World Journal of Pharmaceutical and Life Sciences
WJPLS
www.wjpls.org
SJIF Impact Factor: 7.409

A REVIEW ON THE ROLE OF PYTHON IN REAL-TIME GLUCOSE MONITORING AND DIABETES MANAGEMENT

Monica Samaya N.[1] and Geetha Balasubramani[2]*

[1]B.Tech. Electronics & Communication Engineering, Sastra University, Thanjavur.
[2]*Co-Founder & CRO, Medcuore Medical Solutions Private Limited, Chennai.

***Corresponding Author: Geetha Balasubramani**

Co-Founder & CRO, Medcuore Medical Solutions Private Limited, Chennai.

Article Received on 08/03/2025 **Article Revised on 28/03/2025** **Article Accepted on 18/04/2025**

ABSTRACT
The use of digital monitoring devices has evolved in recent years to better serve patients, particularly those with diabetes. These devices simplify the lives of diabetic individuals by helping them manage their diets and maintain optimal glucose levels. Digitalization of data plays a crucial role, enabling quicker and clearer communication, along with real-time updates for patients. It is also time-saving, as it reduces the effort required for record management. The primary goals include predicting glucose level fluctuations and notifying patients in advance for timely insulin administration. Among individuals with type 1 diabetes, there is an increasing demand for continuous glucose monitoring systems, which provide alerts for high or low blood glucose levels. This review article explores various Python libraries that assist in generating records and visualizations efficiently. These tools are especially valuable in clinical scenarios where obtaining continuous data and comparative findings can be challenging. Python offers simple yet powerful algorithms and a streamlined execution process for such applications. Additionally, this article compares Python-based methods with other glucose monitoring techniques, including machine learning and neural networks, which often utilize Internet of Things (IoT) technology for tracking blood glucose. Based on our comparison, Python demonstrated superior performance over alternative approaches.

KEYWORDS: Diabetes; Digitalization; CGM; Python; Clinical metrics; Machine learning; Neural network; IoT.

INTRODUCTION
Diabetes mellitus, a chronic disease that is grouped based on the level of glucose in the body, which is obtained through the digestion of carbohydrates. In nature, our body is designed to maintain the level of glucose. Beta cells in the pancreas check for the level of blood sugar level, if increases, beta cells release insulin into your bloodstream, and this insulin helps in the process of moving glucose from the blood to cells. Diabetes occurs when the pancreas doesn’t make enough insulin or any at all, or when the body does not respond to the insulin. In humans, glucose levels should be properly maintained for the normal function of nerve cells and the human brain, which consumes nearly 60% of blood sugar. The normal range of fasting blood sugar levels is from 70 mg/dl to 100 mg/dl whereas when it is between 100 to 125 mg/dl monitoring of glycemia is recommended. The causes of insulin disorder can be by birth or due to an unhealthy diet and low physical activity which are responsible for Type 1 and Type 2 diabetes. Most cases of type 1 diabetes (also known as insulin-dependent, juvenile or childhood onset) develop in children and adolescents. Here, the immune system mistakenly attacks and destroys the insulin-producing beta cells in the pancreas, and so the body is unable to produce insulin on its own which requires daily administration of glucose. People with type 2 diabetes have insulin resistance, where the body's cells do not respond effectively to insulin and the pancreas may lose its ability to produce enough insulin to overcome the insulin resistance. More than 95% of people have type 2 diabetes. And the other may be gestational diabetes which occurs during pregnancy. It is vital to deal with hyperglycaemia. If it's not treated, it can become severe and cause serious health problems like vision loss, kidney disease, nerve damage and other such serious conditions. Hypoglycaemia happens when blood sugar level drop below 70 mg/dl which causes lethargy, shaking, twitching, and weakness in arm and leg muscles. The risk factor can be reduced by predicting the blood glucose values of diabetic patients and warning them when these values are not in a safe range. Predictions of future BG levels depend on the amount of past blood glucose level data. It is also important to provide them

with a meaningful response from the data obtained. Many invasive methods are available today, such as Contour Plus and Freestyle Lite, which provide a more accurate blood glucose reading. However, they all involve pricking the finger to extract a tiny amount of blood, leaving scars on the body. This leads to the necessity of a non-invasive method to check blood sugar levels. This can be overcome by continuous glucose monitoring devices (CGM), a non-invasive method for measuring blood sugar levels that provides real time information by tracking glucose values throughout the day. It should also solve the situation where historic data are not available. The structural properties of CGM signals obtained from large quantities of CGM signals are used to predict future blood glucose levels. As a result, more people are using CGMs, which opens up new avenues for treating diabetes. But the data alone will not help diabetics and physicians to understand the pattern of change in their glucose level. In recent years, there have been many advancements in technologies to manage the data and alert the patients, but they all fail to understand how to respond to it and how to modify the diet according to it. In today’s world, open-source data driven programming languages are much simpler to handle than other software. In this article, Python packages and its tools are used, as it is the third most common programming language used globally, yields accurate results of glucose prediction for a vast number of people, and provides a detailed examination that will guide diabetics to follow the correct diet. Python is capable of doing time consuming operations like sending text alerts, sending reminder emails, and updating and formatting data in Excel of any size. A web application is also developed that integrates the Python package to offer its service through the web browser, which will be effortless to handle. Web development can be framed using two primary frameworks: Django and Flask. The availability of these Python libraries made it easier to access.

Glucose monitoring using python

In our rapidly digitising world, managing the data and accessing it through software technology has become the backbone of modern health care. It helps not only in storing data but also in analyzing the relationship between the recorded data and informing patients beforehand to avoid any risky situations. It also becomes easier for the doctors to understand the real cause of the disease, and it becomes easier to cure the disease. Currently, there are many methods used for monitoring blood glucose levels. Continuous glucose monitoring (CGM) systems have become the most adopted technology, in particular for patients who are in need of insulin administration. It is a user-friendly device that gives real-time values of BG levels without any pain. They also help to improve the safety of hypoglycaemic patients by providing other information, like carbohydrate and insulin levels, so that they can modify their diet plan. CGM provides data (may be in a CSV file) and also gives the relationship between those data by using algorithms and some methods that are proprietary and can’t be accessed. They do not contain any clinical metrics. As we have to deal with the swings in blood glucose level, i.e., glucose variability before and after meals for the hypoglycaemia patients, we are in need of examining the other 25 metrics. To resolve this issue, we can use Python packages, which are the most commonly used open software and give a descriptive analysis (Bent et al., 2021). Python can also be used to develop a web application to store and make decisions for diabetics about following their diet based on the data collected by using a CGM device. The following functions can be done with the help of Python packages:

1. Data collection

First, the data is collected from the user by using Freestyle Libre, a CGM device. This CGM device consists of a sensor that is placed on the body. This sensor is scanned every 8 hours, and it stores the data in the web application developed. The data on the device stores it for 90 days. In today's world, this device has been widely used as it is a non-invasive method and helps to get heaps of data in a single day. These data are stored in a web application for further analysis and graphical representation. As Python is a versatile and feasible language, it is used to develop this web application. Development of the web consists of two sections i.e., to generate reports for data and the other to generate patterns and graphs. The web application here uses the Django framework in Python, which follows object relational mapping technique. This technique maps objects to a database, converting the object into data that can be stored, retrieved, and reconstructed by the user. From this web, data are easily retrieved and can be used to tackle the other metrics if needed. The one is the core application. This application uses panda’s library to generate pdf for the obtained series of values.

2. Preprocessing data

Next, the data needs to be pre-processed. Patients are assisted in maintaining their diets by the processing of data and the display of the linkages between them. The web application examines the data file completely and verifies that the data is formatted correctly. The Model object in Python checks the raw data in the preprocessor module and creates the main dataset file. The main dataset file consists of numerous pre-processed files as a single data file. Here, language is used to represent each column. In order to select a particular column in the syntax, a translator object is used to translate the columns. Translator objects generally convert data from one format to another. If the format is not correct, it returns to the web application and reports to the user.

3. Features and Labels

After preprocessing and validating the data, it is inserted into the main dataset file. Here, the user can select their needed features and labels to get their pattern. Libraries like Plotly, Pandas, Seaborn, Matplotlib, and Numpy can be imported from Python packages to give statistical results in an understandable way by creating a pattern for the data obtained from CGM. Following are some of the important statistical tools to be examined, which can be found in Python packages. (Payne et al., 2021)

Numerical measures of summarising CGM data	
Mean BG	Average blood glucose level over two to three months
% time spent within target 70-180 mg/dL ,below 70 mg/dL , above 180 mg/dL	This can be calculated when the time interval is not within these specified ranges of values.
Interquartile range	It is a variability measurement for non-symmetric distributions
Standard deviation for the rate of change of blood glucose level	Is a measure to check for fluctuations in BG levels

Graph visualisation using CGM data	
Glucose Trace	A traditional plot that corresponds to the time spent by BG in the suggested region of values
Poincare plot	They are widespread data points, checking for glucose value fluctuations
Variability and Risk assessment	Highlights essential variances in glucose level

Mean blood glucose value: It is an estimated average of your blood sugar level over a period of time. Here, the describe() function in the Pandas library is used for calculating the mean glucose level. The average mean glucose value should be around 90 to 110 mg/dL for non-diabetic and 70 to 180 mg/dL for diabetic patients.
Syntax: glucose. describe(). transpose()

% of time within each range: here grouper(), fillna(), unstack() functions in Pandas are used to predict the percent of time required for each range of blood glucose level. Generally, a higher value corresponds to lower risk of microvascular complications.

Interquartile range: This range can be detected by box plots and whisker plots, which are created by using the Seaborn library in Python and give a graphical representation of the interquartile range. The advantage of the interquartile range is that it can be used as a measure of variability if the extreme values are not recorded exactly.
Syntax for box plot: box plot(x,data=)

BG rate of change: higher the value of standard deviation, greater the value of blood glucose level. To visualise this, we use histogram by using the Seaborn library in Python. Seaborn is a library for making statistical graphics in Python.

Glucose trace: pyplot can be created by using Python to visualise the fluctuations in blood glucose value**.**

Point care: Poincare plots are used to represent the level of scatteredness of the system. The greater the linearity, the greater the stability of a system. BG(t-1) and BG(t) are represented on the x and y axes, respectively. Samples were collected and compared for different days in a month. The rate of change of BG is calculated using pandas. shift(). The days which have higher scatteredness have much glycaemic variability as compared to less scattered days.

Variability and Grid analysis: Here, the plot is designed in such a way that the x-axis contains the minimum glucose level in 24 hours and the y-axis contains the maximum glucose level in 24 hours, which is obtained by using the min and max functions in the Numpy library. Numpy libraries are used for working with arrays.

4. In solving errors

In measuring the data, when the glucose value ranges >100 mg/dL there may be an error of 15%. This is resolved by smoothing the curves by using Cubic spline technique which can be achieved by the use of the interpolate package present in SciPy library. In order to aid in viewing and interpretation additionally a function is built that permits LOWESS smoothing over the CGM data. The smooth line gives the more accurate representation of data, despite the noisy variables that underlie the original set. This can be implemented by using cgm quantify which has both Python package and R package which has 25 functions and 25 glucose metrics and glycaemic variability (Bent et al., 2024).

5. Designing classifier tree

Machine learning techniques like decision tree classifier algorithms can be used to make classifications based on a series of values using supervised techniques. This decision tree can be implemented by using the package scikit-learn in Python, which consists of a list of libraries that are used for data analysis. The code takes the data file value to register labels like hyperglycaemia, hypoglycaemia, or in range. Next, it uses two classifiers: the prior data and the post data to classify current data in a label. This classification is used to find the current glucose label by using the previous blood glucose values. (José et al., 2020)

6. Model creation

There are numerous parameters in the classifier that are required to prevent overfitting and the creation of complicated patterns. (José et al., 2020). The important parameter is determining the quality of the pattern as it stops splitting the node. The next parameter denotes the depth of the tree. The weight of each feature represents the other parameter. This model creates a Pattern object that has a Rule object.

7. Report generation

At last report is generated in either HTML or pdf format by using the Jinja2 library, which gets its values by using a dictionary that has variables in it. This template is similar to a web page.

Python is also used to implement machine learning algorithms. In python

- *Pandas* are used for exploratory data analysis, and they are great manipulation tools. In the framework, data is shared by using this library.
- *Matplotlib* is used for interactive data visualisation. It is used for creating 2D plots of arrays.
- High level interface is provided by *Seaborn* library for creating plots. It provides high level, customizable plots for statistical data analysis.
- *NumPy* is the base for numerical computations as it facilitates efficient numerical operations on large quantities of data. (Dubosson et al.,2023)

Glucose monitoring using other methods

Neural network

Neural networks can be used in glucose monitoring as they are able to uncover hidden links, patterns, and predictions; draw generalizations and inferences; and learn and model complicated, nonlinear interactions between inputs and outputs. Artificial neural networks are used as a machine learning technique and are the basis of deep learning algorithms. They are defined as a collection of algorithms designed to identify patterns. They identify the patterns by analysing the incoming data, which includes text, music, and image data. There are three parts to it: an input layer, an output layer, and a hidden layer. This hidden layer usually transforms the input into a signal that the output layer can recognise or use. The most important part of ANN is the learning process or training of the network. Our human brain learns from experiences. Similarly, the ANN gains knowledge from a training dataset that contains the input data and the desired result. In this phase, the network examines the incoming data to find the nodes that are associated with the output and correlates them. Artificial neural networks are excellent at pattern recognition, but they also have some disadvantages. We are unaware of the features it takes into account and their corresponding weights to produce a given output. Not only that, the data set required for perfect results from ANN is large and also it takes much time to train ANN. The power consumption for computing is also high. Apart from artificial neural networks, there are many types of neural networks, namely: recurrent neural networks, convolutional neural networks, Hopfield networks, Boltzmann machine networks, etc. These are selected based on the use of the algorithm. (Hossain et al., 2023)

We can find the value of the blood glucose level with the help of IR spectroscopy, as it is portable and less susceptible to physical conditions. First, the blood glucose level was measured before and after the meal for a set of people. PPG (photoplethysmogram) is a non-invasive technique that measures the volumetric fluctuations in blood circulation by using a light source and a photodetector at the skin's surface (Castaneda et al., 2018). Now the ppg signals were recorded with the help of Arduino and Tera term software, which provide the result in csv format. Then the signals are sampled at 115.22 Hz, and the missing data is omitted. Sampling is the process of converting a continuous time signal to discrete time signal to digitally process the data. The data is pre-processed by using two steps: filtering and eliminating. In filtering, a high pass filter is used to omit very low frequency. Eliminating refers to removing the corrupted signals. The PPG signal is normalized, the first derivative is found, and then negative zero crossing points are calculated to find systolic or diastolic peak to find corrupted signal. To determine the most neg points in the curve, notches are determined. Here, CNN is used as it is better at image processing compared to ANN. CNN consists of three layers: convolution layer, the max pooling layer, and the dense layer. The main layer, i.e., the convolution layer, is used in extracting features with the help of filters available in this layer and the first feature extraction layer's output enters into the second. The pooling layer determines the threshold and it reduces the dimension of the layer and at the last the dense layer receives the output in the end which is used to classify the features. The same data is now accessed using RNN, as it is used in processing text data and videos. It can access variable data, and it has memory to store and the ability to access binary data. In recurrent neural networks, the current state's input is fed with the output of the preceding phases. For example, one needs to store the previous letters or words in some sort of memory in order to forecast the next letter in any word or the word in a phrase. The layer that retains some memory for the sequence is the hidden layer. These algorithms are known for their memory, as they store each piece of information in each of their stages. Many sequence modelling and forecasting applications, most notably in language and speech, have shown the effectiveness of RNNs. Long Short-term Memory (LSTM) units are commonly used to create RNNs (Morup et al.). These units offer a gating mechanism that allows the RNNs to retain events from further back in time. This gating mechanism provides high precision while removing input information that is irrelevant. The three gates :input gate, the forget gate, and the output gate are controlled by the memory cell. These gates control the data that is input into, removed from, and added back to the memory cell. The input gate controls what data is added to the memory cell. The forget gate controls what information is removed from the memory cell. Additionally, the output gate regulates the data that the memory cell outputs. Because of this, long-term dependencies can be learned by LSTM networks by allowing them to choose to keep or reject information as it passes through the network. Specifically, the objective is to estimate a target blood glucose value (BGT+τ) from a time series of blood glucose (BG) data represented by (BGt, et), t=1,2,..., T. The following formulas are used to calculate the blood glucose values at each time step t, where σ represents the sigmoid function. (Mirshekarian et al.,)

$$i_t = \sigma(W^{(i)} h_{t-1} + U^{(i)} x_t + b^{(i)})$$
$$f_t = \sigma(W^{(f)} h_{t-1} + U^{(f)} x_t + b^{(f)})$$
$$i_o = \sigma(W^{(o)} h_{t-1} + U^{(o)} x_t + b^{(o)})$$

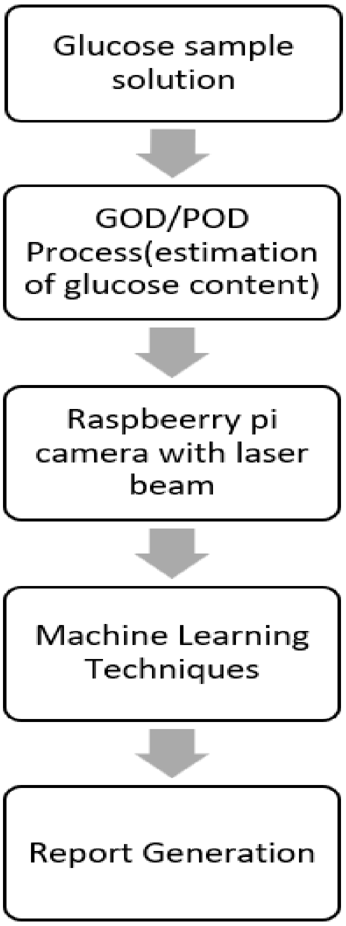
Where f represents the forget gate, o is for output, and i is for input. When the hidden state at time T is simply transformed linearly, the result is the BGL at the time horizon T +τ:

$$BG_{t+\tau} = v^T h_T + b$$

The disadvantage of using neural networks is that they take a long time and require more data. They also consume a lot of computational power. Neural networks are a 'Black Box'; when we give data as input, we get output based on some features, but we don't know how they produce it. In this case, it will be helpful to have decision trees as used in Python. Training to completely learn and understand neural networks takes more time compared to learning other programming languages.

Iot and Machine learning

Algorithms used in machine learning are trained to recognize patterns and relationships in data. To reduce dimensionality, classify information, cluster data points, and make predictions, they employ historical data as input. As machine learning can create models from the given data, it can accurately forecast the data. This makes it a valuable tool. Machine learning frequently involves the use of different algorithms for regression, including decision trees, Random Forests, K-nearest neighbours, adaptive boosting, support vector machines (SVM), and linear regression. Random forest and AdaBoost convert weak predictions to accurate predictions; these methods are commonly utilized for simple regression situations (Anand et al., 2020). First, the blood glucose level is measured using an IOT based glucose monitoring system, and then it is analysed using machine learning techniques. Artificial intelligence's machine learning field of study focuses on creating and analysing statistical algorithms that can learn from data, generalise to new data, and carry out tasks without any instructions. Here, machine learning is used to find the blood glucose value using the GOD-POD method. It is used in maintaining health care reports because the results are accurate and precise. The glucose oxidase (GOD) and peroxidase (POD) methods are used to create the glucose solution, and the colorimetric methodology is used to generate an experimental database. The Raspberry Pi camera takes a picture of the glucose solution, which is then processed through image processing to extract RGB, HSV, and LUX colour space values. To forecast the unknown glucose content, regression methods such as Random Forest, Decision Tree, Multiple Linear Regression, and XG Boost were employed. To create a glucose sample solution, standard glucose solution was diluted in a glucose reagent. In laboratories, the GOD-POD method is a rapid and precise colorimetric technique for estimating glucose content. It involves oxidising d-glucose, releasing hydrogen peroxide, and producing quinonimine dye. The dye concentration is directly proportional to the presence of glucose in the blood. Use a cuvette rather than standard test tubes and take pictures in the proper lighting conditions to produce an image dataset with low reflection. To surround the cuvette and permit light transmission, a white box was constructed. A Raspberry Pi, a tiny computer, is linked to the camera module (Poddar et al., 2023). A Raspberry Pi camera is used to take pictures of the tip of the finger while a laser beam is pointed at the finger. A 5-megapixel Raspberry Pi camera Rev 1.3 was used to capture the photos. The energy source was a visible-wavelength laser with a 2000-hour useful life at 650 nm. Using a laser beam, the amount of light absorption is proportionate to the blood sample's glucose concentration (Paredes et al., 2019). The idea involves putting a 650 nm laser and an RPI camera together in a 3D-printed casing that is fastened to a glove's index fingertip. In order to collect data and characterise the interaction between the laser-beam and the finger, the laser-beam is mounted to the case with its back to the camera lens. For accurate focus, 14 fingertip 640 x 480 px photos are captured at 8 s intervals. The technique ignores the first and last image and only takes into account the core 12 photographs. The idea involves putting a 650 nm laser and an RPI camera together in a 3D-printed casing that is fastened to a glove's index fingertip. In order to collect data and characterise the interaction between the laser-beam and the finger, the laser-beam is mounted to the case with its back to the camera lens. For accurate focus, 14 fingertips 640 x 480 px photos are captured at 8 s intervals. The technique ignores the first and last image and only takes into account the core 12 photographs. Each value in a digital image is a pixel, which is a two-dimensional matrix that indicates the amount of light at a particular location. Cropping is a technique used to remove unwanted borders from an image and obtain the intended region of interest. Images are seen by machine learning algorithms as a matrix of pixels, each of which represents the colour intensity of a single pixel. In order to create a new colour shade, RGB colour is created by combining values from 0 to 255 for each channel. A sort of machine learning known as supervised learning has datasets with inputs like 'R', 'G', 'B', 'H', 'S', 'V', and 'LUX', as target columns. This research uses four supervised machine learning algorithms: multiple linear regression, decision tree, random forest, and XGBoost. A statistical analysis technique called multiple linear regression makes predictions by creating connections between independent and dependent variables. The multiple linear regression approach outperforms all other machine learning techniques and has the highest accuracy values. In people with NGM, pre-diabetes, or type 1 diabetes, machine learning-based glucose prediction type 2 can reliably and safely measure glucose readings for up to 60 minutes (van Doorn et al., 2021). The decision tree approach is often employed for regressions and classification issues but causes overfitting. The scalability factor, optimization, and excellent speed and performance of XG Boost make it a popular decision tree boosting technique. In machine learning, large datasets are needed for training and testing machine learning models, which adds to processing time and expense. It needs much training data and power than Python and thus consumes much time. Further, IoT has complex implementation and a high initial cost.



CONCLUSION

Diabetes mellitus has emerged as one of the most prevalent diseases in the modern world. Continuous glucose monitoring (CGM) has become essential for understanding patterns in blood glucose levels. CGM devices, which are non-invasive and user-friendly, are now widely accessible to the public. Technological advancements in glucose monitoring have led to the development of various techniques and data-processing tools such as Python, AI, ML, IoT, and CNN—each with its own strengths and limitations. Among these, Python stands out as the most accessible and effective solution. As a free, open-source, general-purpose programming language, it features a beginner-friendly syntax and a vast collection of libraries such as Pandas, Flask, and NumPy. These libraries enhance Python's versatility, enabling the analysis and visualization of glucose level data, which helps patients and healthcare providers better understand glucose trends for appropriate intervention. Moreover, even in machine learning, IoT, and neural network applications, Python plays a central role in executing complex algorithms. While understanding and implementing neural network algorithms can be time-consuming and complex, Python simplifies the process with its specialized libraries. Thus, CGM devices serve to record blood glucose levels, while Python facilitates the analysis and reporting of this data in a meaningful and efficient way.

REFERENCES

1. Anand, Pradeep Kumar. "Adaptive Boosting Based Personalized Glucose Monitoring System (PGMS) for Non-Invasive Blood Glucose Prediction with Improved Accuracy." *NCBI*, 2020; 7 May. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7278000/ Accessed 14 January 2024.
2. Bent, Brinnae. "Cgmquantify: Python and R Software Packages for Comprehensive Analysis of Interstitial Glucose and Glycemic Variability from Continuous Glucose Monitor Data." *PubMed*, 2021; 18 August, https://pubmed.ncbi.nlm.nih.gov/35402978/ Accessed, 2024; 6 January.
3. Castaneda, Denisse. "A review on wearable photoplethysmography sensors and their potential future applications in health care." *NCBI*, 2018; 6 August, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6426305/ Accessed, 2024; 14 January.
4. Dubosson, Fabien. ",.", - *YouTube*, 2023; 17. December, https://ieeexplore.ieee.org/abstract/document/7474196. Accessed 11 January 2024.
5. Hossain, Shamima. ",.", - *YouTube*, 2023; 13 November, https://ieeexplore.ieee.org/document/9063187 Accessed, 2024; 14 January.
6. José, F. "Identification of Blood Glucose Patterns through Continuous Glucose Monitoring Sensors and Decision Trees." *medRxiv*, 2020; 30 November, https://www.medrxiv.org/content/10.1101/2020.09.09.20190736v2.full Accessed 10 January 2024.
7. Mirshekarian, Sadegh. "LSTMs and Neural Attention Models for Blood Glucose Prediction: Comparative Experiments on Real and Synthetic Data." *NCBI*, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7890945/ Accessed, 2024; 17 January.
8. Morup, Morten. "Short Term Blood Glucose Prediction based on Continuous Glucose Monitoring Data." *PubMed*,

https://pubmed.ncbi.nlm.nih.gov/33019143/ Accessed 14 January 2024.

9. Paredes, Antonio Alarcon. "An IoT-Based Non-Invasive Glucose Level Monitoring System Using Raspberry Pi." *MDPI*, 2019; 28 July, https://www.mdpi.com/2076-3417/9/15/3046 Accessed 14 January 2024.
10. Payne, Walker. "How to Analyze Blood Glucose Data with Python Data Science Packages." *Towards Data Science*, 2021; 1 September, https://towardsdatascience.com/how-to-analyze-blood-glucose-data-with-python-data-science-packages-4f160f9564be Accessed 6 January 2024.
11. Poddar, Akash. *YouTube*, 2023; 13 November, https://www.sciencedirect.com/science/article/abs/pii/S2214785322062630 Accessed 14 January 2024.
12. van Doorn, William P T M. "Machine learning-based glucose prediction with use of continuous glucose and physical activity monitoring data: The Maastricht Study." *PubMed*, 2021; 24 June, https://pubmed.ncbi.nlm.nih.gov/34166426/ Accessed 14 January 2024.