CSE 535: Group 46
CGM Prediction

Swarali Chine
ASU ID: 1222583687
schine@asu.edu

Yash Deshpande
ASU ID: 1222186498
ydeshpa2@asu.edu

Yash Gandhi
ASU ID: 1222325208
ygandhi4@asu.edu

Sargunjot Singh
ASU ID: 1222357760
ssing369@asu.edu

ABSTRACT

About 10 percent of the World's adult population are affected by Diabetes. With the technology advancement there are more reliable treatments and systems available for diabetes. Continuous blood glucose monitoring (CGM) is one of the major developments. The level of glucose in patients is one of the important aspects of diabetes and Continuous monitoring of glucose level is required. So, with the help of different algorithms like SARIMA, Kalman and Recurrent Neural Network (RNN) we have developed a system for continuous meal detection.

KEYWORDS: Continuous glucose monitoring; auto regression; kalman filter, recurrent neural networks.

INTRODUCTION

Meal detection in Type-1 diabetic patients is very important. When glucose level is high and the body does not release enough insulin can lead to very harmful situations such as hyperglycemia. With the help of technology like CGM sensors which are used to monitor glucose level, we can prevent such harmful situations. But these devices have their own limitations and inaccuracy. So, Meal detection using algorithms can be useful. In this project we have used continuous CGM time series data with bolus ground truth to develop the continuous meal detection method using Autoregression, Kalman filter and RNN based approach. When we detect meals, we can measure insulin levels and maintain glucose levels.

PROJECT SETUP

The data given with CGM value and bolus values data of a patient for every 5 minutes for nearly 6 months. The data contains around 55000 data entries with some empty data. First we have to convert data into a CSV file. The data is from 2 different sensors which give CGM and Bolus values. So we synchronise the data with respect to time and save the data into a csv file which is used in all algorithms. We first used a matlab file - Preprocess_MAT_file.m for getting unsynchronized data in csv format. Then we are using the datacgm2.csv in the jupyter notebook of data_processing.ipynb for creating the finalData.csv which contains the synchronised data. This data is used by the algorithms to train the models in the respective notebook files of the algorithms.

IMPLEMENTATION DETAILS

We implemented three algorithms for meal detection: SARIMA based model, kalman filter based model, and a Recurrent Neural Networks based approach and an initial algorithm as RNN - Gated Recurrent Unit.

A. SARIMA

Seasonal Autoregressive Integrated Moving Average (SARIMA) is a very common and useful technique for time series modelling and forecasting. The fact that SARIMA accounts for the seasonal variation in time series is what makes it so powerful. Our CGM data has seasonal components in it which explains the use of SARIMA for this project. SARIMA comes with trend and seasonality parameters that must be configured before you use it. The hyperparameters for trend elements like

- order of trend (p)
- difference order (d)

- trend moving average order (q)

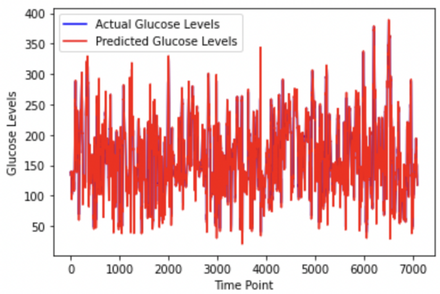
and seasonal elements like

- Seasonal auto-regressive order (P)
- Seasonal differencing order (D)
- Seasonal moving average (Q)
- Seasonal period (m)

need to be configured prior to using the model. Where the specifically chosen hyperparameters for a model are specified; for example, we have used the following parameters for initialising our model: pdq = (1 ,1, 1) PDQm = (1, 0, 1, 12)

Implementation: We used the univariate time series data of glucose that we obtained from the Data processing module. First, the data was reversed so that the first data element is the oldest data (2017) and the most recent data is the last entry. We trained our SARIMA model with 80% of the patient cgm data. We used our model to test the 20% of the patient1 cgm data and calculated the error scores and also plotted the predicted and the actual values of CGM as shown in the figure.

```
Test MSE: 183.373

Test MAE: 6.031

Test RMSE: 13.542
```



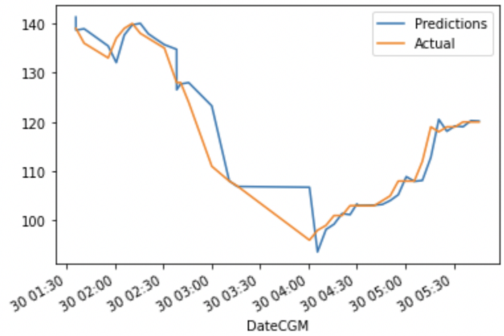
The accuracy of meal detection was calculated by setting a threshold value by empirically studying the cgm values. The threshold which decides meal detection was used as 0.4*mean(cgm) values of the dataset. This was then used to map with the actual ground truth values of the meal given by actual bolus delivered values given to us for that particular patient. The figures below show the accuracy and the results of the SARIMA model. Later the model was tested on patient 2 and we got an accuracy of 83.4319526627219%

```python
cnt = 0
idx = []

for i in range(7099):
  x = res111['Predictions'][i]
  y = truth_values1[i]
  if math.isnan(y) or math.isnan(x):
    continue
  if (abs(x-y)) > threshold["Value CGM"]:
    cnt += 1
    idx.append(i)

print("Test Acc (%):", cnt/b_cnt*100)
```

```
Test Acc (%): 83.4319526627219
```

The figure below shows the predicted and actual cgm values for 8 hours of a day.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fed8ea8a810>
```



B. Kalman Filter

The Kalman filter is one of the common algorithms used for estimations. The Kalman filter provides a prediction of the future based on the past. Kalman observes different values over time such as noise and accuracy and predicts the values of unknown variables.

$$G_s(k+1) = h\left[p_1(k)\left(G_b(k) + \frac{G_s(k)}{hp_1(k)} - G_s(k)\right) - p_2(k)\, I_{eff}(k)\, G_s(k) + R_a(k)\right]$$

The Unscented Kalman filter is a powerful tool for state estimation of nonlinear systems. The drawbacks of linear approximation at an operating point and calculation of Jacobian matrices in the Extended Kalman filter (EKF) are overcome by using a minimal set of carefully chosen sample (sigma) points[1].

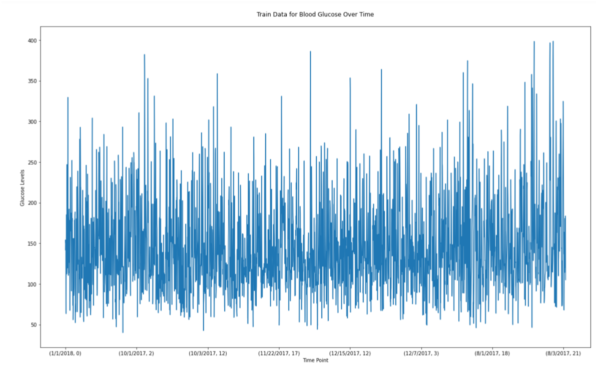
Kalman filter is implement using below given equations:

$$x(k+1) = f(x(k)) + w(k)$$

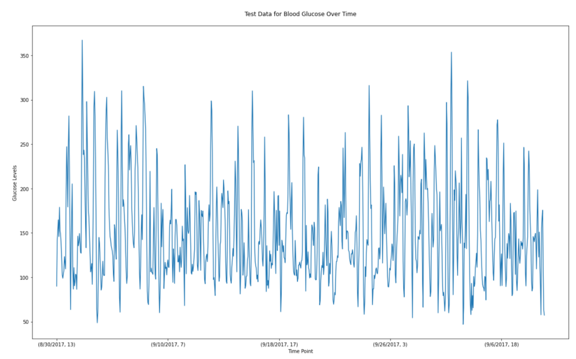
The meal detection algorithm developed above using the kalman filter is able to detect all the tested meals with accuracy of 99.92. Kalman is an optimal estimator and relatively quick and easy to implement and provides an optimal estimate of the condition for normally distributed noisy sensor values under certain conditions.

C. Recurrent Neural Networks

We have implemented Recurrent Neural Network [7] with Long-short Term memory (LSTM) for meal detection. RNN has short term memory so it will encounter a Vanishing Gradient Problem. RNN with LSTM provides a solution to the Vanishing Gradient Problem because it has a forget gate. RNN has 3 gates in its architecture, it contains input gates, output gates and forget gates.



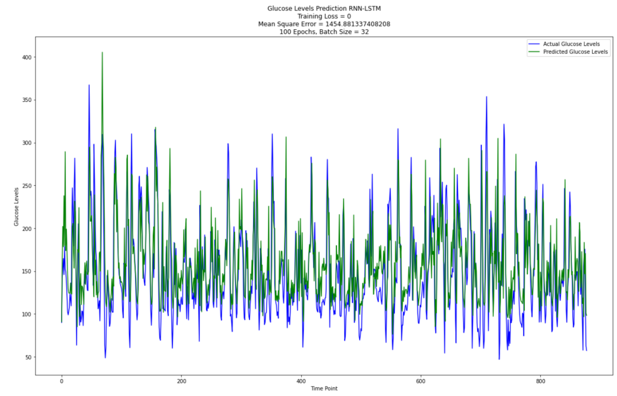
Train data for Blood Glucose over time



Test data for Blood Glucose over time

In this model we have created a feedback like mechanism, where we predict the value and then append the value to input data, to predict the future values. After reading and transforming data we split it into a train and test dataset in ratio of 0.8 and 0.2 as shown in images. Then we feed the data into the model. Model contains 4 LSTM layers containing 50 LSTM units and 4 dropout layers followed by a dense layer. For training the data we used a batch size of 32 and 100 epochs. After training the loss turned out to be 0.01099 .

While prediction using the algorithm here, we used Adam as optimizer and mean squared error as loss function.



Prediction vs Actual glucose level over time

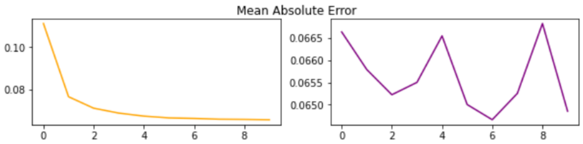
After 100 epochs, the model predicts items as shown in the above image.

D. RNN using Gated Recurrent Units

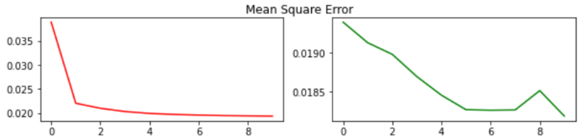
Here, we implemented RNN model for meal detection with the use for Gated Recurrent Units (GRU). GRU works similar to LSTM but has different architecture. GRU has two gates: reset gate and update gate. Reset gate decides how much past information is important and how much to forget. Update gate decides if the information needs to update or not. Due to two gates GRU is less complex than LSTM and works faster.

In CGM we take the time series data of CGM. After that we apply a time shift on CGM and bolus values and prepare the data for the RNN model. Then take input as the statistical features of the data and then split that into train and test data with 80% and 20% data respectively. After applying transformation and scaling in the data, data given to the model. Model has 2 layers of GRU with 100 Units followed by 3 dropouts and a dense layer. In the model we use Adam optimizer and mean squared error as loss function. Also there is an R square, mean absolute error as the evaluation matrix. After 50 epochs the model gives loss value is 0.0194, R square is 0.0397 and mean absolute error is0.0657. For meal prediction we can use the same method used in the Sarima model.As we observed, the model gives less accurate results because of the long sequence to be predicted on time series data.

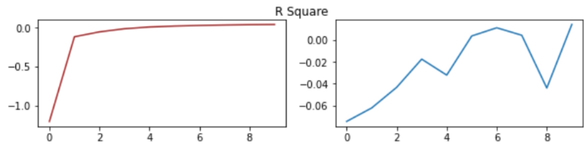
The below graph shows results of GRU evaluation matrices



Mean Absolute Error of plot of GRU



Mean Square Error of plot of GRU.



R square plot of GRU

Execution Time analysis

The time taken for training for each approach is calculated in this analysis.

- SARIMA training the model took less than 1 minute to train (26 seconds for training).
- The Kalman filter took 2 seconds.
- LSTM-RNN took less than 10 minutes (on average 8-10 minutes).
- RNN-GRU took less than 10 minutes (on average 6-8 minutes).

Conclusion

We have implemented methods to monitor glucose level using SARIMA, Kalman filter, Recurrent Neural network using LSTM and GRU. We get good accuracy using these models. We believe that models can be improved to get better accuracy.

Table

Sr	Task	Assign
1	Parse CGM Data (Matlab)	Yash Deshpande / Sargunjot Singh
2	Data Synchronisation	Yash Deshpande/ Sargunjot Singh
3	Development of SARIMA algorithm	Yash Deshpande/ Sargunjot Singh
4	SARIMA Algorithm Instantiation	Yash Deshpande
5	SARIMA Algorithm implementation	Yash Deshpande
6	SARIMA Training and Testing accuracy	Yash Deshpande/ Sargunjot
7	Development of RNN algorithm	Sargunjot/ Yash Deshpande
8	RNN algorithm Instantiation	Sargunjot Singh
9	Algorithm implementation	Sargunjot Singh
10	Training and Testing accuracy	Sargunjot Singh/ Yash Deshpande
11	Development of Kalman Filter algorithm	Swarali Chine/ Yash Gandhi
12	Kalman Filter algorithm Instantiation	Swarali Chine
13	Kalman Filter Algorithm implementation	Swarali Chine
14	Kalman Filter Training and Testing accuracy	Swarali Chine/ Yash Gandhi
15	Development of RNN-GRU algorithm	Yash Gandhi/ Swarali Chine
16	RNN-GRU Algorithm Instantiation	Yash Gandhi
17	RNN-GRU Algorithm implementation	Yash Gandhi
18	RNN-GRU Training and Testing accuracy	Yash Gandhi/ Swarali Chine
19	Testing the Model on Patient 2 data	Yash Gandhi/ Swarali Chine
20	Execution Time analysis	Yash Gandhi/ Swarali Chine

References

[1] J. Feng E. Littlejohn L. Quinn . K. Türksoy, 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 informatics, 20(1):47–54, 2015.

[2] Eyal Dassau, B Wayne Bequette, Bruce A Buckingham, and Francis J Doyle. Detection of a meal using continuous glucose monitoring: implications for an artificial β-cell. Diabetes care, 31(2):295–300, 2008.

[3] Snezhana Georgieva Gocheva-Ilieva, Atanas Valev Ivanov, Desislava Stoyanova Voynikova, and Doychin Todorov Boyadzhiev. Time series analysis and forecasting for air pollution in small urban areas: a sarima and factor analysis approach. Stochastic environmental research and risk assessment, 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. Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series. IEEE Transactions on biomedical engineering, 54(5):931–937, 2007.