

Investigating sparse model for Continuous Glucose Monitoring in Type 2 Diabetes

A Thesis

submitted to

Indian Institute of Science Education and Research Pune
in partial fulfilment of the requirements for the
MS (by Dissertation) Programme

by

Prajjwal



Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

September, 2022

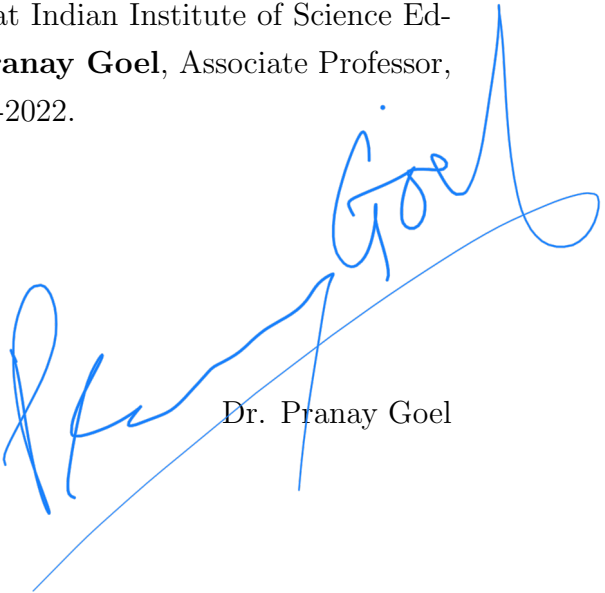
Supervisor: Dr. Pranay Goel

© Prajjwal 2022

All rights reserved

Certificate

This is to certify that this dissertation entitled Investigating sparse model for Continuous Glucose Monitoring in Type 2 Diabetes towards the partial fulfilment of the MS (by Dissertation) programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by **Prajwal** at Indian Institute of Science Education and Research under the supervision of **Dr. Pranay Goel**, Associate Professor, Department of Biology, during the academic year 2021-2022.



Dr. Pranay Goel

Committee:

Dr. Pranay Goel

Dr. M.S. Santhanam Dr. Collins Assisi

Dedicated to diabetic patients all over the world.

Declaration

I hereby declare that the matter embodied in the report entitled Investigating sparse model for Continuous Glucose Monitoring in Type 2 Diabetes are the results of the work carried out by me at the Department of Biology, Indian Institute of Science Education and Research Pune, under the supervision of Dr. Pranay Goel and the same has not been submitted elsewhere for any other degree.



Prajjwal

Prajjwal

Acknowledgements

This thesis and the work I've done over the last year would not have been possible without the assistance of several people, whom I'd like to thank here.

First and foremost, I'd like to thank Dr. Pranay Goel, who has been my mentor for the past year and has provided me with invaluable advice and guidance on how to proceed. He has always encouraged, inspired, and corrected me when necessary. I've had long conversations with him which always provided me with new insights into research and various ideas. It is only because of those discussions that my project has transformed into something with true academic importance. I am grateful to have had a guide who permitted me to be free in many areas as long as the needed work was carried out precisely.

I would like to express my heartfelt gratitude to Dr. Saroj Ghaskadbi and Saurabh D. Kalamkar (Savitribai Phule Pune University) for providing us with the necessary data and for allowing me to collaborate with them. I'd also like to thank Sayantan Majumdar, Sandra Aravind, Arjun KM, and Somashree Chakrobarty, who are all working on their doctorates in Dr. Pranay Goels' lab.

I would also like to mention my parents, Mrs. Archana Srivastav and Mr. Vinod Srivastav, and my sister Harshita Srivastav who have always supported and believed in me, deserve my sincere gratitude. They will be the focus of my entire life, not just for this thesis. I hope that someday I will be able to mark a contribution from my side in healthcare for those who are struggling with similar issues and are in need of assistance, since my mother suffers from both diabetes and high blood pressure, which led to the failure of her kidneys i can understand how deadly this disease can be. And in the end I would also like to thank MS project students from our lab Rashmi Chapke and Suyog Sankhe for their valuable suggestions and comments.

Abstract

Continuous Glucose Monitoring (CGM) is a cutting-edge method for monitoring blood glucose levels at predetermined intervals. Type 2 diabetes is a long-term chronic lifestyle disease brought on by high blood sugar levels. In our investigation, we will use isolated liquid meals and CGM data to predict the glucose level inspiring from a widely used fitting method that is described in this work. Owing to the CGM data's significant nonlinearity and the complexity of adequately modelling it, we restricted the use of non-linear approaches to a single time period during the day. After rigorous research, we discovered that we can draw inspiration from the underdamped case of a damped harmonic oscillator to simulate our data accurately. We then used the same technique to apply and model CGM data available for five patients (2 non-diabetic, 2 diabetic and one pre-diabetic). We have carefully analyzed and shown the data, day by day as well, so that it may be used for more comprehensive analyses of the glucose dynamics and the prescription of a diet or medication for diabetic patients since it demonstrates how glucose varies with simple liquid meals.

Keywords: Non-Linear modelling, Data visualization and analysis, Continuous Glucose Monitoring, Harmonic Oscillator

Contents

Abstract	xi
Introduction	1
1 Preliminaries	3
1.1 Continuous Glucose Monitoring (CGM)	3
1.2 Modelling CGM Data	4
1.3 Objective of current Study	5
2 Deriving the solution	6
2.1 Damped harmonic oscillator	6
2.2 Solution for liquid meals	7
3 Fitting the model	9
3.1 Non-Diabetic	12
3.1.1 Diet one (Non- Diabetic) :	12
3.1.2 Diet two (Non- Diabetic) :	19
3.2 Pre-Diabetic	23
3.2.1 Diet three (Pre-Diabetic) :	23
3.3 Diabetic :	27
3.3.1 Diet four (Diabetic) :	27
3.3.2 Diet five (Diabetic) :	31
4 Further Discussions and Conclusion	35

5 Appendix	37
5.1 MATLAB code for tea time extraction, modelling and plotting	37
5.2 Python code for parameters visualisation and analysis for diet one of dia- betic case.	45
Bibliography	48

Introduction

A sensor, often mounted to the arm, is used in continuous glucose monitoring (CGM), which uses it to provide blood glucose readings every 15 minutes for two weeks. The availability of this high-quality, high-time resolution approach is growing [1]. There has been various model proposed which involves modelling using neural network [2], with derivative free optimization techniques [3], and also minimal model approach [1]. various techniques use particular time range for modelling; one such recent study conducted have modelled the CGM data during sleep [4][5].

For the purposes of our investigation, we first began by modelling complicated time series data using SINDy algorithm [6], where the sparse identification of non-linear dynamical systems is carried out using sparse regression techniques, which is discussed in the last section of this work. After that, we looked at a variety of sparse representation methods; for instance, one in-depth research was conducted on the ISTA algorithm [7], but to the best of our knowledge, none completely explained the dynamics of glucose. As a result, we switched to the most widely used modelling scenario of a damped harmonic oscillator and got our model inspiration from that, which was subsequently applied to all participants but only within a predetermined window during which only the liquid meal was ingested.

Our research objectives :

The data visualisation and analysis for all of the patients for each day was a major component of our study so that we could better understand how and when to do the modelling. We will go over the CGM models and studies that have been done in the first chapter. In the second chapter, we will discuss the equations we are employing and the parameters we will use to fit our CGM data for the non-diabetic, diabetic, and pre-diabetes cases.

Then, in Chapter three, a thorough analysis of every diet diary provided has been carried out along with the discussion of data pre-processing and loading methodologies as well as the step-by-step methodology we employed. In the last section of this work, we had gone into short depth about the different ideas we had previously worked on in the discussion portion.

Chapter 1

Preliminaries

1.1 Continuous Glucose Monitoring (CGM)

CGM is an innovative technique for measuring blood glucose levels at predetermined intervals. For 14 days, the sensor, which is attached to either the arm or the abdomen, measures blood glucose every 15 minutes. The time series obtained from CGM data allows us to gain a broad understanding of the variation of blood glucose throughout the day, particularly after meals and other regular activities. The average blood glucose level in diabetic CGM data is much higher (around 200 mg/dl) than in non-diabetic CGM data (around 100 mg/dl).

Additionally, the diabetic CGM time series has glucose pulses that are wider and stronger than those in the case of non-diabetic patient. CGM is influenced by the individual's physiology as well as the amount of food consumed by the subject while being measured. As a result, the full dynamic modelling of the CGM time series is currently difficult. We wanted to mimic the glucose peaks seen in the CGM data just before breakfast in this investigation.

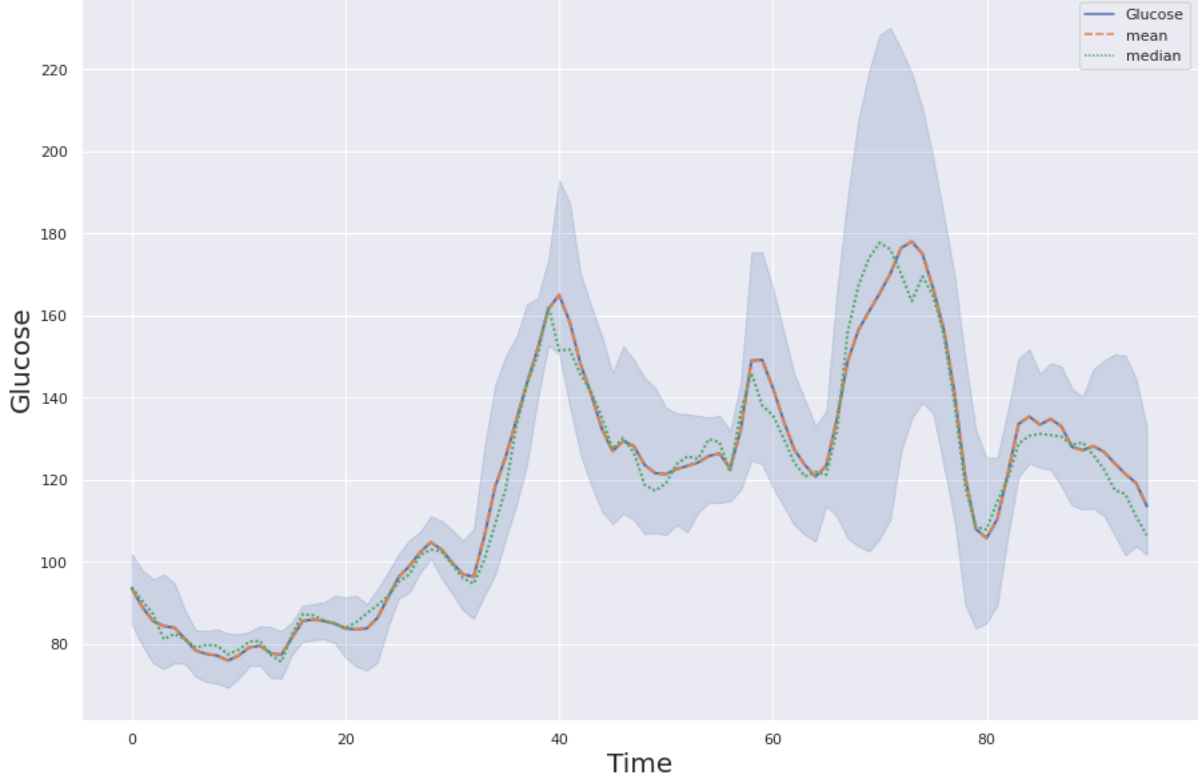


Figure 1.1: Graph of Glucose (mg/dl) vs Time (15 min interval) scattered through a week of diet one from the diabetic data available. Mean and median values are also plotted for a week in orange and green respectively.

1.2 Modelling CGM Data

For modelling CGM various techniques have been used in the past. For example, the Topp model was adapted for glucose-insulin dynamics, and a two compartmental food dynamics model was developed in [1] :

$$\frac{dG}{dt} = R_0 - (EG_0 + S_I I)G + k_{gut}q_{gut} \quad (1.1)$$

$$\frac{dI}{dt} = I_{max} \frac{G^2}{\alpha + G^2} - k_I I \quad (1.2)$$

$$\frac{dq_{sto}}{dt} = -k_{sto}q_{sto} \quad (1.3)$$

$$\frac{dq_{gut}}{dt} = k_{sto}q_{sto} - k_{gut}q_{gut} \quad (1.4)$$

Previously, optimisation was performed in Matlab using a combination of the functions `fmincon` and `patternsearch` with appropriate constraints. [1].

Also The derivative-free optimization and neural network approach has also been

applied to the CGM data [3, 2].

1.3 Objective of current Study

Goel and Ghaskadbi recently proposed an ‘Isolated Liquid Meals Tolerance Test’ (ILMTT) (private communication) which is the intake of a liquid meal, sufficiently separated from another meal, as an alternative to – a CGM version of – the OGTT.

My objective in this thesis is to model the glucose peaks observed after liquid meals, which often occur before the subjects breakfast and can be referred to as ILMTT. As we considered five diets (two non-diabetic, two diabetic, and one pre-diabetic), all of which contain diverse meals like tea, milk, coffee, etc., the liquid meal is not particular and can vary from person to person. For modelling, we analyse that glucose values follow patterns like an underdamped harmonic oscillator where amplitude will oscillate and decay with time, then asymptote to a particular point. And we propose our model by adding an extra intercept term ‘h’ taking an inspiration from that as :

$$G = a \exp(-bt) \cos(wt) + h \quad (1.5)$$

where G is the glucose value at time t and a, b, h and w are unknown parameters to be estimated.

Applying this to all of the available glucose data will help us choose the parameters that will best fit our model. Next, we will compare the original data points with the outcomes of our model to determine the statistical analysis.

Chapter 2

Deriving the solution

The most commonly and widely used model in physics is a harmonic oscillator, and for our modelling, we are using a special case of harmonic oscillator i.e under-damped harmonic oscillator. We will just explain the equations that we are going to use with a simple spring-mass system.

2.1 Damped harmonic oscillator

As it is known from Hooke's law that without an external force acting on the system, the net force on the spring-mass system is given as:

$$F_{net} = ma = -kx \quad (2.1)$$

And for a particular initial conditions system will oscillate forever with the constant force. And the trajectory it follows will be given by the general solution :

$$x(t) = A\sin(w_0t) + B\cos(w_0t) \quad (2.2)$$

where

$$w_0^2 = \frac{k}{m} \quad (2.3)$$

But in the case of damping, net force changes as a function of velocity , thus an additional force of the form $F_b = bv$ will be added and our equation becomes :

$$F_{net} = -kx - bv = ma = m\frac{d^2x}{dt^2} \quad (2.4)$$

2.2 Solution for liquid meals

Rearranging all our terms we get :

$$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + w_0^2x = 0 \quad (2.5)$$

where

$$2\beta = \frac{b}{m}, \quad w_0^2 = \frac{k}{m} \quad (2.6)$$

With a bit of foresight we know that the solution can be of form of :

$$x = A \exp(\alpha t) \quad (2.7)$$

Substituting in the equation above we get the auxiliary equation to be :

$$\alpha^2 + 2\beta\alpha + w_0^2 = 0 \quad (2.8)$$

As the above equation is in the quadratic form hence the solution can be written as:

$$\alpha_1 = -\beta + \sqrt{\beta^2 - w_0^2} \quad (2.9)$$

$$\alpha_2 = -\beta - \sqrt{\beta^2 - w_0^2} \quad (2.10)$$

And we can write general solution of this motion as :

$$x(t) = \exp(-\beta t)[A_1 \exp(\sqrt{\beta^2 - w_0^2}t) + A_2 \exp(-\sqrt{\beta^2 - w_0^2}t)] \quad (2.11)$$

There are special cases to above solution, SHM (oscillations) , under-damped oscillations , critical damped and over-damped.

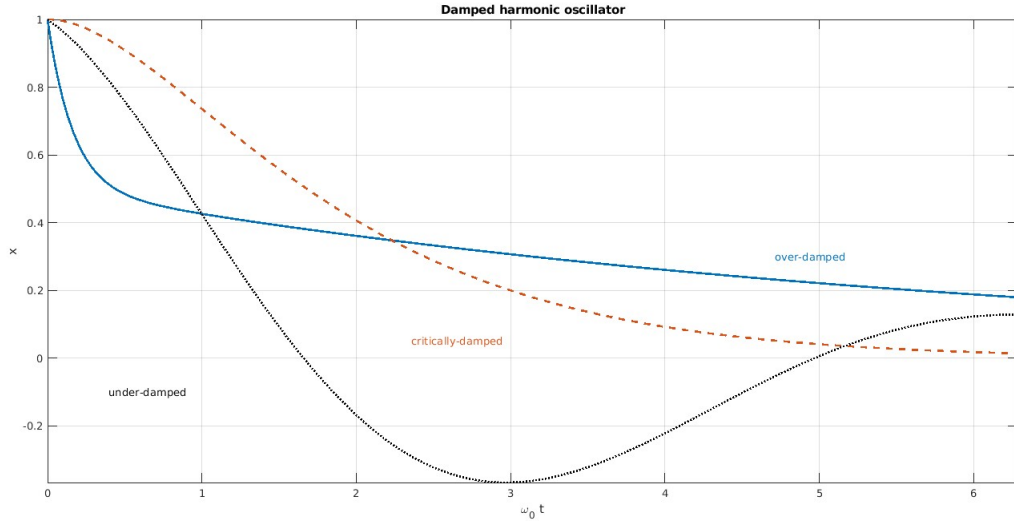


Figure 2.1: Trajectory for all three cases. Black : underdamped , red : critically damped and blue : overdamped.

What we are interested in for our modelling is the underdamped case where :

$$b^2 < 4km \quad \text{or} \quad \beta^2 < w^2.$$

(2.12)

Therefore, in this case, roots are imaginary, which will transform the solution as :

$$x(t) = A \exp(-\beta t) \cos(w_1 t - \delta) \tag{2.13}$$

Hence taking an inspiration from 2.13 we propose our modelling equation as:

$$G = a \exp(-bt) \cos(wt) + h \tag{2.14}$$

where the additional term "h" is added to provide our model with flexibility, allowing it to go up and down now. Next, we will apply this during the liquid meal time window of all the five cases.

Chapter 3

Fitting the model

Five patients' diets have been investigated (2 non-diabetic, one pre-diabetic and two diabetics). All of the diet diaries that are currently available employ the model that we proposed to fit the CGM data with isolated liquid meals.

For the fitting, we examined the tea time for each participant and found that it may resemble an underdamped case of damped harmonic oscillation, in which the wave's amplitude lowers over time and reaches saturation after a predetermined amount of time. We used data sets from the subjects' tea time and used our model there while keeping this in mind.

Our investigation revealed that our algorithm does a good job of capturing liquid meals from the CGM data.

Data :

The daily glucose values that were measured over a 15-minute period are part of the CGM data that we are using to construct our model. Additionally, we have a diet journal to record when the peak meal was had and what a larger peak in the data means. Every case has data for two weeks, and a diet diary includes all meals consumed by the patient, not only liquid meals.

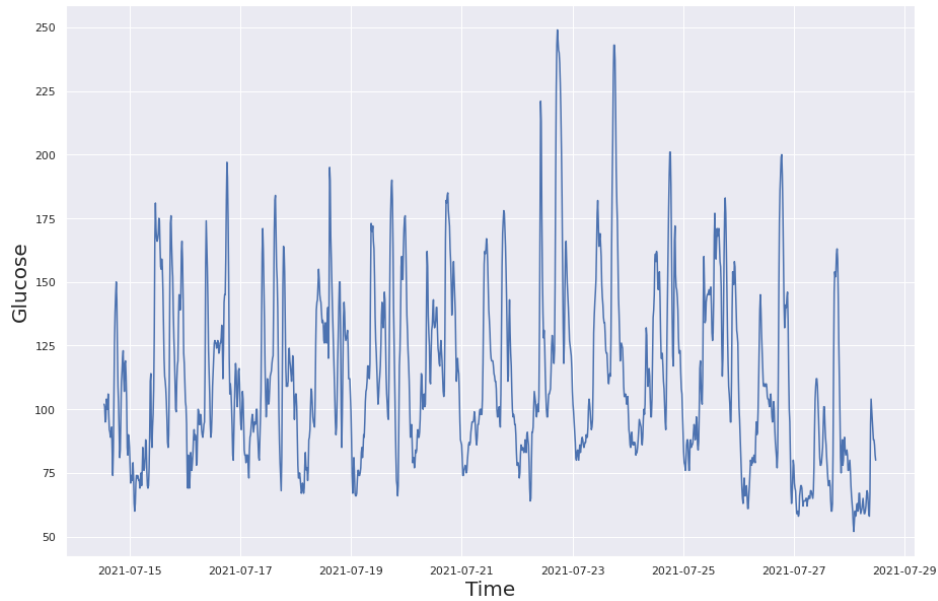


Figure 3.1: Graph between Glucose (mg/dl) and Time (day) variation for 15 days of Diet one (diabetic) data in DateTime format.

Procedure :

For a specific diet, we first loaded our CGM (Continuous Glucose Monitoring) data. You can see in 3.1 how challenging it is to analyse after charting for all days. We began daily analysis to search for patterns because it is practically difficult to analyse this way. For instance, rather than looking at everything at once, we can better grasp the dynamical behaviour of the single day plot as illustrated in figure 3.2

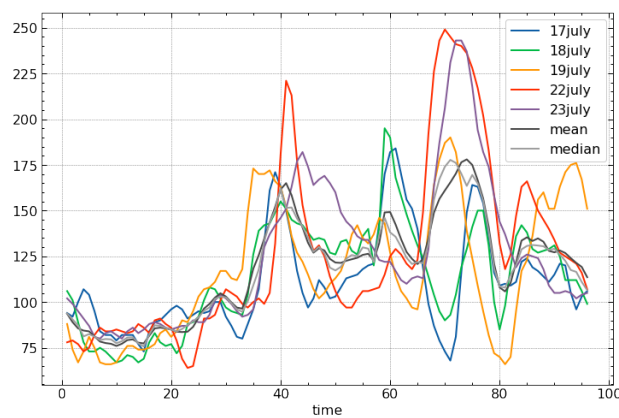


Figure 3.2: Plot of Glucose (mg/dl) vs time (15min gap) variation of diet two (diabetic) data for all days including mean and median of them.

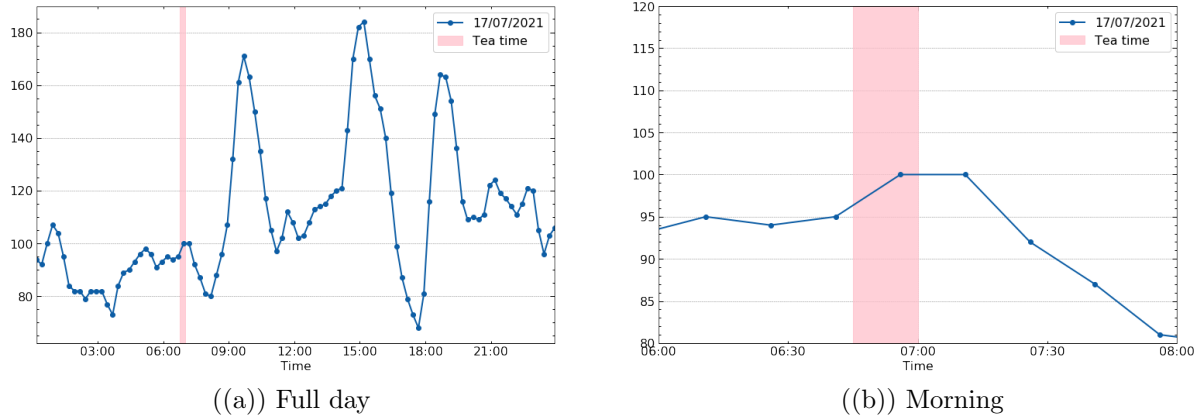


Figure 3.3: Graph between Glucose (mg/dl) and Time (min) for July 17th 2021 from the case two of diabetic data.

Now that we are just interested in one specific aspect of the diet—the liquid meal time—we have realized that, according to most diet charts, this occurs in the morning and typically consists of tea, milk, or coffee. If we take only one day from the diabetic case i.e July 17th 2021 where the liquid meal have tea in the morning it will look like figure 3.3. and we are going to fit that part only through our modelling procedure.

We discovered that it follows a specific harmonic oscillator, or under-damped case, where the amplitude converges to a specific point, after reviewing the data that was accessible day by day for all patients and using various non-linear modelling techniques. The figure 3.4 illustrates how accurately glucose dynamics may be predicted for this specific situation. However, keep in mind that it is diet-specific and depends on the individual’s food intake.

We shall now proceed to actually applying our model to each diet possibility individually. We will begin by outlining the procedure in non-diabetic cases before applying it to all other scenarios. To make things more convenient, we use Matlab for model fitting and interpolation. The model entails loading the data sets first, after which we carefully analyse what meal was consumed when, and then we extract data points for setting up the fit. The coming sections discusses step-by-step procedures from data visualisation and applications, and at the conclusion of each case, a model with the estimated parameters is shown with the data points and newly added ones.

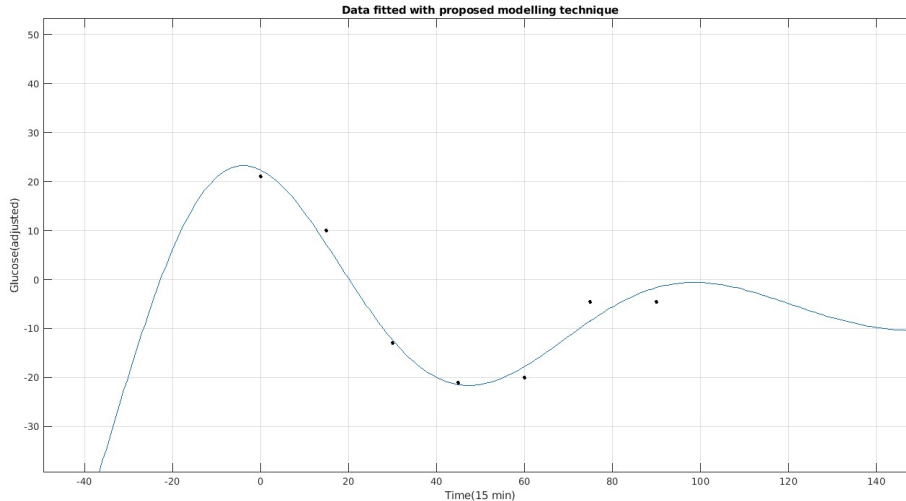


Figure 3.4: Plot of Glucose (Adjusted) vs Time (15 min) of diet two (diabetic). Data points taken from morning time of diet were fitted with the model and it is observed that most of the diets follows this approach reasonably well.

3.1 Non-Diabetic

First, we will apply our modelling methods to two situations of people who do not have diabetes after that, pre-diabetic and diabetic. The non-diabetic cases are discussed as follows :

3.1.1 Diet one (Non- Diabetic) :

We will first go over the data pre-processing and loading methods used:

- The data set containing glucose and time values for case one of non-diabetic CGM provided will be loaded into the matlab workspace by first converting the time column to the appropriate `dateTime` format, which is :

$$dtformat = 'yyyy/MM/dd HH:mm'$$
- Now that the data has been loaded into the matlab programme in 'table' format, we must first convert it to 'timetable' format because we are working with time-series data and need to access a certain window of time. The only information that is necessary is Time and Glucose(mg/dl) values, hence we have also deleted unnecessary columns that were provided for labelling and marking.
- The CGM data available for all diets is for two weeks and after evaluating the first diet, we noticed that liquid meals alone were only consumed for 9 days in the morning, between 7:00 and 9:00 am, and on July 15th from 9:00 to 11:00 am.

- Therefore, we chose a time window from the schedule for all days from 7-9 am, but for July 15 from 9-11 am, and then the glucose readings were retrieved. After extracting the values, we plotted each one separately to determine whether or not it followed our model. For instance, it appears as follows for August 13th in figure 3.5

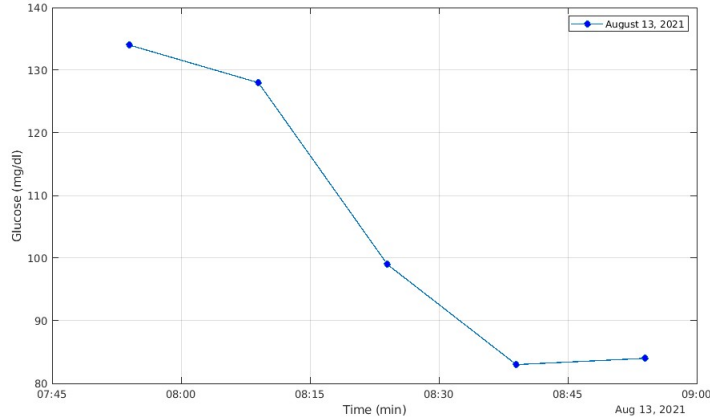


Figure 3.5: Graph between Glucose(mg/dl) vs Time(min) of August 13th, 2021 from case one of a non-diabetic subject.

When it comes to the fitting and modelling process, we are only interested in particular data points that will take the behaviour of the curve into account. And as part of our procedure, we create a new array in the current library using the peak data values and minima for a curve. And this will be the standard technique for all of the diets we follow. The method we'll employ is described in detail below:

- We first extract the data points before and after tea time, typically within a two-hour window for each day, and store them in a separate Matlab timetable. We next pick the maximum value and minimum value for a given day, and compute an average. The updated adjusted values are then obtained by subtracting the computed average value from the glucose measurements for that particular day.
- To ensure that the data points converges, for interpolation we will first take the average of the adjusted values and then, at the very end, add the computed value twice to the adjusted values to store in a new updated array.
- After plotting and visualizing the daily glucose data, we will analyse and determine which day will fit our recommended model the best.
- Repeat the above procedure for each day to see which will produce the good results and remove or add points according to that.

After using the aforementioned procedure, we can now determine that the days that produce the fittest results are 11/07/21 through 18/07/21.

Once more using Matlab to fit the model to the data we generated above with the recommended model discussed in chapter two is:

$$G = a \exp(-bt) \cos(wt) + h \quad (3.1)$$

Our goal will be to find the parameters a, b, w, and h which will best fit our data for each day, and we will do this day by day. First, we load our created array of adjusted glucose for fitting and after looking through optimised parameters for all days we got the following result:

Date	a	b	w	h
11/07/21	38.53	0.014720	0.05885	-6.898
12/07/21	48.87	0.018780	0.05511	-16.460
13/07/21	33.96	0.010360	0.05873	-4.989
14/07/21	25.00	0.006660	0.05658	-2.077
15/07/21	19.93	0.001182	-0.04541	4.000
16/07/21	19.42	0.000558	0.05460	2.996
17/07/21	13.45	0.000210	-0.05760	2.000
18/07/21	36.04	0.020220	0.06337	-9.796

Python is used for the statistical analysis of parameters, it is preferred that the parameter values be loaded and turned into a pandas data frame before analysis. The values for each parameter's mean, standard deviation, range, high, low, and percentile are as follows:

index	a	b	w	h
count	8.0	8.0	8.0	8.0
mean	29.40000	0.009086	0.030529	-3.903
std	11.894565	0.008204	0.050810	7.079028
min	13.45	0.00021	-0.0576	-16.46
25th percentile	19.8025	0.00103	0.0295975	-7.6225
50th percentile	29.48	0.00851	0.055845	-3.533
75th percentile	36.6625	0.015735	0.05876	2.249
max	48.87	0.02022	0.06337	4.0

The correlation graph is plotted as a heatmap in the 3.10:

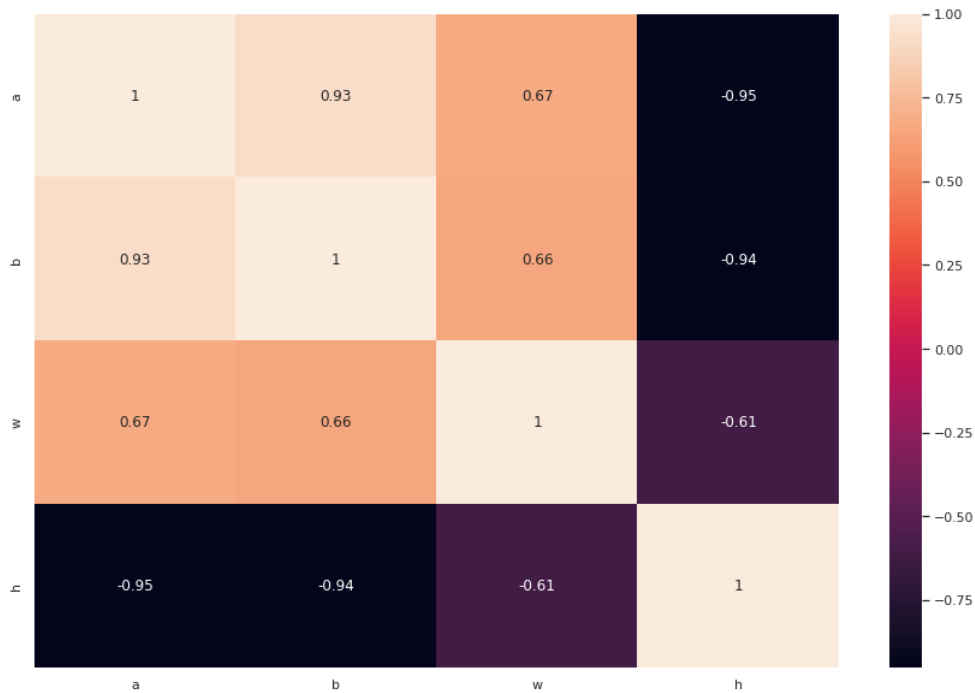
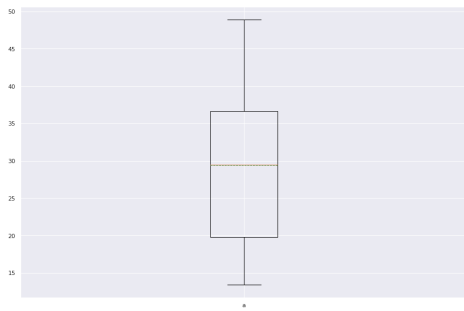
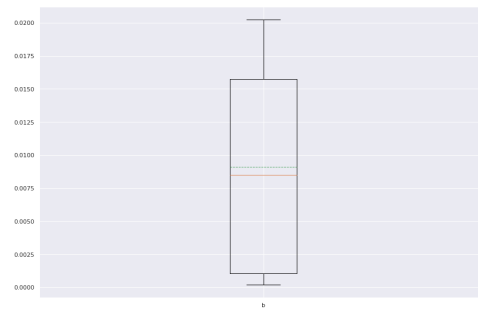


Figure 3.6: Heatmap of correlation between parameters for diet-one of a non-diabetic patient.

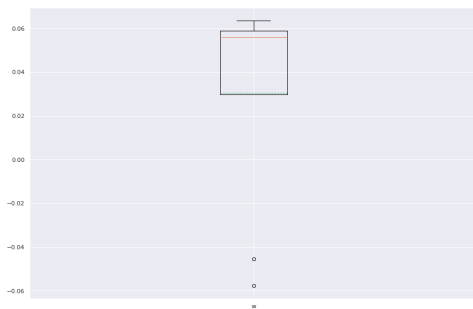
We will now use a **Box-plots** to analyse the parameters , for this diet they are plotted individually in figure 3.7 and combined in figure 3.8. **Box-plots** are used to gauge how evenly dispersed a data set's data are. It creates three quartiles out of the data set. The minimum, maximum, median, first quartile, and third quartile of the data set are shown in this graph. By creating box-plots for each data set, it is also helpful for comparing the distribution of data across data sets. Maximum, minimum, median, and mean values are indicated here. The green and orange lines are where the mean and median line is dotted respectively.



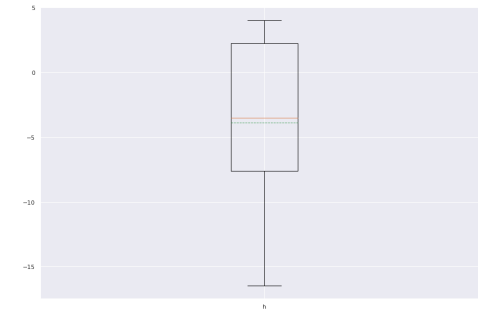
((a)) Box-plot for 'a'



((b)) Box-plot for 'b'



((c)) Box-plot for 'w'



((d)) Box-plot for 'h'

Figure 3.7: Box-plots for all parameters of diet one (non-diabetic) patient.

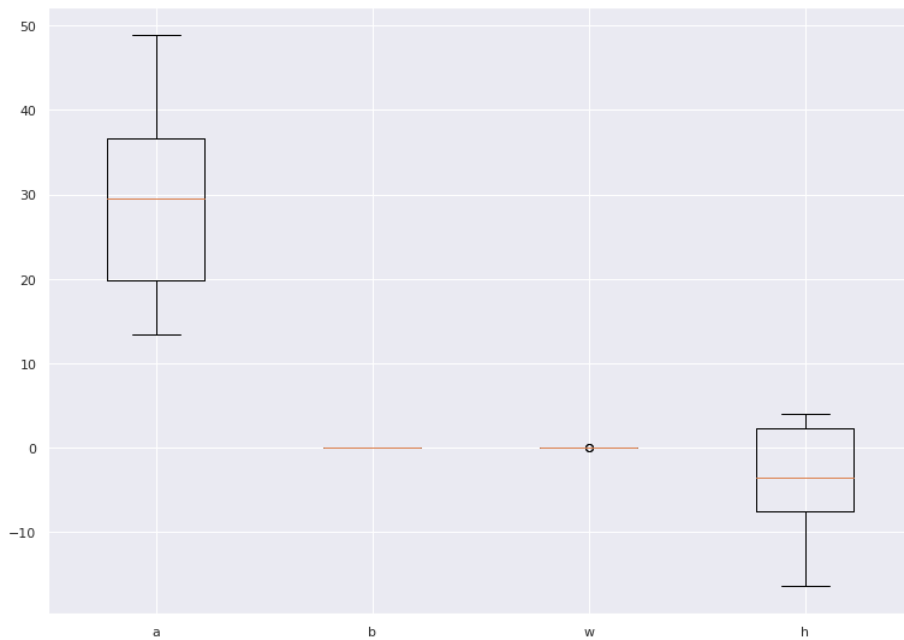


Figure 3.8: Box-plots for all the parameters combined in one scale for comparison of diet one (non-diabetic).

We present the graph in figure 3.9 of actual data from the tea time and the plot from estimated parameters on the same scale for all the selected dates after determining the appropriate parameter values. This summarises our research and findings regarding the liquid meal time for non-diabetic patient one.

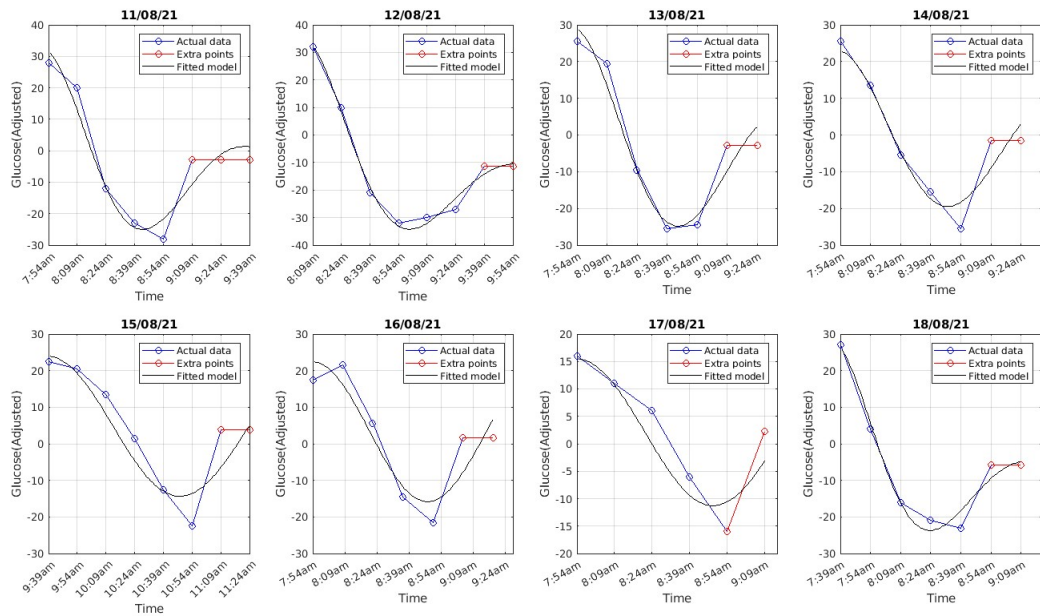


Figure 3.9: A date-wise depiction of the diet dairy’s actual data points (marked with blue) and interpolated points (marked red) overlaid with the black-fitted model after applying our model to case one of non-diabetic patient.

3.1.2 Diet two (Non- Diabetic) :

In order to perform our analysis before loading the data set, we will once more go through all the prior processes that were covered for the first patient instance. The particular liquid meal time will be noted in the diet diary when the time series data has been loaded as CGMND04.csv in which data of diet two of non-diabetic case is available. In this case, the time ranges from 8 to 11 am in the morning, as shown in **Listing 5.1** at the appendix page.

We got fit for July 21, 23, 25, 26, 27, 28, 29, and 30 throughout the course of the two weeks from July 19 to July 21. Additionally, the table below lists the estimated parameters by modelling:

Date	a	b	w	h
21/07/21	-20.0	0.007968	-0.131	0.04872
23/07/21	-1.388	0.0179	0.3639	0.0843
25/07/21	1.483	0.02152	0.0337	0.1277
26/07/21	1.713	0.01779	-1.085	0.102499
27/07/21	2.0	0.01329	-0.2785	0.1008
28/07/21	2.569	0.02122	-0.4837	0.0895
29/07/21	3.0	0.009693	0.3029	0.07043
30/07/21	2.37	0.01327	-4.214	-0.07268

We will acquire statistical insights into our set of parameters after loading them into a Pandas data frame. First, the minimum, maximum, and percentile are provided in the Table. In the table below count indicates the total number of values for each parameter, which is always eight. As usual, min and max are denoted.

index	a	b	h	w
count	8.0	8.0	8.0	8.0
mean	-1.031625	0.015331	-0.6865	0.06891
std	7.7806	0.005072	1.498531	0.061783
min	-20.0	0.007968	-4.214	-0.07268
25th percentile	0.7652500	0.012375750	-0.63402500	0.0650025
50th percentile	1.8565	0.01554	-0.2047500	0.0869
75th percentile	2.41975	0.01873	0.101	0.10122
max	3.0	0.02152	0.3639	0.1277

The correlation graph is plotted as a heatmap in the 3.10:

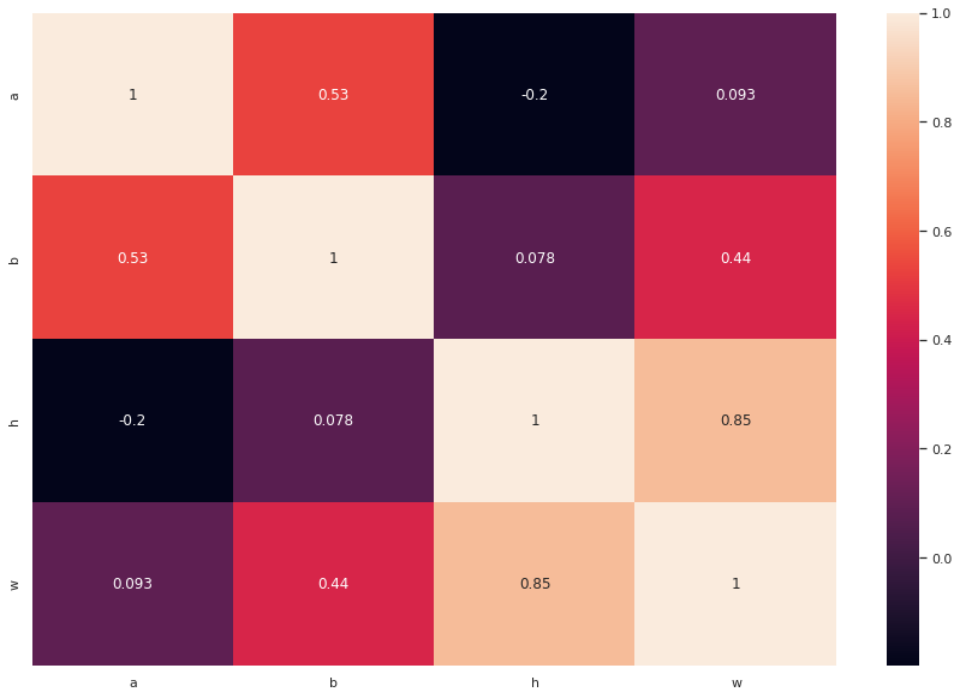


Figure 3.10: Heatmap of correlation between parameters for case two of a non-diabetic patient.

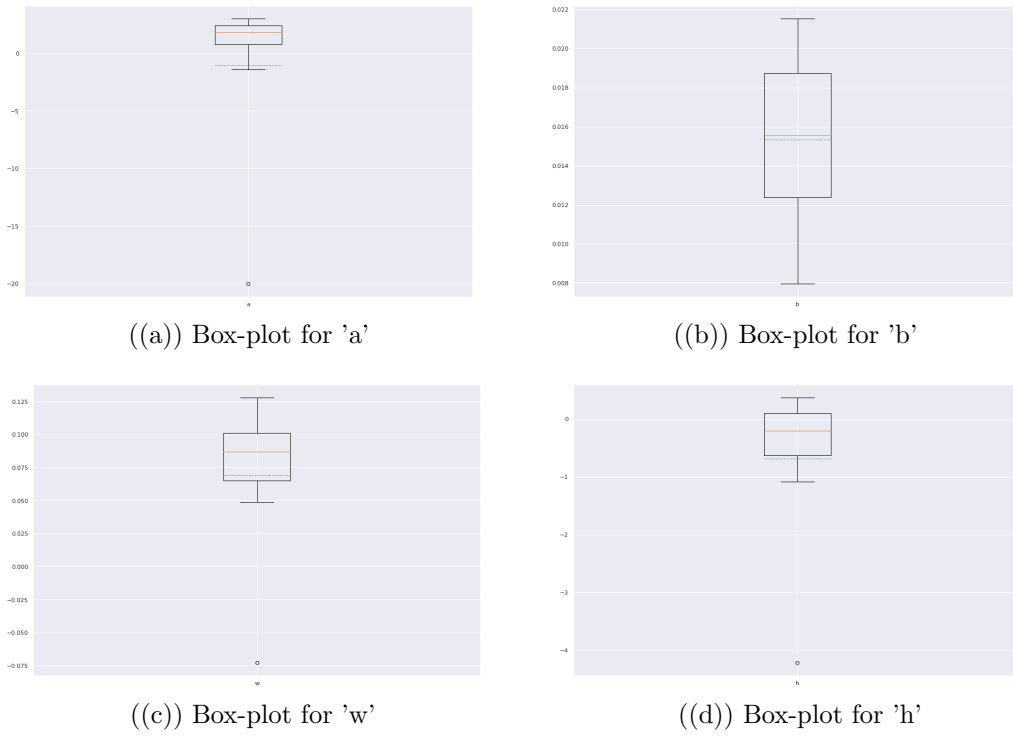


Figure 3.11: Box-plots for all parameters for case two of non-diabetic patient.

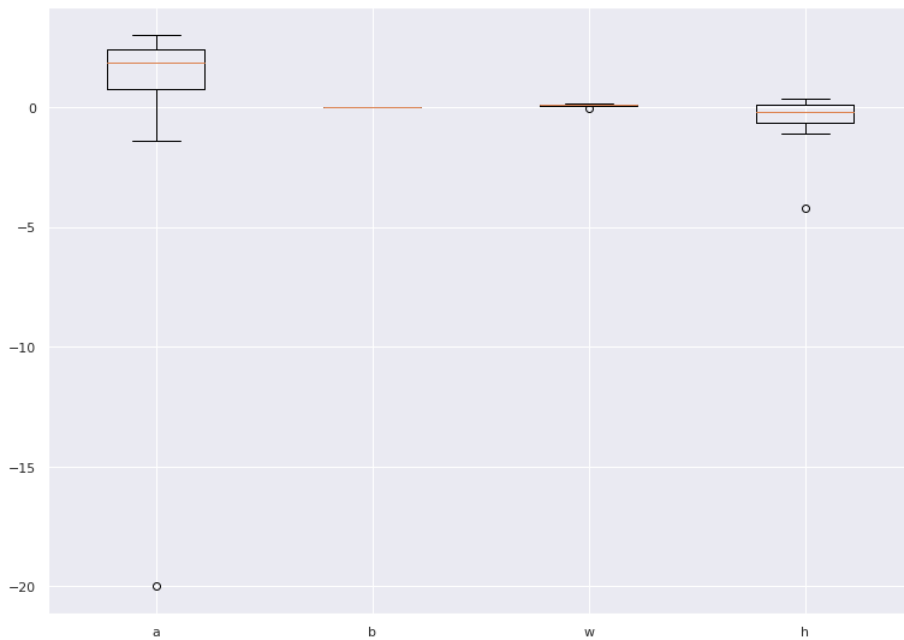


Figure 3.12: Box-plots for all the parameters of case two (non-diabetic) combined in one scale for comparison.

We will now use a **Box-plots** to analyse the parameters , for this diet they are plotted individually in figure 3.7 and combined in figure 3.8.

Finally, we will plot in figure 3.13 the continuous glucose values with the actual data points to see how well it fits a particular time range.

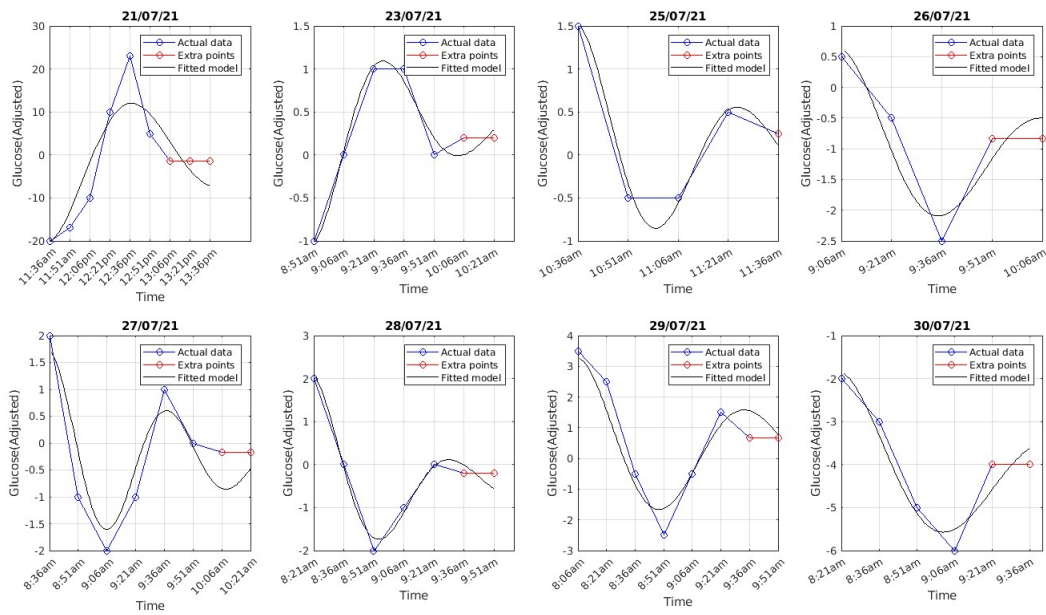


Figure 3.13: A date-wise depiction of the diet dairy’s actual data points (marked with blue) and interpolated points (marked red) overlaid with the black-fitted model for case two (non-diabetic).From the above figure we can see that our proposed model generalizes well as most of the plot in black gives rsquare values above 90 percent.

3.2 Pre-Diabetic

3.2.1 Diet three (Pre-Diabetic) :

Here, we analyse the data for the pre-diabetic patient with the same approach. Its been also performed for eight days from the 15 days available: Additionally, the table below lists the estimated parameters by modelling:

Date	a	b	h	w
6/08/21	-9.000	0.008632	2.0000	0.05903
11/08/21	-25.760	0.016380	5.1450	-0.06430
12/08/21	5.386	0.014940	-1.1050	0.06220
13/08/21	25.950	0.008908	-2.8470	0.05785
14/08/21	-30.090	0.007368	0.3336	0.04970
15/08/21	29.670	0.014780	-7.2970	0.06126
16/08/21	14.670	0.003637	2.0000	0.03995
17/08/21	31.000	0.007554	-4.6880	0.04710

We will acquire statistical insights into our set of parameters after loading them into a pandas data frame. First, the minimum, maximum, and percentile are provided in the Table. Here count indicates the total number of values for each parameter, which is always eight. As usual, min and max are denoted.

index	a	b	h	w
count	8.0	8.0	8.0	8.0
mean	5.228250	0.010275	-0.807300	0.039099
std	24.466415	0.004533	4.034112	0.042497
min	-30.090000	0.003637	-7.297000	-0.064300
25th percentile	-13.190000	0.007508	-3.307250	0.045313
50th percentile	10.028000	0.008770	-0.385700	0.053775
75th percentile	26.880000	0.014820	2.000000	0.059588
max	31.000000	0.016380	5.145000	0.062200

The correlation graph is plotted as a heatmap in the 3.14:

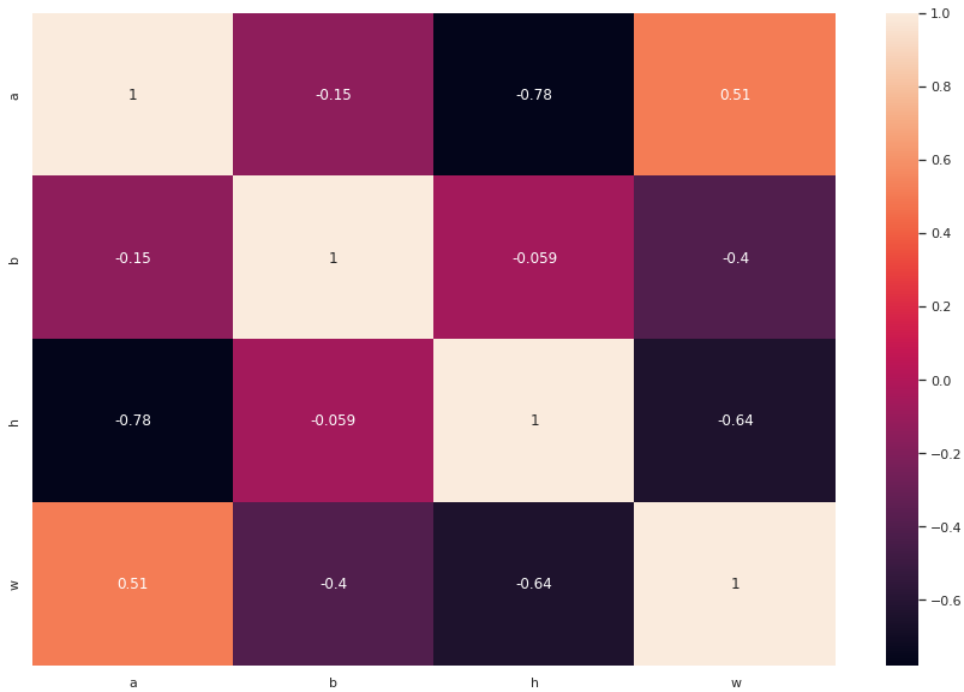


Figure 3.14: Heatmap of correlation between parameters for case one of a pre-diabetic patient.

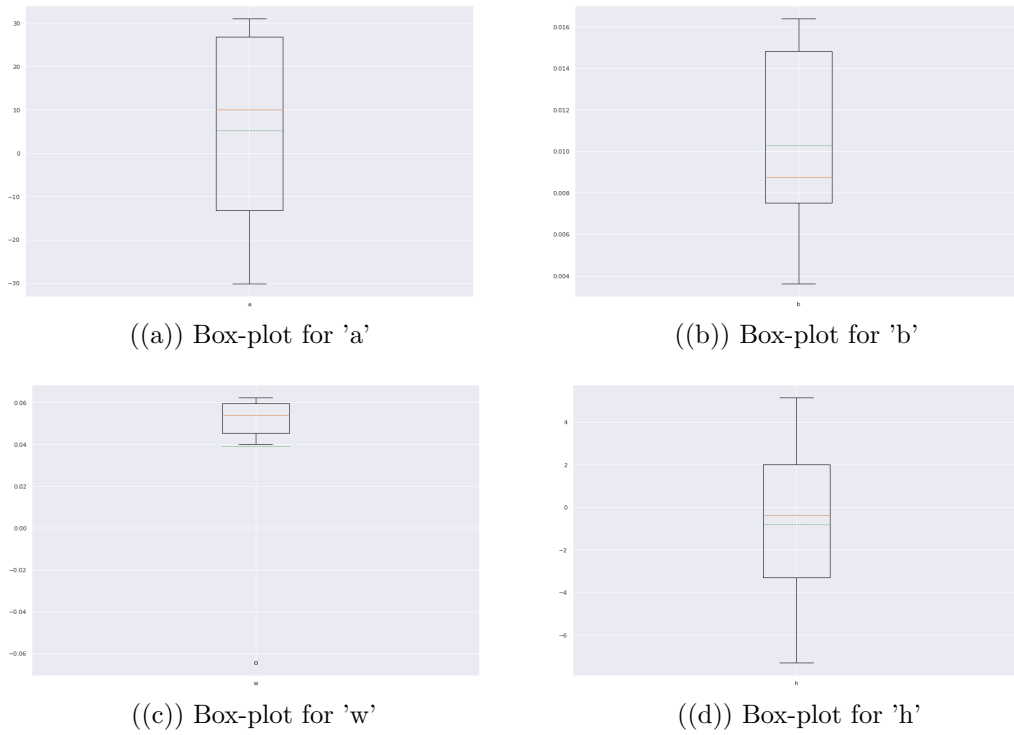


Figure 3.15: Box-plots for all parameters of case one (pre-diabetic) patient.

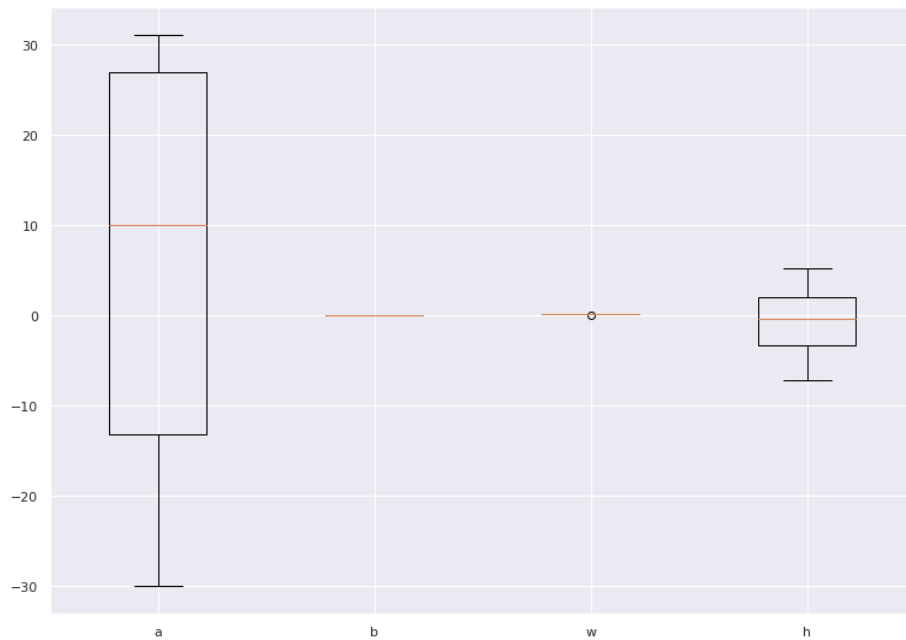


Figure 3.16: Box-plots for all the parameters combined in one scale for comparison of pre diabetic case one.

We will now use a **Box-plots** to analyse the parameters , for this diet they are plotted individually in figure 3.15 and combined in figure 3.16.

Finally, we will plot in figure 3.17 the continuous glucose values with the actual data points to see how well it fits a particular time range.

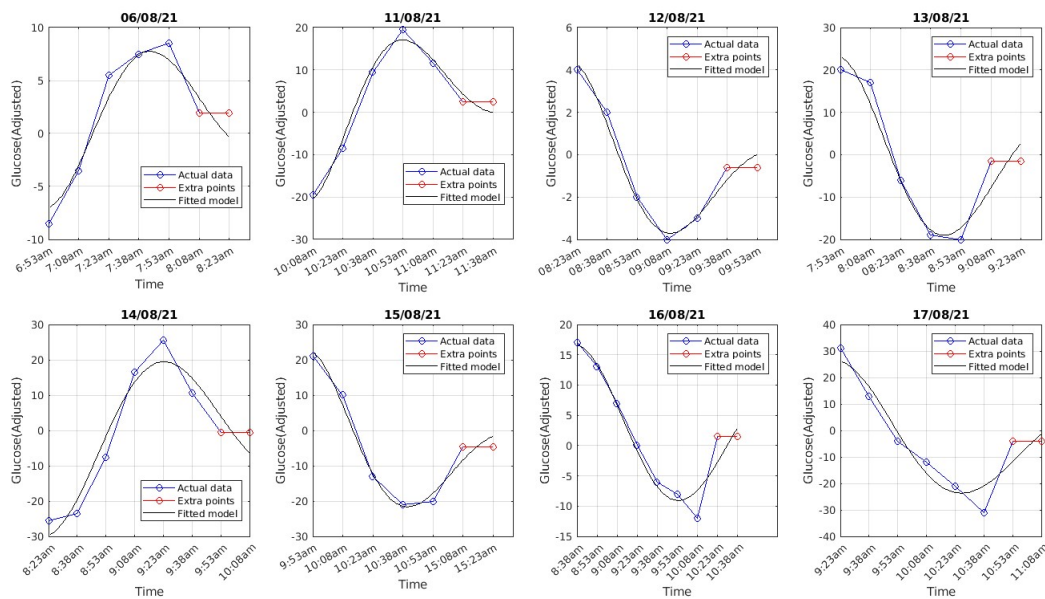


Figure 3.17: A date-wise depiction of the diet dairy’s actual data points (marked with blue) and interpolated points (marked red) overlaid with the black-fitted model applied to case one of pre-diabetic patient. From the above figure we can see that our proposed model generalizes well as most of the plot in black gives rsquare values above 90 percent.

3.3 Diabetic :

3.3.1 Diet four (Diabetic) :

Now, accounting for diabetes patients' diet and glucose data. We have data available from 15 July 21 to 29 July 21; we selected the eight best matches for the dates listed in the table below. Because not all dates include a liquid meal alone in the diet, we only included those dates with liquid alone and not with breakfast, hence the number of days we experience fits is lower than what was specified. Using the same approach again, we estimate the following parameters:

Date	a	b	h	w
17/07/21	10.000	0.003518	-4.4620	0.04662
18/07/21	8.873	0.004573	-2.4920	0.04079
19/07/21	2.500	0.005405	0.1031	0.06826
22/07/21	7.476	0.022060	5.0000	0.03768
23/07/21	7.244	0.007920	0.9821	0.04455
25/07/21	11.480	0.013560	2.0000	0.04379
27/07/21	1.354	0.000573	0.3441	0.05172
28/07/21	6.000	0.006439	0.6854	0.05055

The following statistical insights into our set of parameters are :

index	a	b	h	w
count	8.0	8.0	8.0	8.0
mean	6.865875	0.008006	0.270088	0.047995
std	3.502000	0.006811	2.830933	0.009418
min	1.354000	0.000573	-4.462000	0.037680
25th percentile	5.125000	0.004309	-0.545675	0.043040
50th percentile	7.360000	0.005922	0.514750	0.045585
75th percentile	9.154750	0.009330	1.236575	0.050842
max	11.480000	0.022060	5.000000	0.068260

The correlation graph is plotted as a heatmap in the 3.10:

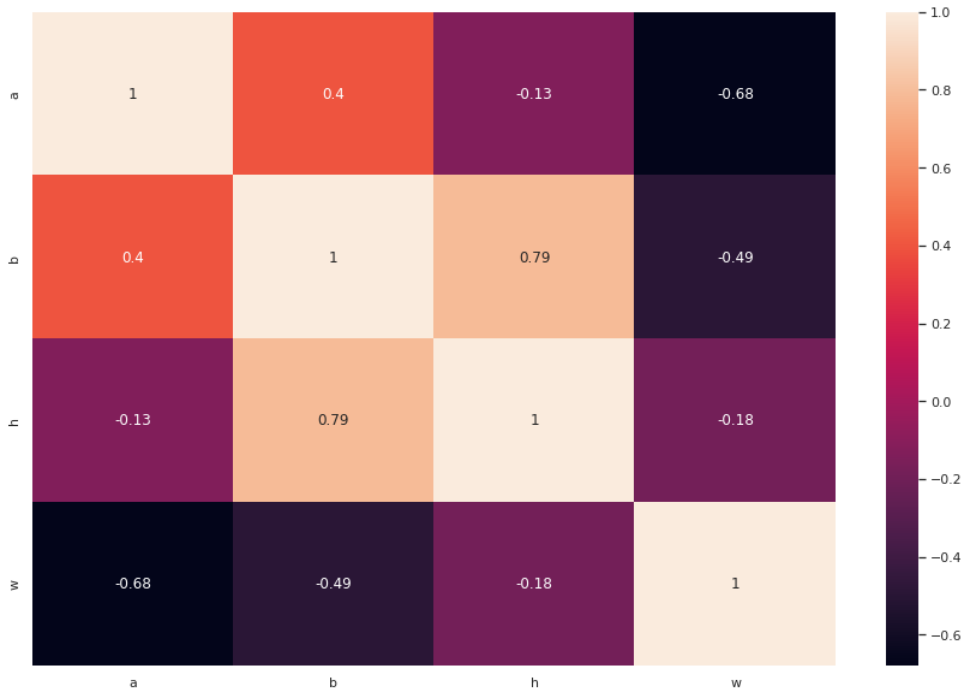


Figure 3.18: Heatmap of correlation between parameters for diet-one of a diabetic patient.

We will now use a **Box-plots** to analyse the parameters , for this diet they are plotted individually in figure 3.7 and combined in figure 3.8.

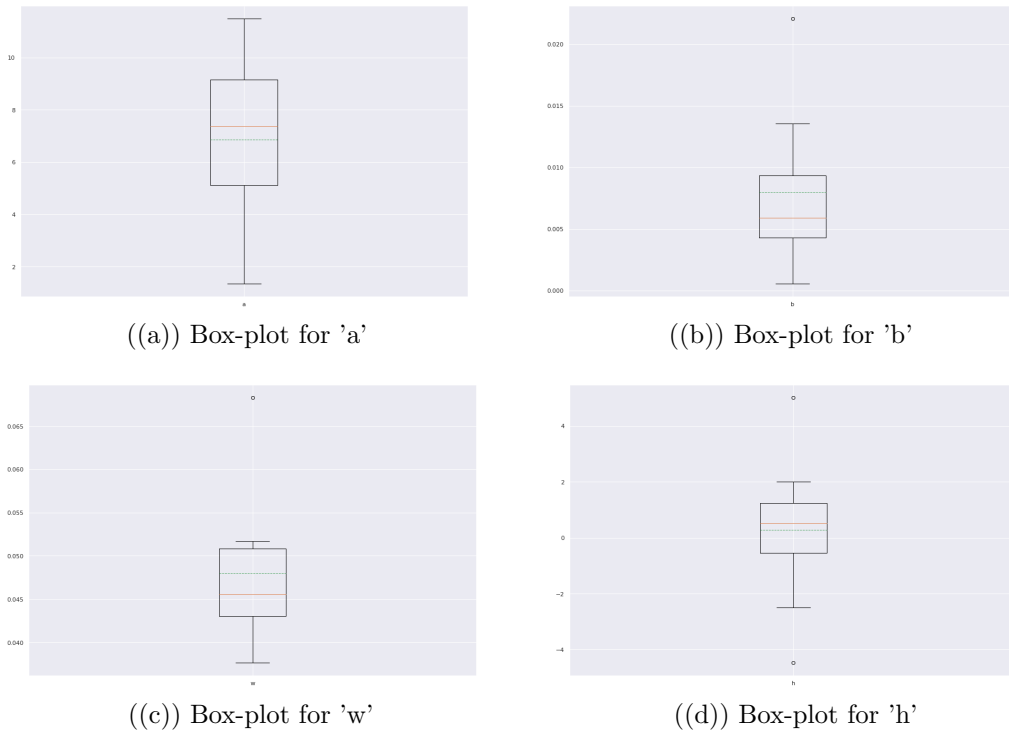


Figure 3.19: Box-plots for all parameters of case one in a diabetic patient data.

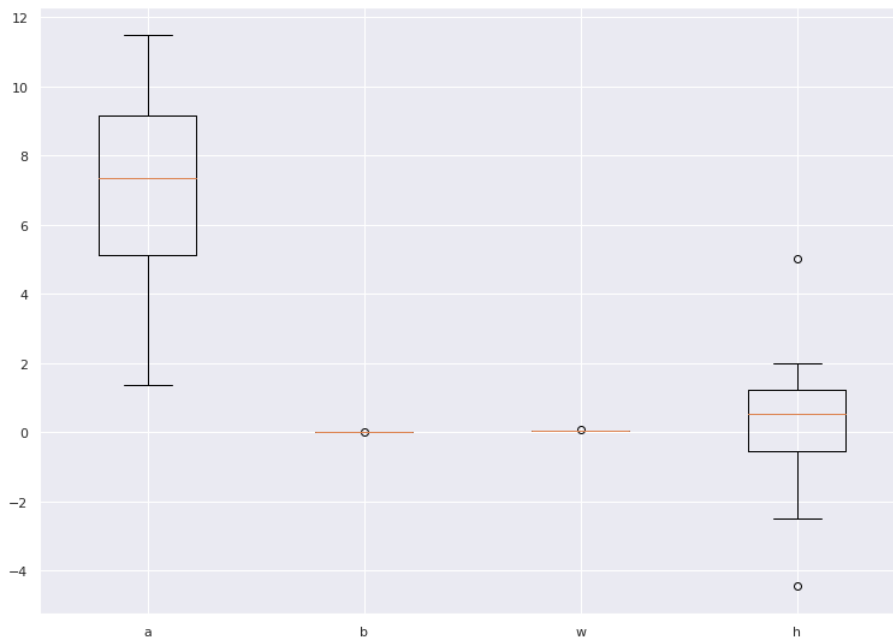


Figure 3.20: Box-plots for all the parameters of case one in diabetic patient data combined at one scale for comparison.

Finally, we will plot in figure 3.13 the continuous glucose values with the actual data points to see how well it fits a particular time range.

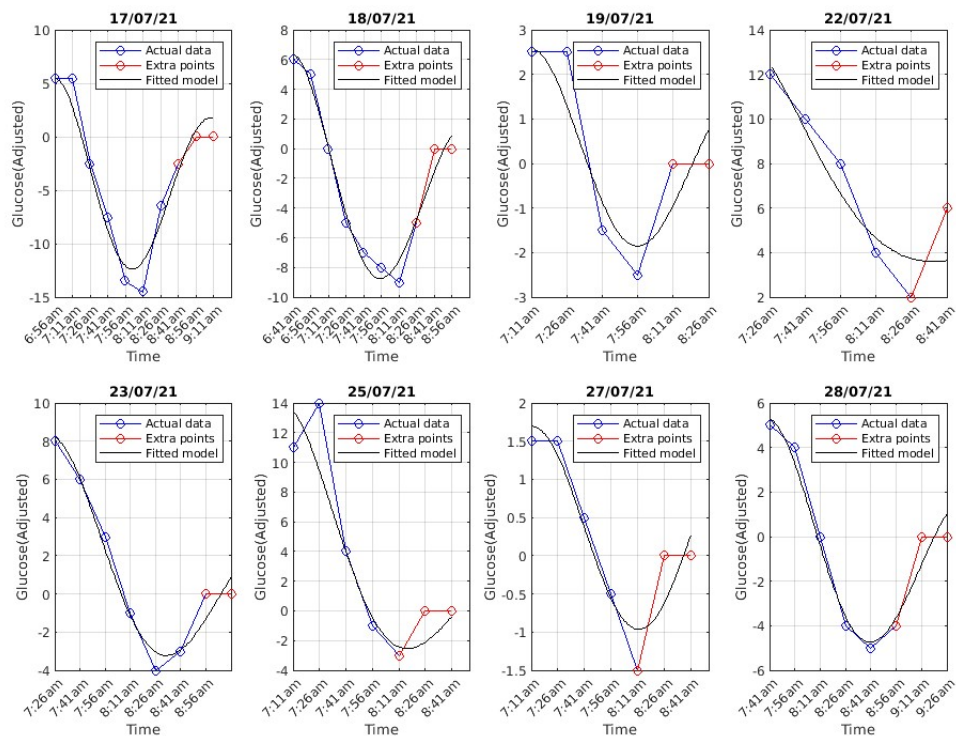


Figure 3.21: A date-wise depiction of the diet dairy's actual data points (marked with blue) and interpolated points (marked red) overlaid with the black-fitted model for case one (diabetic). From the above figure we can see that our proposed model generalizes well as most of the plot in black gives rsquare values above 90 percent.

3.3.2 Diet five (Diabetic) :

Here we analyse last diabetic data and with the same approach again estimated parameters are found as follows :

Date	a	b	h	w
15/07/21	8.759	0.026580	-5.3530	0.08188
16/07/21	-4.952	0.020030	0.0288	0.08146
17/07/21	9.000	0.017000	-1.5600	0.06770
18/07/21	7.968	0.011990	-8.3110	-0.05440
21/07/21	6.000	0.003025	-2.9750	0.04994
22/07/21	2.664	0.023150	-0.6066	0.09097
23/07/21	6.950	0.025580	-2.4050	0.06288
28/07/21	4.756	0.014600	-1.0000	0.06952

The following statistical insights into our set of parameters are :

index	a	b	h	w
count	8.0	8.0	8.0	8.0
mean	5.143125	0.017744	-2.772725	0.056244
std	4.600256	0.007875	2.794391	0.046495
min	-4.952000	0.003025	-8.311000	-0.054400
25th percentile	4.233000	0.013947	-3.569500	0.059645
50th percentile	6.475000	0.018515	-1.982500	0.068610
75th percentile	8.165750	0.023758	-0.901650	0.081565
max	9.000000	0.026580	0.028800	0.090970

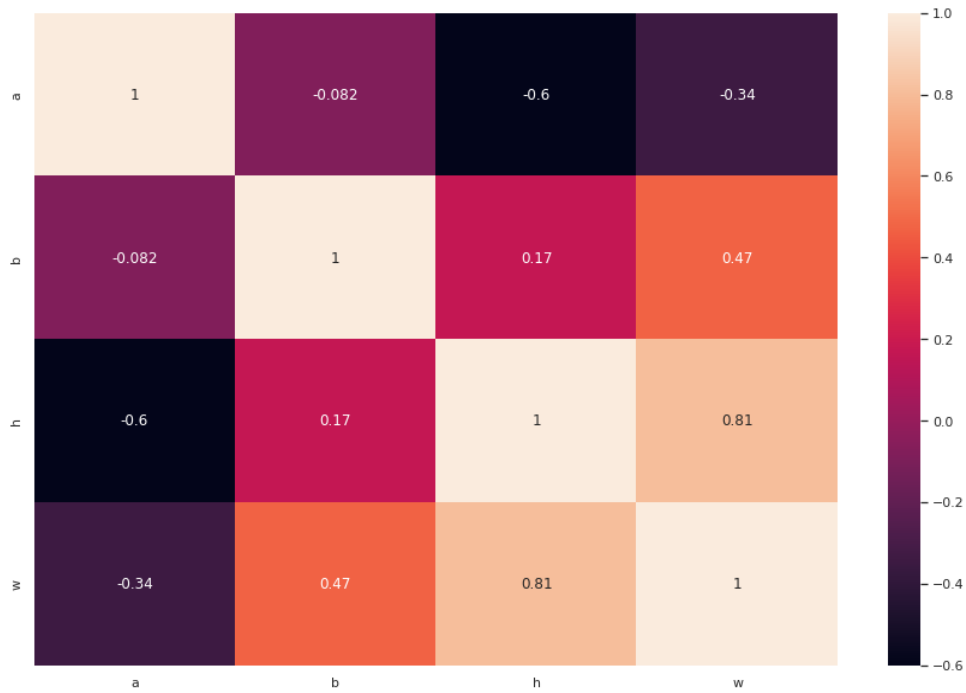


Figure 3.22: Heatmap of correlation between parameters for Diet two of a diabetic patient.

The correlation graph is plotted as a heatmap in the 3.10:

We will now use a **Box-plots** to analyse the parameters , for this diet they are plotted individually in figure 3.7 and combined in figure 3.8.

Finally, we will plot in figure 3.13 the continuous glucose values with the actual data points to see how well it fits a particular time range.

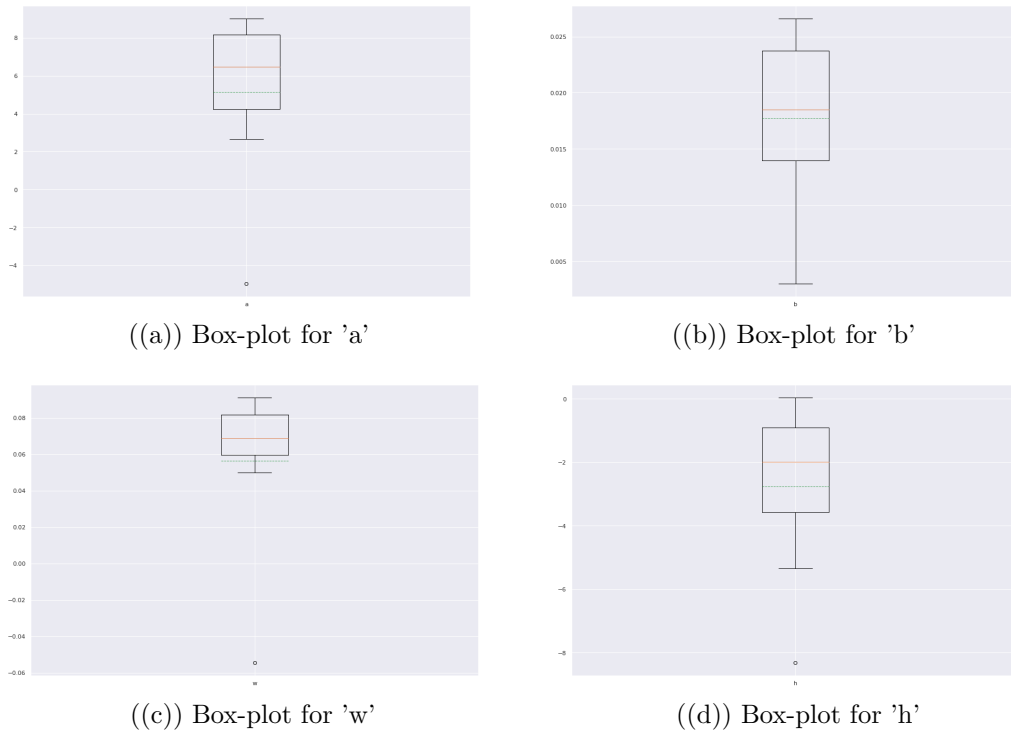


Figure 3.23: Box-plots for all parameters for diet one of a Diabetic patient.

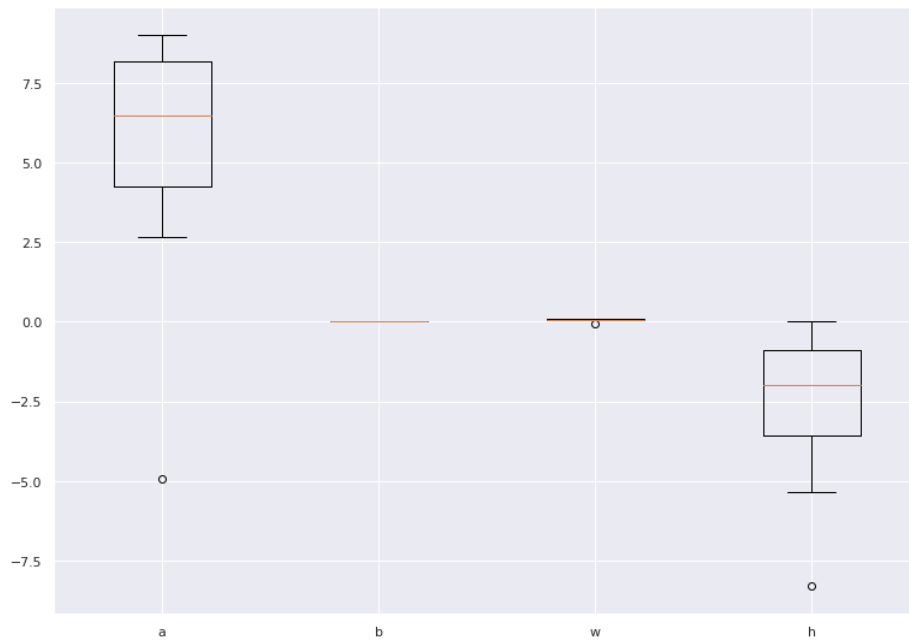


Figure 3.24: Box-plots for all the for diet two (diabetic) where parameters are combined in one scale for comparison.

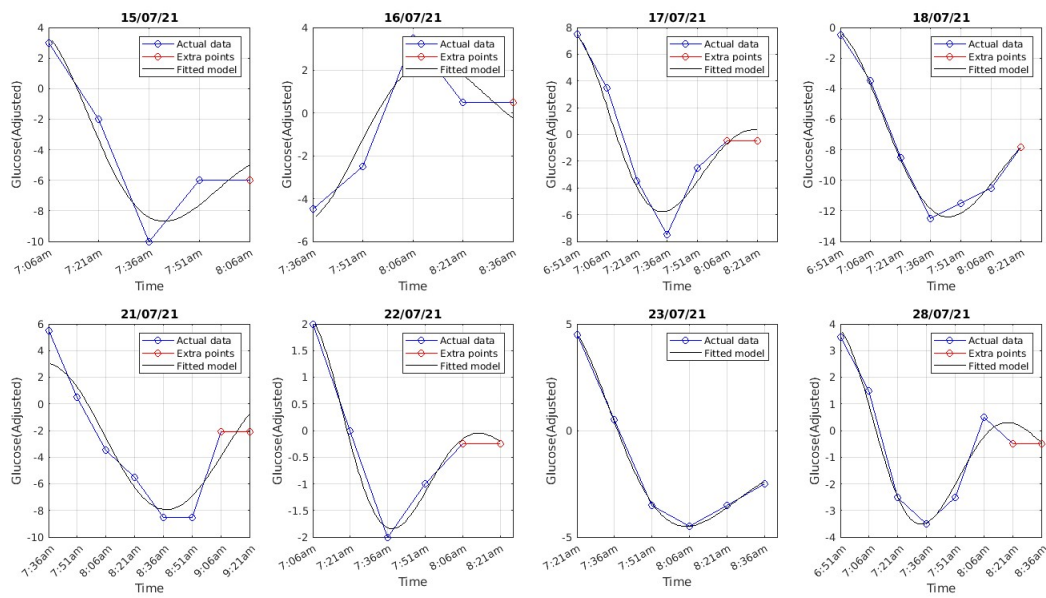


Figure 3.25: A date-wise depiction of the diet dairy's actual data points (marked with blue) and interpolated points (marked red) overlaid with the black-fitted model for diet two of diabetic patient. From the above figure we can see that our proposed model generalizes well as most of the plot in black gives rsquare values above 90 percent.

Chapter 4

Further Discussions and Conclusion

Our strategy focused on the spikes in glucose levels caused by liquid meals that were shown in the CGM data; however, we had previously examined other sparse modelling strategies for the spikes in glucose levels that were seen throughout the day. In the end, we picked the harmonic oscillator model and added an extra intercept term. We explored PySINDy library when perusing the majority of the non-linear sparse models, which was based on a study by [6] that takes time series data as the inputs for all of the states and then apply derivative to it for extracting governing equations from data. The powerful and ground-breaking technique known as SINDy (Sparse Identification of Non-Linear Dynamics) is based on sparse regression and compressed sensing [6]. SINDy is quite capable of identifying dynamics when measurements of every state are provided, although this is not always the case. It is challenging to find the governing equations in our situation of merely glucose, for instance. The identification of latent or partiable variables is always evolving, and if possible, it can be further applied in near future to the CGM data for superior results.

Other sparse approaches were investigated, with the ISTA algorithm serving as the primary algorithm [7]. It is a sophisticated iterative method that produces sparse variables than gradient descent. Additionally, it is a superior option for the optimization issue.

Conclusion

As we draw to a close, we have investigated a number of methodologies, both of whose success and failure might be used to enhance the development of CGM data models in the near future. The same kind of modelling can be investigated for other glucose peaks. We have approximated this case to the liquid meal time in the morning because the underdamped harmonic oscillator while asymptote to particular point with certain oscillations, the glucose data continues to spike with the effect of more food. However, if we expand the same approach throughout the day it will fail so other non-linear techniques can be combined with this study to explore further.

Chapter 5

Appendix

5.1 MATLAB code for tea time extraction, modelling and plotting

Code 5.1: Code applied to case one of a non-diabetic data set named nd04teatimeextraction.m

```
1 CGMND04tb = table2timetable(CGMND04);
2 CGMND04tb = removevars(CGMND04tb, 'ID');
3 CGMND04tb = removevars(CGMND04tb, 'RecordType');
4 CGMND04tb.Properties.VariableNames{1} = 'Glucose';
5 %ndtwo22 = CGMND04tb(timerange('2021-07-21 11:00', '2021-07-21
   14:11'), :);
6 %plot(ndtwo22.Time , ndtwo22.Glucose)
7 %extracting tea time days from data
8 ndtwo21 = CGMND04tb(timerange('2021-07-21 10:50', '2021-07-21
   13:00'), :);
9 ndtwo22 = CGMND04tb(timerange('2021-07-22 10:50', '2021-07-22
   13:00'), :);
10 ndtwo23 = CGMND04tb(timerange('2021-07-23 08:00', '2021-07-23
   10:00'), :);
11 ndtwo24 = CGMND04tb(timerange('2021-07-24 08:00', '2021-07-24
   10:00'), :);
12 ndtwo25 = CGMND04tb(timerange('2021-07-25 10:00', '2021-07-25
   11:30'), :);
13 ndtwo26 = CGMND04tb(timerange('2021-07-26 08:00', '2021-07-26
   10:00'), :);
14 ndtwo27 = CGMND04tb(timerange('2021-07-27 08:00', '2021-07-27
   10:00'), :);
```

```

15 ndtwo28 = CGMND04tb(timerange('2021-07-28 08:00', '2021-07-28
    10:00'), :);
16 ndtwo29 = CGMND04tb(timerange('2021-07-29 08:00', '2021-07-29
    10:00'), :);
17 ndtwo30 = CGMND04tb(timerange('2021-07-30 08:00', '2021-07-30
    09:30'), :);
18 ndtwo01 = CGMND04tb(timerange('2021-08-01 08:00', '2021-08-01
    10:00'), :);
19 ndtwo02 = CGMND04tb(timerange('2021-08-02 09:00', '2021-08-02
    11:00'), :);
20 %plotallglucose values
21 % plot(ndtwo21.Glucose)
22 % hold on
23 % plot(ndtwo22.Glucose)
24 % plot(ndtwo23.Glucose)
25 % plot(ndtwo24.Glucose)
26 % plot(ndtwo25.Glucose)
27 % plot(dtwo26.Glucose)
28 % plot(ndtwo27.Glucose)
29 % plot(ndtwo28.Glucose)
30 % plot(ndtwo29.Glucose)
31 % plot(ndtwo30.Glucose)
32 % plot(ndtwo01.Glucose)
33 % plot(ndtwo02.Glucose)
34 % plot(ndtwo03.Glucose)
35 %plot(ndtwo22.Time , ndtwo22.Glucose)
36 % hold on
37 % plot(ndtwo22.Time , ndtwo22.Glucose)
38 % plot(ndtwo22.Time , ndtwo22.Glucose)
39 % plot(ndtwo22.Time , ndtwo22.Glucose)
40 % plot(ndtwo19.Time , ndtwo19.Glucose)
41 % plont(dtwo20.Time n, dtwo20.Glucose)
42 % plot(ndtwo22.Time , ndtwo22.Glucose)
43 % plot(ndtwo22.Time , ndtwo22.Glucose)
44 % plot(ndtwo23.Time , ndtwo23.Glucose)
45 % plot(ndtwo24.Time , ndtwo24.Glucose)
46 % plot(ndtwo25.Time , ndtwo25.Glucose)
47 % plot(ndtwo26.Time , ndtwo26.Glucose)
48 % plot(ndtwo27.Time , ndtwo27.Glucose)
49 % plot(ndtwo22.Time , ndtwo22.Glucose)

```

```

50 n4h21 = max(ndtwo21.Glucose);
51 n4l21 = min(ndtwo21.Glucose);
52 m4_21 = (n4h21+n4l21)/2;
53 ndtwo21v = ndtwo21.Glucose - m4_21;
54 ndtwo21v = [ndtwo21v(4:end)];
55 ndtwo21v = [ndtwo21v' , mean(ndtwo21v) , mean(ndtwo21v) , mean(
    ndtwo21v)];
56 nt21 = 0:15:(length(ndtwo21v)-1)*15;
57 %plot(ndtwo21v)
58 n4h23 = max(ndtwo23.Glucose);
59 n4l23 = min(ndtwo23.Glucose);
60 m4_23 = (n4h23+n4l23)/2;
61 ndtwo23v = ndtwo23.Glucose - m4_23;
62 ndtwo23v = [ndtwo23v(4:end)];
63 ndtwo23v = [ndtwo23v' , mean(ndtwo23v) , mean(ndtwo23v)];
64 nt23 = 0:15:(length(ndtwo23v)-1)*15;
65 %plot(ndtwo23v)
66 n4h25 = max(ndtwo25.Glucose);
67 n4l25 = min(ndtwo25.Glucose);
68 m4_25 = (n4h25+n4l25)/2;
69 ndtwo25v = ndtwo25.Glucose - m4_25;
70 ndtwo25v = [ndtwo25v(3:end)];
71 ndtwo25v = [ndtwo25v' , mean(ndtwo25v)];
72 nt25 = 0:15:(length(ndtwo25v)-1)*15;
73 %plot(ndtwo25v)
74 n4h26 = max(ndtwo26.Glucose);
75 n4l26 = min(ndtwo26.Glucose);
76 m4_26 = (n4h26+n4l26)/2;
77 ndtwo26v = ndtwo26.Glucose - m4_26;
78 ndtwo26v = [ndtwo26v(5:end-1)];
79 ndtwo26v = [ndtwo26v' , mean(ndtwo26v) , mean(ndtwo26v)];
80 nt26 = 0:15:(length(ndtwo26v)-1)*15;
81 %plot(ndtwo26v)
82 n4h27 = max(ndtwo27.Glucose);
83 n4l27 = min(ndtwo27.Glucose);
84 m4_27 = (n4h27+n4l27)/2;
85 ndtwo27v = ndtwo27.Glucose - m4_27;
86 ndtwo27v = [ndtwo27v(3:end)];
87 ndtwo27v = [ndtwo27v' , mean(ndtwo27v) , mean(ndtwo27v)];
88 nt27 = 0:15:(length(ndtwo27v)-1)*15;

```

```

89 %plot(ndtwo27v)
90 n4h28 = max(ndtwo28.Glucose);
91 n4l28 = min(ndtwo28.Glucose);
92 m4_28 = (n4h28+n4l28)/2;
93 ndtwo28v = ndtwo28.Glucose - m4_28;
94 ndtwo28v = [ndtwo28v(2:6)];
95 ndtwo28v = [ndtwo28v' , mean(ndtwo28v) , mean(ndtwo28v)];
96 nt28 = 0:15:(length(ndtwo28v)-1)*15;
97 %plot(ndtwo28v)
98 n4h29 = max(ndtwo29.Glucose);
99 n4l29 = min(ndtwo29.Glucose);
100 m4_29 = (n4h29+n4l29)/2;
101 ndtwo29v = ndtwo29.Glucose - m4_29;
102 ndtwo29v = [ndtwo29v(1:6)];
103 ndtwo29v = [ndtwo29v' , mean(ndtwo29v) , mean(ndtwo29v)];
104 nt29 = 0:15:(length(ndtwo29v)-1)*15;
105 %plot(ndtwo29v)
106 n4h30 = max(ndtwo30.Glucose);
107 n4l30 = min(ndtwo30.Glucose);
108 m4_30 = (n4h30+n4l30)/2;
109 ndtwo30v = ndtwo30.Glucose - m4_30;
110 ndtwo30v = [ndtwo30v(2:5)];
111 ndtwo30v = [ndtwo30v' , mean(ndtwo30v) , mean(ndtwo30v)];
112 nt30 = 0:15:(length(ndtwo30v)-1)*15;
113 %plot(ndtwo30v)
114 %plot after curve fitting
115 %% july21
116 a = -20; %(fixed at bound)
117 b = 0.007968; %(-0.003895, 0.01983)
118 h = -0.131; %(-6.02, 5.758)
119 w = 0.04872; %(0.03692, 0.06052)
120 nt21_cf = linspace(1,nt21(end));
121 y15 = a*exp(-b*nt21_cf).*cos(w*nt21_cf)+h;
122 subplot(2,4,1)
123 plot(nt21(1:end-2) , ndtwo21v(1:end-2) , '-o' , 'Color','blue')
124 hold on
125 plot(nt21(end-2:end), ndtwo21v(end-2:end) , '-o' , 'Color', 'red'
    )
126 plot(nt21_cf , y15 , 'Color','black')
127 xticks(0:15:(length(nt21) - 1)*15)

```

```

128 xticklabels({'11:36am' '11:51am' '12:06pm' '12:21pm' '12:36pm' '
    12:51pm' '13:06pm' '13:21pm' '13:36pm'})
129 hold off
130 %legend , label and title
131 ylabel('Glucose(Adjusted)')
132 xlabel('Time')
133 legend('Actual data','Extra points', 'Fitted model')
134 title('21/07/21')
135 grid on
136 %% july23
137 a = -1.388 ; % (-2.003, -0.7727)
138 b = 0.0179 ; % (0.000661, 0.03514)
139 h = 0.3639 ; % (0.1207, 0.6071)
140 w = 0.0843 ; % (0.07046, 0.09813)
141 nt23_cf = linspace(1,nt23(end));
142 n4y23 = a*exp(-b*nt23_cf).*cos(w*nt23_cf)+h;
143 subplot(2,4,2)
144 plot(nt23(1:end-1) , ndtwo23v(1:end-1) , '-o', 'Color','blue')
145 hold on
146 plot(nt23(end-1:end), ndtwo23v(end-1:end) , '-o' , 'Color', 'red'
    )
147 plot(nt23_cf , n4y23 , 'Color','black')
148 xticks(0:15:(length(nt23) - 1)*15)
149 xticklabels({'8:51am' '9:06am' '9:21am' '9:36am' '9:51am' '10:06
    am' '10:21am'})
150 hold off
151 %legend , label and title
152 ylabel('Glucose(Adjusted)')
153 xlabel('Time')
154 legend('Actual data','Extra points', 'Fitted model')
155 title('23/07/21')
156 grid on
157 %% july25
158 a = 1.483 ; % (-1.549, 4.515)
159 b = 0.02152 ; % (-0.09361, 0.1366)
160 h = 0.0337 ; %(-1.361, 1.428)
161 w = 0.1277 ;%(0.04161, 0.2138)
162 nt25_cf = linspace(1,nt25(end));
163 n4y25 = a*exp(-b*nt25_cf).*cos(w*nt25_cf)+h;
164 subplot(2,4,3)

```

```

165 plot(nt25(1:end) , ndtwo25v(1:end) , '-o', 'Color','blue')
166 hold on
167 plot(nt25(end:end), ndtwo25v(end:end) , '-o' , 'Color', 'red')
168 plot(nt25_cf , n4y25 , 'Color','black')
169 xticks(0:15:(length(nt25) - 1)*15)
170 xticklabels({'10:36am' '10:51am' '11:06am' '11:21am' '11:36am' '
12:51pm' '13:06pm' '13:21pm' '13:36pm'})
171 hold off
172 %legend , label and title
173 ylabel('Glucose(Adjusted)')
174 xlabel('Time')
175 legend('Actual data','Extra points', 'Fitted model')
176 title('25/07/21')
177 grid on
178 %% july26
179 a = 1.713;% (-9.215, 12.64)
180 b = 0.01779;% (-0.2669, 0.2926)
181 h = -1.085;% (-6.191, 4.02)
182 w = 0.102499 ;% (-0.1694, 0.3743)
183 nt26_cf = linspace(1,nt26(end));
184 n4y26 = a*exp(-b*nt26_cf).*cos(w*nt26_cf)+h;
185 subplot(2,4,4)
186 plot(nt26(1:end-1) , ndtwo26v(1:end-1) , '-o', 'Color','blue')
187 hold on
188 plot(nt26(end-1:end), ndtwo26v(end-1:end) , '-o' , 'Color', 'red'
)
189 plot(nt26_cf , n4y26 , 'Color','black')
190 xticks(0:15:(length(nt26) - 1)*15)
191 xticklabels({'9:06am' '9:21am' '9:36am' '9:51am' '10:06am' '10:21
am'})
192 hold off
193 %legend , label and title
194 ylabel('Glucose(Adjusted)')
195 xlabel('Time')
196 legend('Actual data','Extra points', 'Fitted model')
197 title('26/07/21')
198 grid on
199 %% july27
200 a = 2 ;% (fixed at bound)
201 b = 0.01329 ;% (-0.005131, 0.03172)

```



```

202 h =      -0.2785 ;% (-0.8508, 0.2938)
203 w =      0.1008 ;% (0.08363, 0.1179)
204 nt27_cf = linspace(1,nt27(end));
205 n4y27 = a*exp(-b*nt27_cf).*cos(w*nt27_cf)+h;
206 subplot(2,4,5)
207 plot(nt27(1:end-1) , ndtwo27v(1:end-1) , '-o', 'Color','blue')
208 hold on
209 plot(nt27(end-1:end), ndtwo27v(end-1:end) , '-o' , 'Color', 'red'
    )
210 plot(nt27_cf , n4y27 , 'Color','black')
211 xticks(0:15:(length(nt27) - 1)*15)
212 xticklabels({'8:36am' '8:51am' '9:06am' '9:21am' '9:36am' '9:51am'
    '10:06am' '10:21am'})
213 hold off
214 %legend , label and title
215 ylabel('Glucose(Adjusted)')
216 xlabel('Time')
217 legend('Actual data','Extra points', 'Fitted model')
218 title('27/07/21')
219 grid on
220 %% july28
221 a =      2.569;% (1.581, 3.557)
222 b =      0.02122;% (0.003871, 0.03856)
223 h =     -0.4837;% (-0.8717, -0.09569)
224 w =      0.0895 ;% (0.07537, 0.1036)
225 nt28_cf = linspace(1,nt28(end));
226 n4y28 = a*exp(-b*nt28_cf).*cos(w*nt28_cf)+h;
227 subplot(2,4,6)
228 plot(nt28(1:end-1) , ndtwo28v(1:end-1) , '-o', 'Color','blue')
229 hold on
230 plot(nt28(end-1:end), ndtwo28v(end-1:end) , '-o' , 'Color', 'red'
    )
231 plot(nt28_cf , n4y28 , 'Color','black')
232 xticks(0:15:(length(nt28) - 1)*15)
233 xticklabels({'8:21am' '8:36am' '8:51am' '9:06am' '9:21am' '9:36am'
    '9:51am' '10:06am' '10:21am'})
234 hold off
235 %legend , label and title
236 ylabel('Glucose(Adjusted)')
237 xlabel('Time')

```

```

238 legend('Actual data','Extra points', 'Fitted model')
239 title('28/07/21')
240 grid on
241 %% july29
242 a =          3 ;% (fixed at bound)
243 b =    0.009693 ;% (-0.001457, 0.02084)
244 h =    0.3029 ;% (-0.3766, 0.9824)
245 w =    0.07043 ;% (0.0596, 0.08126)
246 nt29_cf = linspace(1,nt29(end));
247 n4y29 = a*exp(-b*nt29_cf).*cos(w*nt29_cf)+h;
248 subplot(2,4,7)
249 plot(nt29(1:end-1) , ndtwo29v(1:end-1) , '-o', 'Color','blue')
250 hold on
251 plot(nt29(end-1:end), ndtwo29v(end-1:end) , '-o' , 'Color', 'red'
      )
252 plot(nt29_cf , n4y29 , 'Color','black')
253 xticks(0:15:(length(nt29) - 1)*15)
254 xticklabels({'8:06am' '8:21am' '8:36am' '8:51am' '9:06am' '9:21am'
      '9:36am' '9:51am' '10:06am' '10:21am'})
255 hold off
256 %legend , label and title
257 ylabel('Glucose(Adjusted)')
258 xlabel('Time')
259 legend('Actual data','Extra points', 'Fitted model')
260 title('29/07/21')
261 grid on
262 %% july30
263 a =    2.37 ;% (-0.6159, 5.357)
264 b =    0.01327 ;% (-0.0357, 0.06224)
265 h =   -4.214 ;% (-5.592, -2.836)
266 w =  -0.07268 ;% (-0.1061, -0.03923)
267 nt30_cf = linspace(1,nt30(end));
268 n4y30 = a*exp(-b*nt30_cf).*cos(w*nt30_cf)+h;
269 subplot(2,4,8)
270 plot(nt30(1:end-1) , ndtwo30v(1:end-1) , '-o', 'Color','blue')
271 hold on
272 plot(nt30(end-1:end), ndtwo30v(end-1:end) , '-o' , 'Color', 'red'
      )
273 plot(nt30_cf , n4y30 , 'Color','black')
274 xticks(0:15:(length(nt30) - 1)*15)

```

```

275 xticklabels({'8:21am' '8:36am' '8:51am' '9:06am' '9:21am' '9:36am'
    '9:51am' '10:06am' '10:21am'})
276 hold off
277 %legend , label and title
278 ylabel('Glucose(Adjusted)')
279 xlabel('Time')
280 legend('Actual data','Extra points', 'Fitted model')
281 title('30/07/21')
282 grid on

```

5.2 Python code for parameters visualisation and analysis for diet one of diabetic case.

Code 5.2: Python code applied to case one (diabetic) data.

```

1 import numpy as np
2 import pandas as pd
3 #import pysindy as ps
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 %matplotlib inline
7 from scipy.optimize import curve_fit
8
9 import plotly.express as px
10 from scipy.integrate import odeint
11 from scipy.integrate import solve_ivp
12 from sklearn.metrics import mean_squared_error
13 import warnings
14 from scipy.integrate.odepack import ODEintWarning
15 warnings.filterwarnings("ignore", category=ODEintWarning)
16 %config Completer.use_jedi=False
17
18 #Reading data and plotting
19 df = pd.read_csv("Saroj.txt" , sep='\t', header=1, parse_dates= [
    "Time"], usecols=[1,3])
20 df.columns = ['Time', 'Glucose']
21 df
22 sns.set(rc={'figure.figsize':(15,10)})
23 q = sns.lineplot(data = df , x = "Time" , y = "Glucose")

```

```

24 q.set_xlabel("Time", fontsize = 20)
25 q.set_ylabel("Glucose", fontsize = 20)
26
27
28
29 #creating a pandas dataframe for parameters and converting into
    csv.
30
31 data_nd04 = {
32 'a' : [-20, -1.388, 1.483, 1.713, 2, 2.569, 3, 2.37],
33 'b' :
    [0.007968, 0.0179, 0.02152, 0.01779, 0.01329, 0.02122, 0.009693, 0.01327],
34 'h' :
    [-0.131, 0.3639, 0.0337, -1.085, -0.2785, -0.4837, 0.3029, -4.214],
35 'w' :
    [0.04872, 0.0843, 0.1277, 0.102499, 0.1008, 0.0895, 0.07043, -0.07268]
36
37 }
38 # Create the pandas DataFrame with column name is provided
    explicitly
39 parameters_nd04 = pd.DataFrame(data_nd04)
40 parameters_nd04.to_csv('pd_nd04.csv')
41
42 parameters_nd04.describe()
43
44 #create a correlation matrix
45 cor = parameters_nd04.corr() #calculating correlation
46 sns.heatmap(cor , xticklabels = cor.columns , yticklabels = cor.
    columns , annot = True)
47
48 #boxplots
49 plt.boxplot(parameters_nd04.a , meanline= True , showmeans = True
    , labels = 'a')
50
51 plt.boxplot(parameters_nd04.b , meanline= True , showmeans = True
    , labels = 'b')
52

```

```
53 plt.boxplot(parameters_nd04.w ,meanline= True , showmeans = True
    , labels = 'w')
54
55 plt.boxplot(parameters_nd04.h , meanline= True , showmeans = True
    , labels = 'h')
56
57 fig = plt.figure(figsize =(10, 7))
58 data = [parameters_nd04.a , parameters_nd04.b , parameters_nd04.w
    , parameters_nd04.h]
59
60 # First we create an axes instance
61 ax = fig.add_axes([0, 0, 1, 1])
62 # Combining all box-plot
63 bp = ax.boxplot(data , labels = ('a', 'b', 'w' , 'h'))
64 # show plot
65 plt.show()
```

Bibliography

- [1] Pranay Goel et al. “A minimal model approach for analyzing continuous glucose monitoring in type 2 diabetes”. In: *Frontiers in physiology* 9 (2018), p. 673.
- [2] Scott M Pappada et al. “Neural network-based real-time prediction of glucose in patients with insulin-dependent diabetes”. In: *Diabetes technology & therapeutics* 13.2 (2011), pp. 135–141.
- [3] DURGA PARKHI. “Derivative free Optimization to model Continuous Glucose Monitoring in Type 2 Diabetes”. PhD thesis. 2019.
- [4] Irina Gaynanova, Naresh Punjabi, and Ciprian Crainiceanu. “Modeling continuous glucose monitoring (CGM) data during sleep”. In: *Biostatistics* 23.1 (2022), pp. 223–239.
- [5] Louis Monnier et al. “Magnitude of the dawn phenomenon and its impact on the overall glucose exposure in type 2 diabetes: is this of concern?” In: *Diabetes care* 36.12 (2013), pp. 4057–4062.
- [6] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the national academy of sciences* 113.15 (2016), pp. 3932–3937.
- [7] Zheng Zhang et al. “A survey of sparse representation: algorithms and applications”. In: *IEEE access* 3 (2015), pp. 490–530.