Level-Up Insulin: An Automated Insulin Delivery System

Anitra R
CB.AI.U4AIM24004
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu
cb.ai.u4aim24004@cb.students.amrita.edu

Naresh L
CB.AI.U4AIM24028
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu
cb.ai.u4aim24028@cb.students.amrita.edu

Nimisha Patel
CB.AI.U4AIM24029
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu
cb.ai.u4aim24029@cb.students.amrita.edu

Yatish S
CB.AI.U4AIM24050
School of Artificial Intelligence
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu
cb.ai.u4aim24050@cb.students.amrita.edu

***Abstract*—Around 11.4 % of India's population suffers from Diabetes. This rises the demand for a device that can provide continuous monitoring of glucose in blood. Conventional methods use invasive CGM devices and manual insulin injection. To look upon this problem and providing a better approach, this project aims infusing CGM device and insulin pump together, to provide a device which can automatically pump insulin as and when required. This project uses deep learning algorithms and NIR spectroscopy methods to combat the problem. The project aims to provide a compact, portable device that can make lives of diabetic patients easier. It uses Neural Networks and NIR spectroscopy sensor AS7263 to get the continuous glucose monitoring (CGM) non-invasively.The neural network is used to map the values of intensity from the sensor to the glucose values.We have used ESP32 and Arduino Nano microcontroller boards.The device also incorporates the RF transmitter to send the CGM values from the sensor to the pump, hence making the device handy and comfortable to use. The device also incorporates the insulin infusion pump which works on the feedback loop of the non-invasive CGM. It envisions the future scope of automated insulin delivery system based on instantaneous glucose level spikes, into the subcutenous fat layer. In case of hyperglycemia or hypoglycemia, to alert the patient about the condition, a buzzer system is incorporated. Our aim is to provide a better device for the betterment of medico-electrical field which can revolutionize the CGM monitoring and insulin injection for diabetic people.**

***Index Terms*—Automated Insulin injection, AI based insulin injection, Non-invasive glucose monitoring system, Non-invasive glucometer with automated infusion set**

I. Introduction

Diabetes facts and figures show the growing global burden for individuals, families, and countries. The IDF Diabetes Atlas (2021) reports that 10.5 % of the adult population (20-79 years) has diabetes, with almost half unaware that they are living with the condition [1]. Diabetes mellitus (diabetes) is a chronic and potentially life-threatening condition where the body loses its ability to produce insulin, or begins to produce or use insulin less efficiently, resulting in blood glucose levels that are too high (hyperglycemia). Over time, blood glucose levels above the normal range can damage your eyes, kidneys and nerves, and can also cause heart disease and stroke. In long run, it could be fatal. Diabetes is India's one of the fastest-growing chronic condition. The main types of diabetes are type 1, type 2, and gestational diabetes and all are equally harmful to physical and emotional wellbeing. This creates a demand for a device that can inject insulin automatically based upon fluctuating glucose levels. To look at this real-life medical problem, this project provides an efficient system by giving an intertwined system of Continuous Glucose Monitoring (CGM) and insulin pump. The feedback from CGM acts as input for insulin pump making it automated, and as and when required it can inject insulin directly in bloodstream. The key idea is to replace conventional invasive CGM devices with NIR spectroscopy sensor based glucose monitoring system, which provides the intensity of the light reflected and based upon that the glucose values are mapped using deep learning algorithms, feedforward neural networks. Manual insulin shots are tend to replace automated insulin injection into subcutaneous fat layer. Volume of insulin to be injected is determined using volume-time relation, in which volume is the known quantity and using simple volume of cylinder formula (assuming tube to be cylindrical), we can calculate the time for which reservoir needs to be kept open for a desired value of volume of insulin. The pump used for this purpose is peristaltic pump. The non-invasive glucose reading are taken from the superficial areas of the body, most suitably, fingertips. This is so because the sudden change in glucose levels is detected instantly in the superficial areas of the body due to presence of thin capillaries. The device is a closed loop device aiming for basal dosing of insulin into the body. The NIR sensor provides the intensity values as a marker of glucose levels, therefore to map these intensity values to the glucose values and give predictions based on the mapping, we are using neural networks. This helps in providing a reliable and better accuracy for the non-invasive glucose reading. The microcontroller board used for the CGM is Arduino Nano to reduce the size of the device and for the infusion set ESP32 has been used due to its Bluetooth module and versatility. The RF module is used to transmit the values taken by CGM sensor to the infusion pump to complete the feedback loop, hence making the device handy and comfortable to use. Based on the received value, the infusion sets the insulin out if the threshold value is received according to the treat-to-range table values provided in the methodology section. Treat-to -range is basically an algorithm for the dosing of insulin based on the glucose levels. To alert the patient in case of either hyperglycemia or hypoglycemia, buzzer system is incorporated to provide a security check over the abnormal values of glucose reading. The buzzer system activates according to the treat- to-range table values of the glucose. To make it portable and more comfortable, size of the device is kept as small as possible. This project can be a gamechanger to the people who need frequent shots of insulin, and/or for elderly people.

Contributions:
This project models a non-invasive CGM system using AS7263 NIR spectroscopy sensor, based on the fact that glucose absorbs light in the wavelength region 850nm, due to secondary harmonic oscillations of glucose molecules. The AS7263 is interfaced with Arduino Nano, providing compatibility, to make a CGM watch module. The module can be used to take intensity readings directly from superficial areas of the body.The returned intensity values from the sensor are mapped with the real time glucose readings, monitered at that intensity. To predict the glucose readings for a specific intensity, neural networks are introduced into the model. To make the device hassle-free, RF transmitter and receiver is used to transmit the values from the CGM to insulin infusion pump, placed in the abdominal region. The value intensity value received in the insulin infusion set is subjected through the neural network model integrated with ESP32 as a C header file, which inturn maps the intensity values to the blood glucose value. Based on the treat-to-range algorithm, the infusion pump will be triggered at the threshold value of glucose, thereby providing automated insulin injection into the subcutenous fat layer. For combating abnormal insulin levels, or an emergency situation, buzzer activation system is incorporated which provides a check over the abnormal glucose values and notifies immediately in case of any anomaly.

Organisation:
The paper proceed with the literature review done over the taken problem statement. Later the methodology section is introduced which shows our work on dataset generation, neural network implementation and prototype designing. This section deals with the experiments performed for dataset generation, the prototype design, the pin connections and circuit diagram and workflow of each step. The paper concludes with results and comparisons with the existing models, followed by conclusion section.

II. Literature Review

S. Hu, N. Y. Philip et. al [2] in 2011 proposed a model R.S. H. Istepanian, S. Hu, N. Y. Philip, in which basically The non-invasive diabetes sensors from the patients are linked via IPv6 connectivity to the relevant healthcare provider or the diabetes centre. It uses the new concept that matches the functionalities of m-health and IoT for a new and innovative future (4G health) applications.

Bequette, B. W. et. al [3] in 2012 came up with Glucose monitors (GMs), numerous pumps, and very many self monitoring (fingerstick) blood glucose models. In this model, New patch pumps are placed directly on the skin without additional tubing. Algorithms include on–off for prevention of overnight hypoglycemia.

Giovanni Sparacino, et. al [4] in 2013 worked on Real-Time Improvement of Continuous Glucose Monitoring Accuracy: The smart sensor concept. The model proposed is sCGM sensor(Dexco m SEVEN Plus). The methodology involves enhancing a commercial CGM sensor (Dexcom SEVEN Plus) with three real-time software modules for denoising, enhancement, and prediction of glucose levels.The study improves the accuracy and reliability of glucose measurements. And also results in better management of blood glucose levels for individuals with Type 1 Diabetes (T1D).

Messer LH, et. al [5] presented a clinical overview of insulin pump therapy. The model proposed was V-Go 20 V-Go 30. It is a digital device that delivers rapid acting insulin through a small catheter inserted into the body. Simplified pump technology has been developed specifically targeting the needs of people with type 2 diabetes.

Templer S et.al [6] worked on Closed-Loop Insulin Delivery Systems: Past, Present, and Future Directions, based on the model MiniMed 670G (Medtronic hybrid closed loop system). These modern closed loop systems use interstitial glucose sensing, subcutaneous insulin pumps, and increasingly sophisticated algorithms. It potentially improves glycemic control and quality of life for patients.

Neinstein AB, et.al [7] wrote about Smart Insulin Pens: Advancing Digital Transformation and a Connected Diabetes Care Ecosystem. They used Inpens (smart insulin pen that tracks insulin doses). The methodology combines retrospective data analysis, patient education, and collaborative decision-making facilitated by SIP and CGM technology.
Retrospective data reviews serve as powerful educational tools, helping

patients understand the impact of their insulin delivery habits on glucose control.

Tommerdahl, K.Let. al [8] proposed Continuous Glucose Monitor, Insulin Pump, and Automated Insulin Delivery Therapies for Type 1 Diabetes. It uses Continuous Subcutaneous Insulin Infusion or CSII. It includes using Multiple Daily Injections (MDI) with a mix of long-acting and short-acting insulin, or insulin pumps for continuous delivery. Intensive insulin therapy, including insulin pumps, helps maintain blood glucose levels within a target range, reducing the risk of both hyperglycemia and hypoglycemia.

Li, H., et.al [9] published about Skin-Wearable Sensors for Non Invasive Health Monitoring Applications. They used Adhesive radio frequency identification (RFID) sensor patch. These flexible and stretchable non invasive skin-wearable electronics are classified according to their input energy form , i.e., thermoelectrical signals, neural and electrical. These electronics demonstrate the capability of skin-like devices for on-time and continuous long term health monitoring.

, Kim, J. H. et. al [10] proposed Advances in Continuous Glucose Monitoring and Integrated Devices for Management of Diabetes with Insulin-Based Therapy: Improvement in Glycemic Control. They used MiniMed 780G model. The focus is on evaluating how these technologies, when combined with structured education, enhance glycemic outcomes, particularly in achieving target HbA1c while minimizing hypoglycemia. CIPs with built in bolus calculators help reduce missed insulin doses and improve adherence, which directly impacts glycemic control and prevents high blood sugar fluctuations.

Neha Verma et. al [11] worked on Monitoring Technologies-Continuous Glucose Monitoring, Mobile Technology, Biomarkers of Glycemic Control. The Dexcom G6 system model was used by them for this work. CGM improves outcomes by reducing A1C(biomarker), hypoglycemia, and increasing time in target range, with streamlined data communication via mobile and clinic tools. It discusses the use of mobile devices and decision-support tools to empower patients, making diabetes management more convenient and informed.

III. METHODOLOGY

A. *Non-invasive Glucose Monitoring*

Non-invasive glucose monitoring employs an optical technology called Near InfraRed Spectroscopy or NIR spectroscopy. To measure glucose levels using near-infrared light, the light must be absorbed by the glucose molecules and scattered by other chemicals in the tissue. Near-infrared spectroscopy generally selects areas with rich blood vessels and thin skin, such as the fingertips or earlobes, for measurement.

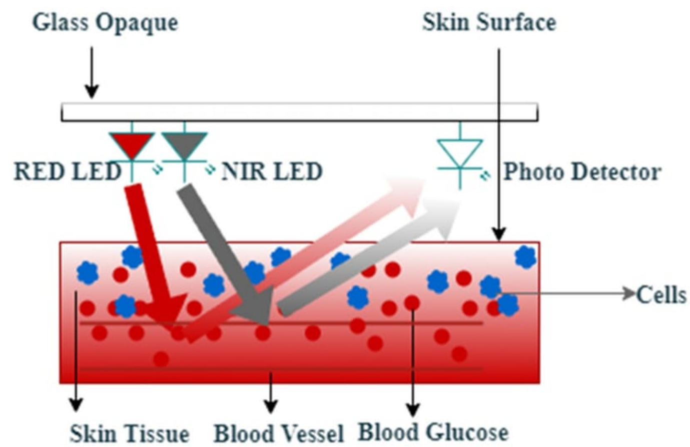

Fig. 1. Working of sensor on blood glucose

As seen above(Fig.1), glucose absorbs and scatters the NIR light that falls on it. This can be used to our advantage in the sensor. When we shine NIR light through the skin, we can estimate the amount of glucose by the intensity of NIR light that is reflected back. More glucose equals less intensity of NIR light reflected back; because the glucose molecules have absorbed most of it and vice versa. The absorption is due to glucose molecules vibrating (NIR's light energy is transferred to kinetic energy) (Fig.2).

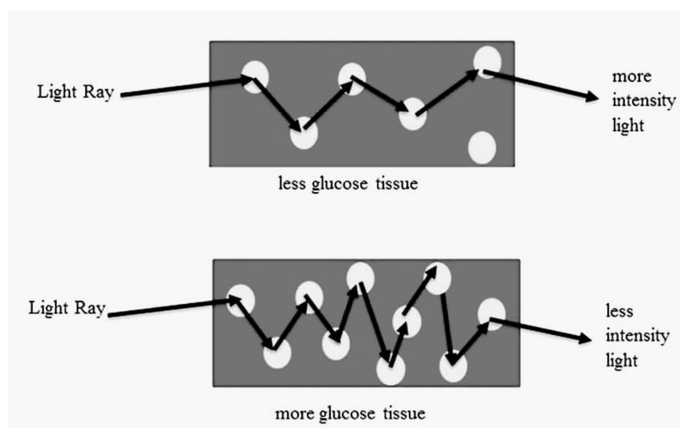

Fig. 2. Interaction of blood glucose and NIR light

Near infrared is a broad spectrum. It goes from 800 nm to 2500 nm. We may wonder at which exact wavelength is glucose absorption maximum. Research has been done in this field. While glucose absorption peaks around 950nm, while there are secondary harmonic oscillations (vibrations) at 850nm range too (Fig.3). This forms the basis as to why we can use infrared rays for the estimation of glucose levels.

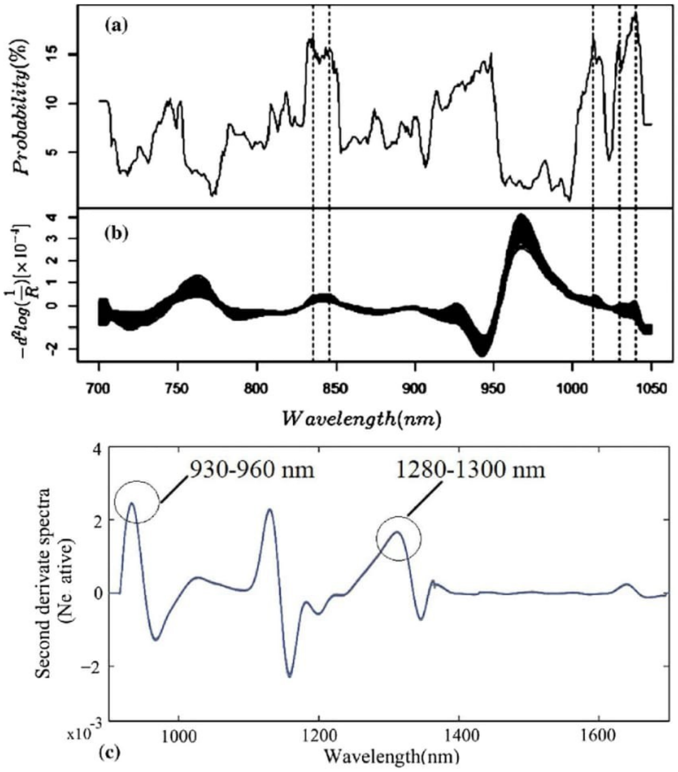

Fig. 3. Probabilty of absorption of glucose at specific wavelengths

We need an NIR emitter to shine light through the skin, and we need a detector (photodiode) to capture the intensity of the light that is reflected back after getting absorbed by the glucose molecules.

The AS7263 NIR sensor would be the perfect fit for capturing blood glucose levels non-invasively using Near Infrared Light. It offers six channels of different wavelengths. All these channels have a ±20 nm bandwidth. These six channels indicate that the AS7263 sensor can shine light of these specific wavelengths and capture the intensity of the light that is received back.

- 610 nm : Violet
- 680 nm : Blue
- 730 nm : Green
- 760 nm : Yellow
- 810 nm : Orange
- 860 nm : Red

We will mostly be interested in the 'Red' channel because it has the 860 ±20 nm wavelength capability and can capture the glucose molecule's secondary harmonic oscillations (vibrations) at 850 nm range as seen above. The sensor can be operated with the GetCalibrated() function which will quickly flash the emitter of the sensor for a fraction of the sensor to get the values and print them to our output.

B. Neural Network for mapping the glucose values

A neural network model consists of 3 components: Input layer, hidden layers and output layers. The input layers would consist of input nodes that determine what we want as an output. In our NN architecture, the input layer would be the AS7263 sensor's analog output values and then the output layer would be the glucose level that is present in our blood. This sensor would be placed on the skin. As we are making a watch-type of sensor module, it would be placed under the watch, pressed against the wrist. Let us see what parameters would actually affect the output of this sensor. For capturing the glucose value in the blood, we can operate the 'Red' channel.

1) Dataset: To generate the dataset, an experimental setup has been created to try and map the analog output of AS7263's 860nm channel to the corresponding blood glucose level. While glucose absorption peaks around 950 nm, there are secondary harmonic oscillations at 850 nm range too. We will be using the 'W-860 nm' channel of the AS7263 NIR sensor because it has the 860 ±20 nm wavelength capability and can capture the intensity differences caused to due glucose molecule's secondary harmonic vibrations at the 850nm range.

a) Experiment perfomed to generate the dataset : Apparatus required:

- Arduino UNO
- AS7263 NIR Spectroscopy sensor
- Invasive Glucometer
- Lancelet pricks
- Testing strip

Hardware Assembly: The circuit diagram for the experimental setup to use the AS7263 to take the intensity readings.

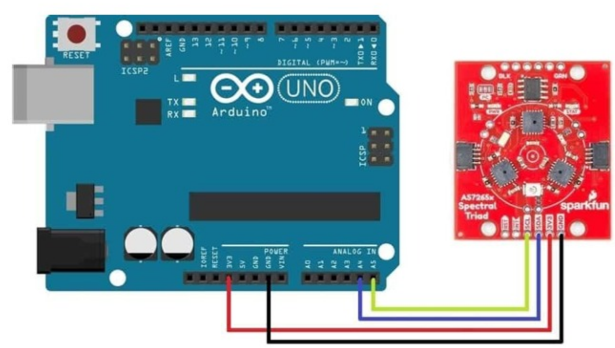

Fig. 4. Circuit diagram for the experimental setup

The pin connections for the above circuit are given in Table 1.

AS7263 Pin	Arduino Uno Pin	Description
VCC	3.3V	Power Supply
GND	GND	Ground
SCL	A5	I2C Clock Line
SDA	A4	I2C Data Line

Table 1. AS7263 NIR sensor pin connections

The real life experimental setup of the hardware assembly for taking intensity readings from AS7263.

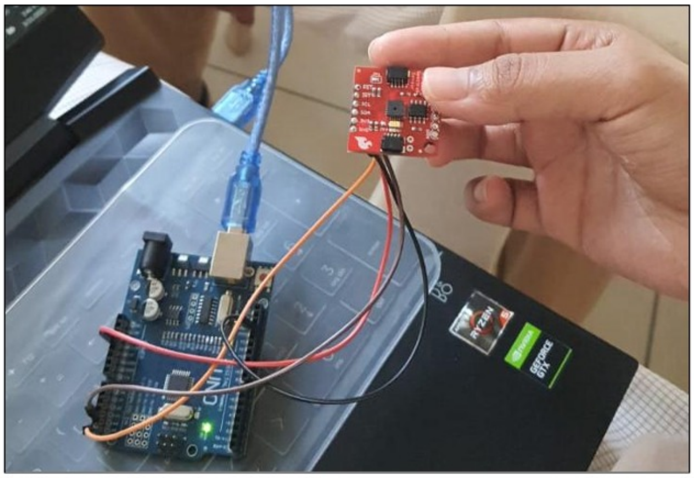

Fig. 5. Hardware Assembly:

Before we got started, we downloaded and installed SparkFun's AS726X Arduino library in the Arduino IDE. The C code was written in Arduino IDE, and the takeMeasurementsWithBulb() and getCalibratedW() functions from the AS726X Arduino library were used to access the W-860 nm NIR channel and get its readings. The takeMeasurementsWithBulb() function illuminates the onboard bulb, calls takeMeasurements() then turns off the onboard bulb. The measuring surface must be properly illuminated, before taking measurements using the sensor, which was not performed in the glucose solution experiment, leading to very small singular digit values.

Procedure:

1) Using the invasive glucometer, we first observed our actual blood glucose value.

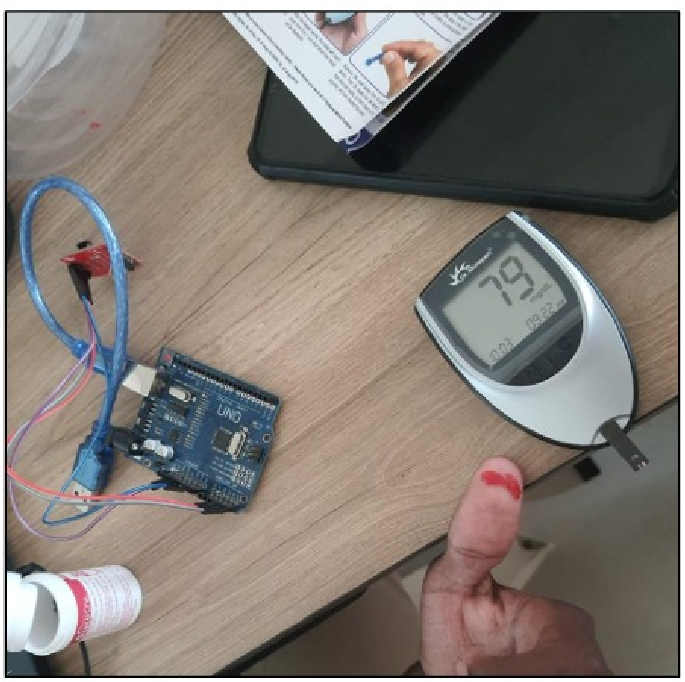

Fig. 6. Measurement of actual blood glucose level

2) The same was performed on multiple people to record a range of blood glucose levels.

Fig. 7. Sampling

3) Once the actual blood glucose level was measured from one person, we immediately recorded the readings from AS7263's W-860 nm channel while the sensor's photodiode array was placed directly on their thumb.

4) The outputs were recorded from the serial monitor of Arduino IDE.

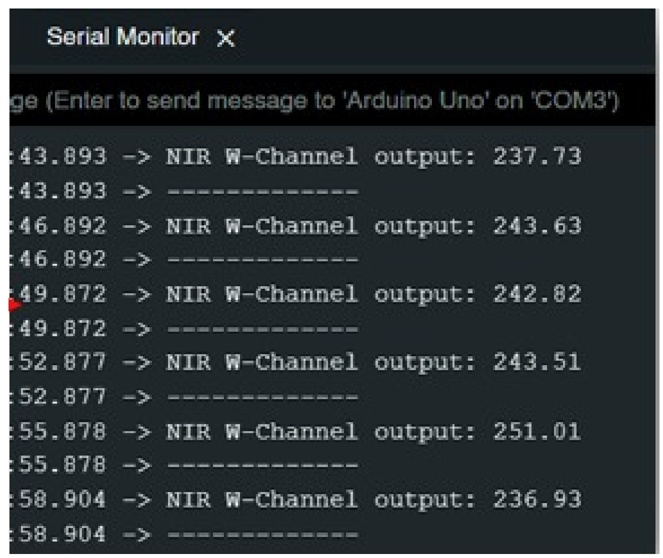

Fig. 8. Serial monitor output

5) Observation was done for 5 blood glucose level corresponding to their respective W-860 nm NIR channel analog output value.

6) The results were then tabulated in Table 2.

7) Once the resultant points were scatter plotted, we fit a curve through it in order to generate noisy points for our synthetic dataset (Fig. 9)

S. No.	Blood Glucose Level	W-860 nm NIR channel analog output
1	79 mg/dL	243.51
2	91 mg/dL	294.38
3	85 mg/dL	274.43
4	116 mg/dL	379.83
5	95 mg/dL	299.71

Table 2. Serial monitor output

After fitting a quadratic curve through the points:

$$y = 0.00009x^2 + 0.2069x + 22.847$$

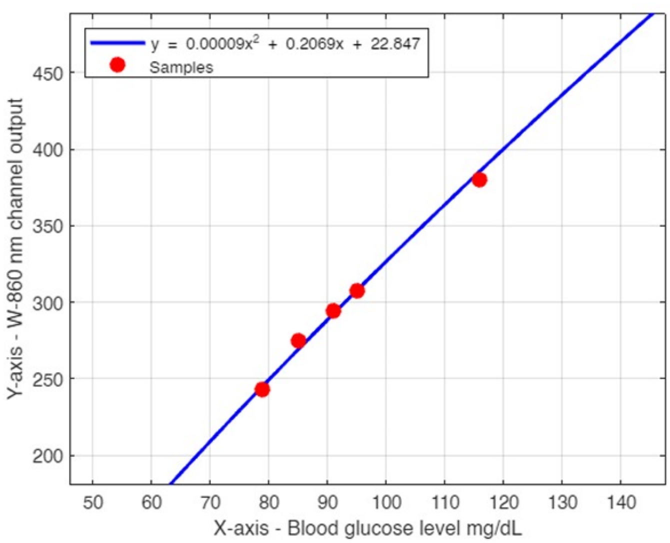

Fig. 9. Plot of the results

Conclusion:
We have successfully performed the mapping of known blood glucose levels to output intensity of NIR sensor at W-860 nm channel. Having mapped the two values, we have plotted the values. From the fitted curve, the datapoints are generated and used for neural network training.

2) Architecture of Neural Network
: The neural network is coded using MATLAB.

Layers of neural network:

```
featureInputLayer(1)
fullyConnectedLayer(64)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(128)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(128)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(256)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(256)
batchNormalizationLayer
reluLayer
dropoutLayer(0.4)
fullyConnectedLayer(256)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(64)
batchNormalizationLayer
reluLayer
fullyConnectedLayer(1)
```

For this project, we have used feed forward neural network to map intensities to the glucose values. Here, the neural network model has been trained using the dataset generated by the above mentioned experiment. The intensity values received via RF transmitter are given as an input to the trained and tested neural network model for prediction of the corresponding glucose values. The dataset of 3000 points used for training and testing of neural network model. The dataset has been split in a ratio of 1 :4 (Test: Train). The loss function used is Huber loss function, which is defined as,

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{1}$$

Where,
$y =$ actual value
$f(x) =$ predicted value
$\delta =$ threshold parameter

The normalization function used here is Batch Normalization and has been incorporated in the neural network layer. The Batch Normalization function used here is-

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mu_B^{(k)}}{\sqrt{(\sigma_B^{(k)})^2 + \epsilon}}, \quad y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{2}$$

where,
x_i: The input activation for the i-th example in a batch μ_B: The mean of the batch (i.e., average of all x_i in the batch)
σ_B^2: The variance of the batch
ϵ: A small constant added to the denominator for numerical stability
$\hat{x}_i$: The normalized value of x_i
γ: A learnable parameter for scaling the normalized value
y: Normalised value of $\hat{x}_i$
β: A learnable parameter for shifting the normalized value

The optimizer used in the neural networks is Adam. It is a combination of both RMSProp and Stochastic gradient descent, to minimize the loss while training the neural network.

C. Prototype Design

Components used:

- Arduino Nano
- Silicon Tube Peristaltic Pump
- 433 MHz RF Transmitter and RF Receiver
- SSD1306 OLED Display Module
- AS7263 NIR Spectroscopy Sensor
- ESP32-WROOM-32
- 5V Relay Module
- 3.3V/5V MB102 Breadboard Power Supply Module

Circuit Diagram and Pin connections:

The schematic circuit diagrams for Insulin pump and CGM watch module are shown in Fig.10 and Fig.11 respectively along with their pin connections.

This CGM watch is place on the wrist and the sensor is placed on the fingertip to get the non- invasive continuous glucose monitoring.

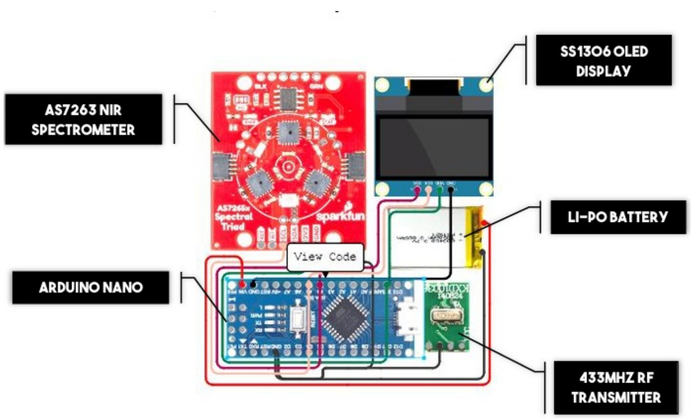

Fig. 10. CGM watch module connections

This insulin pump is to be attached to the abdominal region for the insulin injection.

The readings of AS7263 are to be transmitted to the insulin pump for the injection of insulin. To do so, the pin connections are as follows.

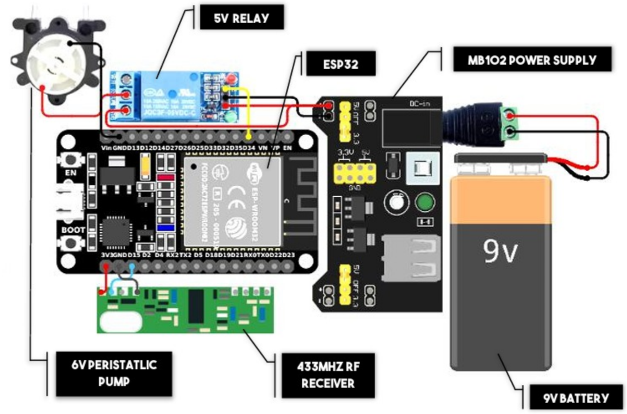

Fig. 11. Insulin pump connections

Interfacing Nano with 433MHz RF Transmitter:
The required pin connects to be made for interfacing Arduino Nano with 433MHz RF Transmitter are listed in Table 3

Pin connections and descriptions:

Transmitter Pin	Arduino Nano Pin	Description
VCC	3.3V or 5V	Power Supply
GND	GND	Ground
DATA	D12	Data Transmission

Table 3. 433 MHz RF Transmitter pin connections

Interfacing ESP32 with 433MHz RF Receiver:
The required pin connects to be made for interfacing ESP32 with 433MHz RF Receiver are listed below in Table 4.

Pin connections and descriptions:

Receiver Pin	ESP32 Pin	Description
VCC	3.3V or 5V	Power Supply
GND	GND	Ground
DATA	D12	Data Transmission

Table 4. 433 MHz RF Receiver pin connections

Exporting neural network as a TensorFlow model:
To export the neural network model as TensorFlow model from MATLAB, a direct function, available in Deep Learning toolbox of MATLAB, exportNetworktoTensorFlow() is used.

Converting from TF to TFLite model:
In order to be able to fit into an ESP32, the model must be quantized and converted into a TFLite (TensorFlowLite) model.

TFLite to C header file (.h file):
To be able to use the TFLite model in the ESP32, we must convert it into a ".h" file. Then, it can be included in our code in Arduino IDE and be uploaded the board.
Uploading the C header to ESP32:

The libraries used for uploading the C header file to ESP32 are-

- EloquentTinyML: This library is to simplify the deployment of Tensorflow Lite for Microcontroller models. It provides an interface to load a model and run inferences.
- TFLM ESP32: Dependency library for EloquentTinyML

This completes the uploading and integration of neural network model to the ESP 32 for received intensity values to glucose mapping. Based on the above readings, the pump will be activated using Treat-to-Range algorithm. Treat-to-Range (TTR) algorithm provides the values of insulin to be injected based upon the predicted glucose values. Table 5 shows the TTR algorithm table. In extreme cases, such as hyperglycemia and hypoglycemia, buzzer is activated to alert the patient of the anomaly as per the TTR algorithm.

S. no.	BGL Readings "mg/dl"	LCD Message	Action Taken
1	50	Range 1	Buzzer activation
2	90	Range 2	No Action
3	120	Range 3	No Action
4	160	Range 4	Insulin Injection (2 ml)
5	220	Range 5	Insulin Injection (4 ml)
6	260	Range 6	Insulin Injection (6 ml)
7	320	Range 7	Insulin Injection (8 ml)
8	360	Range 8	Insulin Injection (10 ml)
9	420	Range 9	Insulin Injection (12 ml), Buzzer activation

Table 5. Treat-to-Range control algorithm

Working of the infusion pump:
For infusion of insulin, peristaltic pump has been used in this project. The flow rate of the pump is 10.5ml/min. The volume of insulin injected into bloodstream is determined by 'time factor' for which reservoir is kept open. For known volume of insulin, height is replaced by time, hence providing the time for which the pump should be active. The formula used is given by equation below.

$$\text{VOL INSULIN} = A = \pi r^2 (t)$$

where,
r=radius of the pump tube
t =the time for which reservoir is kept open

Schematic Design of the prototype:
Fig.12 shows the proposed design of the hardware model. It is a representation showing the real life dynamics of when the device is worn and use. The aim is to make the device compatible and portable for the patient' s convenience. The CGM watch module is placed at the wrist and the insulin infusion pump is placed around the abdomen. Both the modules are connected to each other using RF module for the efficient working of the proposed prototype.

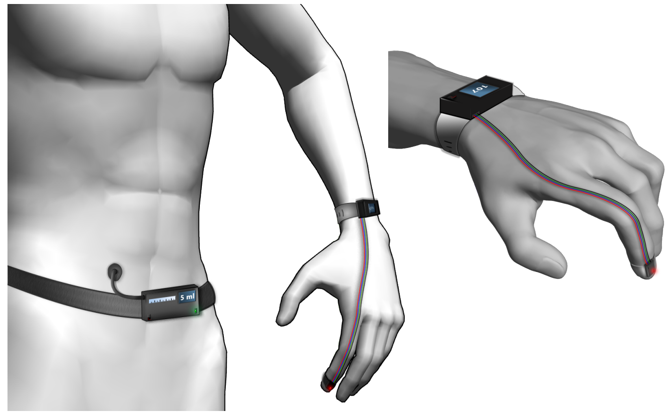

Fig. 12. Design of the wearable model

IV. Results and Comparisons

A. Neural Network

The learning curve for the neural network based glucose model is given as follows in Fig. 13. As seen on the learning curve, the loss over the epochs have been exponentially decreased.

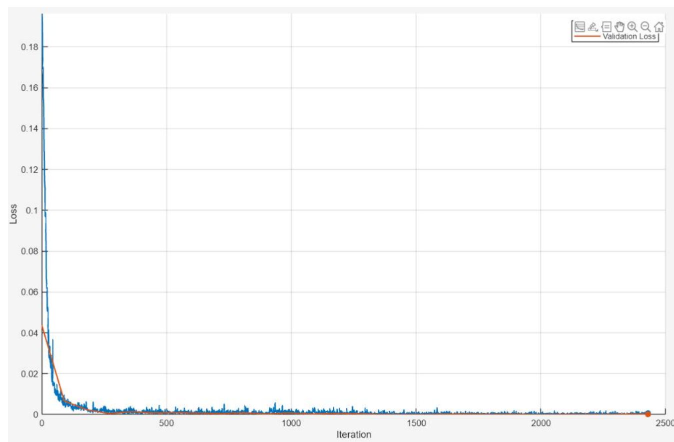

Fig. 13. Neural network learning curve

Upon testing using test data the neural network has given an excellent accuracy of 99.59%. The accuracy of the predictive model is calculated using the below formula.

```
% accuracy calculation
perc_error = abs((y_true - y_pred) ./ y_true) * 100;
accuracy_perc = 100 - mean(perc_error);
```

Fig. 14. Calculation of accuracy of the model

The graph of the actual v/s the predicted glucose values is also plotted based upon the test data.

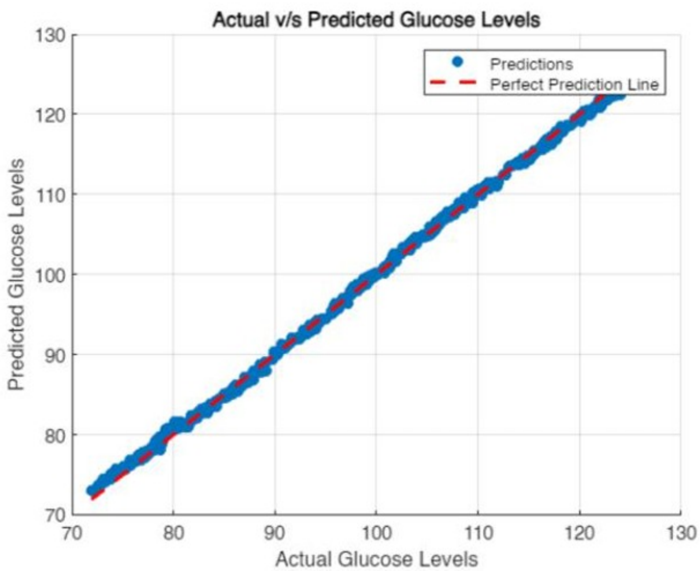

Fig. 15. Neural network learning curve

B. Prototype

Working of CGM module watch:

The model proposed for CGM module watch is shown in Fig.16. Readings can be taken placing the sensor at the fingertip. The RF transmitter then transmits the value of the sensor to the infusion set.

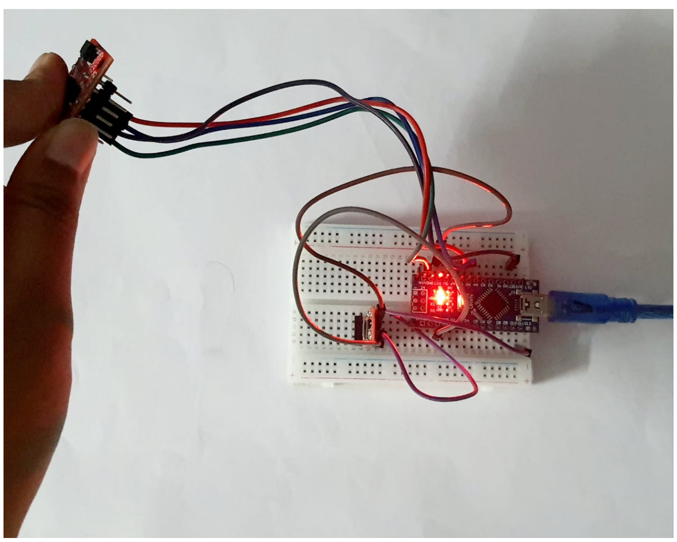

Fig. 16. Prototype CGM module

A brief comparison between the CGM module of existing models in the market and our prototype has been listed in the Table 6.

Model	Bolus and Basal Dosage	Motor/Pump	Activity
Level-Up Insulin	Auto-determined	Peristaltic Pump	Range of targets
Tandem Mobi Control IQ	Bolus – 0.05 Basal – 0.01	Peristaltic Pump	Targets 140 to 160 mg/dl of BGL.
Tandem T-Slim Control IQ	Bolus – 0.05 Basal – 0.01	Peristaltic Pump	Targets 140 to 160 mg/dl of BGL.
Beta Bionics Ilet	Auto-determined	Piston Driven (Not publicly detailed.)	No specific target.
OmniPod 5	Bolus – 0.05 Basal – 0.05	Piston Driven	Targets around 150 mg/dl of BGL.
Animas Vibe	Bolus – 0.05 Basal – 0.025	Diaphragm Pump	150 mg/dl of BGL.

Table 6. Comparison table for CGM module

Working of Insulin infusion pump:

The model proposed for Insulin infusion pump is shown in Fig. 17. It shows the OLED screen with the values of NIR sensor received by the RF receiver and the mapped blood glucose value calculated by the neural network model. Based upon the TTR algorithm the pump will be activated and insulin will be injected. In extreme cases, buzzer system will be activated.

Fig. 17. Prototype insulin infusion pump

A brief comparison between the insulin infusion set of existing models and our prototype has been proposed in the Table 7.

Model	Alert System	Sensor Name	Sensor Type
Level-Up Insulin	Yes- high and low.	AS7263 NIR Spectroscopy Sensor	Non-Invasive Near Infrared
Tandem Mobi Control IQ	Yes- high, low and rapid change alert.	Dexcom G7	Invasive microneedle.
Tandem T-Slim Control IQ	Yes- high, low and battery alerts.	Freestyle Libre 2 Plus	Invasive microneedle
OmniPod 5	Yes- high and low.	Dexcom G6	Invasive microneedle
Bluetooth Low Energy (BLE) (Prototype)	None built-in.	BLE Received Signal Strength Indicator (RSSI)	Non-invasive Electromagnetic
Split Ring Resonator (Prototype)	None built-in.	1.4 GHz Microwave Split Ring Resonator	Non-invasive Electromagnetic

Table 7. Comparison table for Insulin infusion set

C. Experimental results of the functional prototype

For the verification of smooth working of the prototype, five readings have been taken using the prototype which are tabulated in the Table 8. After measuring five readings from the device, it was observed that the system delivers good accuracy of glucose levels if held in place steadily. However, some limitations, for instance, some inaccurate readings take place when the body part being measured is moving around. Cause of this would be due to sensor disturbances or inadequate signals during movement, resulting in noisy data. From this observation , it can be concluded that though the device is efficient under stationary conditions, improvements in motion compensation techniques, such as noise filtering algorithms or motion-resistant sensors are needed.

W-860nm (NIR output)	**BGL** (Glucose level)	**INSULIN** (Output)	**BUZZER** (Activation)
314	96.79 mg/dl	*Nil*	*No*
294	91.66 mg/dl	*Nil*	*No*
161	58.84 mg/dl	*Nil*	**Yes**
439	131.21 mg/dl	**2 ml**	*No*
497	153.75 mg/dl	**4 ml**	*No*

Table 8. Experimental results of the prototype

V. Conclusion

The project aimed at proposing a prototype which provides non- invasive continuous glucose monitoring and automated insulin injection based upon the values received by CGM module. The neural network proposed for this model to map the sensor values of intensities to glucose values, provides an accuracy of 99.53%. Further the readings from the prototype were taken to confirm the working of the model. We can say that this project offers an encouraging potential for real-time diabetes management. The accuracy achieved in stable conditions also holds promise, but challenges like motion-related inaccuracies do need to be sorted out. This device minimizes human intervention by automatically injecting insulin upon observing glucose spikes and it offers a reliable approach toward managing diabetes.

VI. Acknowledgment

We would like to extend our sincere gratitude towards our respected Dean Dr. K P Soman, who gave us this opportunity to do this hands-on project. We would like to thank our esteemed Professors, Dr. Snigdhatanu Acharya and Dr. Amrutha V, who motivated and mentored us throughout the semester for completion of this project. They played pivotal role in completion and execution of our project model, as they guided us for required models and frameworks. Lastly, we would like to thank all the people who contributed in the execution of this project and made it a success.

References

[1] Sun H, Saeedi P, Karuranga S, Pinkepank M, Pavkov. IDF Diabetes Atlas: Global, regional and country-level diabetes prevalence estimates for 2021 and projections for 2045. Diabetes Res Clin Pract. 2022 Jan;183:109119. doi: 10.1016/j.diabres.2021.109119. Epub 2021 Dec 6. Erratum in: Diabetes Res Clin Pract. 2023 Oct;204:110945. doi: 10.1016/j.diabres.2023.110945. PMID: 34879977; PMCID: PMC11057359.

[2] R.S. H. Istepanian, S. Hu, N. Y. Philip and A. Sungoor, "The potential of Internet of m-health Things "m-IoT" for non-invasive glucose level sensing," 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 2011, pp. 5264 5266

[3] Bequette, B. W. (2012). Challenges and recent progress in the development of a closed-loop artificial pancreas. Annual Reviews in Control, 36(2), 255–266.

[4] Andrea Facchinetti, Giovanni Sparacino, Stefania Guerra, Yoeri M. Luijf, J. Hans DeVries, Julia K. Mader, Martin Ellmerer, Carsten Benesch, Lutz Heinemann, Daniela Bruttomesso, Angelo Avogaro, Claudio Cobelli ; Real-Time Improvement of Continuous Glucose Monitoring Accuracy: The smart sensor concept. Diabetes Care 1 April 2013; 36 (4): 793–800.

[5] .Berget C, Messer LH, Forlenza GP. A Clinical Overview of Insulin Pump Therapy for the Management of Diabetes: Past, Present, and Future of Intensive Therapy. Diabetes Spectr. 2019;32(3):194-204.

[6] Templer S (2022) Closed-Loop Insulin Delivery Systems: Past, Present, and Future Directions. Front. Endocrinol. 13:919942

[7] Kompala T, Neinstein AB. Smart Insulin Pens: Advancing Digital Transformation and a Connected Diabetes Care Ecosystem. Journal of Diabetes Science and Technology. 2022;16(3):596-604.

[8] Pauley, M.E., Tommerdahl, K.L., Snell-Bergeon, J.K. et al. Continuous Glucose Monitor, Insulin Pump, and Automated Insulin Delivery Therapies for Type 1 Diabetes: An Update on Potential for Cardiovascular Benefits. Curr Cardiol Rep 24, 2043–2056 (2022)

[9] Mao, P., Li, H., Yu, Z. (2023). A Review of Skin-Wearable Sensors for Non-Invasive Health Monitoring Applications. Sensors, 23(7), 3673.

[10] Yoo, J. H., Kim, J. H. (2023). Advances in Continuous Glucose Monitoring and Integrated Devices for Management of Diabetes with Insulin-Based Therapy: Improvement in Glycemic Control. Diabetes and Metabolism Journal, 47(1), 27–41.

[11] Nihaal Reddy, BS, Neha Verma, MD, and Kathleen Dungan, MD, MPH, Monitoring Technologies- Continuous Glucose Monitoring, Mobile Technology, Biomarkers of Glycemic Control

Appendix A
Images And Schematics

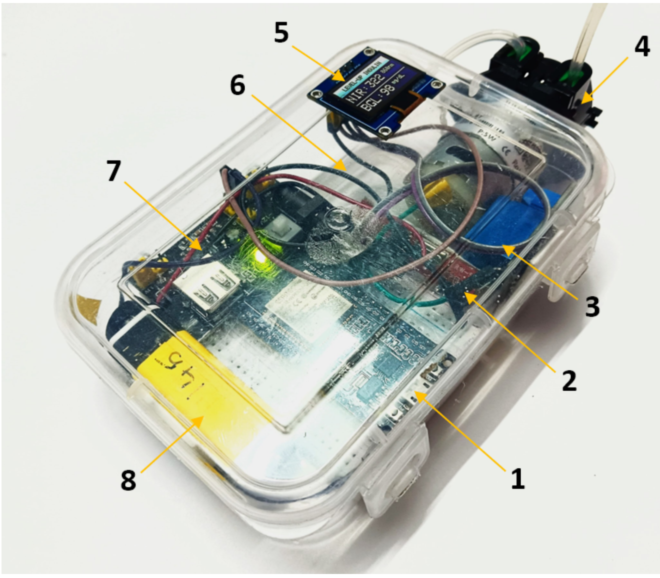

Fig. 18. The Insulin pump model and its peripherals (different parts of hardware used are labeled as follows: 1 shows the ESP 32- WROOM- 32 module, 2 shows the 433 MHz RF Receiver, 3 is the 5V relay switch, 4 is the 6 V peristaltic pump, 5 shows the SSD1306 OLED display, 6 is the space for insulin container and refilling, 7 shows the MB102 power supply module and 8 marks the 9V rechargeable battery).

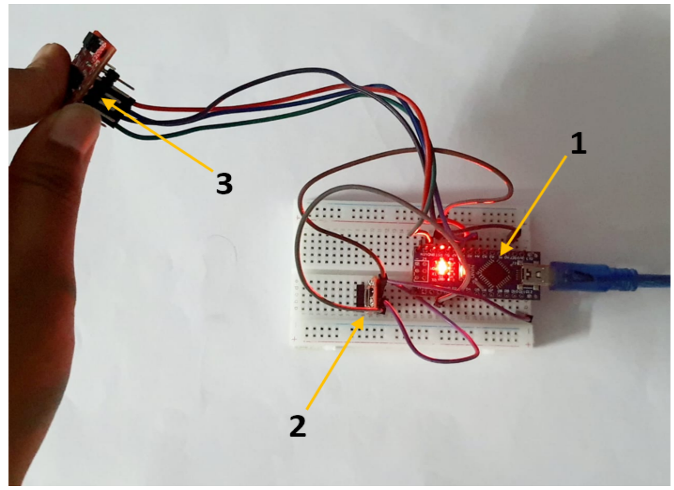

Fig. 19. The CGM sensor model and its peripherals (different parts of hardware used are labeled as follows: 1 shows the Arduino Nano, 2 marks the 433 MHz RF Transmitter and 3 shows the AS7263 NIR Spectroscopy sensor.

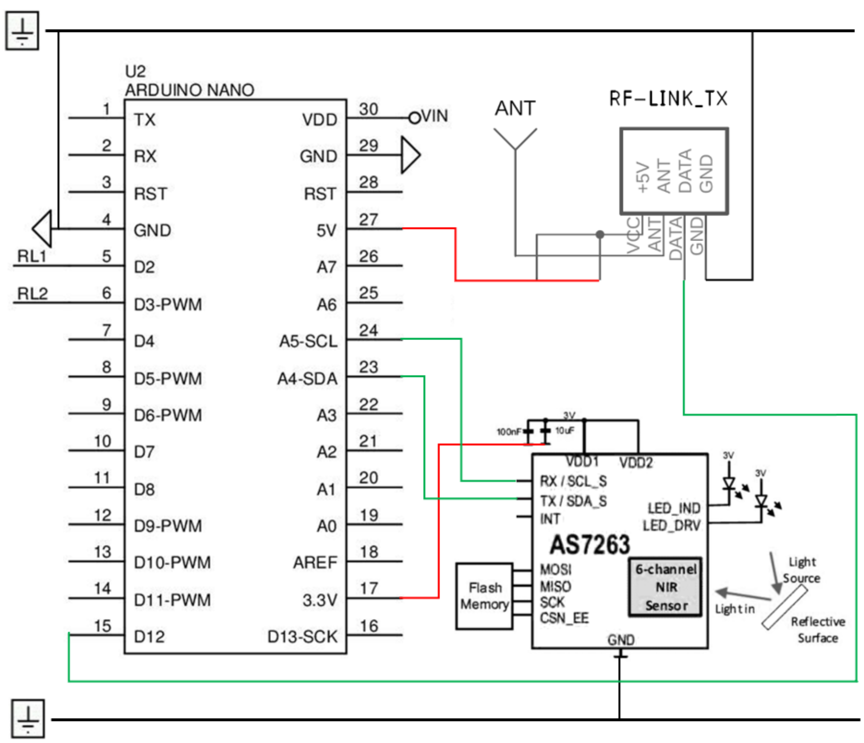

Fig. 20. Schematic diagram of the non-invasive CGM module.

Appendix B
Source Codes

1) `levelup_insulin_gen.mlx` - Dataset Generation

MATLAB

```
% Generation of dataset
x= randi([200,400],3000,1);
y_0=(.00009.*x.*x) + (0.2069.*x) + 22.847;
y=y_0+normrnd(4,0.3,3000,1);
dataset=[x y];


% Normalize the input features and target values
[input_x_norm, input_ps] = mapminmax(x', 0, 1);
input_x_norm = input_x_norm';
[y_0_norm, output_ps] = mapminmax(y', 0, 1);
y_0_norm = y_0_norm';


% Split data into training (80%) and testing (20%) sets
rng(42);     % to make the distribution between test and train fixed
cv = cvpartition(size(input_x_norm, 1), 'HoldOut', 0.2);
XTrain = input_x_norm(cv.training, :);
YTrain =dlarray( y_0_norm(cv.training, :));
XTest = input_x_norm(cv.test, :);
YTest = (y_0_norm(cv.test, :));


% Convert to dlarray format
XTrain = dlarray(XTrain);
XTest = dlarray(XTest);
```

2) `levelup_insulin_gen.mlx` - Neural Network Architecture and Training

MATLAB

```
% Building a neural network
net = dlnetwork;
layers = [
    featureInputLayer(1)
    fullyConnectedLayer(64)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(128)
    batchNormalizationLayer
```

```
    reluLayer
    fullyConnectedLayer(128)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(256)
    batchNormalizationLayer
    reluLayer
fullyConnectedLayer(256)
    batchNormalizationLayer
    reluLayer
    dropoutLayer(0.4)
fullyConnectedLayer(256)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(64)
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(1)
    ];
net = addLayers(net, layers);

plot(net);
% analyzeNetwork(net);

  % defining the loss function - HUBER LOSS
function loss = dlhuber(y_pred, y_true, delta)
    e = y_pred - y_true;
    a = abs(e);
    mask = a <= delta;

    % Quadratic part (resembling L2)
    square_loss = 0.5 * e.^2;

    % Linear part (resembling L1)
    linear_loss = delta * (a - 0.5 * delta);

    % loss function equation
    loss = sum(mask .* square_loss + (~mask) .* linear_loss) /
numel(e);
end


% training the model
options = trainingOptions("adam", ...
    "Plots", "training-progress", ...
    "MaxEpochs", 200, ...
```

```
    "InitialLearnRate", 0.001, ...
    "LearnRateSchedule", "piecewise", ...
    "LearnRateDropFactor", 0.1, ...
    "LearnRateDropPeriod", 50, ...
    "L2Regularization", 0.0005, ...
    "ValidationData", {XTest, YTest}, ...
    "ValidationFrequency", 90, ...
    "ValidationPatience", 10, ...
    "Shuffle", "every-epoch", ...
    "MiniBatchSize", 96);

huberLoss = @(y_pred, y_true) dlhuber(y_pred, y_true, 0.1);   % loss
function

[net, info] = trainnet(XTrain, YTrain, net, huberLoss, options);

% predicting the values on the test data
y_pred_norm = predict(net, XTest);
% inverse to get the actual glucose values from the normalized data
y_pred = mapminmax('reverse', y_pred_norm.extractdata', output_ps)';
y_true = mapminmax('reverse', YTest', output_ps)';


% accuracy calculation
perc_error = abs((y_true - y_pred) ./ y_true) * 100;
accuracy_perc = 100 - mean(perc_error);


fprintf('Accuracy: %.2f%%\n', accuracy_perc);


% Plot actual vs predicted values
figure;
scatter(y_true, y_pred, 'filled');
hold on;
min_val = min([y_true; y_pred]);
max_val = max([y_true; y_pred]);
plot([min_val, max_val], [min_val, max_val], 'r--', 'LineWidth', 2);
hold off;
xlabel('Actual Glucose Levels');
ylabel('Predicted Glucose Levels');
title('Actual v/s Predicted Glucose Levels');
legend('Predictions','Perfect Prediction Line');
grid on;
```

3) `tf_to_tflite.py` - Converting TensorFlow model to TFLite model

Python – VS Code

```
import tensorflow as tf

# Load SavedModel
saved_model_dir = "nir_to_glucose"
converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir)

# Optimize for size (Quantization)
converter.optimizations = [tf.lite.Optimize.DEFAULT]

# Convert to TFLite
tflite_model = converter.convert()

# Save TFLite Model
with open("nir_to_glucose.tflite", "wb") as f:
    f.write(tflite_model)

print("TFLite model conversion successful! File saved as 
'nir_to_glucose.tflite'")
```

4) `insulin_pump.ino` - Mechanism for the Insulin Pump (controlled by ESP32)

Arduino IDE

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <RH_ASK.h>
#include <SPI.h>
#include "nir_to_glucose.h"
#include <tflm_esp32.h>
#include <eloquent_tinyml.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C

#define RELAY_PIN 27
#define BUZZER_PIN 14
```

```
#define NUM_OF_INPUTS 1
#define NUM_OF_OUTPUTS 1
#define ARENA_SIZE 2 * 1024

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

RH_ASK rf_driver;

Eloquent::TinyML::TfLite<NUM_OF_INPUTS, NUM_OF_OUTPUTS, ARENA_SIZE>

tf(nir_to_glucose);

unsigned long lastCheck = 0;
float lastBGL = 0;
int lastNIR = 0;

void setup() {

  Serial.begin(115200);

  pinMode(RELAY_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);

  tf.begin();

  if (!rf_driver.init()) {
    Serial.println("RF init failed");
    while (true);
  }

  if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println("SSD1306 failed");
    while (true);
  }

  display.clearDisplay();
  display.display();
}

void loop() {

  receiveAndPredict();
```

```
  if (millis() - lastCheck >= 60000) {
    controlLogic(lastBGL);
    lastCheck = millis();
  }
}

void receiveAndPredict() {

  uint8_t buf[2];
  uint8_t buflen = sizeof(buf);

  if (rf_driver.recv(buf, &buflen)) {
    int nirValue = (buf[0] << 8) | buf[1];
    float input[1] = { (float)nirValue };
    float glucose = tf.predict(input);
    lastNIR = nirValue;
    lastBGL = glucose;
    displayData(nirValue, glucose);
    Serial.print("NIR: ");
    Serial.print(nirValue);
    Serial.print("  |  BGL: ");
    Serial.println(glucose);
  }
}

void controlLogic(float glucose) {
  if (glucose < 60 || glucose > 360) {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(10000);
    digitalWrite(BUZZER_PIN, LOW);
    return;
  }

  int doseSeconds = calculate_insulin_dose(glucose);

  if (doseSeconds > 0) {
    digitalWrite(RELAY_PIN, HIGH);
    delay(doseSeconds * 1000);
    digitalWrite(RELAY_PIN, LOW);
  }
}
```

```
int calculate_insulin_dose(float glucose_level) {
  if (glucose_level >= 60 && glucose_level < 70) return 2;
  if (glucose_level >= 100 && glucose_level < 160) return 4;
  if (glucose_level >= 160 && glucose_level < 200) return 6;
  if (glucose_level >= 200 && glucose_level < 260) return 8;
  if (glucose_level >= 260 && glucose_level < 300) return 10;
  if (glucose_level >= 300 && glucose_level < 360) return 12;

  return 0;
}


void displayData(int xVal, float yVal) {

  display.clearDisplay();

  display.fillRect(0, 0, SCREEN_WIDTH, 16, SSD1306_WHITE);
  display.setTextColor(SSD1306_BLACK);
  display.setCursor(17, 4);
  display.setTextSize(1);
  display.print("LEVEL-UP INSULIN");

  display.setTextColor(SSD1306_WHITE);
  display.setTextSize(2);
  display.setCursor(2, 20);
  display.print("NIR:");

  display.setCursor(51, 20);
  display.print(xVal);
  display.setTextSize(1);
  display.print(" 860nm");

  display.drawLine(0, 40, SCREEN_WIDTH, 40, SSD1306_WHITE);

  display.setTextSize(2);
  display.setCursor(2, 45);
  display.print("BGL:");

  display.setCursor(51, 45);
  display.print(yVal, 2);
  display.setTextSize(1);
  display.print(" mg/dL");

  display.display();
}
```

5) `CGM_sensor.ino` - Mechanism for the CGM sensor (controlled by Arduino Nano)

Arduino IDE

```
#include "AS726X.h"
#include <Wire.h>
#include <RH_ASK.h>
#include <SPI.h>

AS726X sensor;
RH_ASK rf_driver(2000, -1, 12);  // TX pin = D12

void setup() {
  Serial.begin(115200);
  Wire.begin();

  if (!sensor.begin()) {
    Serial.println("AS726X sensor not detected");
    while (true);
  }
  sensor.disableIndicator();

  if (!rf_driver.init()) {
    Serial.println("RF init failed");
    while (true);
  }
}

void loop() {
  sensor.takeMeasurementsWithBulb();

  int nir = sensor.getCalibratedW();  // 860nm channel
  Serial.print("NIR W-Channel output: ");
  Serial.println(nir);
  Serial.println("-------------");

  // Send NIR as 2-byte message
  uint8_t msg[2];
  msg[0] = (nir >> 8) & 0xFF;
  msg[1] = nir & 0xFF;

  rf_driver.send(msg, sizeof(msg));
  rf_driver.waitPacketSent();

  delay(10000);} // send every 10 seconds
```