Multi-Step ahead Predictive Model for Blood Glucose Concentrations of Type-1 Diabetic Patients

Syed Mohammed Arshad Zaidi (✉ szaidi2@buffalo.edu)

University at Buffalo, State University of New York

Varun Chandola

University at Buffalo, State University of New York

Muhanned Ibrahim

University at Buffalo, State University of New York

Bianca Romanski

RWTH Aachen University

Lucy D. Mastrandrea

University at Buffalo, State University of New York

Tarunraj Singh

University at Buffalo, State University of New York

Research Article

Keywords: Multi-step, predictive model, Blood Glucose

Posted Date: March 24th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-315127/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Scientific Reports on December 1st, 2021.
See the published version at <https://doi.org/10.1038/s41598-021-03341-5>.

Multi-step ahead predictive model for Blood Glucose Concentrations of Type-1 Diabetic Patients

Syed Mohammed Arshad Zaidi^{1,*}, Varun Chandola¹, Muhanned Ibrahim¹, Bianca Romanski⁴, Lucy D. Mastrandrea³, and Tarunraj Singh²

¹University at Buffalo-SUNY, Computer Science and Engineering, Buffalo, 14260, USA

²University at Buffalo-SUNY, Mechanical and Aerospace Engineering, Buffalo, 14260, USA

³University at Buffalo-SUNY, Division of Pediatric Endocrinology, Buffalo, 14203, USA

⁴RWTH Aachen University, Medical Information Technology, Pauwelsstr. 20, 52074 Aachen, Germany

*szaidi2@buffalo.edu

ABSTRACT

Continuous monitoring of blood glucose (BG) levels is a key aspect of diabetes management. Patients with Type-1 diabetes (T1D) require an effective tool to monitor these levels in order to make appropriate decisions regarding insulin administration and food intake to keep BG levels in target range. Effectively and accurately predicting future BG levels at multi-time steps ahead benefits a patient with diabetes by helping them decrease the risks of extremes in BG including hypo- and hyperglycemia. In this study, we present a novel multi-component deep learning model *BG-Predict* that predicts the BG levels in a multi-step look ahead fashion. The model is evaluated both quantitatively and qualitatively on actual blood glucose data for 97 patients. For the prediction horizon (PH) of 30 minutes, the average values for *root mean squared error* (RMSE), *mean absolute error* (MAE), *mean absolute percentage error* (MAPE), and *normalized mean squared error* (NRMSE) are 20.82 ± 4.97 mg/dL, 15.04 ± 3.70 mg/dL, 11.63 ± 2.78 and 0.06 ± 0.01 respectively. When Clarke and Parkes error grid analyses were performed comparing predicted BG with actual BG, the results showed average percentage of points in Zone A of 83.88 ± 6.84 and 87.44 ± 4.97 , respectively. We offer this tool as a mechanism to enhance the predictive capabilities of algorithms for patients with T1D.

Introduction

Diabetes mellitus is a metabolic disease that causes the abnormal regulation of blood glucose (BG) levels in the body. Insulin, an endocrine hormone produced by the pancreas, facilitates uptake of glucose by a variety of cells. Type-1 diabetes (T1D) or insulin-dependent diabetes is an autoimmune condition requiring external administration of insulin for the regulation of blood glucose. Type-2 diabetes (T2D) results from insulin resistance or insulin insensitivity and the regulation of the BG levels require oral medications or insulin. Out of 463 million people with diabetes, approximately 1 million children and adolescents (under age 20) suffer from Type-1 diabetes¹.

Although T1D cannot be cured or prevented, BG management is accomplished with subcutaneous administration of insulin either by injection or continuous infusion. Patients with T1D must monitor their BG level throughout the day and take necessary actions to prevent *hypoglycemia* (low BG levels) and *hyperglycemia* (high BG levels). The ability to accurately predict future BG levels would help patients prevent both low and high blood sugar levels and allow them to meet glycemic targets and decrease the risk of long-term complications. Factors that affect future BG levels include prior BG levels, insulin dose, meal intake, body mass index, physical activity among others. Incorporating all the features to predict BG levels with high accuracy poses a challenging task¹. This is complicated by the physiological yet unpredictable variations seen in the BG levels caused throughout the day (dawn and dusk phenomena). For example, emotional stress can trigger hyperglycemia while physical activity can enhance insulin sensitivity thereby causing hypoglycemia for several hours. Capturing such variations in itself poses a unique challenge for any model to have enough predictive power to forecast BG levels.

Existing BG prediction approaches fall under two paradigms – *physiological-driven* and *data-driven*. Physiological-driven methods typically follow a compartment-based modeling approach, that requires extensive knowledge about the underlying mechanism for each individual². Data-driven approaches use an individual's recorded historical data such as *Continuous Glucose Monitoring* (CGM) readings, meal intake information, programmed basal rate infusion amounts, etc., to learn a predictive model³⁻⁵. Data-driven approaches have the advantage of not having to model the underlying, and often unpredictable, physiological mechanisms. With tremendous advances in acquiring relevant medical data from sources such as CGMs⁶, insulin pumps⁷, and point of care (POC) devices⁸, data-driven approaches have shown promising results for BG prediction. In the past, several statistical and machine learning (ML) techniques, such as Auto Regressive Integrated Moving Average (ARIMA)⁹,

Support Vector Regression (SVR)^{10,11}, and Artificial Neural Networks (ANN)^{12,13} have been used for one-step ahead BG prediction. These models, however, perform poorly for long-term forecasting since they do not effectively capture long-term dependencies. Recently, deep recurrent neural network architectures, such as the *Long Short-Term Memory* (LSTM) networks, have received considerable attention since they handle long-term dependencies well and effectively outperform any of traditional machine learning approach for predicting blood glucose with prediction horizon of 30 minutes^{14–16}.

Though these recurrent neural networks, like LSTMs, have become the preferred tools for sequence modeling tasks, such as language translation¹⁷, they are shown to have some drawbacks, primarily in memorizing long temporal sequences. In fact, recent studies have shown that using a one-dimensional Convolution Neural Network (CNN) can better model long-range temporal dependencies¹⁸. This has led to the use of Temporal Convolution Neural Networks (TCNs) over LSTMs for many time series prediction tasks that involve long-term memories¹⁸. We present a novel TCN based predictive model that can effectively capture long-term history of an individual for BG prediction, and report significant improvements over state of art BG prediction models for a 30 minute prediction horizon. The salient aspects of the proposed model include: a) multi-step prediction that better captures the temporal auto-correlations in the output compared to traditional one-step prediction schemes, b) multi-compartment model that uses varying length histories for different inputs and their combinations, and, c) individual-specific model training to learn the individual characteristics for BG prediction.

To the best of our knowledge this is the first work focusing on modeling future blood glucose based on varying past history information on blood glucose, food intake, basal and bolus insulin using the proposed approach. The illustrated experimental results derived from clinical data for 97 T1D patients show the effectiveness of our proposed model in forecasting BG with prediction horizon of 30 minutes.

Results and Discussion

The experiments were conducted on Tidepool² datasets provided by the Juvenile Diabetes Research Foundation (JDRF)³. The datasets consists of real world data for 97 patients with T1D patients. For each patient, an unseen test data set was used for evaluating the performance of the model. We computed the following quantitative measures of error – *root-mean-square error (RMSE)*, *mean-absolute-error (MAE)*, *mean absolute percentage error (MAPE)* and *normalized root mean square error (NRMSE)*. We also computed these measures for three different categories within the data set. These three subsets pertain to actual blood glucose concentrations falling in different range values. The categories were as follows: 1) **Hypoglycemic** range with blood glucose less than 70 mg/dL 2) **normoglycemic** range with BG values greater than or equal to 70 mg/dL and less than 180 mg/dL, and 3) **hyperglycemic** range with BG readings greater than or equal to 180 mg/dL.

Results

The results for individual patients are presented in Supplementary Table 2. The table provides a summary of both the overall and detailed quantitative performance of the model to predict BG levels 30 minutes in the future. The performance of our model is first compared with other baseline approaches. We then evaluate the performance of our proposed model both quantitatively and qualitatively with respect to different performance measures described in the [evaluation metrics section](#).

Comparison with other approaches

Table 1 presents the results of comparison of our model, BG-Predict with other baseline approaches in terms of RMSE, MAE, MAPE and NRMSE. The units for RMSE, MAE are in mg/dL.

1. **Naive Forecasting** – In naive forecasting, the last recorded observation of blood glucose concentration is copied to the 30 minutes ahead and the model is evaluated with the actual blood glucose concentration. This can be considered as a baseline method in comparing other methods.
2. **Gaussian Processes (GP)** – Gaussian Process is a Bayesian non-parametric approach that generalizes the Gaussian distribution of functions and can be used for both regression and classification. We use the Gaussian Process regression model for forecasting blood glucose concentration 30 minutes ahead.
3. **Kernel Ridge Regression (KRR)** – Kernel Ridge Regression (KRR) is a non-parametric model that combines ridge regression (linear least squares and L_2 -norm regularization) with the kernel trick. The model learns the function in a space induced by the choice of kernel and the data. We used Linear and Radial Basis Function (RBF) kernel for forecasting blood glucose concentration 30 minutes ahead.

Model Approach	RMSE	MAE	MAPE	NRMSE
Naive Forecasting	28.84 $\pm$ 8.83	20.39 $\pm$ 6.32	15.63 $\pm$ 4.94	0.20 $\pm$ 0.06
GP	27.97 $\pm$ 9.16	19.80 $\pm$ 6.24	14.63 $\pm$ 4.03	0.19 $\pm$ 0.05
KRR (Linear)	25.36 $\pm$ 6.45	18.49 $\pm$ 4.78	14.05 $\pm$ 3.48	0.18 $\pm$ 0.04
KRR (RBF)	33.13 $\pm$ 8.79	23.89 $\pm$ 6.38	17.67 $\pm$ 4.14	0.23 $\pm$ 0.04
BG-Predict	20.82 $\pm$ 4.97	15.04 $\pm$ 3.70	11.63 $\pm$ 2.78	0.06 $\pm$ 0.01

Table 1. Comparison with other approaches

Discussion

This study investigates the impact and use of deep learning approaches for modeling blood glucose levels using a multi-step approach. This approach will provide well-informed analysis as well as support to individuals with Type 1 diabetes. For the prediction horizon (PH) of 30 minutes, the proposed model has been evaluated per patient both quantitatively and qualitatively. The detailed results for 30 minutes ahead forecasting can be found as Supplementary Table S2. In the table, some of the patients had no hypo- or hyperglycemic test samples and this has been indicated as *No test samples*. Quantitatively, the proposed method shows the overall average value to be 20.82 $\pm$ 4.97 mg/dL in root mean squared error (RMSE), 15.04 $\pm$ 3.70 mg/dL in mean absolute error (MAE), 11.63 $\pm$ 2.78 in mean absolute percentage error (MAPE) and 0.06 $\pm$ 0.01 in normalized mean squared error (NRMSE). For Hypoglycemia, the average value is 25.10 $\pm$ 9.92 mg/dL in root mean squared error (RMSE), 19.49 $\pm$ 8.66 mg/dL in mean absolute error (MAE), 32.31 $\pm$ 14.46 in mean absolute percentage error (MAPE) and 0.08 $\pm$ 0.03 in normalized mean squared error (NRMSE). For Hyperglycemia, the average value is 28.11 $\pm$ 7.86 mg/dL in root mean squared error (RMSE), 21.65 $\pm$ 6.74 mg/dL in mean absolute error (MAE), 9.99 $\pm$ 3.29 mg/dL in mean absolute percentage error (MAPE) and 0.09 $\pm$ 0.04 in normalized mean squared error (NRMSE). For Normoglycemia, the average value is 19.02 $\pm$ 4.20 mg/dL in root mean squared error (RMSE), 13.73 $\pm$ 3.10 mg/dL in mean absolute error (MAE), 11.55 $\pm$ 2.66 in mean absolute percentage error (MAPE) and 0.06 $\pm$ 0.01 in normalized mean squared error (NRMSE).

In terms of accuracy, the overall true positive rate for hypoglycemia, hyperglycemia and normoglycemia is 0.31 $\pm$ 0.18, 0.75 $\pm$ 0.16 and 0.93 $\pm$ 0.04 respectively. On the other hand, the overall false positive rate for hypoglycemia, hyperglycemia and normoglycemia is 0.02 $\pm$ 0.02, 0.04 $\pm$ 0.02 and 0.30 $\pm$ 0.14 respectively. In general, the model was found to be difficult in capturing the hypoglycemic events while capturing the hyper- and normo-glycemic events well. The false positive rates for all the three events were pretty low indicating that the model didn't raise a large number of false alarms while performing the blood glucose forecasting at 30 minutes ahead. The results of clinical or zonal accuracy for all the patients in terms of Clarke and Parkes error grids can be found as Supplementary Figure S1 and Supplementary Figure S2 respectively. In Clarke and Parkes error grid analysis, the average percentage of points lying in Zone A were 83.88 $\pm$ 6.84 and 87.44 $\pm$ 4.97 respectively. The model consistently provides accurate result as indicated by predicted BG values in Zone A in both the Clarke and Parkes error grid.

Overall, these findings indicate that the blood glucose can be effectively estimated for a 30 minutes prediction horizon for individual patients. This model has an inherent advantage of taking multi-view of the data and we can also leverage other information such as physical activity that can further enhance the predictive accuracy of blood glucose levels at prediction horizon of 45 minutes and 50 minutes. Amongst the related approaches, Marcus et al.¹⁹ presented the use of kernel ridge regression (KRR) approach in predicting the future BG at 30 minutes with just the past history of BG values for individual patient. The model was validated on 11 T1D patients and their reported average RMSE values is 20.48 mg/dL. Mhasker et al.¹⁵ make use of non-patient specific semi-supervised deep learning approach for predicting future blood glucose values. This study only included the subject age group children less than 18 years old and didn't include the prediction of blood glucose measurements. Mirshekarian et al.²⁰ used long short-term memory (LSTM) networks to predict the blood glucose for 10 T1D patients. The subject age group in this study wasn't reported, however, the RMSE for 30 minutes was reported to be 21.4 mg/dL.

Fig. 1 shows the prediction accuracy of the model 30 minutes ahead on validation data for two T1D patients. It can be seen that the model is able to capture the hypoglycemic and hyperglycemic events well.

Methods

Problem Formulation

Given the historical data inputs of **meal intake** (M), **blood glucose** (BG), **basal insulin rate** (Ba) and **bolus insulin** (Bo), we want to predict the future BG values in multi-step manner. The problem can be formulated as:

- Given $\{M_0, \dots, M_t\}$; $\{BG_0, \dots, BG_t\}$; $\{Ba_0, \dots, Ba_t\}$ and $\{Bo_0, \dots, Bo_t\}$

Figure 1. 30 minutes forecasting on validation data for two T1D patients

- Predict $BG_{t+1}, \dots, BG_{t+f}$

We define our model by accounting the different input history length of these input variables. Considering the different input history, we reformulate the problem as:

- Given $\{(M, BG)_{t-p_1}, \dots, (M, BG)_t\}; \{BG_{t-p_2}, \dots, BG_t\}; \{(Ba, BG)_{t-p_3}, \dots, (Ba, BG)_t\}$ and $\{(Bo, BG)_{t-p_4}, \dots, (Bo, BG)_t\}$
- Predict $BG_{t+1}, \dots, BG_{t+f}$

We denote f as the prediction horizon (PH) while the starting position in history for the inputs sequence length in the four components of our proposed model is denoted by p_1, p_2, p_3 and p_4 . Here, $(\cdot, \cdot)$ denotes the interaction term indicating the multiple variables for all time steps to be fed to a component of the proposed model. For example, (M, BG) implies considering both input **meal intake** and **blood glucose** variables for all time steps for sequence length indexed from $t - p_1$ until t .

Data

The experiments in our study are conducted on data for 300 patients with T1D. All methods were carried out in accordance with relevant guidelines and regulations. The experimental analysis was conducted on anonymized datasets obtained from the Tidepool Big Data Donation Project⁴. These datasets do not require IRB approval for use due to the HHS Policy for Protection of Human Research Subjects exemption research guideline, 45 CFR 46.104⁵. Since the purpose of this study is to model the blood glucose dynamics of adult patients with type-1 diabetes, we adopt a strategy as shown in Figure 2 for identifying eligible patients for our experiments. The inclusion of the patients is made on the basis of some Eligibility Criteria (EC). The criterions are: 1) **EC1**: Patients with confirmed type-1 diabetes, 2) **EC2**: Patients of age > 18 years, and 3) **EC3**: Patients whose time since diagnosis > 2 years.

The characteristics of the data for each patient can be found in Supplementary Table S1. Each of the patient's data is divided into training and testing dataset each of which consists of *CGM recordings*, *Basal*, *Normal Bolus*, *Meal Intake*. The table provides the number of CGM samples, frequency of meals taken, hypo-, hyper- and normo-glycemic values in training and test data for each patient. In each of the patient's data, the observation value for CGM measurement, meal intake, basal and bolus insulin is recorded every five minutes.

Temporal Convolutional Networks (TCN)

Sequence-based prediction tasks are solved using Recurrent neural networks (RNN) like (Long short-term memory) LSTM networks and Gated recurrent unit (GRU) networks in the deep learning world. Although these approaches were quite successful in solving several sequence-to-sequence tasks, we encountered certain limitations such as inability to handle long time sequences as well as vanishing and exploding gradients. Temporal Convolutional Networks (TCN) were introduced for video-action segmentation task²¹ to overcome the issues faced by the RNN-based approaches. TCN can be seen as the combination of 1D convolutional neural network with dilated and causal convolutions.

Causal and dilated convolutions

The primary characteristics of temporal convolutional network is that the convolutions are causal and dilated. Causal convolution implies that the convolution operation output at time t in a current layer is performed by considering elements at time t and earlier in the previous layer. Causal convolutions ensure that there is no information leakage from future to present while computing the output prediction at time t . By just performing the simple causal conventional convolution operation, we are only looking back at the history of sequence with size linear in the depth of the network. This makes the task more challenging

Figure 2. Flowchart for processing Tidepool data

for longer input sequences. In order to alleviate this problem, dilated convolutions were introduced to enable large receptive field in ¹⁸.

Use of dilated convolutions implies applying the convolution filter to an area larger than its size by enhancing the filter by dilating it with zeros. Alternatively, the filter can be applied over the larger area in the input by skipping input values with a certain step or dilation factor. Formally, dilated convolution is defined as:

$$F(s) = (\mathbf{x} *_{\mathbf{d}} f) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i}$$

Here, $\mathbf{x}$ is the input sequence, d is the dilation factor, f is the filter and k is the filter size. F is the dilated convolution operation, s is an element of the input sequence, $\mathbf{x}$, $s - d \cdot i$ accounts for the direction of the past. The use of dilated convolution serves two purposes, *firstly* the output of each convolution propagates rich information while tracking very long data sequence. As the dilation factor is increased, the range of inputs is increased and have increased receptive field and *secondly*, it is efficient in computation as the computation cost is less as compare to the use of larger filter size to increase the receptive field. Dilated convolutions are shown in Fig. 3 with dilation factors d and kernel size k .

Figure 3. Dilated Causal Convolutions with dilation factor $d = 1, 2, 4$, and kernel size, $k = 3$

Residual Network

Deep neural networks are popularly used in several studies related to image processing. As we stack more layers, there is great improvement in results, however, there are problems related to vanishing/exploding gradients^{22,23}. Although, these problems can be alleviated with different solutions such as initialization strategies²⁴, etc., there are still issues related to degradation, i.e. arising of higher training error, as we add more layers to the network. With the advent of residual block²⁵, we can achieve more deep layered network without leading to degradation. Instead of stacking layers and learning the desired (true) mapping

function $\mathcal{H}(x)$ like in the conventional neural network block, the layers of the residual block tries to learn the residual function $\mathcal{F}(x)$, i.e., $\mathcal{F}(x) = \mathcal{H}(x) - x$. The output of the residual block will lead to the learning of desired mapping function, which can be recast as $\mathcal{H}(x) = \mathcal{F}(x) + x$. This expression can be realized with the help of shortcut/skip connections. This allows to skip one or more layers without introducing any extra parameter thereby avoiding any further increment to computational complexity. Since, the use of deeper and larger temporal convolutional neural network helps in computation involving larger receptive fields, it would be important to make sure that this large and deep network remains stable throughout training. In order to do this, the use of residual block is kept in the TCN based model.

Figure 4. Residual Block in Temporal Convolutional Network

Figure 4 shows the structure of the residual block that is incorporated in each of the hidden layer in the temporal convolutional network. The residual block in each of the hidden layer differs in terms of dilation factors d . The residual block consists of two 1D convolutional layer with same dilation factor d and kernel size k for better capturing long-term temporal dependencies. Each of which is followed by the normalization, activation and dropout layer. For the normalization block, we use layer normalization²⁶, which is used to normalize the activations of the previous layer for each given example independently in a batch rather than across batch as seen in batch normalization²⁷. The Relu block comprises of the ReLU (Rectified Linear Unit) activation, which is a non-linear and differentiable activation function that is useful for learning complex, non-linear mapping of data. We use a 1D version of spatial dropout²⁸, a regularization technique like dropout but which differs in its working operation. In the normal dropout, individual nodes are dropped off the nodes in the network randomly while 1D version of the spatial dropout drops the entire 1D feature maps which helps in maintaining the independence between feature maps. The 1×1 conv1D layer is to ensure that the width of the output matches with the input.

Seq-to-Seq Models

Sequence-to-Sequence model approach aims at mapping a fixed length input to a fixed length output. Here, input sequence and output sequence may or may not be of same length.

Figure 5. Sequence-to-Sequence approach

This approach have been popularly used in language modeling tasks such as machine translation²⁹, speech recognition³⁰ and video captioning³¹. The structure of a typical encoder-decoder approach involving temporal-convolutional networks is shown in Figure 5.

Here, both the encoder and decoder consists of TCN. The encoder takes the input sequence and processes the input by passing it through different residual blocks having convolution layers with different dilation factors d , collecting all the relevant information and producing an encoded state vector that best summarizes or captures all the input information that were seen by the encoder TCN. This encoded state vector is also called the hidden or latent state. This hidden state is then fed to the decoder part to start the prediction of output sequence one step at a time which is then used to produce the predicted sequence output.

Figure 6. Proposed model

Proposed Model

Our proposed model consists of four components that use temporal convolutional network embedded within encoder-decoder sequence-to-sequence model approaches that take into account different past history of input predictors as shown in Figure 6.

Since, the model predictions are done in multi-step manner and inputs and output sequences are of variable length, we made use of encoder-decoder sequence architecture, which models the sequence to sequence tasks. The purpose of using different input time sequence is because the future blood glucose would be functionally impacted differently by different inputs. For example, the blood glucose levels for an individual with T1D are impacted by food intake (carbohydrates), basal insulin delivery, and bolus insulin administered to cover food and regulate BG levels within a target range.

Given the maximum history of available input information until timestep t , in order to predict BG at future timesteps until prediction horizon f as an output sequence, $\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+f}$, we divide the input temporal sequence history of variables in the following four components that forms the basis of predicting the final sequence output. These *four* components are:

- **History of BG and Meal intake** – In this component, we consider a longer past history of three hours for *BG* and *meal intake* for predicting the future BG. The meal intake in the past impacts the BG levels in the future as meal or the carbohydrates intake usually increases the blood glucose levels. The input sequence for this component is $\langle BG_{t_1}, Meal_{t_1} \rangle$ where $t_1 = \{t - p_1, t - (p_1 - 1), t - (p_1 - 2) \dots, t\}$. Here, we denote the length of past history for this component by p_1 .
- **Recent history of BG** – This component comprises of the recent 30 minutes history of BG. The close history information on the BG history is considered here to aid the improvement in output predictions. The input sequence for this component

is $\langle BG_{t_2} \rangle$ where $t_3 = \{t - p_2, t - (p_2 - 1), t - (p_2 - 2), \dots, t\}$. Here, we denote the length of past history for this component by p_2 .

- **History of BG and basal rate** – This component comprises of inputs of three hours of history of *BG* and *basal rate*. Basal insulin is the background insulin that runs continuously and regulates BG levels between meals, and during times of fasting (ie sleep). The input sequence for this component is $\langle BG_{t_3}, Basal_{t_3} \rangle$ where $t_3 = \{t - p_3, t - (p_3 - 1), t - (p_3 - 2), \dots, t\}$. Here, we denote the length of past history for this component by p_3 .
- **History of BG and bolus insulin** – In this component, we take the input history of three hours for BG and bolus insulin. This insulin is required when the patient consumes a meal or gives insulin to correct a high blood sugar value. The amount of bolus insulin required by the body is largely dependent on amount of meal intake (usually in grams), total daily insulin requirements, and the target range of the BG level. The input sequence for this component is $\langle BG_{t_4}, Bolus_{t_4} \rangle$ where $t_4 = \{t - p_4, t - (p_4 - 1), t - (p_4 - 2), \dots, t\}$. Here, we denote the length of past history for this component by p_4 .

In the model shown in Figure 6, the starting position history for the inputs sequence length in the four components can be determine through p_1, p_2, p_3 and p_4 . The predicted output sequence from these four individual components are then fed into the time distributed layer that further applies the fully-connected or dense layer to every temporal slice of the output sequence. After this, the final output sequences from the time-distributed layers are fused with a weighted parametric layer that looks for the optimum set of contributions that results in accurate future blood glucose values.

Parametric Fusion Layer

In the proposed model architecture, we have four different components that accounts for different histories of variable combinations for predicting the sequence output. We compute final predicted output sequence by fusing the sequence outputs of different components of the model with associated learnable component weighted parameters as below:

$$\hat{\mathbf{Y}} = \mathbf{W}_1 \odot \hat{\mathbf{Y}}_1 + \mathbf{W}_2 \odot \hat{\mathbf{Y}}_2 + \mathbf{W}_3 \odot \hat{\mathbf{Y}}_3 + \mathbf{W}_4 \odot \hat{\mathbf{Y}}_4$$

Here, $\hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2, \hat{\mathbf{Y}}_3$ and $\hat{\mathbf{Y}}_4$ are the predicted sequence output coming out of the four component of the model while $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ and $\mathbf{W}_4$ are the trainable weight parameters that indicates the degree of influence that each of the component has on the final sequence prediction.

Loss Function

We used Huber loss as the objective function for the proposed model. Huber loss is less sensitive to outliers unlike mean squared error. It is defined as:

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

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ are the predicted and actual sequence output. Huber loss is the combination of squared loss and absolute loss in its definition. It becomes quadratic when the error is small else it is linear. The control of boundary that defines the transition from quadratic to linear is done by tuning of hyper-parameter δ . For our experiments we set $\delta = 0.5$.

To improve predictions, we have also incorporated the weighted values for different events. For example, we penalize the cost of incorrect hypoglycemic values by multiplying it with weight, denoted by λ_{hypo} ; the cost of penalizing the incorrect hypoglycemic values is taken by multiplying it with λ_{hyper} with the cost of incorrect normoglycemic values being denoted by λ_{normo} . The illustration of adding the additional cost to the sample sequence is shown in Fig. 7. For our experiments, we use $\lambda_{\text{hypo}} = 100, \lambda_{\text{hyper}} = 10$ and $\lambda_{\text{normo}} = 1$

Experimental Set-up

The proposed model was tested on actual blood glucose data for 97 patients as discussed in Data section. The experiments were conducted on a computing cluster available through Centre for Computational Research (CCR) in University at Buffalo. The nodes were equipped with NVIDIA Tesla V100 GPUs with 16GB memory. We used *Keras* library³² with the *Tensorflow* library³³ as backend.

Data Preprocessing

The data has been normalized using Min-Max normalization method to normalize or scale the data into the range $[0, 1]$ before feeding into the model. The output predictions from the model are then de-normalized or re-scaled back to permit comparing and evaluating with groundtruth values using different performance measures. This is an important step since normalizing the values is required before feeding into the model while de-normalizing is required for correct evaluation of model predictions.

Figure 7. Weighted Loss Function

Training and Parameter Learning

Parameters of the proposed model are learned through back propagation through time (BPTT) with **Adam** as optimizer with learning rate of 0.0001. The training batch size is set to 32. For the 1D convolution layers, the kernel size for the convolution layers is set to 2 while the number of filters is set to 64. The dilation factors d in these layers is changed with the following values – [1, 2, 4, 8, 16, 32]. The 1D spatial dropout is set to 0.2. We use one stack of residual block in each of the encoder and decoder module within each each of four components of our model. Based on the sensitivity analysis of history size and impact of each input variables on the forecasting skill of the model, we have set these values as $p_1 = p_3 = p_4 = 36$ and $p_2 = 6$, since $36 * 5 = 180$ minutes and $6 * 5 = 30$ minutes. We choose the past history of first, third and fourth component as 36 hours to account for a better long-term dependencies. The second component has the input of recent history of 30 minutes for BG as the recent values have great impact on the values that are followed in the output sequence.

Evaluation

In this subsection, we describe the evaluation measures used for evaluating the performance of our model. The performance evaluation metrics is the key indicator of the prediction power of a developed model.

Quantitative evaluation metrics

The performance metrics used in our experiment for evaluating the prediction accuracy of the proposed model are listed as below:

1. **Root Mean Squared error (RMSE)**

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}$$

2. **Mean Absolute error (MAE)**

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

3. **Mean Absolute Percentage error (MAPE)**

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} * 100\%$$

4. **Normalized Root Mean Squared Error (NRMSE)**

$$NRMSE = \frac{RMSE}{y_{\max} - y_{\min}}$$

Here, N is the number of BGC values, $\hat{y}_i$ and y_i are the predicted and actual BGC values respectively, while $y_{\max}$ and $y_{\min}$ are the maximum and minimum of the actual BGC values respectively. These evaluation metrics describes the quantitative accuracy for blood glucose predictions. While they are good indicators of the model's performance quantitatively, they are unable to clearly represent the qualitative performance like clinical accuracy of modeled blood glucose meter. In order to represent this qualitative measure of the model, we make use of error grid analysis which is described in the following section.

Error Grid Analysis

To measure the clinical accuracy of our model predictions, we used both the Parkes³⁴ and the Clarke³⁵ error grid system. In each of these grid systems, the errors are categorized into zones and each zone covers a certain range of prediction and reference values. The pairing of the corresponding values in the actual blood glucose and predicted blood glucose levels is plotted in the error grid. Each of these pairs falls in one of the zones in the error grid. Each zone represents the degree of risk of an outcome arising due to the difference in error between predicted and actual values. The performance of the model can be best assessed by having more number of pairs of predicted and actual points appearing in Zone A. Zone A is defined as the grid region where predicted values fall within $\pm 20\%$ of the actual data. In clinical practice, predictions which fall within this Zone would not be expected to alter clinical action by the patient.

Conclusion and Future work

We present here a novel deep learning based model to predict future BG of T1D patients in a multi-step ahead manner. This model takes the past history of BG, insulin (basal and bolus) and meal intake in order to predict the multi-step ahead BG values. The novelty of the the proposed model lies in accounting of different view of past history for different variables like meal intake, basal rate and insulin boluses. Results with validation data for 97 T1D patients demonstrate that our proposed model performs consistently on both qualitative and quantitative metrics. Moreover, our proposed model outperforms other competitive and baseline approaches. In the future, we would incorporate other patient-related information to further enhance the prediction accuracy. The other direction can be considered in incorporating a probabilistic layer that can quantify the uncertainty of input and output.

Notes

1. IAF diabetes atlas, 9th edition 2019, <https://diabetesatlas.org/en/>
2. More information on <https://www.tidepool.org/>
3. More information on <https://www.jdrf.org/>
4. More information on <https://www.tidepool.org/bigdata>
5. More information on <https://www.hhs.gov/ohrp/regulations-and-policy/regulations/45-cfr-46/common-rule-subpart-a-46104/index.html>

References

1. Bazaev, N., Pletenev, A. & Pozhar, K. Classification of factors affecting blood glucose concentration dynamics. *Biomed. Eng.* **47** (2013).
2. Oviedo, S., Vehí, J., Calm, R. & Armengol, J. A review of personalized blood glucose prediction strategies for t1dm patients. *Int. journal for numerical methods biomedical engineering* **33**, e2833 (2017).
3. Contreras, I. & Vehí, J. Artificial intelligence for diabetes management and decision support: literature review. *J. medical Internet research* **20**, e10775 (2018).
4. Rigla, M., García-Sáez, G., Pons, B. & Hernando, M. E. Artificial intelligence methodologies and their application to diabetes. *J. diabetes science technology* **12**, 303–310 (2018).
5. Frandes, M., Timar, B., Timar, R. & Lungeanu, D. Chaotic time series prediction for glucose dynamics in type 1 diabetes mellitus using regime-switching models. *Sci. Reports* **7**, 6232 (2017).
6. Foster, N. C., Miller, K. M., Tamborlane, W. V., Bergenstal, R. M. & Beck, R. W. Continuous glucose monitoring in patients with type 1 diabetes using insulin injections. *Diabetes Care* **39**, e81 (2016).
7. Pickup, J. Insulin pumps. *Diabetes technology & therapeutics* **16**, S–17 (2014).
8. Blake, D. R. & Nathan, D. M. Point-of-care testing for diabetes. *Critical care nursing quarterly* **27**, 150–161 (2004).
9. Yang, J., Li, L., Shi, Y. & Xie, X. An arima model with adaptive orders for predicting blood glucose concentrations and hypoglycemia. *IEEE journal biomedical health informatics* **23**, 1251–1260 (2018).
10. Bunescu, R., Struble, N., Marling, C., Shubrook, J. & Schwartz, F. Blood glucose level prediction using physiological models and support vector regression. In *2013 12th International Conference on Machine Learning and Applications*, vol. 1, 135–140 (IEEE, 2013).
11. Hamdi, T. *et al.* Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm. *Biocybern. Biomed. Eng.* **38**, 362–372 (2018).

12. Ali, J. B. *et al.* Continuous blood glucose level prediction of type 1 diabetes based on artificial neural network. *Biocybern. Biomed. Eng.* **38**, 828–840 (2018).
13. Bertachi, A., Biagi, L., Contreras, I., Luo, N. & Vehí, J. Prediction of blood glucose levels and nocturnal hypoglycemia using physiological models and artificial neural networks. In *KHD@IJCAI*, 85–90 (2018).
14. Sun, Q., Jankovic, M. V., Bally, L. & Mougiakakou, S. G. Predicting blood glucose with an lstm and bi-lstm based deep neural network. In *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, 1–5 (IEEE, 2018).
15. Mhaskar, H. N., Pereverzyev, S. V. & van der Walt, M. D. A deep learning approach to diabetic blood glucose prediction. *Front. Appl. Math. Stat.* **3**, 14 (2017).
16. Daskalaki, E., Prountzou, A., Diem, P. & Mougiakakou, S. G. Real-time adaptive models for the personalized prediction of glycemic profile in type 1 diabetes patients. *Diabetes technology & therapeutics* **14**, 168–174 (2012).
17. Vaswani, A. *et al.* Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010 (2017).
18. Bai, S., Kolter, J. Z. & Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).
19. Marcus, Y. *et al.* Improving blood glucose level predictability using machine learning. *Diabetes/Metabolism Res. Rev.* e3348 (2020).
20. Mirshekarian, S., Bunescu, R., Marling, C. & Schwartz, F. Using lstms to learn physiological models of blood glucose behavior. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2887–2891 (IEEE, 2017).
21. Lea, C., Flynn, M. D., Vidal, R., Reiter, A. & Hager, G. D. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 156–165 (2017).
22. Bengio, Y., Simard, P., Frasconi, P. *et al.* Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* **5**, 157–166 (1994).
23. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256 (2010).
24. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034 (2015).
25. Li, H., Xu, Z., Taylor, G., Studer, C. & Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, 6389–6399 (2018).
26. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
27. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
28. Tompson, J., Goroshin, R., Jain, A., LeCun, Y. & Bregler, C. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 648–656 (2015).
29. Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112 (2014).
30. Prabhavalkar, R. *et al.* A comparison of sequence-to-sequence models for speech recognition. In *Interspeech*, 939–943 (2017).
31. Venugopalan, S. *et al.* Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, 4534–4542 (2015).
32. Chollet, F. *et al.* Keras (2015).
33. Abadi, M. *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems (2015). Software available from tensorflow.org.
34. Parkes, J. L., Slatin, S. L., Pardo, S. & Ginsberg, B. H. A new consensus error grid to evaluate the clinical significance of inaccuracies in the measurement of blood glucose. *Diabetes care* **23**, 1143–1148 (2000).
35. Clarke, W. L., Cox, D., Gonder-Frederick, L. A., Carter, W. & Pohl, S. L. Evaluating clinical accuracy of systems for self-monitoring of blood glucose. *Diabetes care* **10**, 622–628 (1987).

Acknowledgements

The authors would like to acknowledge and appreciate the efforts made by JDRF, a non-profit organization, in providing the datasets for the Type-1 diabetic patients. This work was partially supported by JDRF (1-SRA-2019-823-S-B) and the National Science Foundation (Directorate of Engineering - CMMI 1537210 and Office of Advanced Cyberinfrastructure - OAC 1910539). Computing facilities were provided by the University at Buffalo's Center for Computational Research.

Author contributions statement

S.M.A.Z., V.C., M.I., T.R.S., L.D.M. and B.R. have contributed to the writing of this paper. S.M.A.Z., V.C. and T.R.S. proposed the predictive model. S.M.A.Z. implemented the model and conducted the experiments. M.I. and B.R. performed the data processing. M.I. contributed to the comparison of the proposed model with other approaches. V.C., T.R.S. and L.D.M. contributed in writing reviewing and editing. The authors have reviewed the manuscript before submission.

Additional information

Competing Interests: The authors declare that they have no competing interests.

Figures

Figure 1

30 minutes forecasting on validation data for two T1D patients

Figure 2

Flowchart for processing Tidepool data

Figure 3

Dilated Causal Convolutions with dilation factor $d = 1, 2, 4$, and kernel size, $k = 3$

Figure 4

Residual Block in Temporal Convolutional Network

Figure 5

Sequence-to-Sequence approach

Figure 6

Proposed model

Figure 7

Weighted Loss Function

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Supplementary.pdf](#)