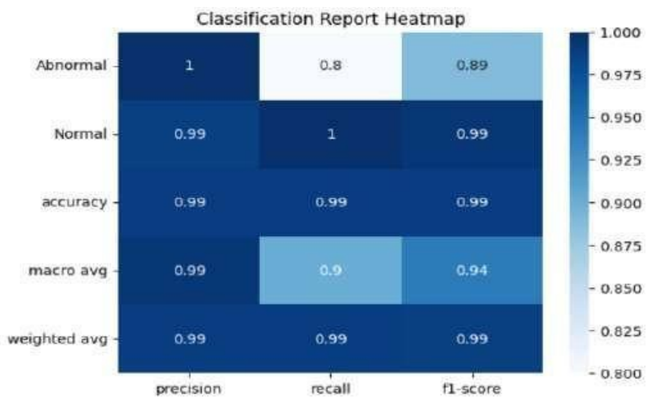
(b) SVM



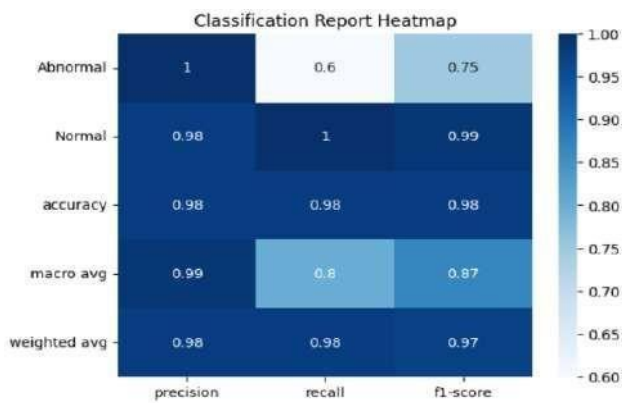
(c) DT



(d) NB



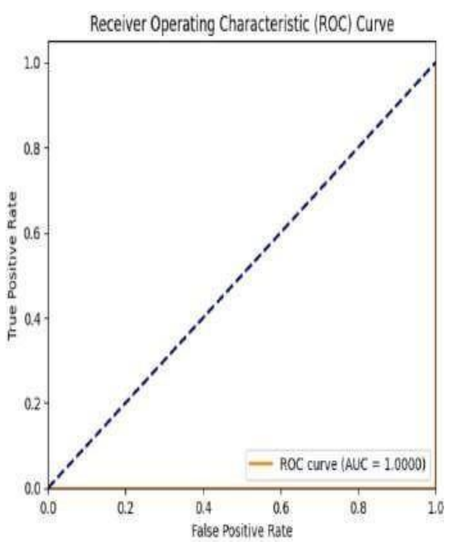
(e) MLP



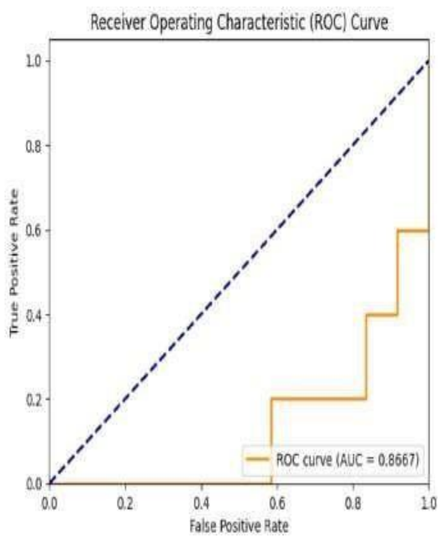
(f) RF

Figure 5.10 Visualization of Heat Map for Training 90 and Testing 10 Datasets

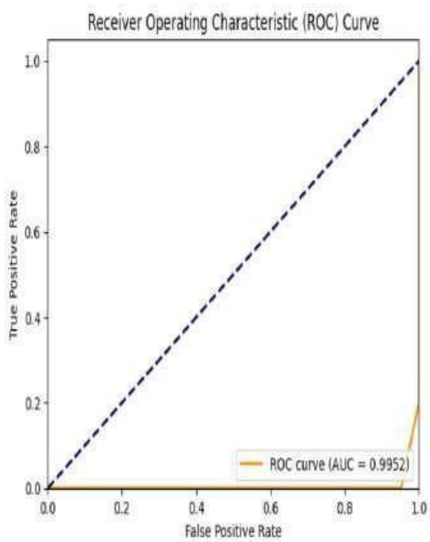
In this study, our results shows that the Gradient Boosting, MLP, and Random Forest models achieved perfect scores of 1.000 for the AUC, indicating excellent performance in correctly classifying positive and negative instances in the dataset. The Decision Tree and Naïve Bayes also demonstrated good performance with AUC scores of 0.9952 and 0.9881, respectively. However, the SVM model had a lower AUC score of 0.8667, indicating relatively weaker performance in identifying positive instances. Our findings suggest that the Gradient Boosting, MLP, and Random Forest models are highly effective in classifying positive instances in this dataset. The Decision Tree and Naïve Bayes models also demonstrate good performance, while the SVM model may not be the optimal choice for this particular classification task. These results have important implications for the development of machine learning algorithms and can inform future research in this area and visualization of roc curve is shown in Fig 5.11.



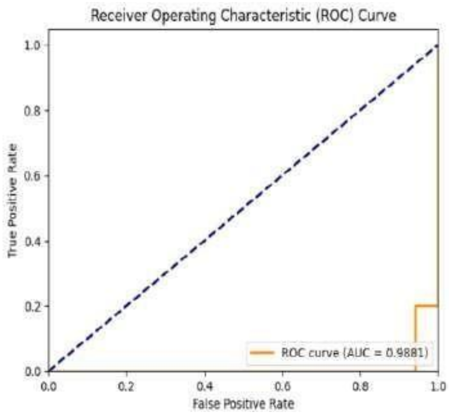
(a) GB



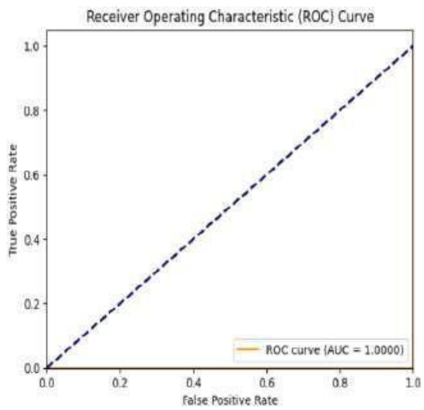
(b) SVM



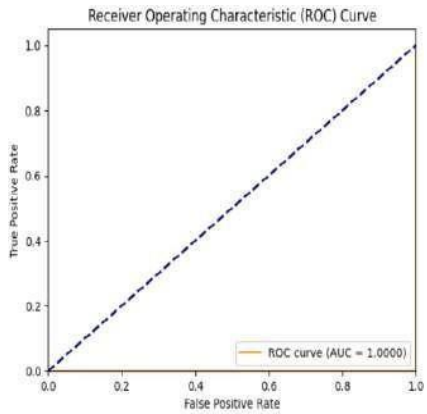
(c) DT



(d) NB



(e) MLP



(f) RF

Figure 5.11 Visualization of ROC Curve for Training 90 and Testing 10 Datasets

Naïve Bayes and SVM models have achieved an accuracy score, precision score, recall score, and f1-score of 0.93, which indicates that these models have classified 93% of the observations correctly and have achieved a precision, recall, and f1-score of 0.94. The Decision Tree model has achieved an accuracy score of 0.95 and precision, recall, and f1-scores of 0.96, indicating that this model has classified 95% of the observations correctly and has achieved a precision, recall, and f1-score of 0.96.

5.2.3 Training 50 and testing 50 datasets:

The results as shown in Table 5.5, can see that Gradient Boosting, Random Forest, and MLP models have achieved an accuracy score, precision score, recall score, and f1-score of 1.00, which indicates that these models have classified all the observations correctly and have achieved perfect precision, recall, and f1-scores. Overall, it appears that Gradient Boosting, Random Forest, and MLP models have performed the best on this dataset, achieving a perfect score on all metrics. These models can be considered as the best models for this particular classification problem and it can visualized more briefly in Figure 5.12 of Heatmap.

TABLE 5.5: Accuracy Score for Training 50 and Testing 50 Datasets

Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Gradient Boosting	1.00	1.00	1.00	1.00
Random Forest	1.00	1.00	1.00	1.00
MLP	1.00	1.00	1.00	1.00
Naïve Bayes	0.93	0.94	0.93	0.92
SVM	0.93	0.94	0.93	0.92
Decision Tree	0.95	0.96	0.96	0.95

Naïve Bayes and SVM models have achieved an accuracy score, precision score, recall score, and f1-score of 0.93, which indicates that these models have classified 93% of the observations correctly and have achieved a precision, recall, and f1-score of 0.94. The Decision Tree model has achieved an accuracy score of 0.95 and precision, recall, and f1-scores of 0.96, indicating that this model has classified 95% of the observations correctly and has achieved a precision, recall, and f1-score of 0.96.

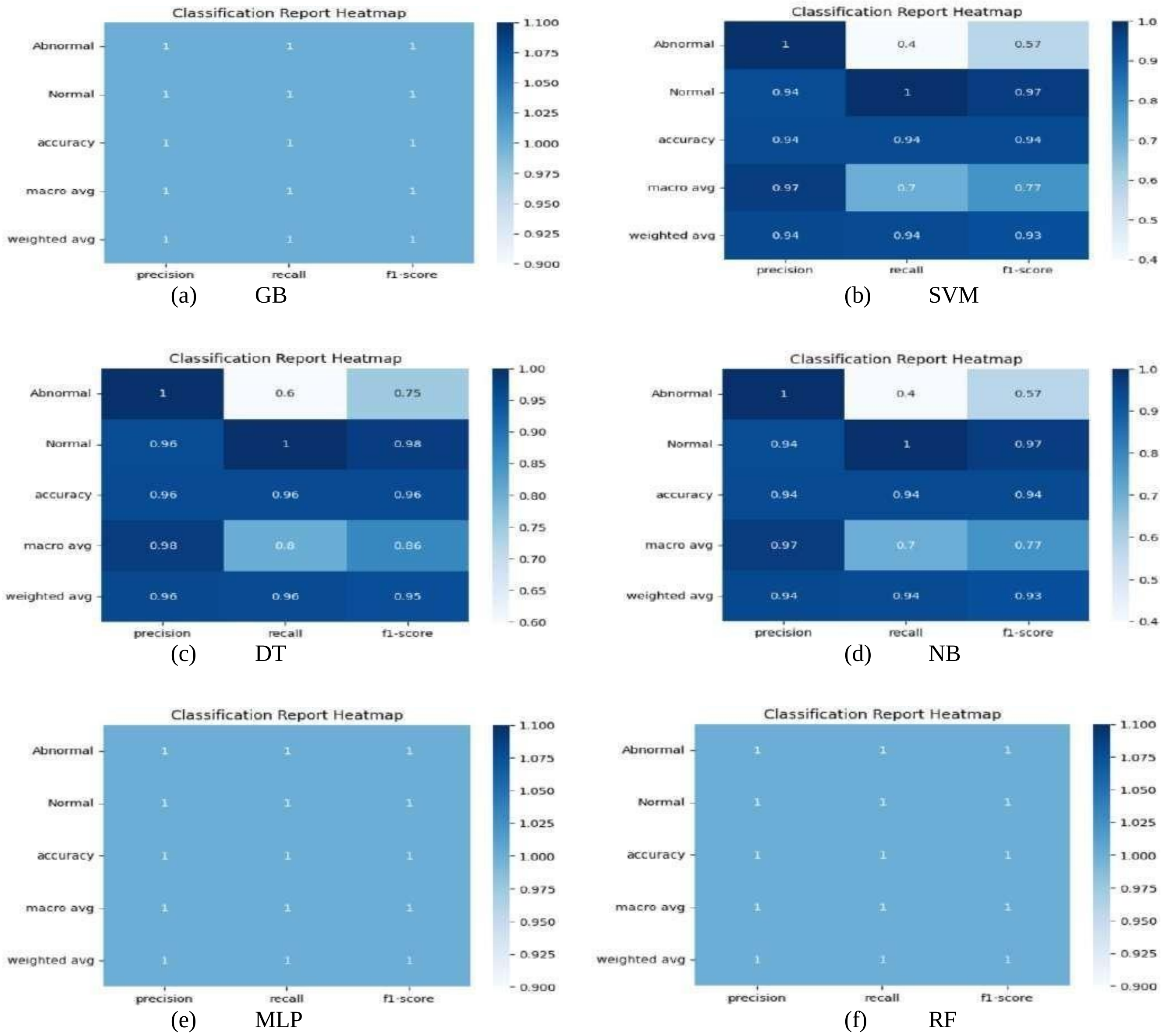
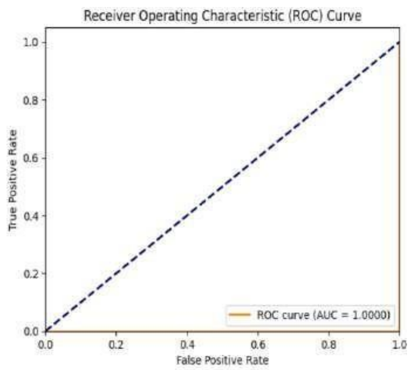
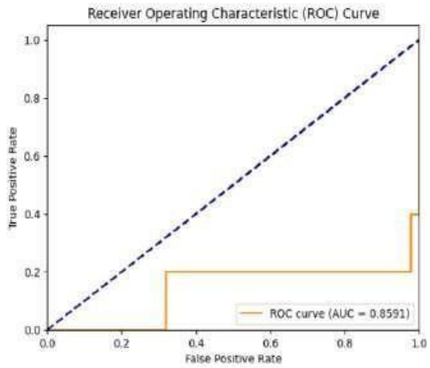


Figure 5.12 Visualization of Heat Map for Training 50 and Testing 50 Datasets

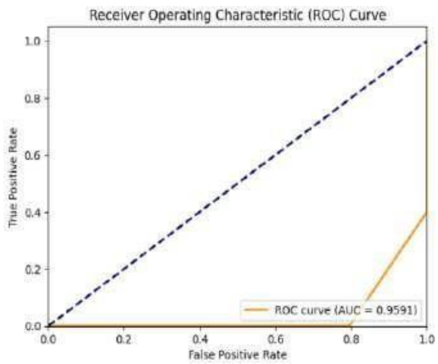
Overall, it appears that Gradient Boosting, Random Forest, and MLP models have performed the best on this dataset, achieving a perfect score on all metrics. These models can be considered as the best models for this particular classification problem and it can visualized more briefly in Fig 5.12 of Heatmap.



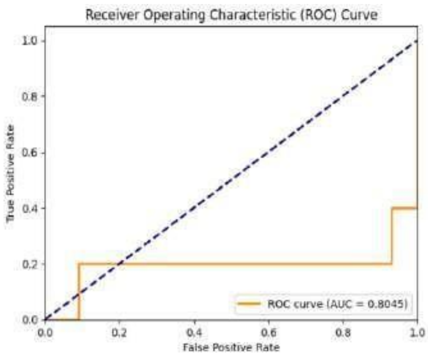
(a) GB



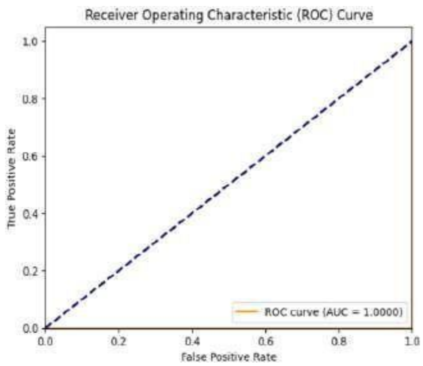
(b) SVM



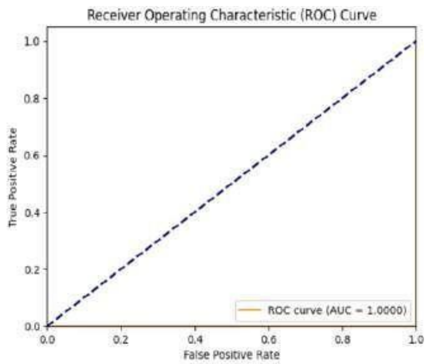
(c) DT



(d) NB



(e) MLP



(f) RF

Figure 5.13 Visualization of Heat Map for Training 50 and Testing 50 Datasets

In this study as shown in Fig 5.13 three models namely Gradient Boosting, MLP, and Random Forest, achieved a perfect AUC score of 1, implying that they can accurately discriminate between the two classes. The decision tree model performed well with an AUC score of 0.9591, indicating a strong ability to distinguish between the classes. The Support Vector Machine (SVM) model achieved an AUC score of 0.8591, which implies that it performed well but as not as well as the top-performing models. In contrast, the Naïve Bayes model has the lowest AUC score of 0.8045, indicating that it performed the least effectively among the models evaluated. These findings demonstrate that the Gradient Boosting, MLP, and Random Forest models are the most effective in distinguishing between the two classes, while the SVM, Decision Tree, and Naïve Bayes models are also effective but to a lesser extent.

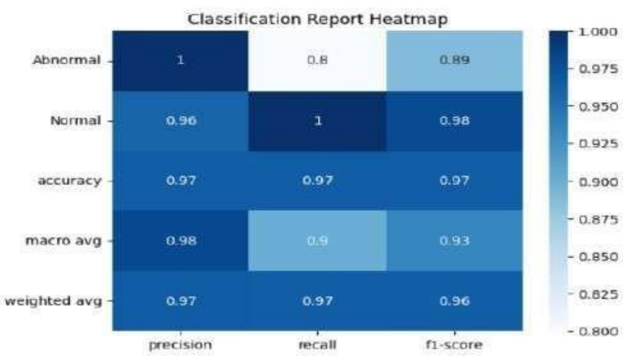
5.2.4 Training 30 and testing 70 datasets:

Based on the classification report, as shown in Table 5.6 the performance of six different classification models were evaluated in terms of their accuracy score, precision score, recall score, and f1-score. The result shows that the Gradient Boosting model outperformed the other models with an accuracy score of 0.96 and f1-score of 0.95. The Random Forest and MLP models also performed well with similar accuracy scores of 0.93 and f1-scores of 0.92. On the other hand, the Naïve Bayes and SVM models had similar performance with an accuracy score of 0.89 and an f1-score of 0.87.

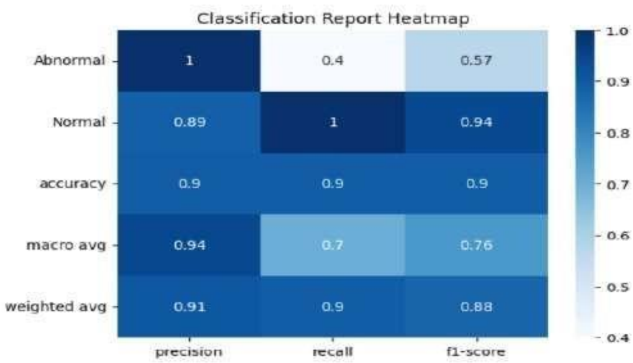
TABLE 5.6: Accuracy Score for Training 70 and Testing 30 Datasets

Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Gradient Boosting	0.96	0.96	0.96	0.95
Random Forest	0.93	0.93	0.93	0.92
MLP	0.93	0.93	0.93	0.92
Naïve Bayes	0.89	0.90	0.89	0.87
SVM	0.89	0.90	0.89	0.87
Decision Tree	0.86	0.84	0.86	0.84

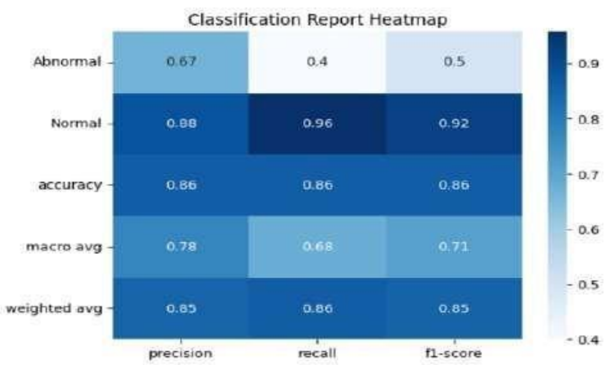
Finally, the Decision Tree model has the lowest performance with an accuracy score of 0.86 and an f1-score of 0.84. The highest accuracy and f1-scores of the Gradient Boosting model suggest that it is the best model for predicting the health condition of unknown patients. The accuracy score shows the overall correctness of the model's predictions, while the f1-score combines the precision and recall scores, which measures the balance between false positives and false negatives. Therefore, the higher f1-score of the Gradient Boosting model indicates that it achieved a better balance between precision and recall, which is crucial in predicting the health condition of unknown patients and it can be visualized clearly in Fig 5.14 of Heat Map



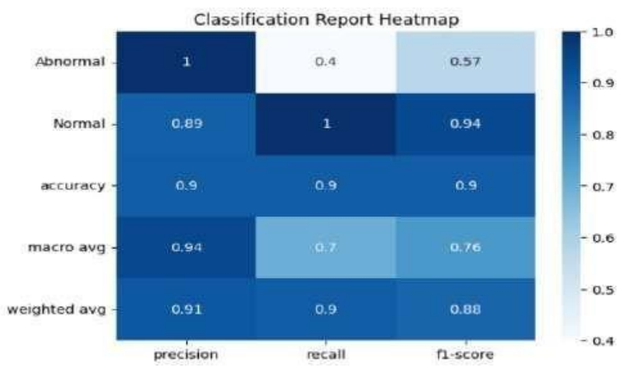
(a) GB



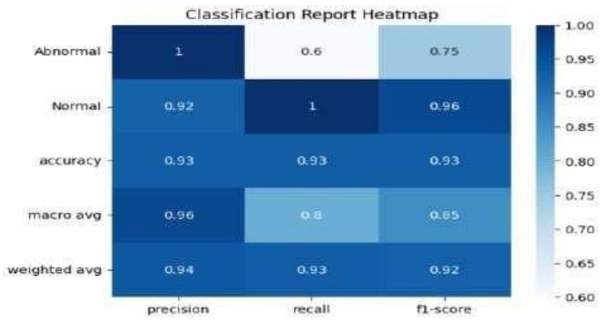
(b) SVM



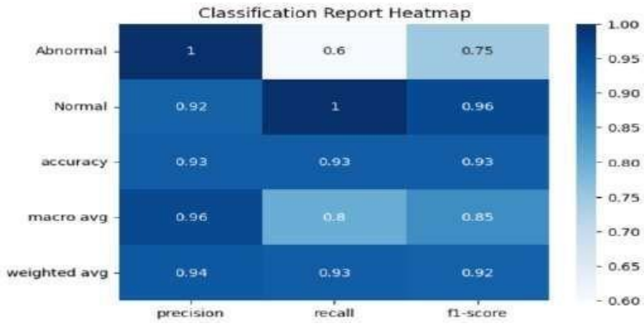
(c) DT



(d) NB



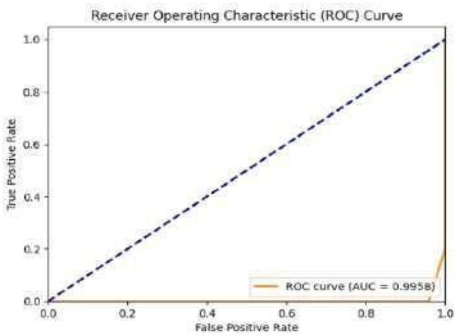
(e) MLP



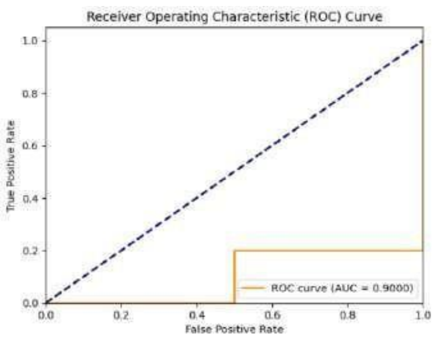
(f) RF

Figure 5.14 Visualization of Heat Map for Training 30 and Testing 70 Datasets

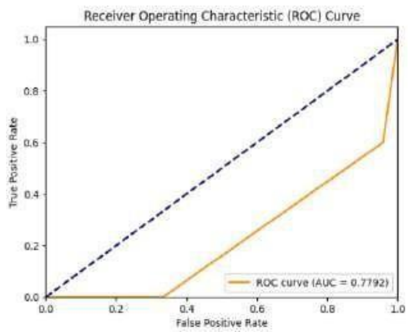
By visualizing this ROC Curve as shown in Figure 5.15, the AUC values, the Gradient Boosting and Random Forest models had the highest discriminative power with AUC values of 0.9958, while the MLP and Naïve Bayes models performed well with AUC values of 0.9833 and 0.9667, respectively. The Support Vector Machine model achieved a moderate AUC value of 0.9000, while the Decision Tree model has the lowest AUC value at 0.7792. These results suggest that the Gradient Boosting and Random Forest models are best performing models for accurately classifying patient health condition, followed by the Multilayer Perceptron and Naïve Bayes models.



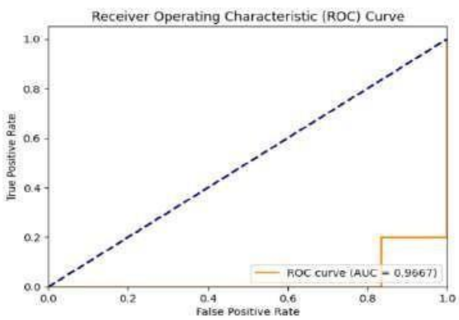
(a) GB



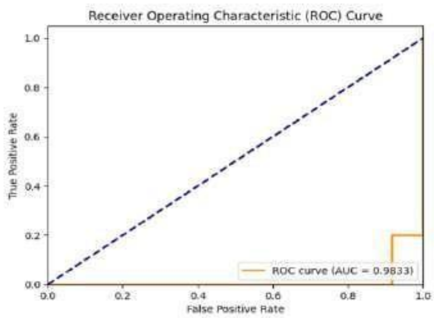
(b) SVM



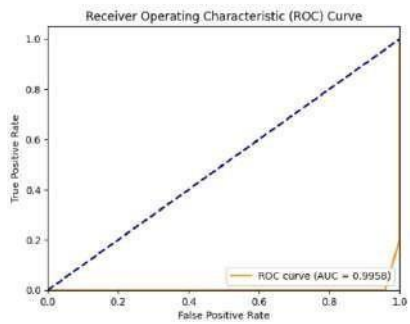
(c) DT



(d) NB



(e) MLP



(f) RF

Figure 5.15 Visualization of ROC Curve for Training 30 and Testing 70 Datasets

By visualizing this ROC Curve as shown in Figure 11, the AUC values, the Gradient Boosting and Random Forest models had the highest discriminative power with AUC values of 0.9958, while the MLP and Naïve Bayes models performed well with AUC values of 0.9833 and 0.9667, respectively. The Support Vector Machine model achieved a moderate AUC value of 0.9000, while the Decision Tree model has the lowest AUC value at 0.7792. These results suggest that the Gradient Boosting and Random Forest models are best performing models for accurately classifying patient health condition, followed by the Multilayer Perceptron and Naïve Bayes models. Based on the given classification reports, it can be concluded that the Gradient Boosting, Random Forest, MLP, Naïve Bayes, SVM, and Decision Tree models were used to classify the datasets into multiple classes, and the performance of these models was evaluated using various metrics such as accuracy, precision, recall and f1-score.

In the first experiment where 70% of the dataset was used for training and 30% for testing, it was observed that the Gradient Boosting model performed the best with an accuracy, precision, recall, and f1-score of 1.0, indicating that the model predicted all the classes correctly. The Random Forest, MLP, Naïve Bayes, SVM and Decision Tree models also performed well, with accuracy scores ranging from 0.95 to 0.98.

In the second experiment where 90% of the dataset was used for training and 10% for testing, the Gradient Boosting model again performed the best with an accuracy, precision, recall, and f1-score of 1.0. The Random Forest, MLP, Naïve Bayes, SVM, and Decision Tree models also performed well, with accuracy scores ranging from 0.95 to 0.98.

In the third experiment, where 50% of the dataset was used for training and testing, the Gradient Boosting, Random Forest, and MLP models performed perfectly with an accuracy, precision, recall, and f1-score of 1.0. The Naïve Bayes and SVM models performed better with an accuracy score of 0.95. In the fourth experiment where 30% of the dataset was used for training and 70% for testing, the Gradient Boosting model had the highest accuracy score of 0.96, followed by the Random Forest and MLP models with an accuracy score of 0.93.

The Naïve Bayes and SVM models had an accuracy score of 0.89, while the Decision Tree model performed the worst with an accuracy score of 0.86. Finally, it can be concluded that the Gradient Boosting, Random Forest, and MLP models consistently performed well in all experiments, while the Naïve Bayes and SVM models performed relatively well in some experiments and poorly in others. The Decision Tree model consistently performed the worst in some cases and best in most experiments. Therefore, it is recommended to use Gradient Boosting, Random Forest, or MLP models for similar classification tasks.

Chapter 6

CONCLUSION

The proposed system will monitor the real time health status of a person by using various sensors such as MAX30102 sensor senses the heart rate and blood oxygen, LM35 sensor senses the body temperature, DHT11 sensor senses the room temperature and room humidity along with MQ135 sensor senses the air quality of our surrounding environment. It is portable due to its minimal in size and very reliable to use. It costs very minimum for our proposed system. As a result, the system we proposed is implemented efficiently for real-time health monitoring for the critical body condition patients or persons so that doctors, nurses and caretaker can have alert in taking care of their patients by using SMS emergency alert. We can able to send these real time sensed data to the cloud and monitor the sensed data through the thing speak cloud platform efficiently. The sensed values will be displayed in the 16x2 LCD display that is interfaced with ESP-32 microcontroller Wi-Fi module if there was any internet connection issue occurred. We can also view these sensed values through our created website by using any devices such as laptops, mobile phones and remote access to the system by specialized medical professionals will minimize exposure time, enabling immediate action to be taken.

The access to the created website will only authorized by the users who have been authenticated with a valid username and password. Our system can effectively transmit data from the sensor nodes to the cloud servers and end-users but the security model is limited and not designed for wide networks. The COVID pandemic has altered the way of living, as people are more time-constrained. This virtual real time health monitoring would allow individuals to avoid visiting medical facilities with high footfall, reduce direct contact with others, and maintain a safer and more comfortable experience. Therefore, this study offers a promising solution to the problems discussed earlier in the paper. The proposed system makes use of four sensors to track heart rate, blood oxygen, body temperature, room temperature, room humidity, and air quality. Nevertheless, the system can be expanded with more sensors with more sensors to measure blood pressure, ECG, and blood glucose levels in the future.

The proposed model can be further improvised with reducing energy consumption by implementing deep sleep features in the future and also improving the security model of the prototype systems. Additionally, machine learning algorithms and data analytics methods can be employed to predict diverse health information, such as the risk of a person experiencing a heart attack, stroke, or cardiovascular diseases, and to classify whether a patient’s health condition is within the normal range or outside it. In this paper, the current and future states of real time health monitoring system that rely on Internet of Things, including their requirements, challenges, recommendations, and directions for future research.

This project explored that machine learning algorithms are effective in classifying health conditions using real-time sensor data. In particular, the Gradient Boosting algorithm consistently outperformed other algorithms in all experiments, achieving 100% accuracy in two out of four experiments and 96% and 100% accuracy in the other two experiments. This indicates that Gradient Boosting is a promising approach for the early detection of abnormal health conditions using real-time sensor data.

The study also evaluated the performance of five other algorithms – Support Vector Machines, Random Forest, Decision Tree, Naïve Bayes, and Multilayer Perceptron neural networks and found that these algorithms also performed well in most experiments, with accuracy scores ranging from 86% to 100%. This suggests that these algorithms could also be considered for health condition classification using real-time sensor data, depending on the specific use case and requirements. Overall, the findings of this study are significant for the development of automated systems for health condition classification and early detection of abnormal health conditions in real time monitoring scenarios.

There are several potential areas of future research that could build upon the findings of this study. One possibility is to further optimize the machine learning models developed in this study to improve their performance, or hyperparameters to enhance the accuracy and precision of the models. Another promising application of this research is the development of real-time health monitoring systems that utilize these models for early detection of abnormal health conditions. Such systems can leverage sensor data from wearable or IoT devices, combined with machine learning algorithms, to enable remote monitoring and timely medical interventions. Further various machine learning algorithms can be implemented and tested for improving the accuracy of results. Moreover, the proposed approach has broader implications for other healthcare applications, including disease diagnosis, drug discovery, and personalized medicine. The models developed in this study can be adopted to classify other types of health conditions and used to guide medical decisions or drug treatments. Finally, the techniques and algorithms used in this study can be applied to other domains beyond healthcare, such as finance, marketing, and social media analysis. This could help extract insights for various business applications using the same machine learning models and techniques used in this study.

Chapter 7

5 Summary

The proposed system is a portable and reliable real-time health monitoring system that uses various sensors to track heart rate, blood oxygen, body temperature, room temperature, room humidity, and air quality. The system transmits the sensed data to the cloud and displays it on an LCD screen or website, enabling doctors, nurses, and caretakers to monitor their patients' health remotely. The system can be expanded with additional sensors to measure blood pressure, ECG, and blood glucose levels. In addition, machine learning algorithms and data analytics methods can be employed to predict diverse health information and classify whether a patient’s health condition is within the normal range or outside it. The study demonstrates that machine learning algorithms, particularly Gradient Boosting, are effective in classifying health conditions using real-time sensor data. The findings have significant implications for the development of automated systems for health condition classification and early detection of abnormal health conditions in real-time monitoring scenarios, and have broader implications for other healthcare applications, such as disease diagnosis and personalized medicine.

Explore:

- Proposed model
- Scope Healthcare IoT
- The current state of Healthcare IoT
- Connectivity's role in many Healthcare capabilities
- The smart health and IoT

References

[1] Prof. N. D. Gedam, Pranav Rajurkar, Nayan Hatwar, and Atharva Sawai, "IoT based E-Health Monitoring and Room Environment Controlling System," in International Research Journal of Engineering and Technology (IRJET), 2021.

[2] Khairul Islam , Farabi Alam , Abid Ibna Zahid , Mohammad Monirujjaman Khan ,and Muhammad Inam Abbasi, " Internet of Things- (IoT-) Based Real-Time Vital Physiological Parameter Monitoring System for Remote Asthma Patients, " in Research Article of Hindawi Wireless Communications and Mobile Computing (WILEY), 2022.

[3] Prasun Biswas and Shreyashi Haldar, "Remote Health Monitoring System using Internet of Things," in International Research Journal of Engineering and Technology, 2020.

[4] Riyazulla Rahman .J, Shridhar Sanshi and N. Nasurudeen Ahamed, "Health Monitoring and Prediction using Internet of Things and Machine Learning, " in International Conference on Advanced Computing and Communication Systems, 2021.

[5] Nada Y. Philip and Honggang Wang, "Internet of Things for In-Home Health Monitoring Systems: Current Advances, Challenges and Future Directions, " in IEEE Journal on Selected Areas in Communications, Vol.39, No.2, 2021.

[6] Geng Yang, Mingzhe Jiang, Wei Ouyang, Guangchao Ji, Haibo Xie, Amir M.Rahmani, Pasi Liejeberg, Hannu Tenhunen, "IoT- based Remote Pain Monitoring System: from Device to Cloud Platform" in IEEE Journal of Biomedical and Health Informatics, 2017.

[7] JunHo Jo, ByungWan Jo, JungHoon Kim, SungJun Kim and WoonYong Han, "Devolopment of an IoT-Based Indoor Air Quality Monitoring Platform, " in Hindawi Journal of Sensors, 2020.

[8] Afzaal Hussain, Kashif Zafar and Abdul Rauf Baig, "Fog-Centric IoT based Framework for Healthcare Monitoring, Management and Early Warning System, " in IEEE Access, 2021.

[9] Sachchidanand Jha and Dr. V. Natarajan, "Real Time Patient Health Monitoring and Alarming Wireless Sensor Network, ", in IJESC, Vol.6, No.12, 2016

[10] Hesham A. El Zouka and Mustafa M. Hosni, "Secure IoT communications for smart healthcare monitoring system, " in Elsevier, 2019.

[11] Prajoona Valsalan, Tariq Ahmed Barham Baomar and Ali Hussian Omar Baabood, "IoT Based Health Monitoring System, " in Journal of Critical Reviews, Vol7, Issue 4, 2020.

[12] Prachil Patil, Swapnil Patil, Gaurav Parab, Mugdha Salvi and Ananthu Nair, "IoT based Patient Health Monitoring System, " in International Journal of Engineering Research and Technology, 2020.

[13] Suliman Abdulmalek, Abdul Nasir, Waheb A.Jabbar, Mukarram A.M.Almuhaya, Anupam Kumar Bairagi, Md. Al-Masrur Khan and Seong-Hoon Kee, "IoT-Based Healthcare-Monitoring System towards Improving Quality of Life, " in MDPI, 2022.

[14] R.Alekya, Neelima Devi Boddeti, K. Salomi Monica, Dr.R.Prabha, Dr.V.Venkatesh, "IoT based Smart Healthcare Monitoring Systems, " in European Journal of Molecualar and Clinical Medicine, Vol7,Issue 11, 2020.

[15] Mohit Yadav, Aditya Vardhan, Amarjeet Singh Chauhan and Sanjay Saini, "IoT Based Health Monitoring System, " in International Journal of Creative Research Thoughts, Vol10, Issue 1, 2022

[16] Muhammad Irfan, Husnain Jawad, Barkoum Betra Felix, Saadullah Abbasi, Annum Nawaaz, Saeed Akbarzadeh, Muhammad Awais, Li Chen, Tomi Westerlund, Wei Chen "Non-Wearable IoT-Based Smart Ambient Behavior Observation System, " in IEEE SENSORS JOURNAL, 2022.

[17] Nnamdi, Micky Chisom, Joboson, Peter Kanene, Dyaji, Charles Bala, "Monitoring Health Using IoT and Thingspeak, " in International Journal of Information Processing and Communication (IJIPC), Vol10, No.1&2 2020.

[18] Radwa Sameh, M.Genedy, A Abdeldayem, Mohammad H.Abdel azeem, "Design and Implementation of an SPO2 Based Sensor for Heart Monitoring Using an Android Application, " in Journal of Physics, 2020.

[19] Shubham Banka, Isha Madan, S.S.Saranya, "Smart Healthcare Monitoring using IoT, " in International Journal of Applied Engineering Research, Vol.13, No.15, 2018.

[20] Mohammad Nuruzzaman Bhuiyan, MD Masum Billah, Farzana Bhuiyan, MD Ashikur Rahman Bhuiyan, Nazmul Hasan, MD Mahbubur Rahman, MD Sipon Miah, Mohammad Alibakhshikenari, Farhad Arpanaei, Francisco Falcone, Minbo Niu, "Design and Implementation of a Feasible Model for the IoT Based Ubiquitous Healthcare Monitoring System for Rural and Urban Areas, " in IEEE Access, 2022.

[21] J.Turner, C.Zellner, T.Khan, and K.Yelamarthi, "Continuos heart rate monitoring using smartphone, " in Proc. IEEE Int. Conf.Electro Inf.Technol.(EIT), 2017.

[22] V.Tamilselvi, S.Sribalaji, P.Vigneshwaran, P.Vinu, J.GeethaRamani, "IoT based health monitoring system, " in Proc. ICACCS, 2020.

[23] Y.Yang, X.Liu, R.H.Deng, "Lightweight break-glass access control system for healthcare Internet -of-Things, " in IEEE Trans. Ind. Informat.. Vol. 14, No.8, 2018.

[24] F.Wu, T.Wu, M.R.Yuce, "Design and implementation of a wearable sensor network system for IoT-connected safety and health applications, " in Proc. IEEE 5th World Forum Internet of Things (WF-IoT), 2018.

[25] K.Haripriya, Chaganti M.Aravind, V.Karthigayen, P.Ganesh, " Patient Health Monitoring Using IoT and Cloud Based, " in Indian Journal of Science and Technology, 2016.

[26] Mahmood Hussain Mir, Sanjay Jamwal, Abolfazl Ehbodniya, Tanya Garg, Ummer Iqbal, and Issah Abubakari Samori, "IoT-Enabled Framework for Early Detection and Prediction of COVID-19 Suspects by Leveraging Machine Learning in Cloud," *Journal of Healthcare Engineering*, vol.2022, Article ID 7713939, 2022

[27] Shahab S.Band, Sina Ardabili, Atefeh Yarahmadi, Bahareh Pahlevanzadeh, Adiqa Kausar Kiani, Amin Beheshti, Hamid Alinejad-Rokny, Iman Dehzangi, Arthur Chang, Amir Mosavi, and Massoud Moslehpour, "A Survey on Machine Learning and Internet of Medical Things-Based Approaches for Handling COVID-19: Meta-Analysis," *Frontiers in Public Health,* PMID 35812486, 2022.

[28] Yedla Vineetha, Yogesh Misra, and K.Krishna Kishore, "A Real Time IOT Based Patient Health Monitoring System Using Machine Learning Algorithms," *European Journal of Molecular and Clinical Medicine,*vol.7, Issue 4, 2020.

[29] Suliman Abdulmalek, Abdul Nasir, Waheb A.Jabbar, Mukkarram A.M. Almuhaya, Anupam Kumar Bairagi, Md. Al-Masrur Khan, and Seong-Hoon Kee, "IoT-Based Healthcare-Monitoring System towards Improving Quality of Life," *healthcare MDPI,* 2022.

[30] Xingdong Wu, Chao Liu, Lijun Wang and, Muhammad Bilal, "Internet of things-enabled real-time health monitoring system using deep learning," Neural Computing and Applications, 2021.

[31] Kamlesh Kumar, Prince Kumar, Dipankar Deb, Mihaela-Ligia Unguresan, and Vlad Muresan, "Artificial Intelligence and Machine Learning Based Intervention in Medical Infrastructure," healthcare MDPI, *PMID 36673575,* 2023.

[32] Farida Sabry, Tamer Eltaras, Wadha Labda, Khawla Alzoubi, and Qutaibah Malluhi, "Machine Learning for Healthcare Wearable Devices," Journal of Healthcare Engineering, *Volume 2022, Article ID 465923,* 2022.

[33] A.V.L.N. Sujith, Guna Sekhar Sajja, V. Mahalakshmi, Shibli Nuhmani, and B. prasanalaksmi, "Systematic review of smart health monitoring using deep learning and Artificial intelligence," Neuroscience Informatics, 2021.

[34] Abderrahmane Ed-daoudy and Khalil Maalmi, "A new Internet of Things architecture for real-time prediction of various diseases using machine learning on big data environment," Journal of Big Data, 2019.

[35] Deva Priya Isravel, Vidya Priya Darcini S, and Salaja Silas, "Improved Heart Disease Diagnostic IoT Model Using Machine Learning Techniques," International Journal of Scientific and Technology Research, *Volume 9, Issue 02,* 2020.

[36] Jiang Huang, Jung Li, Zheming Li, Zhu Zhu, Chen Shen, Guoqiang Qi, and Gang Yu, "Detection of Diseases Using Machine Learning Image Recognition Technology in Artificial Intelligence," Computational Intelligence and Neuroscience, *Volume 2022, Article ID 5658641,* 2022.

[37] Yuequan Bao, and Hui Li, "Machine learning paradigm for Structural health monitoring," Structural Health Monitoring, *Volume 20, Issue 4,* 2020.

[38] Zeeshan Ahmed, Khalid Mohamed, Saman Zeeshan, and XingQi Dong, “Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine,” The Journal of Biological Databases and Curation, *Volume.2020, Article ID baaa010,* 2020.

[39] Nikos Fazakis, Otilia Kocsis, Elias Dritsas, Sotiris Alexiou, Nikos Fakotakis, Konstantinos Moustakas, “Machine Learning Tools for Long-Term Type 2 Diabetes Risk Prediction,” IEEE Access, *Volume.9*, *Electronic ISSN.2169-3536*, 2021.

[40] Md. Milon Islam, Ashikur Rahaman, and Md. Rashedul Islam, “Development of Smart Healthcare Monitoring System in IoT Environment,” SN Computer Science, 2020.
Ozer Celik, “A Research on Machine Learning Methods and its Applications,” Journal of Educational Technology and Online

List of publications

The paper titled "Analysis and Development of IoT-Based Health Monitoring Systems" was published in the European Chemical Bulletin Scopus journal.

Analysis and Development of Iot-Based Health Monitoring Systems *Section A-Research paper*

ANALYSIS AND DEVELOPMENT OF IOT-BASED HEALTH MONITORING SYSTEMS

Prakash P [1], Dr. Jaffino G [2,*]

Article History: Received: 12.12.2022 **Revised:** 29.01.2023 **Accepted:** 15.03.2023

Abstract

With increasing growth of new healthcare technology IoT is rapidly revolutionizing the healthcare industry. Nowadays there are lot of Internet of Things devices which is used to monitor the health over internet. By using these smart devices doctors and caretakers also taking advantages to keep an eye on the health conditions of the patients. In this work the proposed health monitoring of a patient using IoT based sensors which records the patient heartrate, blood oxygen using MAX30102 sensor, body temperature using LM35 sensor, room temperature and room humidity using DHT11 sensor and quality of air using MQ135. ESP32 Microcontroller board which controls these sensors sends these sensed values to the cloud and these sensed values can be viewedthrough the created website using various devices. The sensed values are also displayed in 16x2 i2c LCD displaymodule. If the Heath condition is normal/abnormal it sends emergency message to the doctors/caretakers.

Keywords: Health, Internet of Things (IoT) Sensors; Analytics platform service; Digital automation platform

[1]School of Electronics Engineering, Vellore Institute of Technology, Vellore
[2*]School of Electronics Engineering, Vellore Institute of Technology, Vellore

Email:[1] prakashpgmj@gmail.com , [2*]jaffino22@yahoo.com;

DOI: 10.31838/ecb/2023.12.s3.254

Curriculum vitae

Name: Prakash P

Registration Number: 21MTS0017

Branch: MTech IoT and Sensor Systems

Department: Sensor and Biomedical Technology

School: School of Electronics Engineering

Phone no.: +91**9092261168**

Email – ID: prakash.p2021@vitstudent.ac.in

Vellore Institute of Technology, Vellore.