NON-INVASIVE GLUCOMETER USING BREATH ACETONE

A MINI PROJECT REPORT

Submitted by

KAUSHALYA R (113020104039)
KAVYASHRI (113020104041)
SUNPREET KAUR (113020104098)

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

VEL TECH HIGH TECH
Dr. RANGARAJAN Dr. SAKUNTHALA ENGINEERING COLLEGE
An Autonomous Institution
Approved by AICTE-New Delhi, Affiliated to Anna University, Chennai
Accredited by NBA, New Delhi & Accredited by NAAC with “A” Grade & CGPA of 3.27

NOVEMBER 2022

VEL TECH HIGH TECH

Dr. RANGARAJAN Dr. SAKUNTHALA ENGINEERING COLLEGE

An Autonomous Institution

Approved by AICTE-New Delhi, Affiliated to Anna University, Chennai

Accredited by NBA, New Delhi & Accredited by NAAC with "A" Grade & CGPA of 3.27

BONAFIDE CERTIFICATE

Certified that this mini project entitled **"NON-INVASIVE GLUCOMETER USING BREATH ACETONE"**is the bonafide work of"**KAUSHALYA R (113020104039) KAVYASHRI (113020104041)SUNPREET KAUR (113020104098)"**who carried out the work under my supervision.

SIGNATURE	SIGNATURE
Dr. DURGA DEVI, B.E., M.E., PhD.,	**Dr. V. R RAVI., B.E.,M.E,Ph.D.,**
HEAD OF THE DEPARTMENT	**MENTOR, Dr. V. R. RAVI**
Computer Science and Engineering	**DEAN - SoEC**

SIGNATURE

Dr. V. R RAVI., B.E.,M.E,Ph.D.,

PROFESSOR

DEAN - SoEC

ACKNOWLEDGEMENT

We wish to express our sincere thanks to the people who extended their help during the course of our Mini Project work.

First of all we would like to express our deep gratitude to our beloved and respected **FOUNDER & CHAIRMAN Col. Prof. Vel. Shri. Dr. R. RANGARAJAN** B.E. (Elec.), B.E (Mech.), M.S (Auto.), DSC, and**Foundress& VICE CHAIRMAN Dr. SAKUNTHALA RANGARAJAN** M.B.B.S. We also record sincere thanks to our Honorable Principal, **Dr. E.KAMALANABAN B.E., M.E., Ph.D.,** for his kind support for his Mini Project.

We are thankful and extremely grateful to our Dean-SoEC**Dr. V. RAVI** and Head of the department **Dr. S. DURGADEVI,** B.E., M.E., Ph.D., for their support and motivation feed to us for doing the Mini Project.

We would like to extend our sincere thanks to our guide Dean-SoEC**Dr. V. RAVI** for his technical support during this project. His stupendous encouragement enabled us to complete our Mini Project successfully.

We would like to extend our sincere thanks to our Project Coordinator **Dr. M. MALATHY** B.E., M.E., Ph.D., and **Mrs. A. KRISHNAVENI** B. E., M. E., for their continuous support over this Mini Project

TABLE OFCONTENTS

Abstract:

Acetone is one of the vocs which is obtained from fatty acid metabolism and acts as an important biomarker for diabetes detection. It was reported that the breath of diabetes patients consists of acetone in abnormal concentrations and gradually increases according to the patient's blood glucose value. Acetone concentration can vary between 560 ppm to >1000 ppm in diabetes patients. MQ-138 semiconductor gas sensor has a high sensitivity to acetone and is used in this project to design a breath acetone detection system. Measurement of Acetone: Breath acetone detection chamber consists of an input valve, gas room which breath acetone is drawn into it and MQ-138 sensor to measure breath acetone. The analog output from MQ-138 sensor is converted to digital output by using 24-bit A/D converter and interfaced to Raspberry PI. The Raspberry PI will measure breath acetone value. Development of Machine Learning Model to estimate Blood Sugar Level from Acetone: Breath Acetone values are measured using breath acetone detection chamber from 500 different diabetes and non-diabetes patients and also blood sugar values are measured from lab for the same 500 patients. A suitable Machine Learning model will be developed using exhaled acetone (collected from breath acetone detection chamber) and Blood Sugar value (collected from the Lab) of 500 different diabetes and non-diabetes patients. The developed Machine Learning Model will corelate exhaled acetone with the blood sugar level. Deployment of ML Model in Raspberry PI: The developed ML model will be deployed in Raspberry PI and placed inside blood sugar measuring chamber along with OLED display. If breath acetone of a new patient is applied to this non-invasive glucometer, it will display blood glucose value on OLED display. The entire programme will be developed using MATLAB..

Keywords—acetone,blood glucose level, machine learning, gas sensor, non invasive

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRO-NYM	ABBRIVIATIONS
SVOR	Support Vector Ordinal Regression
VOC	Volatile Organic Compound
OGTT	Oral glucose tolerance test
T1DM	Type-1 diabetes mellitus
BOHB	β-hydroxybutyrate
OLED	Organic light emitting devices
A/D	Analog/Digital
TCP	Transmission Control Protocol
IP	Internet Protocol
MATLAB	Matrix Laboratory
IR	Infrared Radiation

CHAPTER 1
INTRODUCTION

Blood glucose monitoring is particularly important for diabetic patients to control their conditions. Typically, they measure their blood glucose levels by piercing a finger to obtain a blood sample. This method is accurate but is painful, invasive, and unsafe. Therefore, it does not suit everyone, especially in the case where it needs several samplings each day. As a result, it highly requires non-invasive continuous glucose monitoring. Recently, several non-invasive glucose monitoring approaches were studied, including spectrophotometric (IR light, fluorescence-based, etc.), electrical impedance, photoacoustic, light scattering, and iontophoretic. These approaches can provide a painless and convenient procedure. However, all of them work through skin measurements. They are susceptible to environmental variations and subjects' physical and chemical parameters, such as changes in temperature and humidity, variations in subjects' blood pressure, skin hydration, and skin pigmentation. Breath analysis is supposed to provide useful information about the blood glucose levels of diabetic patients. The analysis of breath acetone associated with diabetes was the most extensively studied area in the field of breath analysis since the 1960s. The key reason is that diabetes is the most common pathological cause of elevated blood ketone, and, plasma acetoneamd has been proven to be linearly related to breath acetone. Therefore, we can smell the 'sweet odor' of the breath of diabetics due to the elevated presence of acetone in blood and breath.

CHAPTER 2

LITERATURE SURVEY

2.1 Non-invasive blood glucose monitoring for diabetics by means of breath signal analysis

Collection and analyzation of the breath samples of diabetic patients by using self-designed analysis system and dividing the samples into four categories marked with ‘well controlled’, ‘somewhat controlled’, ‘poorly controlled’, and ‘not controlled’, respectively, according to the simultaneous bloodglucose values. Then attemptedtopredict the condition of an input diabetic sample by using an ordinal regression technique, SVOR. Since the output of SVOR is an absolute value to indicate the sample’s condition, it does not involve the probability of a prediction. Therefore, discovered a method to map the output of SVOR to probabilities to decide which levels the input sample belongs to. This approach enabled to make a flexible judgment about the real condition of the disease, which is more suitable for medical applications than general classifiers.

2.2 The clinical potential of exhaled breath analysis for diabetes mellitus

Breath analysis methods appears on the verge of major breakthroughs that will hopefully exponentially accelerate the transition of past theoretical concepts into practical clinical devices relevant to diabetic populations. However, much still remains to be discovered regarding the origins, pathways, and pathophysiological roles of breath components; fortunately, many new analytical tools that can isolate VOC production from specific cells and tissues are available to now help answer these questions. Furthermore, many of the currently available findings were obtained in highly controlled experimental conditions, and their applicability to real life will need to be confirmed in larger scale clinical trials taking into account several confounding variables.

2.3 Application of MQ-138 Semiconductor Sensor for

Breath Acetone Detection

MQ-138 semiconductor sensor is one of the amongst metal oxide semiconductor based gas sensors which has high sensitivity to Acetone, Toluene and Formaldehyde. Besides, it has some advantages as gas sensor such as wide detecting scope, fast response, good stability, long lifetime, and simple drive circuit [6]. MQ-138 semiconductor sensor has been applied for sensor array in E-nose system to recognize gases and odors [7-9], food analysis [10]. However, this sensor has not been found in a single sensor system to detect specific gases or odors. Thus, a new application as a sensor system of MQ-138 semiconductor sensor to sense particular gases or odors is highly expected.

2.4 Monitoring rapid metabolic changes in health and type-1 diabetes with breath acetone sensors

The selective monitoring of breath markers (e.g., acetone for lipolysis) with sensors to track metabolic changes that can reveal disease-related abnormalities remains challenging. Here, subtle breath acetone changes during fasting, exercise and glucose ingestion are tracked in two model situations: Patients suffering from type-1 diabetes mellitus and healthy subjects (total: 19 volunteers) were monitored using chemoresistive sensors based on Si/WO3 nanoparticles. Specifically, each subject cycled after overnight fasting to stimulate fatty acid oxidation followed by an oral glucose tolerance test, as monitored by capillary blood glucose and β-hydroxybutyrate (BOHB) concentrations. The sensor recognized accurately the individual breath acetone patterns before and after OGTT (both R2 = 0.9) at negligible interference, for instance, from glucose ingestion-associated volatiles (e.g., ethanol) or isoprene, as confirmed by high-resolution mass spectrometry. Furthermore, distinct differences in the breath acetone patterns of T1DM over healthy subjects were revealed including higher (t-test, $p = 0.006$) breath acetone ratio 2 h after starting the OGTT.

2.5 Continuous Monitoring of Breath Acetone, Blood Glucose and Blood Ketone in 20 Type 1 Diabetic Outpatients Over 30 Days

It would be insightful to study longitudinal variations of breath acetone concentration in given individuals that have no baseline effect resulting from individual physiological heterogeneity. We carried out a daily-based continuous monitoring of BA, blood glucose , and blood ketone (BK) in 20 type 1 diabetic outpatients and 5 healthy volunteers over a period of 30 days. 600 breath samples from the T1D outpatients and the healthy subjects were collected and tested. BA was measured using a cavity ringdown BA analyzer. Simultaneous BG and BK levels were also measured using a standard BG/ketone meter. Our findings include: The T1D subjects have elevated mean BA concentrations as compared to the controls. There exists a positive correlation ($R=0.57$, $P<0.05$) between the individual mean BA concentration and the individual mean BK in the 20 T1D outpatients. Some adverse or abnormal physiological conditions such as diabetic ketoacidosis, low body mass index or a special daily activity (e.g., exercises) can be identified via an abnormal BA concentration. This study suggests that continuous monitoring BA is useful in monitoring some abnormal physiological status such as T1D outpatients with very high BG or ketone bodies.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Currently,the most widely used technique for measuring the glucose level is with the help of enzymatic strips. The level of testing results are not constant and varies depending upon the enzymatic strips and the quality of the instrument used. This use invasive method to collect the blood samples.

3.2 DISADVANTAGES

- It is Painful.
- It can Lead to Non-Compliance.
- It is Costly.
- It is not Always Accurate.
- It can Cause Infection.
- It can Lead to Skin Changes.

3.3 PROPOSED SYSTEM

Acetone is one of the VOC which obtained from fatty acid metabolism and act as an important biomarker for diabetes detection. In this System, Machine Learning Model will be developed to correlate exhaled acetone with the blood sugar level .This model will be developed using exhaled acetone (collected from the test chamber) taken as input and Blood Sugar value (collected from the Lab) taken as output of different volunteers . The person(with /without blood sugar) blows in the designed test chamber and their acetone reading is collected in parts per billion . The acetone reading will be initially analog so, we will use A/D convertor to convert it into digital . Then we will give this reading to Machine Learning based artificial neural network connected to Raspberry PI .To insure accuracy, volunteer will be requested to blow several times so as to get the average value for acetone .

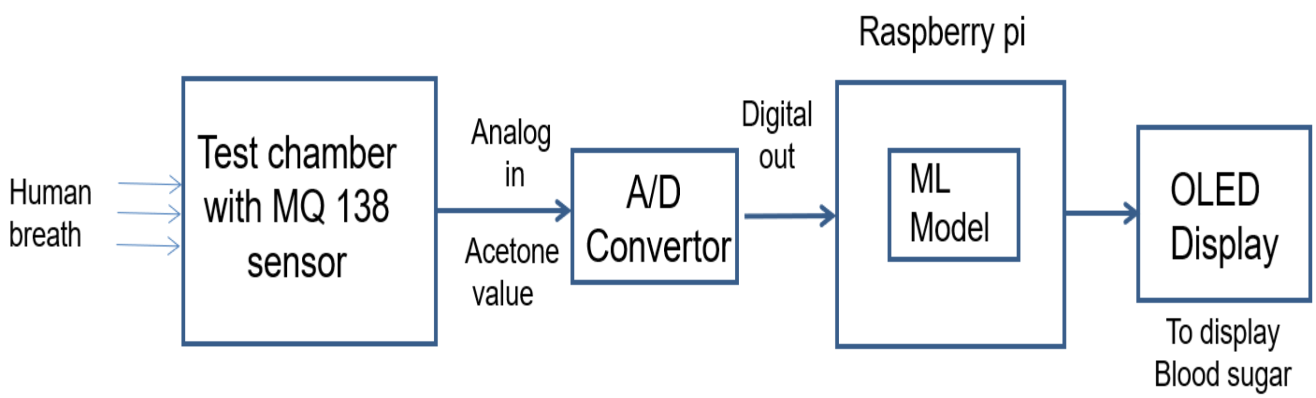
3.4 ADVANTAGES

Non-invasive glucometer is an advancement in the range of glucometers that do not invade the skin. The working principle of non-invasive glucometer is based on three techniques: spectroscopy, transdermal and other techniques . A saliva-based non-invasive glucometer is an imminent glucometer which uses saliva as an analyte to help diabetic patients to succumb to various difficulties. These are implanted with the glucose sensor based on the electrochemical principle which is powered wirelessly to give an accurate amount of glucose.

CHAPTER 4

PROJECT DISCRIPTION

4.1 OVERALL ARCHITECTURE



4.1 Overall architecture

MQ-138: MQ-138 semiconductor sensor is one of the amongst metal oxide semiconductor based gas sensors which has high sensitivity to Acetone, Toluene and Formaldehyde. Besides, it has some advantages as gas sensor such as wide detecting scope, fast response, good stability, long lifetime, and simple drive circuit. MQ-138 semiconductor sensor has been applied for sensor array in E-nose system to recognize gases and odors, food analysis. However, this sensor has not been found in a single sensor system to detect specific gases or odors. Thus, a new application as a sensor system of MQ-138 semiconductor sensor to sense particular gases or odors is highly expected.

A/D CONVERTOR: An A/D converter is a device that converts analog signals (usually voltage) obtained from environmental (physical) phenomena into digital format.

ML MODEL: A machine learning model is defined as a mathematical representation of the output of the training process. Machine learning is the study of different algorithms that can improve automatically through experience & old data and build the model. A machine learning model is similar to computer software designed to recognize patterns or behaviors based on previous experience or data. The learning algorithm discovers patterns within the training data, and it outputs an ML model which captures these patterns and makes predictions on new data.

OLED DISPLAY: An OLED is a solid-state device consisting of a thin, carbon-based semiconductor layer that emits light when electricity is applied by adjacent electrodes. It can produce some of the best picture quality of any display technology.

CHAPTER 5
SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

- MQ138 Sensor
- Raspberry PI
- A/D Convertor
- OLED Display
- Test Chamber

The **hardware requirements** may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design.

5.2 SOFTWARE REQUIREMENTS

- Language :Python
- Documentation :Microsoft office
- Operating System :Raspbian OS
- Software: MATLAB

The **software requirements** provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing asksand tracking the team's and tracking the team's progress throughout the development activity.

CHAPTER 6
SYSTEM ANALYSIS

6.1 PYTHON

Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high-level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming. Python is a high-level general-purpose programming language that can be applied to many different classes of problems. The language comes with a large standard library that covers areas such as string processing (regular expressions, Unicode, calculating differences between files), internet protocols (HTTP, FTP, SMTP, XML-RPC, POP, IMAP), software engineering (unit testing, logging, profiling, parsing Python code), and operating system interfaces (system calls, filesystems, TCP/IP sockets).There are probably millions of users, though it’s difficult to obtain an exa bct count. Python is available for free download, so there are no sales figures, and it’s available from many different sites and packaged with many Linux distributions, so download statistics don’t tell the whole story either. The comp.lang.python newsgroup is very active, but not all Python users post to the group or even read it.

6.2RASPBERRY PI

Raspberry Pi is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom.[14] The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries.[15][16][17] The original model became more popular than anticipated,[18] selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring,[19] because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of the HDMI and USB standards.

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz

- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

Operating system: Linux (Raspberry Pi OS)

CPU: Pi 3 A+: 1.4 GHz quad-core A53 64-bit

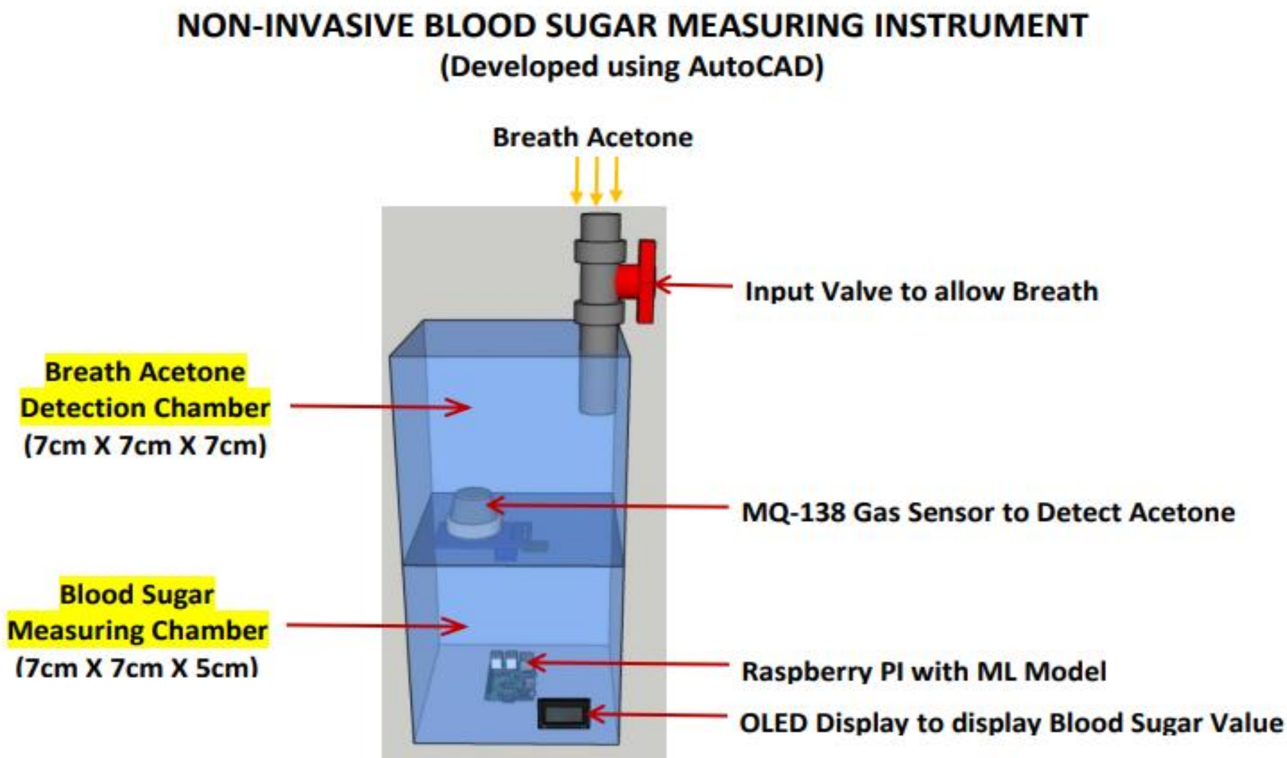
Graphics: Pi 3 A+: Broadcom Video Core IV 400

CHAPTER 7
MODULES

7.1 MODULE 1:CONSTRUCTION OF CHAMBER

Acetone detection chamber consists of an input valve, gas room which breath acetone is drawn into it and MQ-138 sensor to measure breath acetone. The analog output from MQ-

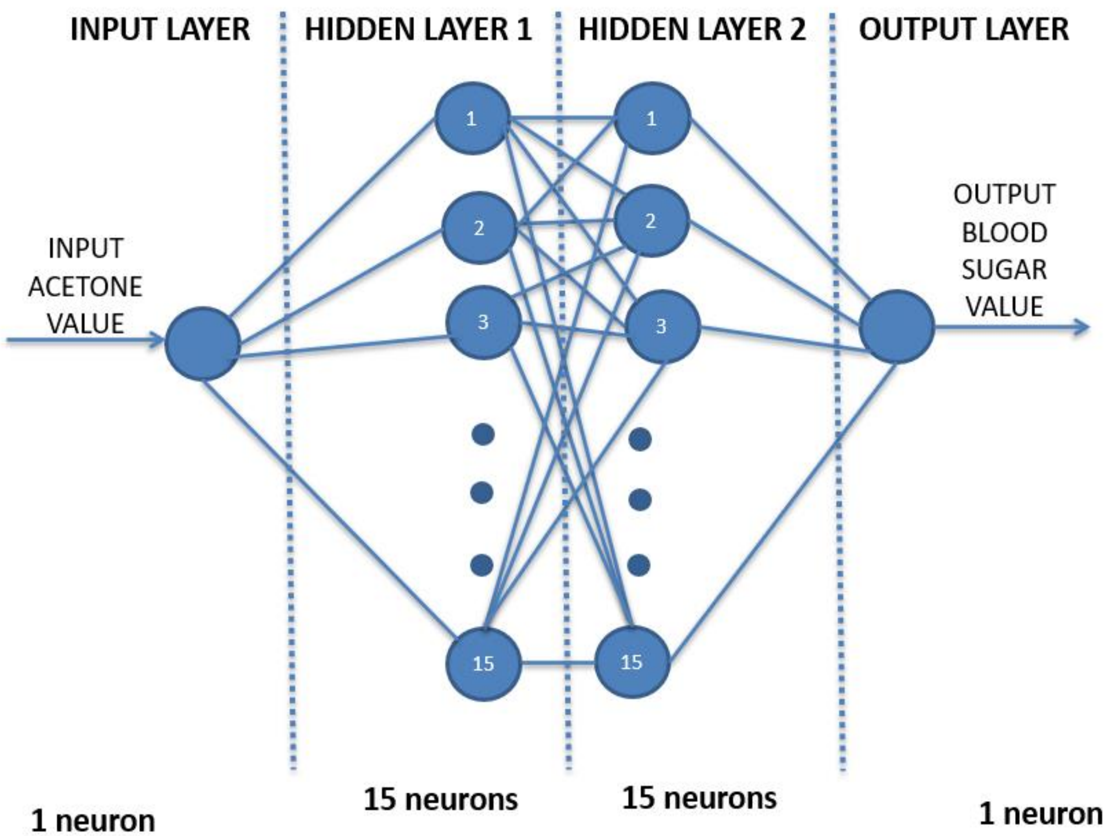
138 sensor is converted to digital output by using 24-bit A/D converter and interfaced to Raspberry PI. The Raspberry PI will measure breath acetone value.



7.1 CHAMBER MODEL

7.2 MODULE 2: DEVELOPMENT OF MACHINE LEARNING MODEL TO ESTIMATE BLOOD SUGAR LEVEL FROM ACETONE

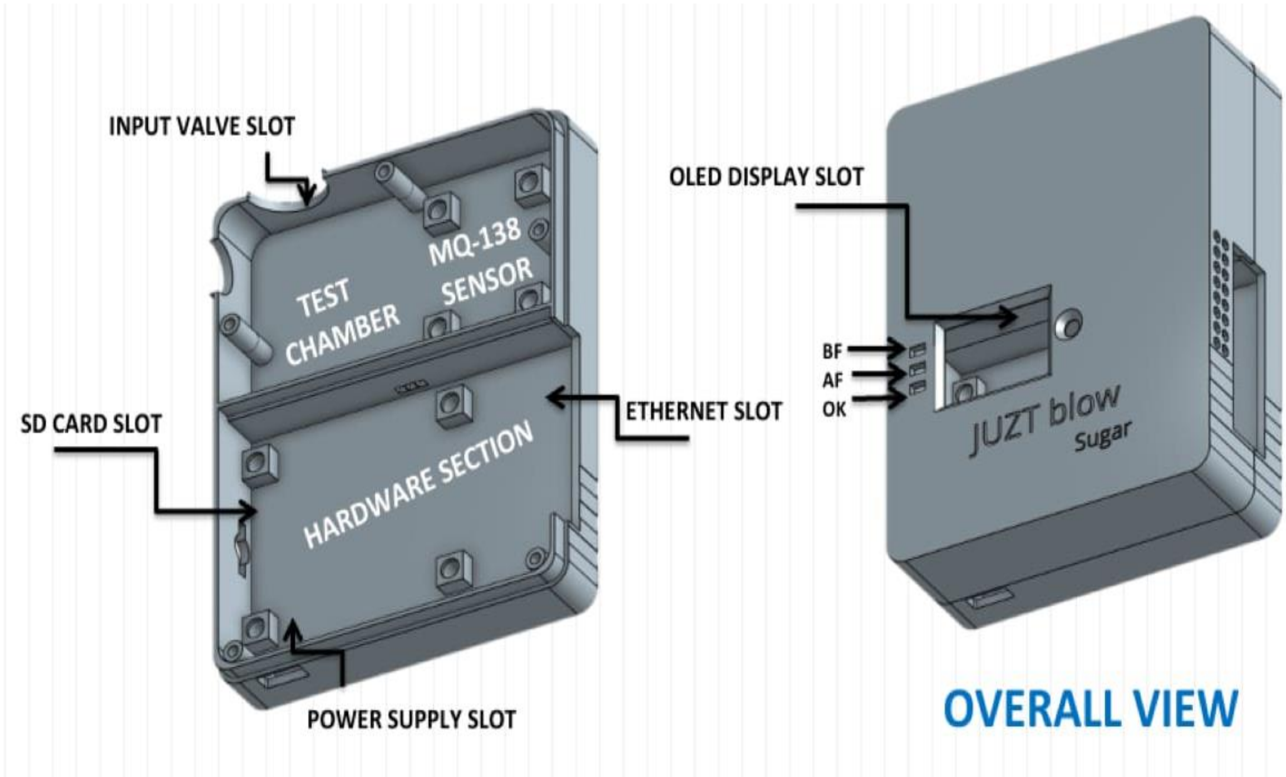
Breath Acetone values are measured using breath acetone detection chamber from 500 different diabetes and non-diabetes patients and also blood sugar values are measured from lab for the same 500 patients. A suitable Machine Learning model will be developed using exhaled acetone (collected from breath acetone detection chamber) and Blood Sugar value (collected from the Lab) of 500 different diabetes and non-diabetes patients. The developed Machine Learning Model will correlate exhaled acetone with the blood sugar level.



7.2 ARTIFICIAL NEURAL NETWORK

7.3 MODULE 3: DEPLOYMENT OF ML MODEL IN RASPBERRY PI

The developed ML model will be deployed in Raspberry Piand which is placed inside blood sugar measuring chamber along with OLED display. If breath acetone of a new patient is blown in to this non-invasive glucometer, it will display blood glucose value on OLED display. The entire programme will be developed using Artificial Neural Network.



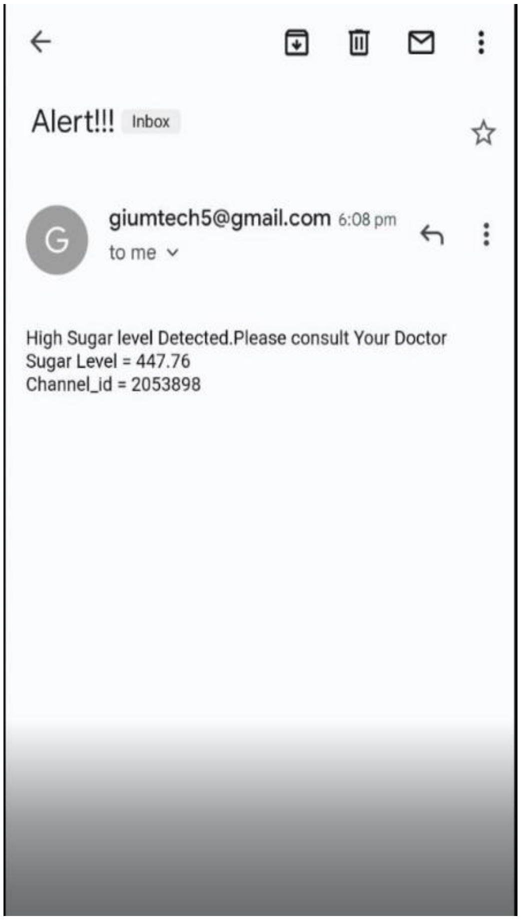
7.3 3D PRINTING DEMO

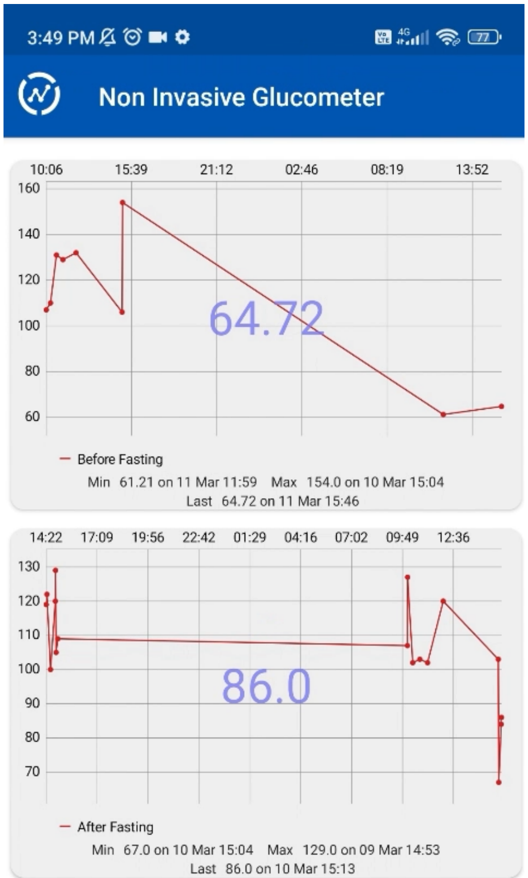
MODULE : 4

ADDITIONAL FEATURES

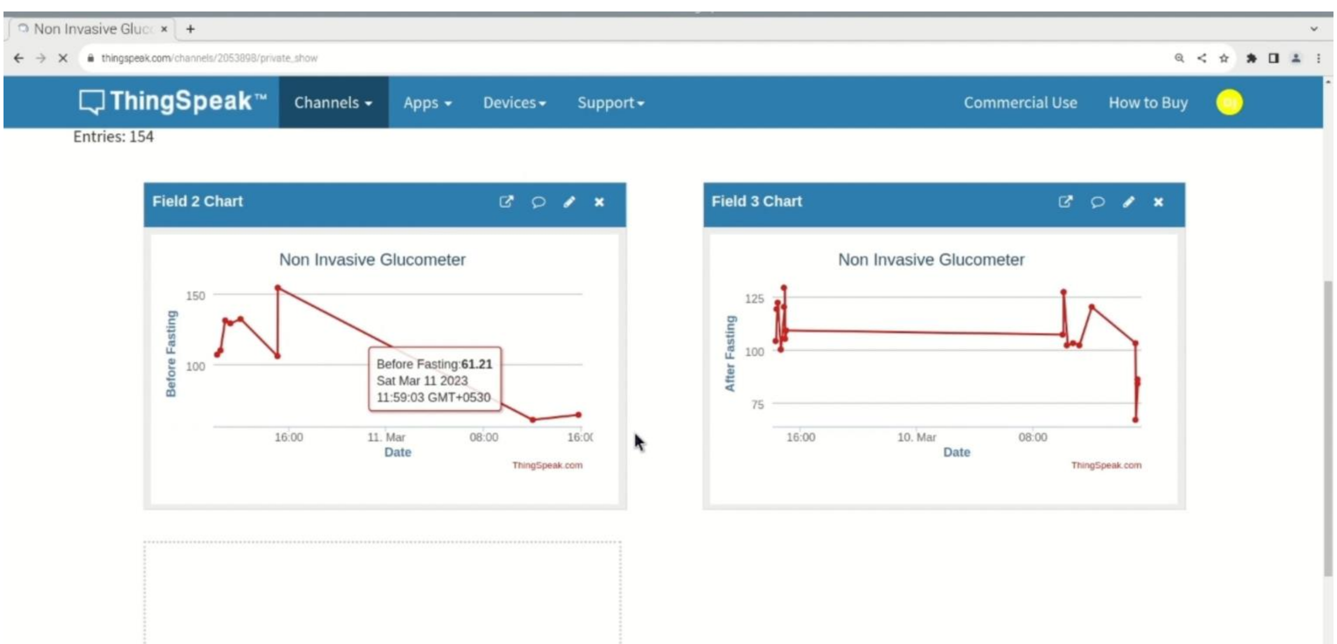
- Blood sugar data is updated in ThingSpeak cloud using IoT and can be accessed
- Email alerts are sent to family members and doctors if the sugar value falls below or exceeds the threshold.

7.4 EMAIL DEMO



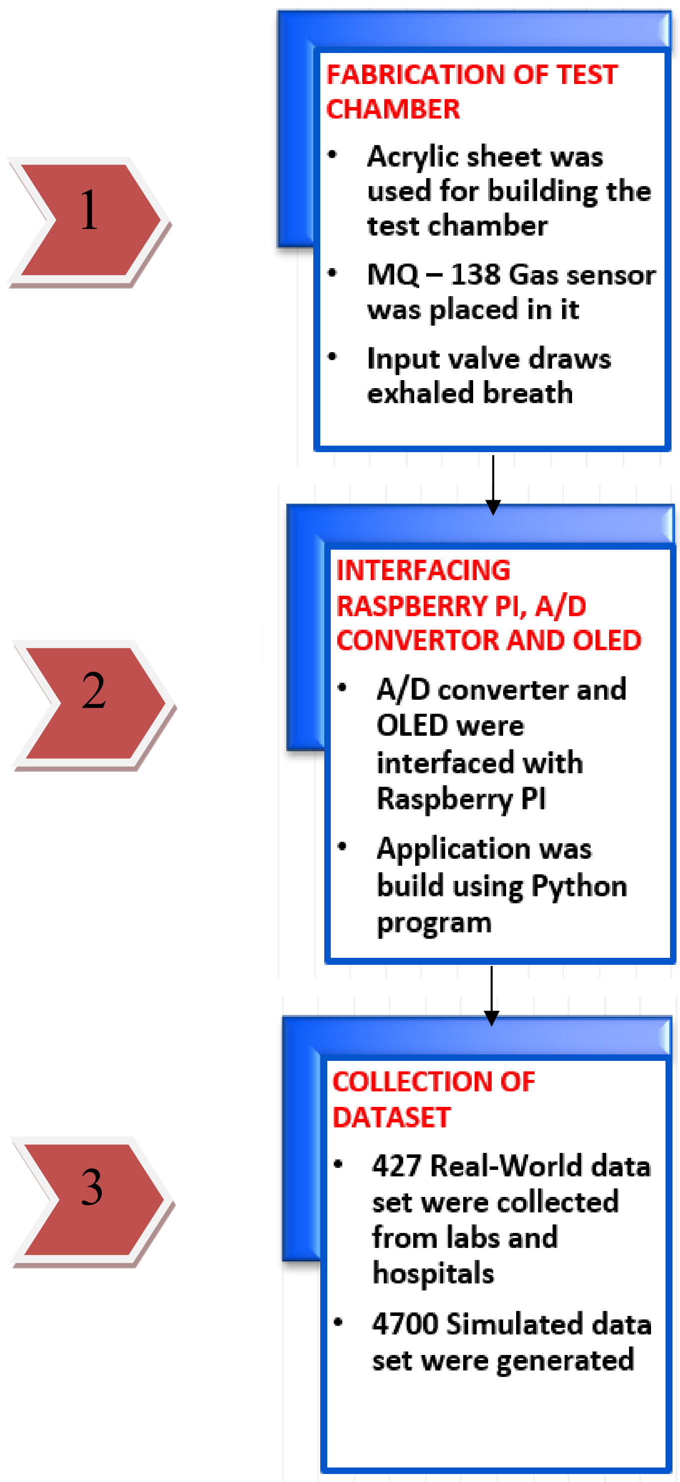


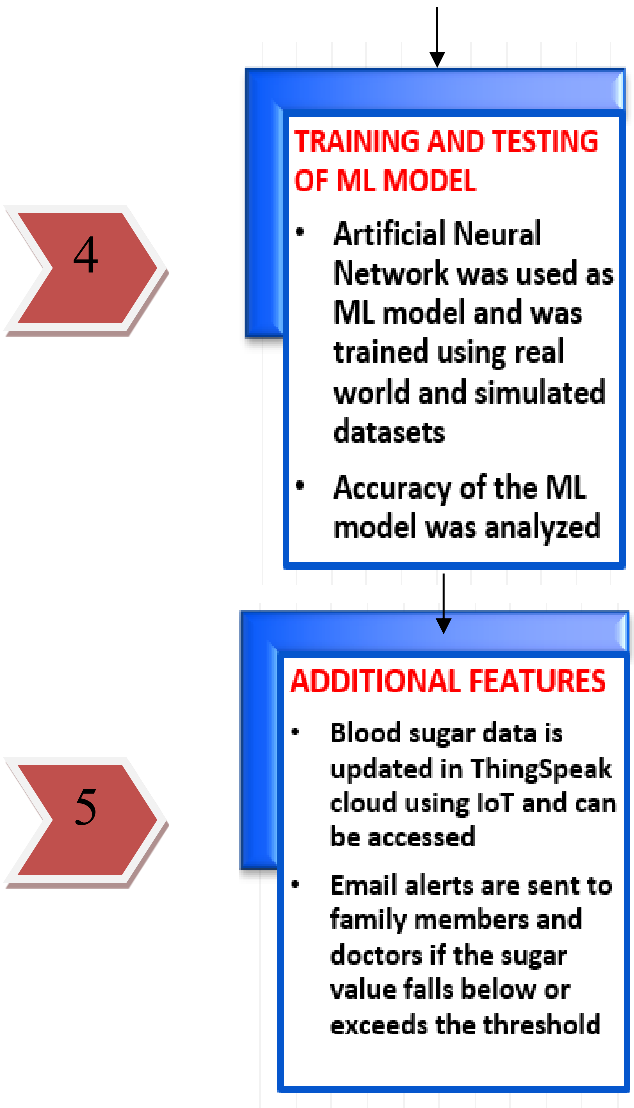
7.4 THINGSSPEAK IOT DEMO



CHAPTER 8

IMPLEMENTATION

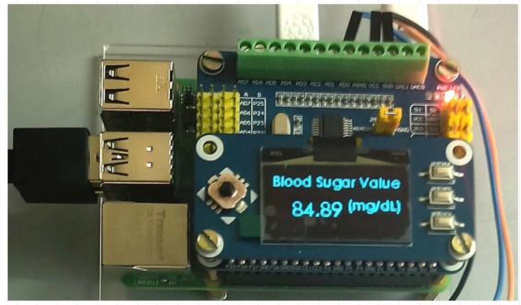




Acetone range : 300 - 5000 ppb

Blood sugar range : 60 - 600 mg/dL

Breath Acetone value is measured using MQ – 138 sensor and converted to 24 bit Digital value by high precision A/D converter and displayed on OLED display.



Breath Acetone value is given as direct input to ANN based ML model and the output is Blood Sugar value and it is displayed on OLED display.

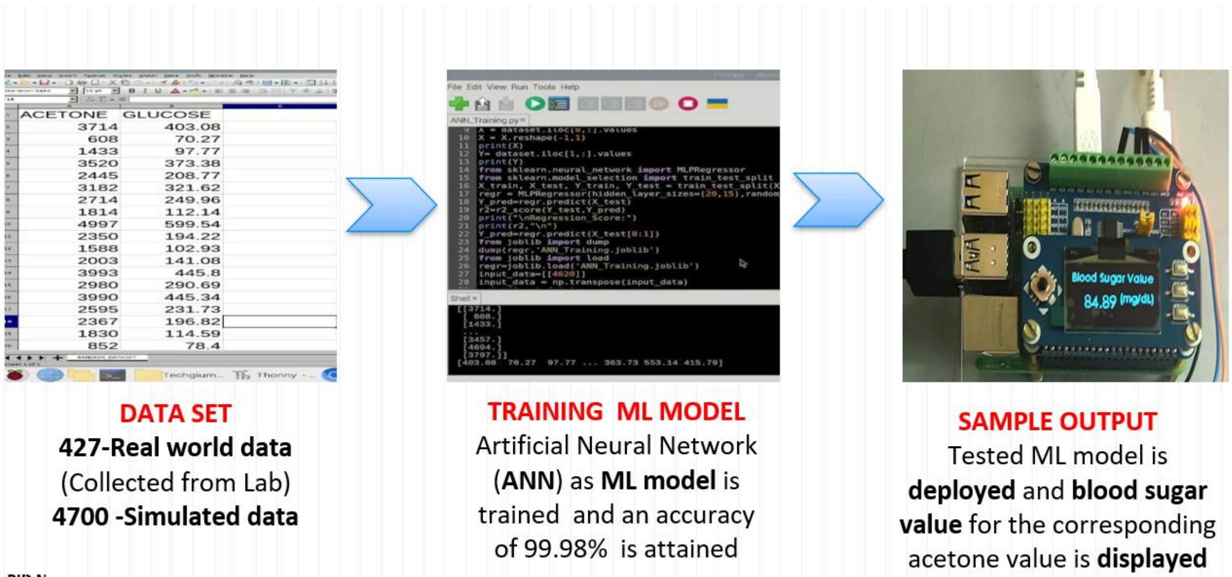
8.1 ACETONE AND BLOOD SUGAR VALUE

CHAPTER 9

TESTING AND ANALYSIS

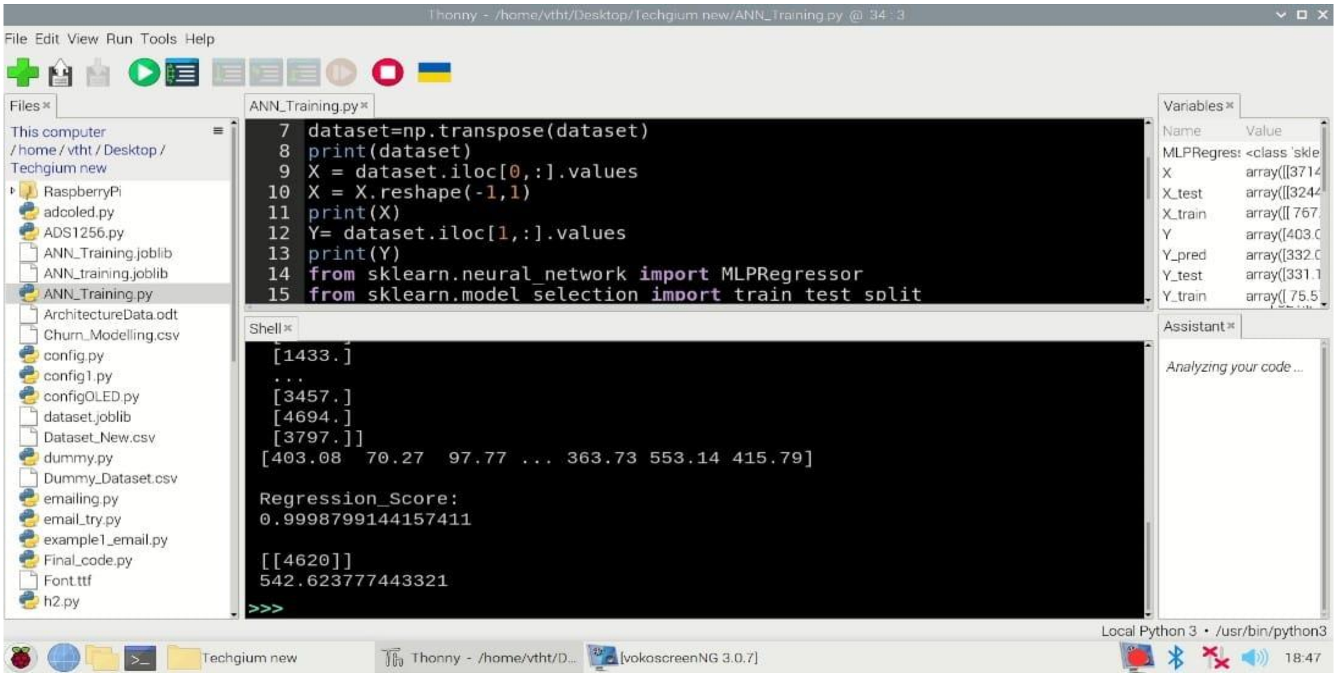
We have used 427 real world and 4700 simulated dataset and trained the Artificial Neural Network as Machine Learning Model and we got an accuracy of 99.98 percentage.

After training we tested the result by blowing into the chamber and adjusted the measurements. The tested ML Model is deployed in the raspberry pi and blood sugar value for the corresponding acetone value is diaplayed.



9.1 TRAINING OF ML MODEL

we have used regression testing so we got the regression score.



9.2 REGRESSION SCORE

CHAPTER 10

CODE

```
import ADS1256
import SH1106
import config1
import subprocess
import RPi.GPIO as GPIO
import SH1106OLED
import time
import configOLED
import traceback
import numpy as np
import pandas as pd
import joblib
import httplib2
import urllib3
import thingspeak
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage
from PIL import Image,ImageDraw,ImageFont

channel_id= 2053898
write_key1='XSQN721HU79GG18H'
a=2

ADC = ADS1256.ADS1256()
```

```
ADC.ADS1256_init()
disp = SH1106OLED.SH1106()
ADC_Value = ADC.ADS1256_GetAll()

#GPIO define
RST_PIN        = 25
CS_PIN         = 8
DC_PIN         = 24
#joystick
KEY_UP_PIN     = 6
#button key
KEY1_PIN       = 21
KEY2_PIN       = 20
KEY3_PIN       = 16

disp = SH1106OLED.SH1106()
disp.Init()

# Clear display.
disp.clear()

GPIO.setmode(GPIO.BCM)
GPIO.setup(KEY_UP_PIN,     GPIO.IN, pull_up_down=GPIO.PUD_UP)    # Input with pull-up
GPIO.setup(KEY1_PIN,       GPIO.IN, pull_up_down=GPIO.PUD_UP)      # Input with pull-up
GPIO.setup(KEY2_PIN,       GPIO.IN, pull_up_down=GPIO.PUD_UP)      # Input with pull-up
```

GPIO.setup(KEY3_PIN,   GPIO.IN, pull_up_down=GPIO.PUD_UP)   # Input with pull-up

 # Create blank image for drawing.

image = Image.new('1', (disp.width, disp.height), "WHITE")

 # Get drawing object to draw on image.

draw = ImageDraw.Draw(image)

font = ImageFont.truetype('Font.ttf',25)

font10 = ImageFont.truetype('Font.ttf',10)

draw.text((6,10),'JUZTblow', font = font, fill = 0)

draw.text((95,38),'Sugar', font = font10, fill = 0)

disp.ShowImage(disp.getbuffer(image))

time.sleep(5)

disp.clear()

image1 = Image.new('1', (disp.width, disp.height), "WHITE")

draw = ImageDraw.Draw(image1)

font = ImageFont.truetype('Font.ttf', 15)

font10 = ImageFont.truetype('Font.ttf',10)

font20 = ImageFont.truetype('Font.ttf',10)

draw.text((25,5), 'Instructions ', font = font, fill = 0)

draw.text((12,25),'To Be Followed',font = font, fill = 0)

disp.Init()

disp.ShowImage(disp.getbuffer(image1))

time.sleep(5)

disp.clear()

image2 = Image.new('1', (disp.width, disp.height), "WHITE")

```
draw = ImageDraw.Draw(image2)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((15,20), 'Open the Knob', font = font, fill = 0)
draw.text((2,1), '1', font = font, fill = 0)
draw.text((4,5),'chamber carefully ',font = font10, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image2))
time.sleep(5)
disp.clear()

image3 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image3)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',15)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((1,1), '2', font = font, fill = 0)
draw.text((35,20),'Blow into',font = font10, fill = 0)
draw.text((35,40),'Chamber',font = font, fill = 0)
draw.text((4,5),'chamber carefully ',font = font10, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image3))
time.sleep(5)
disp.clear()

image4 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image4)
```

```
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((1,1), '3', font = font, fill = 0)
draw.text((15,20), 'Close the Knob', font = font, fill = 0)
draw.text((18,40),  'while blowing',font = font, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image4))
time.sleep(5)
disp.clear()

image7 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image7)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((5,12), 'Before fasting - BF ', font = font, fill = 0)
draw.text((5,32),'After fasting  - AF',font = font, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image7))
time.sleep(5)
disp.clear()

image6 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image6)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
```

```
draw.text((20,1), 'After Blowing', font = font, fill = 0)
draw.text((14,25),'Select BF or AF',font = font, fill = 0)
draw.text((22,47),'to continue',font = font, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image6))
time.sleep(5)



while 1:

    if GPIO.input(KEY1_PIN): # button is released
        print()
    else:
        draw.ellipse((70,0,90,20), outline=255, fill=1) #A button filled

        ADC_ch0 = ADC_Value[0]*3.3/0x7fffff*1000
        ADC_value=[[round(ADC_ch0,2)]]
        ADC_value1=str(ADC_value)
        from joblib import load
        regr=joblib.load('ANN_Training.joblib')
        predicted_output=regr.predict(ADC_value)
        predicted=np.round(predicted_output,2)
        for output in predicted:
            glucose=str(output)
            print(glucose)
        a=output
        image1 = Image.new('1', (disp.width, disp.height), "WHITE")
        draw = ImageDraw.Draw(image1)
```

```
    font = ImageFont.truetype('Font.ttf', 10)
    font10 = ImageFont.truetype('Font.ttf',13)
    font20 = ImageFont.truetype('Font.ttf',10)
    draw.text((5,5),'Before fasting is selected', font = font, fill = 0)
    draw.text((20,22),'Are you sure?', font = font10, fill = 0)
    draw.text((1 ,45),'press OK to continue',font = font10, fill = 0)
    disp.ShowImage(disp.getbuffer(image1))
    time.sleep(1)


if(GPIO.input(KEY2_PIN)):
    print()
else:
    draw.ellipse((100,20,120,40), outline=255, fill=1) #B button filled
    ADC_ch0 = ADC_Value[0]*3.3/0x7fffff*1000
    ADC_value_A=[[round(ADC_ch0,2)]]
    ADC_value1_A=str(ADC_value_A)
    from joblib import load
    regr_A=joblib.load('ANN_Training.joblib')
    predicted_output_A=regr_A.predict(ADC_value_A)
    disp.Init()
    # Clear display.
    disp.clear()
    predicted_A=np.round(predicted_output_A,2)
    for output_A in predicted_A:
        glucose_A=str(output_A)
        print(glucose_A)
    output=0
```

```
    a=output_A
    image1 = Image.new('1', (disp.width, disp.height), "WHITE")
    draw = ImageDraw.Draw(image1)
    font = ImageFont.truetype('Font.ttf', 11)
    font10 = ImageFont.truetype('Font.ttf',13)
    font20 = ImageFont.truetype('Font.ttf',10)
    draw.text((5,1), 'After fasting is selected', font = font, fill = 0)
    draw.text((25,22), 'Are you sure?', font = font10, fill = 0)
    draw.text((1,45),'press OK to continue',font = font10, fill = 0)
    draw.text((60,30), '(mg/dL)', font = font10, fill = 0)
    disp.ShowImage(disp.getbuffer(image1))
    time.sleep(1)


if GPIO.input(KEY3_PIN):# button is released
    print()
else:
    if(a==output):
        image2 = Image.new('1', (disp.width, disp.height), "WHITE")
        draw = ImageDraw.Draw(image2)
        font = ImageFont.truetype('Font.ttf', 15)
        font10 = ImageFont.truetype('Font.ttf',10)
        font20 = ImageFont.truetype('Font.ttf',10)
        draw.text((20,22), 'Measuring...', font = font, fill = 0)
        disp.ShowImage(disp.getbuffer(image2))
        time.sleep(5)
        image1 = Image.new('1', (disp.width, disp.height), "WHITE")
        draw = ImageDraw.Draw(image2)
```

```
image1 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image1)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',12)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((1,1), 'Before fasting:', font = font10, fill = 0)
draw.text((10,23), 'Blood Sugar Value:', font = font10, fill = 0)
draw.text((20,45),str(glucose),font = font, fill = 0)
draw.text((60,45), '(mg/dL)', font = font10, fill = 0)
disp.ShowImage(disp.getbuffer(image1))
time.sleep(1)
print("Acetone(ppb): " + ADC_value1)
print("Blood Sugar Value(mg/dL): " + glucose)
def measure(channel):
    try:
        response = channel.update({'field2': output})
    except:
        print("Connection Failure")
if _name_ =="_main_":
    channel = thingspeak.Channel(channel_id,write_key1)
measure(channel)
sender_email = "giumtech5@gmail.com"
sender_password = "hmpd hhjc fayd oxpe"
# Email details
recipient_email = "divyachitra391@gmail.com"
subject = "Alert!!!"
output=40
if output>=250:
```

```
    body = "\nHigh Sugar level detected.\nPlease consult your Doctor."
    sugar="Sugar level = "
    url1="\nChannel_ID = "
    url='2053898'
    message1 = MIMEMultipart()
    message1["From"] = sender_email
    message1["To"] = recipient_email
    message1["Subject"] = subject

    # Add the message body
    message1.attach(MIMEText(sugar + glucose + body + url1 + url, "plain"))

    with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
        smtp.starttls()
        smtp.login(sender_email, sender_password)

        # Send the email
        smtp.sendmail(sender_email, recipient_email, message1.as_string())

    print("Email sent successfully!")

elif output<=50:
    body = "\nLow Sugar level detected.\nPlease consult your Doctor."
    sugar="Sugar level = "
    url1="\nChannel_ID = "
    url='2053898'
    message1 = MIMEMultipart()
    message1["From"] = sender_email
```

```
        message1["To"] = recipient_email
        message1["Subject"] = subject
        message1.attach(MIMEText(sugar + glucose + body + url1 + url, "plain"))

        # Add the message body
       message.attach(MIMEText(body, "plain"))
        with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
            smtp.starttls()
            smtp.login(sender_email, sender_password)

            # Send the email
            smtp.sendmail(sender_email, recipient_email, message1.as_string())

        print("Email sent successfully!")

elif(a==output_A):
    image2 = Image.new('1', (disp.width, disp.height), "WHITE")
    draw = ImageDraw.Draw(image2)
    font = ImageFont.truetype('Font.ttf', 15)
    font10 = ImageFont.truetype('Font.ttf',10)
    font20 = ImageFont.truetype('Font.ttf',10)
    draw.text((20,22), 'Measuring...', font = font, fill = 0)
    disp.ShowImage(disp.getbuffer(image2))
    time.sleep(5)
    image1 = Image.new('1', (disp.width, disp.height), "WHITE")
    draw = ImageDraw.Draw(image1)
    font = ImageFont.truetype('Font.ttf', 14)
    font10 = ImageFont.truetype('Font.ttf',12)
```

```
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((1,1), 'After fasting:', font = font10, fill = 0)
draw.text((10,25), 'Blood Sugar Value:', font = font10, fill = 0)
draw.text((20,45),str(glucose_A),font = font, fill = 0)
draw.text((60,45), '(mg/dL)', font = font10, fill = 0)
disp.ShowImage(disp.getbuffer(image1))
time.sleep(1)
print("Acetone(ppb): " + ADC_value1_A)
print("Blood Sugar Value(mg/dL): " + glucose_A)
def measure(channel):
    try:
        response = channel.update({'field3': output_A})
    except:
        print("Connection Failure")
if _name_ =="_main_":
    channel = thingspeak.Channel(channel_id,write_key1)
measure(channel)
sender_email = "giumtech5@gmail.com"
sender_password = "hmpd hhjc fayd oxpe"
# Email details
recipient_email = "divyachitra391@gmail.com"
subject = "Alert!!!"
output_A=45
if output_A>=250:
    body = "\nHigh Sugar level detected.\nPlease consult your Doctor."
    sugar="Sugar level = "
    url1="\nChannel_ID = "
    url='2053898'
```

```
        message1 = MIMEMultipart()
        message1["From"] = sender_email
        message1["To"] = recipient_email
        message1["Subject"] = subject

        # Add the message body
        message1.attach(MIMEText(sugar+ glucose_A + body + url1 + url, "plain"))
        # Create a SMTP connection
        with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
            smtp.starttls()
            smtp.login(sender_email, sender_password)

            # Send the email
            smtp.sendmail(sender_email, recipient_email, message1.as_string())

        print("Email sent successfully!")
    elif output_A<=50:
        body = "\nLow Sugar level detected.\nPlease consult your Doctor."
        sugar="Sugar level = "
        url1="\nChannel_ID = "
        url='2053898'
        message1 = MIMEMultipart()
        message1["From"] = sender_email
        message1["To"] = recipient_email
        message1["Subject"] = subject

        # Add the message body
        message1.attach(MIMEText(sugar+ glucose_A + body + url1 + url, "plain"))
```

```
# Create a SMTP connection
with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
    smtp.starttls()
    smtp.login(sender_email, sender_password)

    # Send the email
    smtp.sendmail(sender_email, recipient_email, message1.as_string())

print("Email sent successfully!")
```

CHAPTER 11

CONCLUSION AND FUTURE WORK:

This approach enabled us to make a flexible judgment about the real condition of the disease, which is more suitable for medical applications than general classifiers. Frankly speaking, the current prediction accuracies are not quite high comparing with blood test, the results are promising because it does provide a possibility of non-invasive blood glucose measurement and further investigation on this topic will be continued

CHAPTER 12

REFERENCES

- Meixiu Sun, Zhennan Wang, Yuan Yuan, Zhuying Chen, Xiaomeng Zhao, Yingxin Li and Chuji Wang:Continuous Monitoring of Breath Acetone, Blood Glucose and Blood Ketone in 20 Type 1 Diabetic Outpatients Over 30 Days:Sun et al., J Anal Bioanal Tech 2017, 8:5
- Andreas T. Güntner, Ines C. Weber b, Stephanie Schon b, Sotiris E. Pratsinis b, Philipp, A. Gerber:Monitoring rapid metabolic changes in health and type-1 diabetes with breath acetone sensors :Sensors and Actuators: B. Chemical 367 (2022) 132182.
- Dongmin Guoa, David Zhanga, Lei Zhanga, Guangming Lub: Non-invasive blood glucose monitoring for diabetics by means of breath signal analysis:Sensors and Actuators B 173 (2012) 106–113.
- T I Nasution, R Asrosa and I Nainggolan: Application of MQ-138 Semiconductor Sensor for Breath Acetone Detection:Journal of Physics: Conf. Series 1116 (2018) 032023.
- Timothy Do Chau Minh a, Donald Ray Blake b, Pietro Renato Galassetti:The clinical potential of exhaled breath analysis for diabetes mellitus: 2012 Elsevier Ireland Ltd.
- Saraswati Kulkarni , Ruma Ghosh:A simple approach for sensing and accurate prediction of multiple organic vapors by sensors based on CuO nanowires: Department of Electrical Engineering, Indian Institute of Technology Dharwad, Karnataka, 580011, India
- Chenshan Gao, Xu Liu , Huiru Yang , Quan Zhou , Yingying Zhang , Anli Yang ,Huaiyu Ye, Yufei Liu a, Lei Zhang, Guoqi Zhang:A theoretical study of Ti–MoSe2 as a noninvasive type-1 diabetes diagnosis material for detecting acetone from exhaled breath:Vacuum 182 (2020) 109729

- Sadia Afreen, Jun-Jie Zhu:Rethinking EBAD: Evolution of smart noninvasive detection of diabetes:School of Chemistry and Chemical Engineering, Nanjing University, Nanjing, 210023, China