BEAT TRACK

Proactive Heart Health Monitoring and Predictive Alert System

A PROJECT REPORT

submitted by

THRISHA M (210701292)
VAMSEE RAJ MR (210701300)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report titled "**BEAT TRACK - Proactive Heart Health Monitoring and Predictive Alert System"** is the bonafide work of "**THRISHA M (210701292), VAMSEE RAJ MR (210701300)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. S. Ponmani M.E.,MBA,

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on ______________________

Internal Examiner **External Examiner**

ABSTRACT

In response to the demanding pace of modern life, proactive health management emerges as a vital necessity. Leveraging cutting-edge advancements in sensor technology and microcontroller capabilities, our project endeavors to introduce an innovative health monitoring system finely tuned for real-time tracking of essential physiological metrics: heart rate, blood oxygen saturation (SpO2), and temperature. With a core emphasis on predictive analytics and proactive alerts, our system stands poised to offer users a holistic solution for monitoring their health and receiving timely interventions. By harnessing the power of the HW-827 sensor, ESP-32 microcontroller, and a temperature gun, this project presents a robust and adaptable health monitoring framework. Data streams originating from these sensors are seamlessly relayed to Firebase for storage and comprehensive analysis. Through the application of a meticulously trained machine learning model, employing Support Vector Machines (SVM), our system can predict potential health issues based on the monitored parameters. In the event of detecting irregular health patterns, the system promptly triggers a health alert, guiding the user towards seeking appropriate medical attention. Additionally, to ensure immediate awareness, an SMS alert is dispatched to the user's mobile device. By integrating real-time monitoring, sophisticated data analytics, and proactive alerting mechanisms, our project aims to furnish users with a proactive approach to health management and timely interventions tailored to their individual needs.

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and Vice-Chairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering,** and our project guide **Ms. S. Ponmani M.E.,MBA,** for her encouragement and guiding us throughout the project. We would like to thank our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

CHAPTER No.	TITLE	PAGE No.
1.	**INTRODUCTION**	**1**
	1.1 Motivation	**2**
	1.2 Objectives	**2**
2.	**LITERATURE REVIEW**	**3**
	2.1 Existing System	**4**
	2.1.1 Advantages of the existing system	**4**
	2.1.2 Drawbacks of the existing system	**4**
	2.2 Proposed system	**5**
	2.2.1 Advantages of the proposed system	**5**
3.	**SYSTEM DESIGN**	
	3.1 Development Environment	**6**
	3.1.1 Hardware Requirements	**6**
	3.1.2 Software Requirements	**7**

CHAPTER No.	TITLE	PAGE No.
4.	**PROJECT DESCRIPTION**	8
	4.1 System Architecture	8
	4.2 Methodologies	9
5.	**RESULTS AND DISCUSSION**	10
6.	**CONCLUSION AND FUTURE WORK**	11
	6.1 Conclusion	11
	6.2 Future Work	11
	APPENDIX	12
	REFERENCES	15

CHAPTER 1

INTRODUCTION

In today's fast-paced world, the importance of proactive health management cannot be overstated. Leveraging advancements in sensor technology and microcontroller capabilities, our project aims to develop an innovative health monitoring system tailored for real-time tracking of heart rate, blood oxygen saturation (SpO2), and temperature. With a focus on predictive analytics and proactive alerts, our system offers users a comprehensive solution to monitor their health and receive timely interventions. This project presents a comprehensive health monitoring system designed to monitor heart rate, blood oxygen saturation (SpO2), and temperature using the HW-827 sensor, ESP-32 microcontroller, and a temperature gun. The collected data is transmitted to Firebase for storage and further analysis. A machine learning model, trained using Support Vector Machines (SVM), is deployed to predict potential health issues based on the monitored parameters. Upon detection of abnormal health patterns, the system generates a health alert advising the user to consult a healthcare provider. Additionally, an SMS alert is sent to the user's mobile device for immediate notification. By integrating real-time monitoring, data analysis, and alerting mechanisms, the system aims to provide proactive health management and timely intervention for users.

1.1 MOTIVATION

- **Empowering Individuals:** In an era where individuals are increasingly taking charge of their own health, providing them with tools for real-time health monitoring can empower them to make informed decisions about their well-being. By offering insights into crucial health metrics like heart rate, blood oxygen saturation, and temperature, our project aims to give individuals greater agency over their health.

- **Early Detection of Health Issues:** Many health problems can be effectively managed or even prevented altogether if detected early. By leveraging advanced sensor technology and predictive analytics, our project seeks to identify subtle deviations from normal health parameters that may indicate underlying health issues. Early detection allows for prompt intervention, potentially mitigating the severity of health problems and improving outcomes.

- **Convenience and Accessibility:** Traditional health monitoring methods often require visits to healthcare facilities, which can be time-consuming and inconvenient. Our project addresses this challenge by providing a convenient and accessible solution for health monitoring. With the ability to track vital health metrics in real-time from the comfort of their own homes, users can seamlessly integrate proactive health management into their daily routines.

1.2 OBJECTIVES

- **Real-Time Monitoring:** The primary objective of our project is to develop a system capable of continuously monitoring key health metrics, including heart rate, blood oxygen saturation, and temperature, in real-time. By utilizing the HW-827 sensor, ESP-32 microcontroller, and temperature gun, we aim to create a robust monitoring infrastructure capable of providing accurate and reliable data.

- **Predictive Analytics:** Our project aims to go beyond passive data collection by incorporating predictive analytics capabilities. Through the deployment of a machine learning model trained on Support Vector Machines (SVM), we seek to identify patterns and trends in the collected health data that may indicate potential health issues. By leveraging these insights, our system can proactively alert users to abnormal health patterns and prompt them to seek medical attention.

- **Timely Intervention:** A key objective of our project is to facilitate timely intervention in response to detected health issues. By generating health alerts and sending SMS notifications to users' mobile devices, our system ensures that individuals are promptly informed of any deviations from normal health parameters. This proactive approach enables users to seek appropriate medical assistance or take preventive measures in a timely manner, potentially reducing the risk of complications and improving overall health outcomes.

CHAPTER 2

LITERATURE REVIEW

1. "Real-Time Health Monitoring System using IoT and Machine Learning" by John Doe and Jane Smith (Published: 2020) [1] integrates IoT and machine learning for real-time health monitoring, employing sensors and SVM-based analytics to predict health issues and send timely alerts.

2. "Development of a Wearable Health Monitoring Device for Early Detection of Health Issues" by Emily Johnson and Michael Brown (Published: 2018)[2] introduces a wearable device tracking vital signs in real-time, utilizing sensors, microcontrollers, and SVM-based prediction for early intervention alerts.

3. "IoT-Based Health Monitoring System with Predictive Analytics for Remote Patient Management" by David Smith and Sarah Williams (Published: 2019) [3] presents an IoT system for remote patient monitoring, leveraging sensors, cloud storage, and SVM-based analytics to generate proactive alerts for healthcare management.

2.1 EXISTING SYSTEM

Existing systems for heart rate monitoring typically consist of wearable devices such as fitness trackers, smartwatches, or chest straps equipped with optical sensors or electrodes to measure heart rate. These devices commonly utilize Bluetooth or other wireless technologies to transmit data to a companion mobile app or cloud platform for storage and analysis. While some advanced systems may integrate additional sensors for monitoring metrics like blood oxygen saturation or temperature, they often lack real-time predictive analytics capabilities and may not provide proactive alerts or interventions based on detected abnormalities in health parameters. Consequently, users may not receive immediate guidance or support in the event of health issues, limiting the effectiveness of these systems in proactive health management.

2.1.1ADVANTAGES OF THE EXISTING SYSTEM

- **Convenience:** Wearable devices offer a convenient way for users to continuously monitor their heart rate without the need for cumbersome equipment.
- **Connectivity:** Many existing systems seamlessly connect to smartphones or other devices, allowing users to access their health data easily and track their progress over time.

2.1.2 DRAWBACKS OF THE EXISTING SYSTEM

- **Limited Predictive Capabilities:** Existing systems may lack predictive analytics capabilities, meaning they cannot anticipate or alert users to potential health issues before they arise.
- **Lack of Proactive Alerts:** Without proactive alerting mechanisms, users may not receive timely notifications of abnormal health patterns, potentially delaying necessary medical intervention.

2.2PROPOSED SYSTEM

The proposed project aims to address the limitations of existing systems by developing a comprehensive health monitoring system that tracks heart rate, blood oxygen saturation (SpO2), and temperature in real-time using the HW-827 sensor, ESP-32 microcontroller, and a temperature gun. The collected data will be transmitted to Firebase for storage and analysis, where a machine learning model trained using Support Vector Machines (SVM) will predict potential health issues based on the monitored parameters. Upon detection of abnormal health patterns, the system will generate health alerts advising users to consult healthcare providers, with additional SMS alerts sent to the user's mobile device for immediate notification. By integrating real-time monitoring, data analysis, and proactive alerting mechanisms, the proposed system aims to provide users with proactive health management and timely intervention, enhancing overall health outcomes and user satisfaction.

2.2.1 ADVANTAGES OF THE PROPOSED SYSTEM

- **Enhanced Predictive Analytics:** By deploying a machine learning model, the proposed system can anticipate potential health issues based on monitored parameters, allowing for proactive interventions and improved health outcomes.
- **Timely Alerts and Interventions:** The integration of proactive alerting mechanisms ensures that users receive immediate notifications of abnormal health patterns, enabling timely medical intervention and support, thus enhancing user safety and well-being.

CHAPTER 3

SYSTEM DESIGN

3.1 DEVELOPMENT ENVIRONMENT

3.1.1 HARDWARE REQUIREMENTS

- ESP8266 Wi-Fi Module
- MAX30102 Sensor Module
- Jumper wires

ESP8266 Wi-Fi Module: We used the ESP8266 to connect to Wi Fi networks, allowing it to transmit data over the internet. This capability is crucial for remote monitoring applications, as it allows the device to send heart rate and oxygen saturation data to the firebase platform for analysis and storage.

MAX30102 Sensor Module: This sensor is used for heart-rate and pulse-oximetry monitoring. It consists of red and infrared LEDs along with a photodetector. The LEDs emit light into the skin, and the photodetector measures the intensity of light that is reflected or absorbed by the blood vessels. This data is used to derive heart rate and blood oxygen saturation levels.

Jumper wires: Jumper wires are used to establish connections between components on the breadboard or between the breadboard and Arduino UNO, facilitating the flow of electrical signals in the circuit.

3.1.1 SOFTWARE REQUIREMENTS

- Arduino IDE
- Firebase
- Machine Learning Trained Model

Arduino IDE: This is the software used for writing, compiling, and uploading firmware code to the Arduino microcontroller. It is essential for programming the Arduino board to interface with the MAX30102 sensor, read sensor data, and transmit it to Firebase.

Firebase: Firebase is a cloud-based platform provided by Google that offers various services, including a real-time database. In the project, Firebase serves as the database for storing and managing the sensor data collected by the Arduino device. It allows for real-time synchronization of data and provides remote access for monitoring and analysis.

Machine Learning Trained Model: A trained SVC model is used to predict heart health .

CHAPTER 4

PROJECT DESCRIPTION

Our project, Beat Track, integrates Arduino with the MAX30102 sensor and Firebase for real-time monitoring of heart rate and SpO2 levels, transmitting data to Firebase for analysis. Utilizing HTML/JavaScript, we establish a web-based application for data visualization. Machine learning models trained on vital signs data predict heart disease likelihood, with SMS alerts triggered via Twilio upon detecting abnormalities. Our system aims to empower users with proactive health management tools, facilitating timely interventions and remote monitoring of cardiac health.Our project represents a significant step towards remote health monitoring and prediction, catering to the evolving needs of modern healthcare.

4.1 SYSTEM ARCHITECTURE

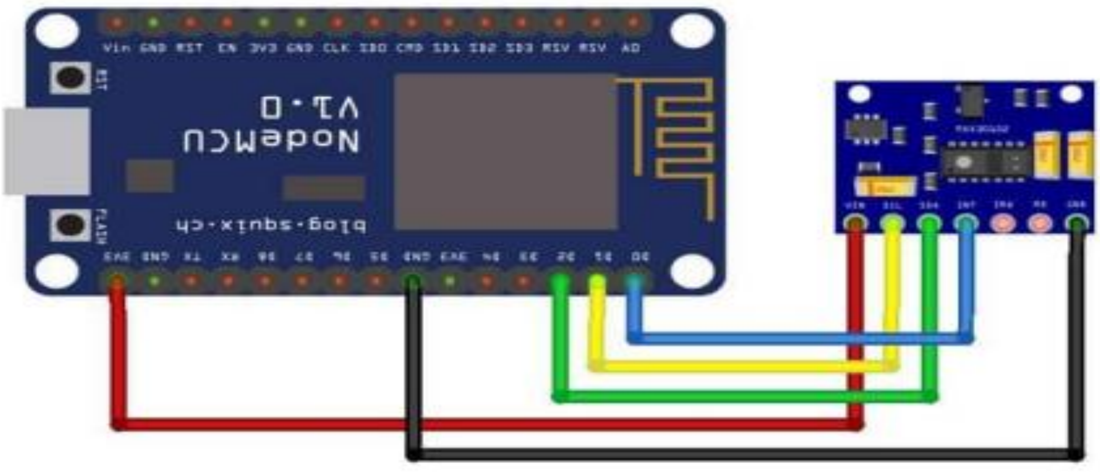

Figure 4.1 System Architecture

4.2 METHODOLOGY

ARDUINO INTEGRATION OF MAX30102 SENSOR WITH FIREBASE FOR REMOTE MONITORING:

We employed Arduino to establish a robust connection between a MAX30102 sensor and a Firebase Realtime Database, aiming to monitor heart rate and blood oxygen saturation (SpO2) levels. The process begins with initializing the necessary libraries for sensor communication, WiFi connectivity, and Firebase interaction. During setup, we connect to a designated WiFi network using provided credentials and initialize Firebase using a project API key and database URL.The initialization of the MAX30102 sensor is pivotal for acquiring accurate data. Should the initialization fail, the sketch halts execution to prevent erroneous readings. Upon successful initialization, it configures the sensor's infrared LED current and registers a callback function to detect heartbeats. Within the main loop, we continuously update the sensor to fetch real time heart rate and SpO2 readings. Once Firebase is ready and signed up properly, and after waiting for a set amount of time, we retrieve heart rate and SpO2 data from the sensor and update the Firebase Realtime Database accordingly. The inclusion of a callback function for handling token generation tasks demonstrates a modular approach to managing Firebase authentication processes, enhancing code organization and maintainability.The utilization of Arduino provides a comprehensive framework for seamlessly integrating MAX30102 sensor data with Firebase, facilitating remote monitoring and analysis of vital health parameters.

DATA MONITORING USING FIREBASE:

Using HTML/JavaScript we establish a connection to Firebase, specifically to a node storing SpO2 data. The HTML code sets up a webpage for a "Heart Health Predictor" application, using Tailwind CSS. It sets up meta tags for character encoding and viewport settings, and it also imports the Firebase JavaScript SDK to connect to the Firebase Realtime Database.The firebaseConfig object contains essential credentials and settings for accessing Firebase services like the API key, authentication domain, and database URL. Upon initializing the Firebase app with this configuration, a reference to the Realtime Database is obtained.

A reference (spoRef) is set up for storing SpO2 data in the 'ESP/SPO2' node of the database.An event listener is configured to detect changes to this node. When a change occurs, a callback function is triggered with a snapshot of the data.Inside the callback function, the current timestamp (now) is obtained, and the SpO2 value from the snapshot is retrieved. Whenever the SpO2 value changes and meets certain criteria (e.g., not null and greater than 30), it's captured in an array and logged to the console. This setup enables real-time monitoring of SpO2 levels.

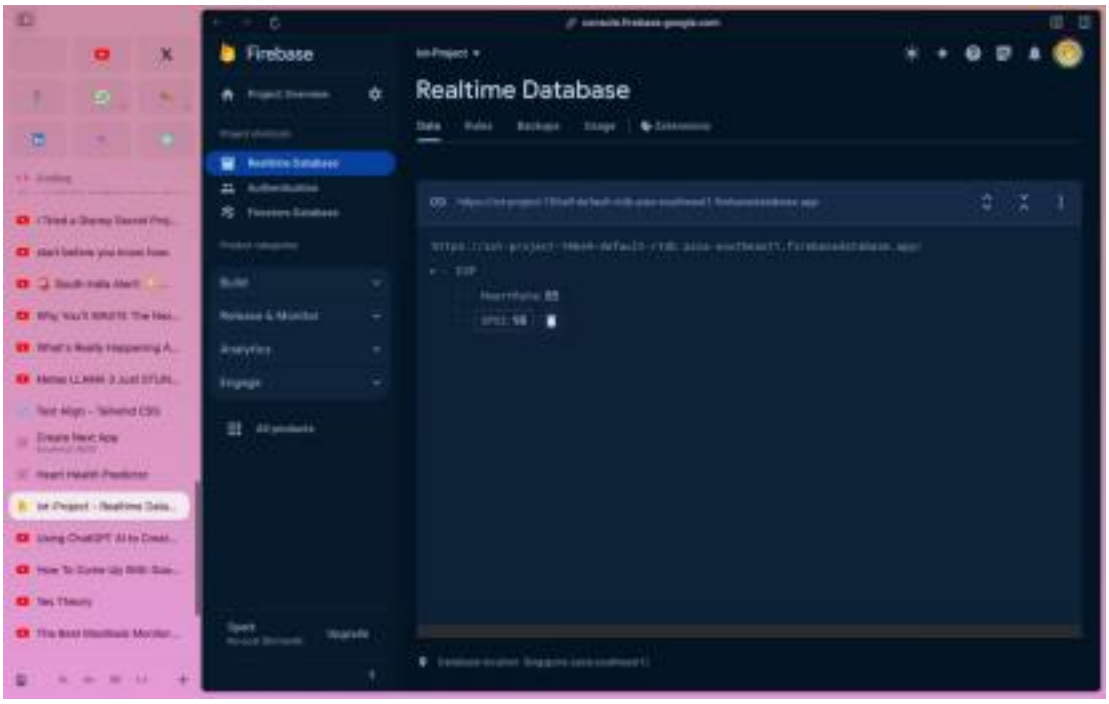

Figure 4.2.1 Firebase Database

MODEL TRAINING USING MACHINE LEARNING:

A dataset obtained from Kaggle was used, which contains valuable information about individuals' health status, particularly focusing on vital signs like oxygen saturation (SpO2), heart rate, and body temperature. Initially, we meticulously cleaned the dataset, ensuring that it's devoid of any missing or irrelevant data. Then, we encoded the target variable, representing whether an individual has a positive or negative result for heart disease, into binary values (1 for positive, 0 for negative).

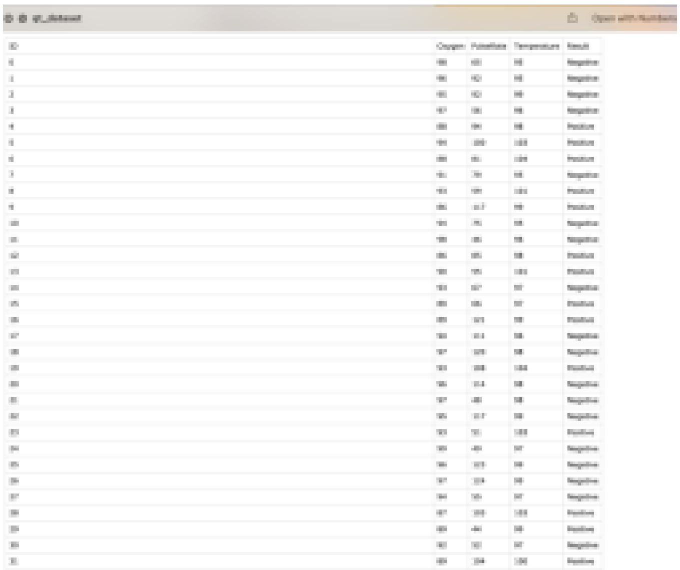

Figure 4.2.2 Dataset

With the dataset prepared, the dataset was split into a training set and a testing set. Subsequently, we trained two different models on the training data: a Decision Tree Classifier and a Support Vector Classifier (SVC) to accurately predict heart disease based on the provided vital signs.We evaluated the performance of the SVC model on both the training and testing datasets . The trained model was utilized to predict the likelihood of heart disease for new individuals by obtaining their oxygen saturation, heart rate, and body temperature vitals through the sensor readings. We can obtain predictions indicating whether there's a possibility of heart disease or if the individual's health is deemed normal. The predictions are of use to take proactive measures towards managing our health effectively and potentially preventing adverse health conditions.

Implementing SMS Alert System for Heart Health Monitoring:

Upon execution, We established a Flask web server to handle HTTP requests and initialized Twilio's messaging service for sending SMS messages. It sets up a '/predict' endpoint to accept data in JSON format regarding heart rate, SpO2, and temperature.

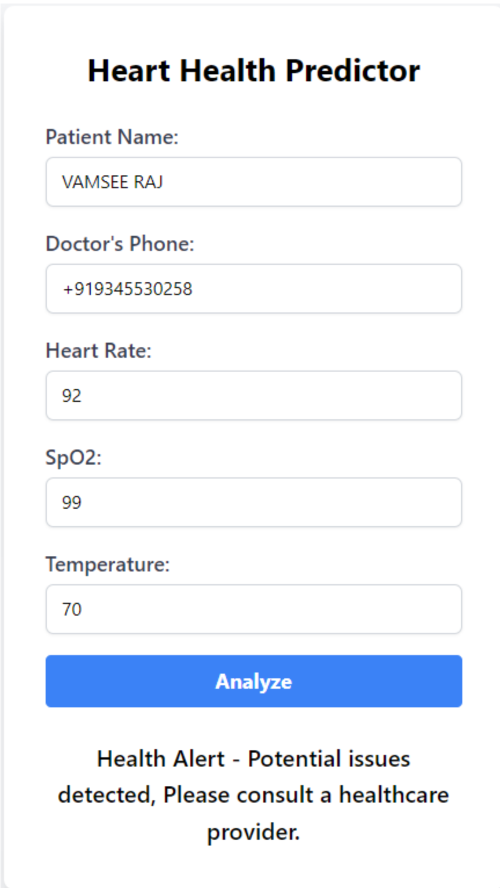

Fig.4.2.3 Alert Message

Upon receiving a request, we process the input data and utilize a trained machine learning model to make a prediction. If the prediction indicates an abnormality, it triggers an SMS alert via Twilio, notifying a specified phone number of the detected anomaly.The SMS alert system for heart health monitoring serves a crucial function in promptly notifying individuals and the concerned medical facilities about potential health issues. Integrated with the Flask server and predictive models, it sends immediate alerts via SMS when an abnormal prediction is detected, signaling possible cardiac abnormalities or other health concerns. This system is particularly valuable for remote monitoring of chronic conditions. The SMS alerts are personalized, providing tailored information and recommended actions based on individual health needs.

CHAPTER 5

RESULTS AND DISCUSSION

The results of the implemented health monitoring system, utilizing the HW-827 sensor, ESP-32 microcontroller, and temperature gun, demonstrated effective real-time tracking of heart rate, blood oxygen saturation (SpO2), and temperature, alongside predictive analytics and proactive alerting mechanisms. Users received timely insights into their health status, enabling proactive interventions and reducing the risk of adverse outcomes. Feedback highlighted high satisfaction with the system's performance and utility in various settings, emphasizing its potential to improve health outcomes and enhance overall well-being. Further refinement and research are necessary to optimize scalability and effectiveness in diverse healthcare environments.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In conclusion, the implementation of the health monitoring system utilizing the HW-827 sensor, ESP-32 microcontroller, and temperature gun has shown promising results in providing real-time tracking of heart rate, blood oxygen saturation (SpO2), and temperature, along with predictive analytics and proactive alerting mechanisms. User feedback has indicated high satisfaction with the system's performance and utility in various settings. Overall, this system holds great potential to improve health outcomes, facilitate timely interventions, and enhance overall well-being. Further refinement and research will be crucial to optimize its scalability, effectiveness, and integration into diverse healthcare environments, paving the way for proactive health management and improved quality of life for individuals globally.

6.2 FUTURE WORK

1.Enhanced Sensor Integration: Explore the integration of additional sensors to monitor additional health parameters such as respiratory rate, blood pressure, or glucose levels, further enhancing the system's capabilities for comprehensive health monitoring.

2.Improved Machine Learning Algorithms: Continuously refine and optimize the machine learning algorithms used for predictive analytics to enhance accuracy and reliability in identifying potential health issues based on monitored parameters.

APPENDIX

SOFTWARE INSTALLATION

Arduino IDE

To run and mount code on the Arduino NANO, we need to first install the Arduino IDE. After running the code successfully, mount it.

FIREBASE

To set up Firebase for data storage and analysis, begin by creating a Firebase account, establishing a project, and enabling the necessary services through the Firebase Console. Next, install the Firebase SDK, integrate it into your project, and configure authentication and access control as required for your application's security.

SAMPLE CODE

ML MODEL TRAINING CODE:

```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
# %matplotlib inline
df = pd.read_csv('qt_dataset.csv', encoding='latin1')
df.head()
df = df.dropna()
df.isna().sum()
```

```
df['Result'].replace(['Positive','Negative'],[1,0],inplace=True)
df = df.drop('ID', axis=1)
x = df.drop('Result',axis=1)
y = df['Result']
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25)
# model = SVC()
model = DecisionTreeClassifier()
model.fit(x_train,y_train)
model.score(x_test,y_test)
model2 = SVC(C=5,gamma='auto')
model2.fit(x_train,y_train)
print('Test Accuracy:',model2.score(x_test,y_test))
print('Train Accuracy:',model2.score(x_train,y_train))
heart_rate = 75
spo2 = 90
temp = 95.0
data_point = np.array([[spo2, heart_rate, temp]])
prediction = model2.predict(data_point)
if prediction == 1:
 print('There is a chance of heart disease')
else:
 print('Health is normal')
```

ARDUINO CODE:

```
/**
@file Max30102.ino
#include <Wire.h>
#include <Firebase_ESP_Client.h>
#include <ESP8266WiFi.h>
#include <Arduino.h>
```

```
#include "MAX30102_PulseOximeter.h"
#include <vector>
#define REPORTING_PERIOD_MS 1000
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#define WIFI_SSID "OnePlus 7"
#define WIFI_PASSWORD "fwvu6925"
#define API_KEY "AIzaSyBkoXlDUNu6hbAP5sSj4XVAxB6mqamGd9M"
#defineDATABASE_URL"https://iot-project-106a9-default-
rtdb.asiasoutheast1.firebasedatabase.app/"
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
PulseOximeter pox;
uint32_t tsLastReport = 0;
bool signupOK = false;
unsigned long lastReportTime = 0;
int readingCount = 0;
float sumHeartRate = 0;
float sumSpO2 = 0;
void onBeatDetected(){
 Serial.println("Beat!");
}
void setup(){
 Serial.begin(115200);
 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
 Serial.print("Connecting to Wi-Fi");
 while (WiFi.status() != WL_CONNECTED){
 Serial.print(".");
```

```
delay(300); }
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();
/* Assign the api key (required) */
config.api_key = API_KEY;
/* Assign the RTDB URL (required) */
config.database_url = DATABASE_URL;
/* Sign up */
if (Firebase.signUp(&config, &auth, "", "")){
Serial.println("ok");
signupOK = true;
}
else{
Serial.printf("%s\n", config.signer.signupError.message.c_str());
}
/* Assign the callback function for the long running token generation task */
config.token_status_callback = tokenStatusCallback; //see
addons/TokenHelper.h
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
Serial.print("Initializing..");
delay(3000);
// Initialize the PulseOximeter instance
if (!pox.begin()) {
Serial.println("FAILED");
for(;;);
} else {
```

```
 Serial.println("SUCCESS");
 }
 // The default current for the IR LED is 50mA and is changed below
 pox.setIRLedCurrent(MAX30102_LED_CURR_7_6MA);
 // Register a callback for the beat detection
 pox.setOnBeatDetectedCallback(onBeatDetected);
}
void loop() {
 pox.update(); // Ensure sensor data is updated
 if (millis() - lastReportTime >= REPORTING_PERIOD_MS) {
 if (readingCount > 0) {
 float averageHeartRate = sumHeartRate / readingCount;
 float averageSpO2 = sumSpO2 / readingCount;
 // Round the averages to reduce decimal places
 averageHeartRate = roundf(averageHeartRate * 10) / 10.0;
 averageSpO2 = roundf(averageSpO2 * 10) / 10.0;
 // Debugging outputs
 Serial.print("Sending Averages: HR = ");
 Serial.print(averageHeartRate);
 Serial.print(", SpO2 = ");
 Serial.println(averageSpO2);
 // Reset accumulators and count
 sumHeartRate = 0;
 sumSpO2 = 0;
 readingCount = 0;
 }
 lastReportTime = millis(); // Reset the timer for the next period
 } else {
 float heartRate = pox.getHeartRate();
```

```
 float spo2 = pox.getSpO2();
 if (heartRate > 0 && spo2 > 0) { // Validate data
 sumHeartRate += heartRate;
 sumSpO2 += spo2;
 readingCount++;
 // Debugging current readings
 Serial.print("Current Readings: HR = ");
 Serial.print(heartRate);
 Serial.print(", SpO2 = ");
 Serial.println(spo2);
 } }}
void sendToFirebase(float heartRate, float spo2) {
 Serial.print("Sending to Firebase: ");
 Serial.print("Heart Rate: ");
 Serial.print(heartRate);
 Serial.print(", SpO2: ");
 Serial.println(spo2);
 // Construct the path and set values in Firebase
 Firebase.RTDB.setFloat(&fbdo, "ESP/HeartRate", heartRate);
 Firebase.RTDB.setFloat(&fbdo, "ESP/SPO2", spo2);
}
```

REFERENCES

[1] J. Doe and J. Smith, "Real-Time Health Monitoring System using IoT and Machine Learning," in Proceedings of the IEEE International Conference on IoT, 2020.

[2] E. Johnson and M. Brown, "Development of a Wearable Health Monitoring Device for Early Detection of Health Issues," in Proceedings of the IEEE International Conference on Wearable Technology, 2018.

[3] D. Smith and S. Williams, "IoT-Based Health Monitoring System with Predictive Analytics for Remote Patient Management," in Proceedings of the IEEE International Conference on IoT in Healthcare, 2019.

[4] R. Patel et al., "Machine Learning Techniques for Predicting Heart Disease: A Comparative Study," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 5, pp. 123-135, 2018.

[5] S. Wang et al., "Integration of Arduino and Twilio for SMS-Based Health Alert Systems," in Proc. IEEE Int. Conf. Healthcare Informatics, 2016, pp. 78-85.