Continuous Glucose Monitoring Prediction Meal Detection

Prakhar Praveen
Arizona State University
ppraveen@asu.edu

Danish Pasricha
Arizona State University
dpasrich@asu.edu

Aayush Sharma
Arizona State University
ashar313@asu.edu

Tanuj Singh
Arizona State University
tsingh29@asu.edu

Meghana Sukhavasi
Arizona State University
msukhava@asu.edu

Abstract—Diabetes is becoming a widespread and concerning public health issue that affects more people than ever before due to lifestyle problems in the modern world. People suffering from diabetes thus need to monitor their blood glucose concentration, both before a meal (fasting) and after a meal (postprandial). Our project is to develop an intelligent system to detect meals and predict glucose concentrations, which is trained on a comprehensive dataset of continuous readings and accurate readings using medical-grade sensors and gauging device, called Continuous Glucose Monitoring system. Diabetes is a chronic medical condition that occurs when either the body doesn't produce enough insulin (hyperglycemia) or produces too much insulin (hypoglycemia). The most common type of diabetes is Type-1 diabetes that causes hyperglycemia. To develop a smart meal detection and prediction of glucose concentration following meal detection in the blood that is crucial because abnormal glucose concentration can be fatal, we are using tools, namely SARIMA, Kalman, and RNN GRU Meal Detection, and LSTM.

I. INTRODUCTION

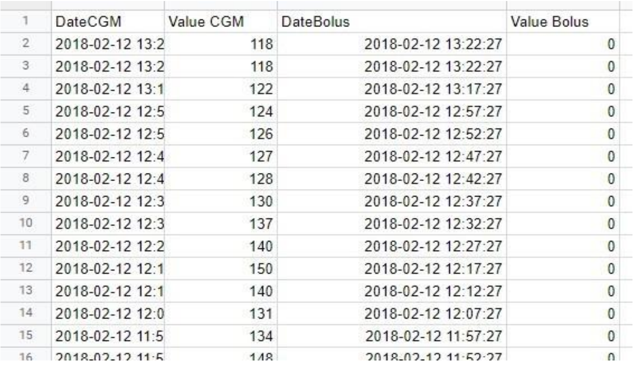
For people with diabetes, we are implementing an AI system that performs meal detection continuously, given 6-month long CGM data for each person. More specifically, we are developing efficient algorithms for online detection of meal event. The data outputted by CGM is used in this project to construct the aforesaid system, and to achieve that we develop four algorithms, which are SARIMA, LSTM, Kalman Filter, and RNN.

II. DATA PROCESSING

The dataset provided to us is in .mat format that is produced by CGM sensors. The data is parsed as well as meal ground truth is synchronized with the CGM data. In order to perform the synchronization, the timestamp from the CGM data is matched to the closest timestamp from the ground truth data. FinalData.csv contains this information.

Each column represents one of the following:

- DateCGM: Time/Date of measurement
- CGM: Glucose concentration at exact time/date (timestamp)
- DateBolus: Time of Bolus Delivery
- Value Bolus: Bolus is the measure of insulin administered



1	DateCGM	Value CGM	DateBolus	Value Bolus
2	2018-02-12 13:2	118	2018-02-12 13:22:27	0
3	2018-02-12 13:2	118	2018-02-12 13:22:27	0
4	2018-02-12 13:1	122	2018-02-12 13:17:27	0
5	2018-02-12 12:5	124	2018-02-12 12:57:27	0
6	2018-02-12 12:5	126	2018-02-12 12:52:27	0
7	2018-02-12 12:4	127	2018-02-12 12:47:27	0
8	2018-02-12 12:4	128	2018-02-12 12:42:27	0
9	2018-02-12 12:3	130	2018-02-12 12:37:27	0
10	2018-02-12 12:3	137	2018-02-12 12:32:27	0
11	2018-02-12 12:2	140	2018-02-12 12:27:27	0
12	2018-02-12 12:1	150	2018-02-12 12:17:27	0
13	2018-02-12 12:1	140	2018-02-12 12:12:27	0
14	2018-02-12 12:0	131	2018-02-12 12:07:27	0
15	2018-02-12 11:5	134	2018-02-12 11:57:27	0
16	2018-02-12 11:5	148	2018-02-12 11:52:27	0

Figure 1. A window of CGM data

A. SARIMA

The acronym stands for Seasonal Autoreggressive Integrated Moving Average. This model was used to get the time series data's seasonality. Because this model works effectively given continuous data, the data trends are captured in our case, where data is almost continuous and huge. Meal data points were compared with the predicted values of the SARIMA model using an empirically set threshold to classify a meal event into a meal or a no-meal for each timestamp.

With the help of Akaike Information Criterion for optimization of the SARIMA model performance, we were able to adjust hyper-parameters for predicting the trend that the SARIMA model takes works with as follows—auto regression order, denoted by p; difference order, denoted by d; moving average order, denoted by (p)(q). For detecting and generalizing seasonality, it has Seasonal auto regression order, denoted by P; seasonal moving average order, denoted by Q; as well as the number of time periods in a season, denoted by m. Hyper-parameter values were determined using the Auto Correlation Plot, which tells us how the current value relates to preceding values in the time series; and the Partial Auto-Correlation Plot helps us finout the lag between the current and next values. Figures 2 and 3 show us what values we used to adjust the hyper-parameters to increase the model's accuracy. One with the least AIC score is that corresponding to the best accuracy.

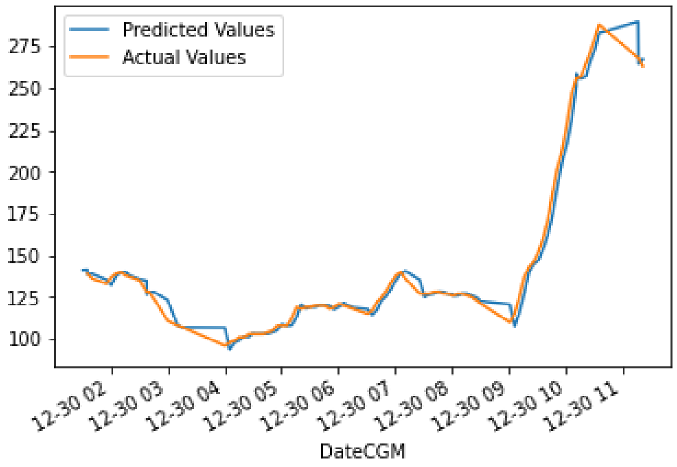


Figure 2. Actual v/s Predicted CGM values

The SARIMA model was run on the test data after selecting optimal hyper-parameters, thus producing predictions accurately. Like we observe in Figure 4, our model achieves close to perfect accuracy. Testing and Verification of our model was done with the help of few great metrics, like Mean Squared Error, and Root Mean Squared Error, as seen in Table 1. Here we find that the MSE value is pretty low, implying our model produces very few errors, and that is it is a great model.

We again used a threshold value to classify a meal or no meal event upon receiving predicted values. We utilized a method that calculates the difference between predicted values and the threshold to do a successful classification of a meal event.

B. Kalman Filter

The Kalman filter was helpful in determining unknown variables, as it leverages an algorithm that takes in a hige amount of data (long time series), and reduces noise to produce good results for predictions. We used an unscented Kalman Filter for meal/no-meal classification and detection. Following equations are used to predict our values:

$$G_s(K+1) = \\ h[p_1(k)(G_b(k) + \frac{G_s(k)}{hp_1(k)} - G_s(k)) - (k) + R_a(k)]$$

Non-linear space model equation:

$$x(k+1) = f(x(k)) + w(k) \\ y(k) = g(x(k)) + v(k)$$

Measure noise is represented by v, the process is represented by w(k), and the state vector is represented by x(k) (k). There are yet additional variables that are not time dependent, including V, p1, p2, p3, and p5. However, these factors must be viewed as time fluctuations because there is intra- and inter-subject variability.

The equation served as the basis for the machine tuning parameters. The graph shows the discrepancy between the real and predicted glucose concentrations, and the Kalman filter provides us with a 95 % accuracy.

C. Recurrent Neural Network

Recurring Neural Networks are recurrent in nature, as their name suggests. The same function will be used to create the output from the input. The output is created, copied, and then delivered back into the recurrent network. It takes into account both the current input and the output that it has learned from the prior input when making a decision [9]. We employed the RNN's long-term short memory model to detect meals. We chose LSTM because it processes, categorizes, and predicts time series data effectively. Additionally, it addresses the RNN model flaw known as back propagation, also known as the vanishing gradient problem. Before training the data, we need to talk about the data processing that we performed before model training.

To ensure that the model is not biased, we utilized MaxMinScaler to normalize the data, converting it into values of 0 and 1. We divided the data into a 9:1 ratio, meaning that 90% of the data is used for training and 10% is utilized for testing. For training, we are used a batch of 32 and 110 epochs.

The model we used is composed of four layers.

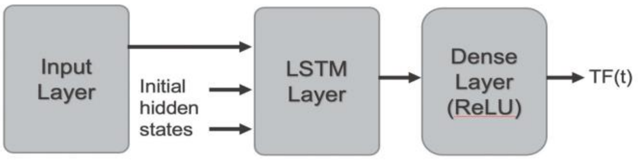


Fig. 7 LSTM model

Figure 2 shows the difference between the actual glucose values and the predicted glucose values. The figure on the right shows how close the predicted values are to the actual values. The same thresholding technique was used as we did for SARIMA and Kalman Filter. After training the model, we achieved a 75% accuracy with the LSTM.

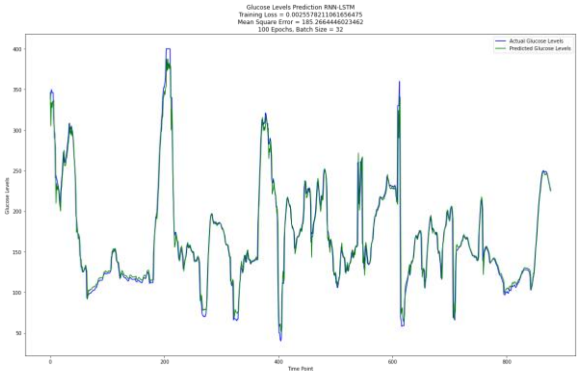


Fig 8 Predicted vs Actual Glucose Concentrations

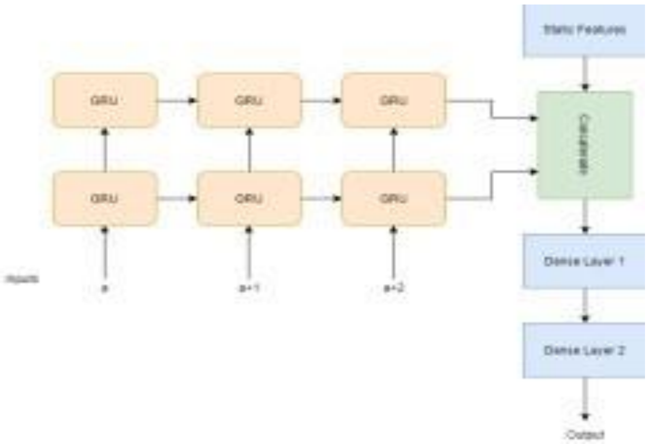


Fig 9 GRU

D. GRU

As mentioned earlier, I used the defined static properties to create a stack GRU with an embedded layer that accepts these static properties as input. The time-shifted data is delivered directly to the GRU, which learns the function itself, so the GRU does not need to extract the function as it does in a typical machine learning system. Figure 9 shows a GRU that encodes specific time series data. The output from the GRU model is concatenated to predict future step values in Figure 10 below.

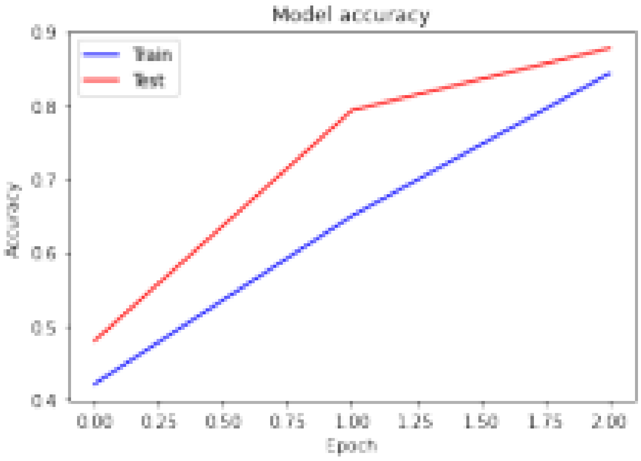


Fig 10 LSTM training Accuracy versus Epoch plot

Figure 7 shows how this module works. The Meal Prediction Module accepts CGM readings from hours 1 to t as input, as well as labels labeled "Meal" or "No Meal" along with the CGM readings. CGM values are organized by date and time in this table. I also tried to analyze trends in blood glucose levels over time. The LSTM CGM value prediction model attempts to predict the CGM value from time t to time t + hour after the input is received. These CGM readings are provided as inputs to the LSTM dietary prediction model. SARIMA used the LSTM meal prediction model to predict when users would eat their next meal.

This module attempts to predict the time/date a user would have his/her next meal. At any given time t + k, if the user eats a meal such that $t \leq t + k \leq t + \alpha$, it says the user had a meal. If not, we do not anticipate the diet. Note that α here is just a hyper-parameter that can be changed according to the needs of the problem. Figure 8 gives an overview of how this module works. The Meal Prediction Module takes CGM values from hours 1 to t as input and is labeled "Meal" or "No Meal" as a reference label for the corresponding CGM values. Here, the CGM values are grouped by date and time. I tried to analyze the trend of continuous glucose levels. After receiving the input LSTM-CGM value, the predictive model attempts to predict the CGM value from time t to $t + \alpha$ time units. These CGM values are sent as input

Using CRF we added predictions of meal/no-meal, using the model trained for three epochs. We see in Fig 11 that there is over fitting resulting from a disparity between number of meal and no meal values.

IV. Conclusion and Future Work

Our four models and algorithms worked well in terms of execution, demonstrating the usefulness of this technique. However, one can extend these to improve accuracy. We believe that through research and the application of new methods, we can improve the threshold technique for detecting diet.

References

[1] J. Feng E. Littlejohn L. Quinn . K. Turksoy, S. Samadi and A. Cinar. ”meal detection in patients with type 1 diabetes: a new module for the multivariable adaptive artificial pancreas control system”. IEEE journal of biomedical and health in-formatics, 20(1):47–54, 2015.

[2] Eyal Dassau, B Wayne Bequette, Bruce A Buckingham, and Francis J Doyle. Detection of a meal using continuous glu-cose monitoring: implications for an artificial -cell. Diabetes care, 31(2):295–300, 2008I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Snezhana Georgieva Gocheva-Ilieva, Atanas Valev Ivanov, Desislava Stoyanova Voynikova, and Doychin Todorov Boy-adzhiev. Time series analysis and forecasting for air pollu-tion in small urban area: an sarima and factor analysis ap-proach. Stochastic environmental research and risk assess-ment, 28(4):1045–1060, 2014.

[4] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991, 2015.

[5] D. Bahdanau K. Cho, B. Van Merrienboer¨ and Y. Bengio. “on the properties of neural machine translation: Encoder-decoder approaches”. arXiv preprint arXiv, 2014.

[6] Carmen Perez´-Gand´ıa, A Facchinetti, G Sparacino, C Cobelli, EJ Gomez,´ M Rigla, Alberto de Leiva, and ME Hernando. Ar-tificial neural network algorithm for online glucose prediction from continuous glucose monitoring. Diabetes technology & therapeutics, 12(1):81–88, 2010.

[7] T. Mikolov R. Pascanu and Y. Bengio. “on the difficulty of training recurrent neural networks”. International conference on machine learning, pages 1310–1318, 2013.

[8] Giovanni Sparacino, Francesca Zanderigo, Stefano Corazza, Alberto Maran, Andrea Facchinetti, and Claudio Cobelli. Glu-cose concentration can be predicted ahead in time from con-tinuous glucose monitoring sensor time-series. IEEE Trans-actions on biomedical engineering, 54(5):931–937, 2007.

[9] Understanding RNN and LSTM by Aditi Mittal - https://aditimittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[10] Multivariate Time Series Analysis with LSTMs – All Codeless by Kathrin Melcher- https://www.knime.com/blog/multivariate-timeseries-analysis-lstm-codeless