Neural Computing and Applications (2021) 33:4191–4203
https://doi.org/10.1007/s00521-020-05248-0

ORIGINAL ARTICLE

Long short-term memory neural network for glucose prediction

Jaime Carrillo-Moreno[1] · Carmen Pérez-Gandía[1,2] · Rafael Sendra-Arranz[1] · Gema García-Sáez[1,2] · M. Elena Hernando[1,2] · Álvaro Gutiérrez[1]

Received: 27 March 2020 / Accepted: 25 July 2020 / Published online: 3 August 2020

Abstract

Diabetes is a chronic disease that affects a high percentage of the world population and produces different and serious complications to patients. Most diabetes complications may be avoided by controlling the blood glucose levels exhaustively. Moreover, a prediction of future glucose levels has shown to be fundamental in helping patients to plan and modify their treatment in real-time. In this paper, a glucose predictor based on long short-term memory neural networks is designed. Three input parameters are fed to the predictor: past glucose levels obtained from a continuous glucose monitoring sensor, the insulin units administered by an insulin pump and the patient's carbohydrates intake. Different prediction times and input dimensions have been evaluated in order to provide the best prediction to patients. Results encourage the use of glucose predictions to avoid the occurrence of hypoglycemias, anticipate correction actions, and to increase the quality of life of these patients.

Keywords Artificial neural network · Long short-term memory (LSTM) · Type 1 diabetes · Times-series forecasting · Glucose prediction

1 Introduction

Diabetes mellitus (DM), also known as diabetes, is a disease mainly distinguished by high blood glucose levels [1]. It is classified in four groups: type 1, type 2, gestational and other specific types of diabetes [2]. The disease is mainly related to insulin, one of the hormones that regulates blood glucose concentration in the body.

Diabetes is an important health care problem worldwide. According to the last available global report by the World Health Organization [3], around 422 million adults were diagnosed with diabetes in 2014. The disease's prevalence and the number of cases of diabetes continues to increase annually, expecting to reach 693 million by 2045 [4]. In 2012, diabetes was found to be the direct cause in 1.5 million deaths, in addition to 2.2 million deaths caused indirectly by complications of the disease [3]. Heart attacks, strokes and kidney failure are some complications that may arise from diabetes [5]. These complications are very costly to healthcare systems and could be drastically reduced with better control of the disease [6–8].

Type 2 diabetes is the most common form of diabetes. However, type 1 diabetes, the second most common, requires more attention during daily activities. The treatment to manage type 1 diabetes is based on the exogenous administration of insulin supported by exhaustive blood glucose monitoring, with which most of the complications can be prevented or minimized [9]. In addition to insulin therapy administration, nutritional therapy and physical

✉ Álvaro Gutiérrez
a.gutierrez@upm.es

Jaime Carrillo-Moreno
jaime.carrillo.moreno@alumnos.upm.es

Carmen Pérez-Gandía
cperez@gbt.tfo.upm.es

Rafael Sendra-Arranz
r.sendra@alumnos.upm.es

Gema García-Sáez
gema.garcia.saez@upm.es

M. Elena Hernando
mariaelena.hernando@upm.es

[1] E.T.S. Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain

[2] Centro de Investigación Biomédica en Red de Bioingeniería, Biomateriales y Nanomedicina (CIBER-BBN), Madrid, Spain

exercise are essential for controlling the blood glucose levels [5]. Carbohydrates intake and physical exercise are the main disturbances in blood glucose concentration.

The goal of blood glucose control is to reach normoglycemia levels, normal concentration of blood glucose, minimizing the occurrences of hypoglycemia and hyperglycemia, low and high blood glucose levels, respectively. The American Diabetes Association (ADA) defines hypoglycemia as a measurable glucose concentration 70 mg/dl and hyperglycemia as glucose concentration 180 mg/dl. The 2019 ADA revision recommended the range 80–130 mg/dl as premeal glucose target [10].

Self-monitoring carried out by the patient is necessary to track and control blood glucose levels [11]. Patients need to measure their capillary glucose levels discretely, several times per day with a glucose meter, or using a continuous glucose monitoring (CGM) sensor [12, 13]. CGM sensors provide a continuous flow of information about the glucose concentration, allowing a better control of the disease in real time.

A glucose predictor would allow patients to plan their actions and therefore, deal with the disease better [14]. However, to have a useful predictor, two factors must be taken into account [15]. First, the prediction horizon (PH) has to be long enough (at least 15 min), so patients have time to adjust their treatment. Secondly, the fact that CGM systems measure glucose levels in interstitial fluid in the subcutaneous adipose tissue and not directly in blood means that there is a delay between the glucose concentration measured by CGM sensors and the blood glucose levels [16], so it is important that the PH is longer than the delay.

Artificial neural networks (ANNs) have been widely used to address forecasting problems in different fields. ANNs excel in comparison with traditional algorithms for time series forecasting because they are able to infer the complex relationships of the data and its nonlinearity [17–20].

ANNs have been used in the past, showing good performance in the prediction of blood glucose levels. Specifically, different techniques have been followed to predict the blood glucose level, such as (MLPs) [15], (CNNs) [21] and recurrent neural networks (RNNs) [22].

Pérez-Gandía et al. [15] proposed a MLP to predict glucose in real time from two different sensors. The model only used CGM measurements as input data. The algorithm provided three different PHs for 15, 30 and 45 min. Moreover, they compared their model with the auto-regressive model (ARM) of [23], obtaining better results using the MLP. Zhu et al. [21] proposed a model that was built from CNN layers and employed WaveNet algorithms [24]. They used the type 1 diabetes OhioT1DM dataset developed by [25] with patients' glucose levels, insulin bolus, carbohydrates intake and time as input. They also used other input data such as heart rate and temperature. However, these new parameters hindered the performance of their model. A RNN, proposed by [22], used input data from a CGM sensor. They compared their RNN model and a MLP model with four different PHs: 15, 30, 45 and 60 min. As expected, the results showed that the RNN architecture was better in making predictions than the MLP model, because their feedback connections allow them to learn temporal dependencies [26].

Given the importance of short- and long-term prediction needs for diabetes patients, this paper presents a model using long short-term memory (LSTM) neural networks which extends the work of [15]. LSTMs are a type of RNNs that are able to maintain both short and long time dependencies through their states [27]. They are composed of gates that control the information that is stored in the states by altering the input and output.

Therefore, the main goal of this paper is to design and implement a set of predictors with the aim of predicting accurately the glucose level of type 1 diabetes and improving previous results in the literature. First, a main predictor model developed will consist of an LSTM fed with previous values of glucose, insulin bolus and meal intake. Based on the main predictor model, a set of predictors specialized in forecasting for different PHs will be deployed. Furthermore, the use of different (IDs) in the performance of the predictors are explored.

The paper is structured as follows. In Sect. 2, an introduction to the basic concepts of LSTM neural networks is carried out. Section 3 presents the dataset and the process followed to develop the set of predictors that are evaluated in this paper. Section 4 is dedicated to assess the set of predictors designed in Sect. 3. Section 5 provides conclusions of the paper and outlines possibilities for future research.

2 Theoretical background

LSTM neural networks were firstly introduced by [27]. LSTm is a type of RNNs developed specifically to solve the vanishing and exploding gradient problem [28, 29]. During the training of the RNNs with backpropagation through time (BPTT) [30], when the gradients are propagated over time, they tend to vanish or become unstable. This problem makes it very difficult for RNNs to learn long time dependencies. This issue is addressed by the LSTMs, being able to react to long- and short-term dependencies.

An LSTM cell (see Fig. 1) is composed of 4 layers, also called gates [26, 31]. The four gates are the forget gate (f_k), the input gate (i_k), the new cell state candidate gate ($\widetilde{c_k}$) and the output gate (o_k). Each gate has its own parameters,

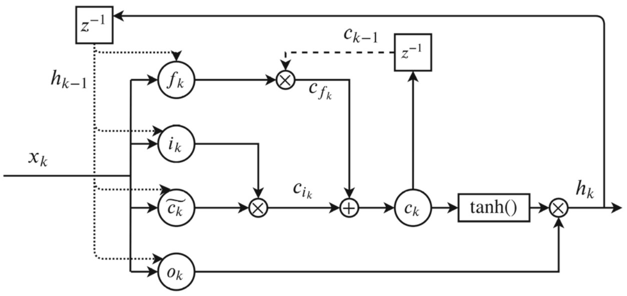

Fig. 1 Computational graph of a LSTM cell [32]

biases (b), weight for the input value (W) and weight for the hidden state value (U).

The cell state is the memory of the cell and maintains the information through time. The inputs to the four gates are the correspondent x_k vector, where k is a sequence from $k = \{1, \ldots, \tau\}$, τ being a number of time steps that belongs to $\mathbb{N}$, and the previous outputs (hidden states) from the LSTM (h_{k-1}).

The behavior of an LSTM cell is as follows:

1. The forget layer decides what percentage of the information from the cell state has to be left behind and which should remain in it. The activation function of this gate is a sigmoid (σ):

$$f_k = \sigma\left(W_f x_k + U_f h_{k-1} + b_f\right) \tag{1}$$

where W_f is the input weight matrix, x_k is the input vector, U_f is the previous output weight matrix, h_{k-1} is the previous output and b_f is the bias.

The output of the forget layer (f_k) is multiplied element-wise by the cell state (c_k).

$$c_{f_k} = f_k \circ c_{k-1} \tag{2}$$

This process is the one that erases the information of the cell state and occurs for the input and output gates.

2. The next step consists of updating the new information to the cell state, c_{i_k}. The new cell state candidate gate ($\widetilde{c_k}$) proposes the new information that might be added to the memory of the cell, with a hyperbolic tangent (tanh) as the activation function:

$$\widetilde{c_k} = \tanh\left(W_i x_k + U_i h_{k-1} + b_i\right) \tag{3}$$

On the contrary, the input gate uses a sigmoid function:

$$i_k = \sigma\left(W_i x_k + U_i h_{k-1} + b_i\right) \tag{4}$$

Next, the cell state is calculated:

$$c_{i_k} = i_k \circ \widetilde{c_k} \tag{5}$$

3. The new cell state is updated with the information provided by the two previous steps:

$$c_k = c_{f_k} + c_{i_k} \tag{6}$$

4. Next, the output gate calculates its output by means of a sigmoid activation function:

$$o_k = \sigma\left(W_o x_k + U_o h_{k-1} + b_o\right) \tag{7}$$

The cell state (c_k) is modified with a hyperbolic tangent (tanh) activation function. Finally, the output h_k is calculated as:

$$h_k = o_k \circ \tanh\left(c_k\right) \tag{8}$$

3 Design process

3.1 Dataset

The dataset includes time-dependent information about the glucose levels obtained with the Guardian® Real Time CGM sensor, the insulin data administered by each patient's insulin pump, the food intake converted into grams of carbohydrates and the capillary blood glucose levels from a glucometer (used for the calibration of the CGM sensor). The glucose levels are sampled every 5 min, while the insulin data and the food intake are adjusted to this sampling period. Therefore, all the glucose profiles have a sampling period of 5 min, meaning each profile has 288 samples per day.

The first three variables (glucose, insulin and carbohydrates intake) from the aforementioned database will be used for prediction (see Fig. 2 for an example). The fourth

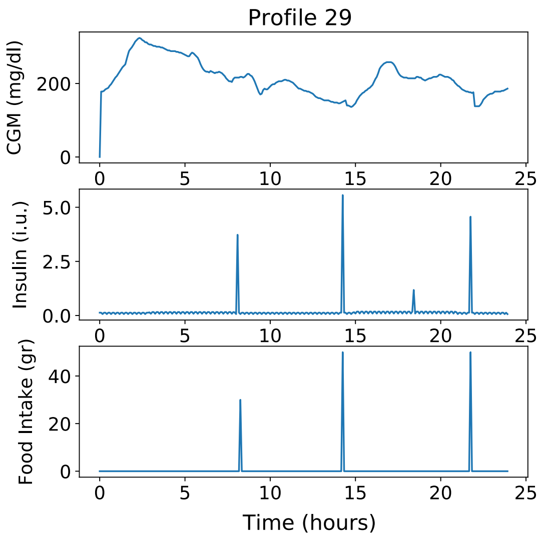

Fig. 2 Representation of the raw parameters to feed the RNN. The top graph represents the glucose values obtained by the CGM sensor, the middle graph represents the insulin units provided, and the bottom graph shows the food intake in grams of carbohydrates

variable, capillary blood glucose levels, were automatically used by the Guardian® Real Time to calibrate the output.

3.1.1 Data preprocessing

3.1.2 Glucose preprocessing

CGM profiles usually present three main artifacts that need to be fixed before the dataset can be used for glucose prediction:

1. There are missing values (values equal to zero), mainly caused by a problem with the connection between the CGM sensor and the receptor or problems that happen during calibration. A linear interpolation has been used to eliminate missing data.
2. There are non-derivative values that appear when a blood glucose calibration is entered to the CGM (see step in Fig. 3). A threshold of 30 mg/dl has been defined to compensate these distortions. Only steps greater than or equal to the threshold are compensated. The preprocessing to compensate works as follows: if a distortion greater than or equal to the threshold is found, then there is a check for an associated calibration value that has produced this distortion. If it is confirmed, a linear transformation is applied to the signal to minimize the distortion, see Eq. 9. The transformation technique is applied before and after the calibration point (CP), where the calibration is inserted into the CGM sensor measurement ($k_a = 0$). This process is applied until a number of time steps (N_p, N_a) is reached or when the distortion decreases to a threshold of 20 mg/dl.

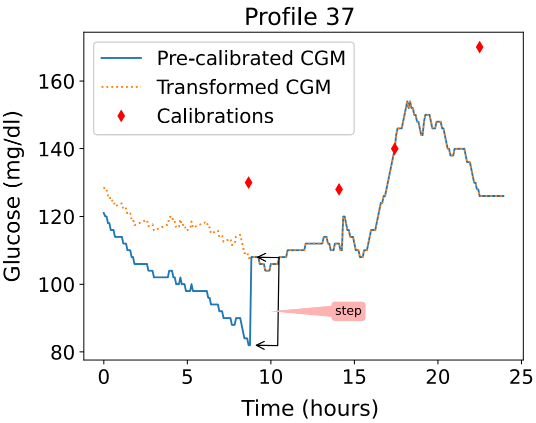

Fig. 3 Step produced by the calibration. This graph represents the glucose values from the pre-calibrated CGM sensor, the calibration measures made by glucometer and the transformed CGM after the transformation

$$\begin{cases} y'_{\mathrm{CGM}}[i_p] = y_{\mathrm{CGM}}[i_p] + \mathfrak{C}\left(\dfrac{\mathfrak{N}_\mathfrak{p} - \mathfrak{k}_\mathfrak{p}}{\mathfrak{N}_\mathfrak{p}}\right) \\ y'_{\mathrm{CGM}}[i_a] = y_{\mathrm{CGM}}[i_a] + \mathfrak{C}\left(\dfrac{\mathfrak{N}_\mathfrak{a} - \mathfrak{k}_\mathfrak{a}}{\mathfrak{N}_\mathfrak{a}}\right) \end{cases} \tag{9}$$

where

$$\begin{cases} i_p = (\mathrm{CP} + k_p) \\ i_a = (\mathrm{CP} - k_a) \\ k_p = \{1, \ldots, N_p\} \\ k_a = \{0, \ldots, N_a\} \end{cases} \tag{10}$$

where N_p is the number of time steps where the transformation is going to be applied before the CP, whereas N_a corresponds to the number of time steps after the CP. The corrective measure ($\mathfrak{C}$) is applied to the CGM signal from the CP and its effect decreases the further it moves from that point. $\mathfrak{C}$ is directly proportional to the difference between the glucometer calibration measure $\left(y[\mathrm{CP}]_{\mathrm{calibration}}\right)$ and the CGM value $\left(y[\mathrm{CP}]_{\mathrm{CGM}}\right)$, see Eq. 11.

$$\mathfrak{C} = \theta\left(\mathfrak{y}[\mathfrak{CP}]_{\mathrm{calibration}} - \mathfrak{y}[\mathfrak{CP}]_{\mathrm{CGM}}\right) \tag{11}$$

where θ is a constant that corresponds with the percentage of the corrective measure that is going to be applied to the signal. Its value is chosen between 0 and 1. Therefore, θ, N_p and N_a are selected for each profile to minimize distortion of the original signal, maintaining its trend and shape. Figure 3 shows an example of the results of the process compared to the pre-calibrated CGM signal and the calibration measures. When performing this process, the 57th profile, out of 58, showed not to have calibration data. Therefore, this profile has been dropped from the study because distortions could not be corrected.
3. Finally, there is noise in the CGM signal. To reduce the CGM signal noise and smooth the signal, a moving average filter is applied:

$$\hat{x}[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[k-n] \tag{12}$$

where N is the length of the window, x is the raw signal and $\hat{x}$ is the filtered signal. The moving average filter is a finite impulse response (FIR) filter. It only depends on the input parameters, and it is a time-dependent filter. For this application, the selected filter should fulfill the requirement of keeping the relevant information in the filtered signal. However, the filtering process causes a delay in the output signal and a potential loss of information. Taking all this into account, the chosen moving average filter has a length of $N = 5$.

3.1.3 Insulin preprocessing

Insulin pumps try to emulate the functioning of a healthy pancreas with two kinds of insulin injections: basal insulin bolus and preprandial insulin bolus. The basal profile is intended to emulate the insulin secretion of the pancreas in fasting periods, and the preprandial insulin bolus is intended to control hyperglycemia after each meal. The preprandial bolus is several orders of magnitude bigger than the basal micro-boluses, usually infused every 3–5 min. The preprandial insulin boluses are represented as an impulse function (see Fig. 2). This may produce the ANN to ignore the value of these boluses of insulin or another unforeseen behavior. Therefore, a preprocessing technique is needed.

The chosen solution is to transform the preprandial bolus of insulin, that is an impulse function, into a rectangular function with the same area. The postprandial bolus is distributed during 12 samples that correspond to 1 h, simulating the delivery of an insulin bolus over an extended period of time (known as the extended bolus in insulin infusion systems), which avoids a high initial dose of insulin and allows an extended insulin action [33].

3.1.4 Food intake preprocessing

The food intake was manually provided by the patient. These values were transformed to grams of carbohydrates, representing the amount of carbohydrates eaten by the patient in the meal. Each registered meal is represented with only one value. The representation of this parameter shows that the food intake is also an impulse function (see Fig. 2).

The food intake parameter is transformed into a rectangular function with the same area that the impulse function with 12 samples equivalent to 1 h. The value of 1 h has been defined as the average absorption time of glucose for the different types of carbohydrates.

3.1.5 Normalization

LSTMs have sigmoid and tanh activation functions in their gates (see Sect. 2). The value of the sigmoid output is in the range between 0 and 1, while the tanh function range is between -1 and 1. Therefore, all input parameters to the ANN have been normalized in the range of 0–1 to prevent unforeseen results and to facilitate the proper learning of the neural network. This process is also done to the targets of the ANN model. The min–max feature scaling is applied to normalize the input parameters and targets. Figure 4 shows the result of all the preprocessing techniques applied in the dataset.

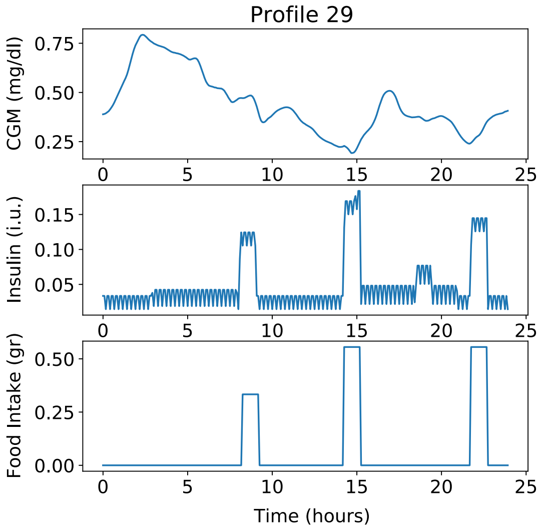

Fig. 4 Parameters after the application of the preprocessing techniques. The top graph represents the glucose values obtained by the CGM sensor. The middle graph represents the insulin units provided, while the bottom graph shows the food intake in grams of carbohydrates

3.1.6 Data arrangement

The data are to be arranged in input vectors, $\mathfrak{I} = \{\mathfrak{I}, \mathfrak{I}, \ldots, \mathfrak{I}_{\mathfrak{n}}\}$, and targets vectors, $\mathfrak{T} = \{\mathfrak{T}, \mathfrak{T}, \ldots, \mathfrak{T}_{\mathfrak{n}}\}$, where n is the number of tuples of input and target vectors in the dataset. The input time series are: glucose ($\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$), insulin ($\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n\}$) and food intake ($z = \{z_1, z_2, \ldots, z_n\}$). These values are converted in a supervised learning problem using the sliding window or windowing method [34, 35], and it has been applied as follows:

- The ID, also known as the sequence of values, determines how many past values of glucose, insulin or carbohydrates intake are fed as an input to the predictor. The PH states the time at which the prediction will be made in minutes.
- The past values of glucose are used as input to the predictor, while the future values of glucose are the target of the predictor. Therefore, the time series have to be re-arranged in input and target values. The insulin and the carbohydrates intake parameters are only inputs to the system, so they only have to be re-arranged as input values.
- $\mathfrak{I}_{\mathrm{i}}$ represents an input time series to the predictor at a specific time instant, and $\mathfrak{T}_{\mathrm{i}}$ is the corresponding future glucose value for that input. For specific values of ID and PH, every input ($\mathfrak{I}_{\mathrm{i}}$) is arranged as a sequence of $\tau = ID - 1$ values in the past, that is:

$$\mathfrak{I}_{\mathrm{i}} = \begin{bmatrix} \mathfrak{x}[\mathfrak{k}-\tau] & \dots & \mathfrak{x}[\mathfrak{k}-] & \mathfrak{x}[\mathfrak{k}] \\ \mathfrak{w}[\mathfrak{k}-\tau] & \dots & \mathfrak{w}[\mathfrak{k}-] & \mathfrak{w}[\mathfrak{k}] \\ \mathfrak{z}[\mathfrak{k}-\tau] & \dots & \mathfrak{z}[\mathfrak{k}-] & \mathfrak{z}[\mathfrak{k}] \end{bmatrix} \tag{13}$$

The correspondent target ($\mathfrak{T}_{\mathrm{i}}[\mathfrak{k}]$) is the glucose value for the expected PH in the future, that is:

$$\mathfrak{T}_{\mathrm{i}} = \mathfrak{x}[\mathfrak{k} + \mathfrak{P}\mathfrak{H}] = \mathfrak{y}[\mathfrak{k}] \tag{14}$$

The dataset is divided into three smaller subsets: training, validation and the test datasets. It is important to note that the data belonging to each patient should be only in one of the datasets, because these data are correlated among them and this could bring misleading results as the data would be biased. Taking all of this into consideration, the different subsets are distributed as follows: training dataset has 36 profiles from five patients, validation dataset has 6 profiles from one patient and test dataset has 15 profiles from two patients.

3.2 Neural network architecture

The chosen framework to implement the predictor is *PyTorch* [36]. *PyTorch* is an open source platform to develop neural network models, and it is deeply integrated with Python.

A predictor example, shown in Fig. 5, is composed of three layers: two LSTM layers and one final linear layer. The LSTM layers (see Sect. 2) have a variable number of neurons to be optimized, and the linear layer is a single perceptron with only one neuron and a linear activation function. Figure 5 also presents the unfolding and the data flow of the model. First, the input sequence ($\mathfrak{I}_{\mathrm{i}}$) comes in the first LSTM layer. In this layer, the unfolding is processed and the output (h_k^1) is the input to the following LSTM layer. Note that the superscript denotes the layer of the model. The second LSTM layer processes the information and provides its output for the last value of the sequence, h_τ^2 to the linear layer. Finally, the linear layer computes the h_τ^2 to produce the expected value of glucose concentration (y_{PH}) at the desired PH.

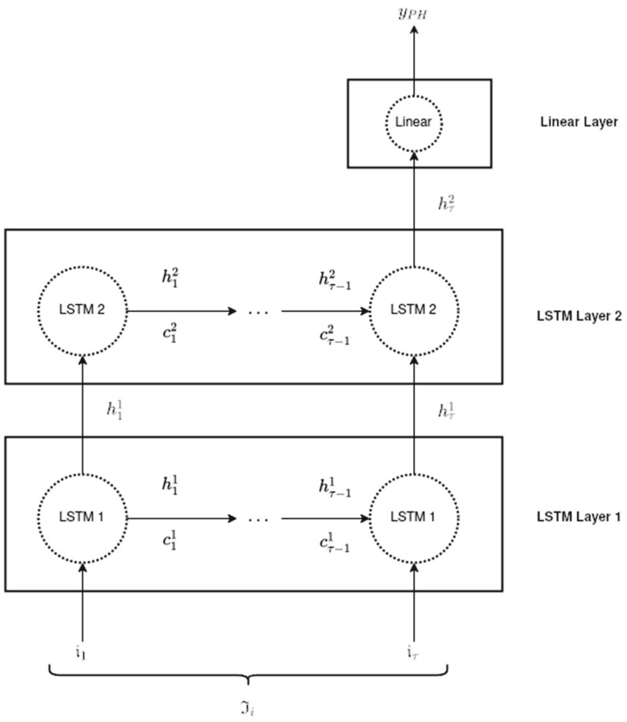

Fig. 5 Generic architecture version of the predictor proposed

3.3 Parameter optimization

All IDs and PHs are combined to search the parameters that produce the best performance. In this paper, three values of ID are set for the network optimization: 5, 20 and 50 samples, defining a small, medium and large input vector. The search of the optimal parameters is divided in two parts, the architecture parameters and the parameters related to the training of the proposed predictor.

For the architecture optimization, the output linear layer will not be modified and its structure remains in one layer with one neuron. Therefore, during this Section when any reference is made to the number of layers or neurons, it refers to the LSTM layers of the model.

All LSTM layers will have the same number of neurons. The number of neurons that is considered in this study is from 10 to 100 in steps of 10. The number of LSTM layers is 1, 2 and 3. This gives a total of 30 combinations that are searched to find the best performance for the 3 IDs (5, 20 and 50 samples) and four PH (5 min, 15 min, 30 min and 45 min).

Moreover, two optimizers are chosen to train the proposed predictor: RMSprop [37] and Adam [38]. These optimizers are used to update the weights and biases of the ANN in the backpropagation algorithm. On the other hand, five different learning rates (α) are considered $\alpha = [0.0001, 0.0005, 0.001, 0.005, 0.01]$ to search the best parameters for the training of the ANN.

This parameter setup makes a total of 3600 combinations to be tested. From them, 12 different models shown in Table 1 are selected considering the best root mean square error (RMSE) value obtained in the prediction. The motivation of the decision to use this metric for parameter selection is twofold. Firstly, in contrast to Mean Square Error (MSE) it provides a statistic error that maintains the scale of the time series under assessment. Secondly, compared to other metrics such as the mean absolute error, it is sensitive to large errors that must be mitigated in potential

Table 1 Optimal parameters for each ID and PH

ID	PH (min)	LSTM layers	Neurons	Optimizer	α
5	5	2	60	Adam	0.001
5	15	3	10	Adam	0.001
5	30	2	10	Adam	0.0001
5	45	3	10	Adam	0.0005
20	5	2	80	Adam	0.0005
20	15	1	70	Adam	0.001
20	30	3	10	Adam	0.0001
20	45	3	90	Adam	0.0005
50	5	3	30	Adam	0.0001
50	15	2	10	Adam	0.0001
50	30	2	10	Adam	0.001
50	45	2	20	Adam	0.001

All these models have one output linear layer with one neuron

medical applications involving patients. Results of these models are analyzed in detail in Sect. 4.

4 Results

4.1 Metrics

For all the metrics, let the prediction be $Y = \{y_1, \ldots, y_n\}$, the expected output or targets $T = \{t_1, \ldots, t_n\}$, and N the number of samples to be evaluated.

The MSE has been used as the loss function to be minimized, see Eq. 15. In the backpropagation algorithms, it is typically used to optimize artificial neural networks. The MSE simplifies the gradient calculations, and it is a globally differentiable function and provides an optimal and computationally simple optimization improvement.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2 \qquad (15)$$

Nonetheless, the following metrics are used in the evaluation of the performance of the proposed predictor:

- The RMSE, see Eq. 16:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2} \qquad (16)$$

where t_i is the expected output (target) and y_i is the prediction made by the model. The RMSE, unlike the MSE, presents the error in the same unit as the variables that it is evaluating. Additionally, it is the most frequently used metric to quantify accuracy for glucose prediction models in the literature [15, 23, 39, 40].

- The Pearson correlation coefficient ($\mathrm{r}_{t,y}$), see Eq. 17:

$$\mathrm{r}_{t,y} = \frac{\sum_{i=1}^{N}(t_i - \bar{t})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(t_i - \bar{t})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \qquad (17)$$

where $\bar{t}$ and $\bar{y}$ are the mean defined as $\bar{t} = \frac{1}{N}\sum_{i=1}^{n} t$ and $\bar{y} = \frac{1}{N}\sum_{i=1}^{n} y$. This coefficient gives information about the similarities between the target signal and the predicted signal. The Pearson correlation coefficient has been used previously by other authors, complementing RMSE to evaluate prediction models' accuracy [41–43].

- The delay associated with the prediction. The purpose of this metric is to assess the lag produced in the prediction with respect to the original glucose signal from the CGM sensor. The presence of delay reduces the accuracy when it is calculated as RMSE and reduces the expected PH. Therefore, it is necessary to calculate the delay to characterize the performance of a glucose predictor model. Several authors calculate this metric [15, 23, 41, 42, 44–46], although the algorithm to set the delay is not exactly the same for all them. To estimate the delay associated with prediction, the concept of mean delay presented by [23, 44] is extended by calculating every sample between the 25% and 75% of the slope length. The slope length is defined as the difference between the lowest point and the highest point. This metric measures the delay in the positive and negative slopes individually. Therefore, there are two values for this metric, one corresponds with the rising trend (upward delay) and the other with the falling trend (downward delay). The upward and downward delay are finally computed as the average of all rising trends and all the falling trends,

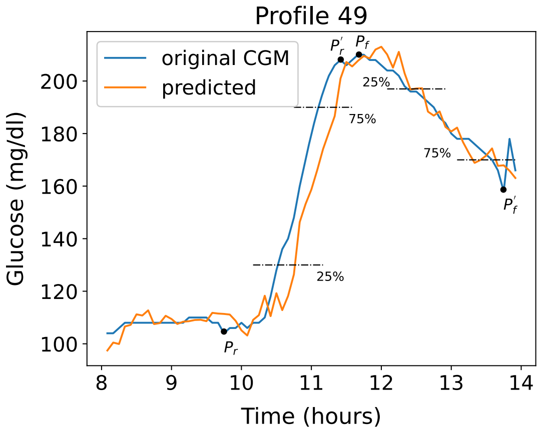

Fig. 6 Example of the delay calculation, each percentage is calculate for the length of each slope. For the rising slope, P_r is the lowest point and P_r' is the highest point. In the falling slope, P_f is the highest point and P_f' is the lowest point

respectively. Figure 6 shows an example of the slope points at which the delay is calculated.

4.2 Results

In Sect. 3.3, twelve predictors were tuned to predict the glucose concentration by using three input parameters. Each of the twelve predictors are specialized for an ID and a PH (see Table 1).

This study is divided in two parts: (1) an evaluation of the accuracy of the 12 different predictors for their corresponding PH; and (2) assessment of the effect of the prediction when different sequences of previous glucose, insulin and carbohydrates intake values or IDs are entered as input.

In the first part, the twelve predictors are evaluated with the test dataset. The process followed to prevent overfitting, called early stopping, stops the training when the performance for the validation dataset begins to worsen [47]. The results from these metrics are shown in Tables 2 and 4 for the training and test dataset, respectively. In addition, the predictions for a glucose profile of the test dataset for each model are displayed.

It is important to note that the prediction starts once the first ID values are fed. An example of this is presented in Fig. 7 with an ID of 5 and a PH of 45 min. The model starts to predict when it receives the first 5 values. Therefore, the samples are separated by 5 min, the model of an ID of 5 will wait 25 min to begin the predictions. Once the model has generated the first prediction, the model will output a value for each new input received.

First, it is important to verify that the models do not incur in overfitting. Tables 2, 3 and 4 show the results for the training, validation and test datasets, respectively. The RMSE is very similar for the three datasets, being slightly lower for the validation dataset. There are some exceptions where the training dataset is slightly lower, these cases are the models for IDs 20 and 50 with PH 45 min. The other metrics have similar results for the three datasets.

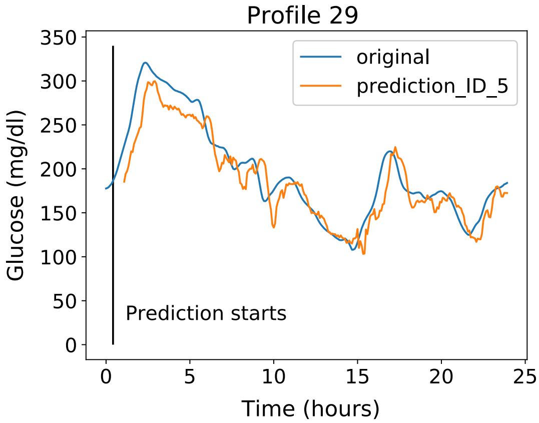

Fig. 7 Example of the effect of an ID of 5 in the prediction

There are three models specialized in the prediction of each PH. Firstly, we assess the performance of the models that predict with a PH of 5 min. Figure 8 shows the prediction made by the predictor with IDs 5, 20 and 50 for a specific glucose profile. These three models have a high accuracy with error lower than 1 mg/dl, see Table 4, and the time series are almost identical. No delay is obtained from the output of these models.

The performance of the models with a PH of 15 min for the same glucose profile is shown in Fig. 9. These models improve the RMSE results presented in [15] for a PH of 15 min in a 55.44%, 56.57% and 62.32%, for the ID of 5, 20 and 50, respectively. Moreover, Fig. 9 shows that the predicted glucose signals resemble with the original values

Table 2 Metrics for all the models for the training dataset

Model	ID	PH (min)	RMSE (mg/dl)	$r_{t,y}$	U. delay (min)	D. delay (min)
60-60-1	5	5	1.03	0.99	0	0
10-10-10-1	5	15	4.4	0.99	5	5
10-10-1	5	30	12	0.96	15	10
10-10-10-1	5	45	20.06	0.9	30	15
80-80-1	20	5	0.9	0.99	0	0
70-1	20	15	4.39	0.99	5	0
10-10-10-1	20	30	11.97	0.96	15	5
90-90-90-1	20	45	19.6	0.9	25	10
30-30-30-1	50	5	0.86	0.99	0	0
10-10-1	50	15	3.66	0.99	5	0
10-10-1	50	30	11.59	0.96	15	5
20-20-1	50	45	17.94	0.90	25	10

The models are identified by the numbers of neurons in each layer and the ID is the input dimension. U. delay and D. delay are defined as the upwards delay and the downward delay, respectively

Table 3 Metrics for all the models for the validation dataset

Model	ID	PH (min)	RMSE (mg/dl)	$r_{t,y}$	U. delay (min)	D. delay (min)
60-60-1	5	5	0.95	0.99	0	0
10-10-10-1	5	15	4.14	0.99	0	5
10-10-1	5	30	11.51	0.96	10	15
10-10-10-1	5	45	18.79	0.89	25	25
80-80-1	20	5	0.79	0.99	0	0
70-1	20	15	3.86	0.99	0	5
10-10-10-1	20	30	11.37	0.96	10	10
90-90-90-1	20	45	20.66	0.89	10	25
30-30-30-1	50	5	0.85	0.99	0	0
10-10-1	50	15	3.59	0.99	5	5
10-10-1	50	30	11.72	0.96	10	15
20-20-1	50	45	19.51	0.87	25	25

The models are identified by the numbers of neurons in each layer and the ID is the input dimension. U. delay and D. delay are defined as the upwards delay and the downward delay, respectively

Table 4 Metrics for all the models for the test dataset

Model	ID	PH (min)	RMSE (mg/dl)	$r_{t,y}$	U. delay (min)	D. delay (min)
60-60-1	5	5	0.96	0.99	0	0
10-10-10-1	5	15	4.34	0.99	5	5
10-10-1	5	30	12.6	0.96	15	10
10-10-10-1	5	45	21.40	0.89	30	15
80-80-1	20	5	0.86	0.99	0	0
70-1	20	15	4.23	0.99	5	0
10-10-10-1	20	30	12.51	0.96	15	5
90-90-90-1	20	45	21.52	0.89	25	20
30-30-30-1	50	5	0.85	0.99	0	0
10-10-1	50	15	3.67	0.99	5	0
10-10-1	50	30	12.20	0.96	15	10
20-20-1	50	45	20.76	0.88	25	15

The models are identified by the numbers of neurons in each layer and the ID is the input dimension. U. delay and D. delay are defined as the upwards delay and the downward delay, respectively

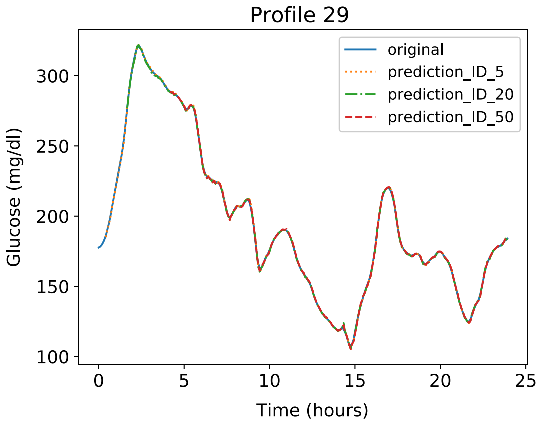

Fig. 8 Predictions made at 5 min by the models with an ID of 5, 20 and 50

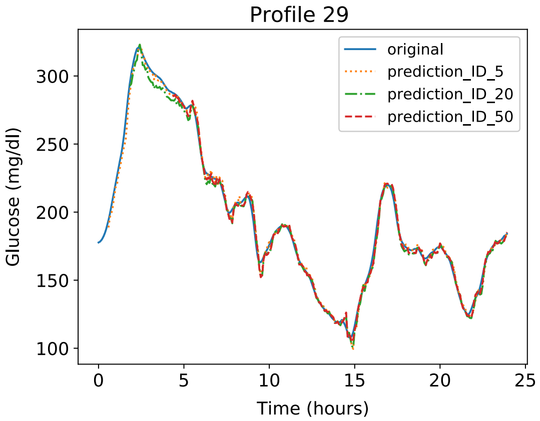

Fig. 9 Predictions made at 15 min by the models with an ID of 5, 20 and 50

from the CGM sensor. This idea is reinforced by the value of the correlation, $r_{t,y}$, that is very close to 1 (see Table 4). Furthermore, the models present a delay of 5 min in the upward slopes. However, the model is faster in the downward slopes, which explains why the models for IDs 20 and 50 have a null downward delay.

The glucose predictions made for a PH of 30 min and the same glucose profile are shown in Fig. 10. Comparing the results with those obtained in [15] for a PH of 30 min, these predictors improve the RMSE in a 27.79%, 28.31% and 30.09%, for the ID of 5, 20 and 50, respectively. The $r_{t,y}$ is very close to 1 (see Table 4), demonstrating that the predicted signal is very close to the original signal. The delays obtained are similar for all the models that predict using a PH of 30 min.

Lastly, the models with a PH of 45 min for the same profile are shown in Fig. 11. The results from these predictors, see Table 4, improve the RMSE obtained in [15] by a 17.54%, 16.98% and 20.53%, for the ID of 5, 20 and 50, respectively. The similarity between the predicted signal and the original signal from the CGM sensor have worsened with regard to the previous cases where the $r_{t,y}$ was very close to 1. The presented models are faster in the downward slopes than in the upward slopes.

After the study of the performance of all the predictors, the most accurate models are the ones with an ID of 50 (see Table 4). Moreover, these models are faster in both rising and falling slopes. This result was expected, because it is the model that receives more information in their inputs. However, the evaluation metrics in Table 4 present a similar performance between models with the same PH, independent of the ID.

Figures 8, 9, 10 and 11 also present a high similitude between the prediction at the same PH. Some differences in the model performance start to appear when the PH is 30 min or 45 min. Specifically, the differences are better identified when there is a steep slope. In these cases, the predictors with ID of 5 and 20 tend to overshoot a bit more than the predictors with IDs of 50. Nevertheless, these differences are difficult to notice and are not very remarkable.

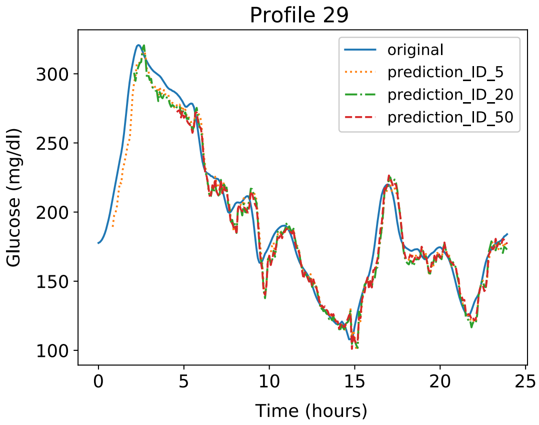

Fig. 10 Predictions made at 30 min by the models with an ID of 5, 20 and 50

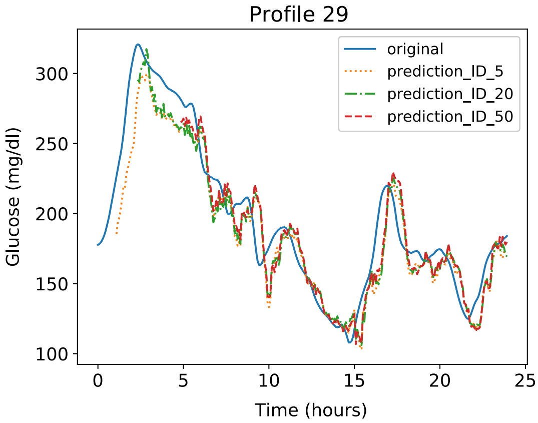

Fig. 11 Predictions made at 45 min by the models with an ID of 5, 20 and 50

4.3 Discussion

Twelve models have been deployed to achieve the objective of predicting glucose concentration in patients with type 1 diabetes. These models may be grouped by their PH: 5 min, 15 min, 30 min and 45 min.

The predictors with a PH of 5 min are the most accurate models, but they do not provide enough time to anticipate therapeutic actions to predict adverse events (hypoglycemia or hyperglycemia) due to the delayed action of insulin infusions [15]. Consequently, the predictors with a PH of 5 min are not useful in a clinical scenario.

Predictors specialized to forecast at 15 min have shown a high accuracy; they can be used to improve the glucose control supported by artificial pancreas systems, which are designed automatic and real-time adjustments of insulin delivery with no human intervention [48]. The models with a PHs of 30 min or longer are good enough to allow a patient to do the necessary adjustments in insulin delivery and consequently to prevent the occurrence of hypoglycemia or hyperglycemia events. Both predictors with a PH of 30 and 45 min will provide a sufficient time interval to anticipate the patient's action, both being useful to predict the glucose concentration. However, predictors with a PH of 30 min have the best compromise between available time interval to allow therapeutic adjustments and the accuracy of the predictions.

In Sect. 4.2, the performance of the predictors with the same PH and different ID was compared, assessing the effect of the ID in the predictor's output. It is important to consider all the characteristics of the predictors before choosing one. These characteristics are the ID, PHs and the results of the metrics evaluation. Taking all this into account, the results from Table 4 are very similar for each predictor at the same PH. However, the predictors with the larger IDs have a longer initialization time. Therefore, with all the models having similar levels of accuracy, it would be advisable to choose models with smaller IDs, because they start to predict sooner.

Nevertheless, the most accurate predictors were those with an ID of 50 that corresponds with a period of 250 min. This happens because they are the models which more information received. However, the results from the other models are similar. The improvement of adding more information in the input is small. This effect is explained hereafter for several reasons.

First, patients with type 1 diabetes does not have endogenous insulin secretion; hence, their metabolisms do not have the mechanisms that regulate the glucose concentration. The LSTM neural networks are fed with the past glucose concentration values. The predictor could be giving priority to the recent values in time because they have a bigger impact in the future values of glucose. Hence, the most important characteristic taken into account by the predictors is the glucose trend, so they tend to ignore the oldest values that have a lower impact in the trend of the signal.

The mechanisms that patients have to change the glucose concentration are mainly the injection of insulin, which diminishes the glucose value, and the consumption of carbohydrates, which increases the glucose concentration. However, the effects of both insulin and carbohydrates intakes are not immediate. This leads to another possible problem: insulin starts to act 1 h after its injection and the absorption of the carbohydrates consumed takes between 30 min and 2 h depending on the type of carbohydrates. Therefore, these are the long term dependencies that the predictor should learn to improve the prediction.

The problem is that the insulin bolus and the food intake is presented a few times in each profile, 3 or 4 times on average, corresponding with main meals. Then, considering that the training dataset has 36 daily profiles, the number of available main meals is around one hundred. This might not be a sufficient number of training samples to learn the effects of possible long term dependencies in the future glucose values. Therefore, the oldest input values are being partially ignored.

In addition, LSTM have at least four more weights and biases than a multilayer perceptron. This is translated in LSTM needing bigger datasets for training and the proposed predictors could benefit from enlarging the number of profiles.

Moreover, long term dependencies in the glucose concentration might exist. Future research with profiles longer than 1 day would be needed to assess the impact of this long term dependencies.

Nevertheless, the twelve models presented are between 17 and 62% more accurate than those obtained in [15]. In addition, the twelve models are slower in the upwards trend than the models presented in [15] while performing faster in the downwards trends. In fact, this is a desirable characteristic because diabetic patients are usually more worried about hypoglycemia events than hyperglycemia events.

5 Conclusions

The aim of this paper was to continue the work presented in previous studies [15] with the design of LSTM based predictors. The results obtained are encouraging and support the use of LSTM for predictions of glucose concentration in patients with diabetes type 1.

The evaluation of the predictors was done at four different PHs 5, 15, 30 and 45 min, and the predictors were fed with three different ID 5, 20 and 50. From this architecture, twelve predictors were deployed, each one specialized in a PH with a specific ID.

Results show that predictors with a PH of 30 min provide the best compromise between the amount of time that a patient has to modify the treatment and the performance of the model predictions. In addition, it is recommended to use an ID of 5 because it has a much shorter initialization time. Nonetheless, the rest of the models could be also used depending on the prediction needs that have to be covered.

Results presented in this paper open new lines of future work. The most important aspect is the number of patients and the amount of data. Some reasons have been proposed to explain why the different IDs give such similar results. The most probable cause is the need for more data to train the predictor based on LSTM. Nonetheless, a complementary approach using the paradigm of few-shot learning, where the models are trained with small amounts of data, is recommended (see [49, 50]).

Moreover, in this paper only three parameters are used to calculate the futures values of glucose. However, the process that regulates the glucose concentration is a complex mechanism and other parameters such as physical activity has an impact. Therefore, a future research line might be to add new parameters to feed the predictors and assess the importance of these parameters in the performance of the predictor. Finally, an automatic approach to correct the sensor artifacts and an automatic optimization of all ANN parameters would be desired.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. DeFronzo RA, Ferrannini E, Zimmet P, Alberti KGMM (2015) International textbook of diabetes mellitus, 4th edn. Wiley-Blackwell, Hoboken
2. Alberti KG, Zimmet PZ (1998) Definition, diagnosis and classification of diabetes mellitus and its complications Part 1: diagnosis and classification of diabetes mellitus provisional report of a WHO consultation. Diabet Med 15(7):539–553. https://doi.org/10.1002/(SICI)1096-9136(199807)15:7<539::AID-DIA668>3.0.CO;2-S
3. Roglic G (ed) (2016) Global report on diabetes. World Health Organization, Geneva
4. Cho NH, Shaw JE, Karuranga S, Huang Y, da Rocha Fernandes JD, Ohlrogge AW, Malanda B (2018) Idf diabetes atlas: global estimates of diabetes prevalence for 2017 and projections for 2045. Diabet Res Clin Pract 138:271–281. https://doi.org/10.1016/j.diabres.2018.02.023
5. Williams G, Pickup JC (2004) The handbook of diabetes, 3rd edn. Wiley-Blackwell, Malden
6. Bommer C, Sagalova V, Heesemann E, Manne-Goehler J, Atun R, Bärnighausen T, Davies J, Vollmer S (2018) Global economic burden of diabetes in adults: projections from 2015 to 2030. Diabet Care 41(5):963–970. https://doi.org/10.2337/dc17-1962
7. Economic costs of diabetes in the u.s. in 2017, Diabetes Care 41 (5) (2018) 917–928. arXiv:https://care.diabetesjournals.org/content/41/5/917.full.pdf, https://doi.org/10.2337/dci18-0007. https://care.diabetesjournals.org/content/41/5/917
8. Kähm K, Laxy M, Schneider U, Rogowski WH, Lhachimi SK, Holle R (2018) Health care costs associated with incident complications in patients with type 2 diabetes in Germany. Diabet Care 41(5):971–978. https://doi.org/10.2337/dc17-1763
9. Control D, C. T. of Diabetes Interventions, C. D. R. Group* (2009) Modern-day clinical course of type 1 diabetes mellitus after 30 years' duration: the diabetes control and complications trial/epidemiology of diabetes interventions and complications and pittsburgh epidemiology of diabetes complications experience (1983–2005), Arch Intern Med 169(14):1307–1316. https://doi.org/10.1001/archinternmed.2009.193
10. American Diabetes Association, Diabetes Care: 42 (Supplement 1). Diabetes Care 42(Supplement 1). http://care.diabetesjournals.org/content/42/Supplement_1
11. Benjamin EM (2002) Self-monitoring of blood glucose: the basics. Clin Diabet 20(1):45–47. https://doi.org/10.2337/diaclin.20.1.45
12. Keenan DB, Mastrototaro JJ, Voskanyan G, Steil GM (2009) Delays in minimally invasive continuous glucose monitoring devices: a review of current technology. J Diabet Sci Technol (Online) 3(5):1207–1214
13. Medtronic (2019) Guardian Sensor 3 | Medtronic Diabetes, https://www.medtronicdiabetes.com/products/guardian-sensor-3. Accessed May 27, 2019
14. Pérez-Gandía C, García-Sáez G, Subías D, Rodríguez-Herrero A, Gómez EJ, Rigla M, Hernando ME (2018) Decision support in diabetes care: the challenge of supporting patients in their daily living using a mobile glucose predictor. J Diabet Sci Technol 12(2):243–250. https://doi.org/10.1177/1932296818761457
15. Pérez-Gandía C, Facchinetti A, Sparacino G, Cobelli C, Gómez EJ, Rigla M, de Leiva A, Hernando ME (2010) Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring. Diabet Technol Therap 12(1):81–88. https://doi.org/10.1089/dia.2009.0076
16. Kulcu E, Tamada JA, Reach G, Potts RO, Lesho MJ (2003) Physiological differences between interstitial glucose and blood glucose measured in human subjects. Diabet Care 26(8):2405–2409. https://doi.org/10.2337/diacare.26.8.2405
17. Zhang GP (2012) Neural networks for time-series forecasting. In: Rozenberg G, Bäck T, Kok JN (eds) Handbook of natural computing, Ch. 14. Springer, Berlin, pp 461–477
18. Nguyen HH, Chan CW (2004) Multiple neural networks for a long term time series forecast. Neural Comput Appl 13(1):90–98. https://doi.org/10.1007/s00521-003-0390-z
19. Donate JP, Li X, Sánchez GG, de Miguel AS (2013) Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm. Neural Comput Appl 22(1):11–20. https://doi.org/10.1007/s00521-011-0741-0
20. I. Business Forecast Systems (2019) Forecast pro®. http://www.forecastpro.com/solutions/forecast-pro/. Accessed Sep 21, 2019
21. Zhu T, Li K, Herrero P, Chen J, Georgiou P (2018) A deep learning algorithm for personalized blood glucose prediction. In: Proceedings of the 2018 international joint conference on artificial intelligence, international joint conference on artificial intelligence, Freiburg, Germany, pp 1–5
22. Allam F, Nossai Z, Gomma H, Ibrahim I, Abdelsalam M (2011) A recurrent neural network approach for predicting glucose concentration in type-1 diabetic patients. In: Iliadis L, Jayne C (eds) Engineering applications of neural networks. IFIP advances in information and communication technology. Springer, Berlin, pp 254–259
23. Sparacino G, Zanderigo F, Corazza S, Maran A, Facchinetti A, Cobelli C (2007) Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series. IEEE Trans Bio-med Eng 54(5):931–937. https://doi.org/10.1109/TBME.2006.889774
24. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) Wavenet: a generative model for raw audio (2016) arXiv:1609.03499
25. Marling CR, Bunescu RC (2018) The OhioT1dm dataset for blood glucose level prediction. In: Proceedings of the 2018 international joint conference on artificial intelligence, Freiburg, Germany, pp 1–4
26. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
27. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
28. Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertain Fuzziness Knowl Based Syst 06(02):107–116. https://doi.org/10.1142/S0218488598000094
29. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166. https://doi.org/10.1109/72.279181
30. Werbos PJ (1990) Backpropagation through time: what it does and how to do it. Proc IEEE 78(10):1550–1560. https://doi.org/10.1109/5.58337
31. Olah C (Aug. 2015) Understanding LSTM networks—colah's blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed May 15, 2019
32. Sendra Arranz R (2018) Design and implementation of an HVAC consumption prediction system based on LSTM neural networks. Ph.D. Thesis, Universidad Polit'cnica de Madrid, Madrid

33. O'Connell MA, Gilbertson HR, Donath SM, Cameron FJ (2008) Optimizing postprandial glycemia in pediatric patients with type 1 diabetes using insulin pump therapy. Diabet Care 31(8):1491–1495
34. Kapoor P, Bedi SS (2013) Weather forecasting using sliding window algorithm. ISRN Signal Process 2013:1–5
35. Kotu V, Deshpande B (2019) Chapter 12—Time series forecasting. In: Kotu V, Deshpande B (eds) Data science, 2nd edn. Morgan Kaufmann, Burlington, pp 395–445
36. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in PyTorch. NIPS-W. https://openreview.net/forum?id=BJJsrmfCZ
37. Hinton G, Srivastava N, Swersky K (2012) Neural networks for machine learning—Lecture 6a: overview of mini-batch gradient descent. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. Accessed May 30, 2019
38. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980 [cs]
39. Georga EI, Protopappas VC, Polyzos D, Fotiadis DI (2015) Evaluation of short-term predictors of glucose concentration in type 1 diabetes combining feature ranking with regression models. Med Biol Eng Comput 53(12):1305–1318. https://doi.org/10.1007/s11517-015-1263-1
40. Robertson G, Lehmann ED, Sandham W, Hamilton D (2011) Blood glucose prediction using artificial neural networks trained with the AIDA diabetes simulator: a proof-of-concept pilot study. J Electr Comput Eng 2011:1–11. https://doi.org/10.1155/2011/681786
41. Gani A, Gribok A, Rajaraman S, Ward W, Reifman J (2009) Predicting subcutaneous glucose concentration in humans: data-driven glucose modeling. IEEE Trans Biomed Eng 56(2):246–254. https://doi.org/10.1109/TBME.2008.2005937
42. Daskalaki E, Prountzou A, Diem P, Mougiakakou SG (2012) Real-time adaptive models for the personalized prediction of glycemic profile in type 1 diabetes patients. Diabet Technol Therap 14(2):168–174. https://doi.org/10.1089/dia.2011.0093
43. Georga EI, Protopappas VC, Polyzos D, Fotiadis DI (2012) A predictive model of subcutaneous glucose concentration in type 1 diabetes based on random forests. In: Annual international conference of the IEEE engineering in medicine and biology society. IEEE, San Diego, CA, pp 2889–2892. https://doi.org/10.1109/EMBC.2012.6346567
44. Sparacino G, Zanderigo F, Maran A, Cobelli C (2006) Continuous glucose monitoring and hypo/hyperglycaemia prediction. Diabet Res Clin Pract 74:S160–S163. https://doi.org/10.1016/S0168-8227(06)70023-7
45. Shanthi S, Kumar D (2012) Prediction of blood glucose concentration ahead of time with feature based neural network. Malays J Comput Sci 25(3):136–148
46. Facchinetti A, Sparacino G, Trifoglio E, Cobelli C (2011) A new index to optimally design and compare continuous glucose monitoring glucose prediction algorithms. Diabet Technol Therap 13(2):111–119. https://doi.org/10.1089/dia.2010.0151
47. Prechelt L (1998) Early stopping—but when? In: Orr GB, Müller K-R (eds) Neural networks: tricks of the trade. Springer, Berlin, pp 55–69
48. Capel I, Rigla M, García-Sáez G, Rodríguez-Herrero A, Pons B, Subías D, García-García F, Gallach M, Aguilar M, Pérez-Gandía C, Gómez EJ, Caixàs A, Hernando ME (2014) Artificial pancreas using a personalized rule-based controller achieves overnight normoglycemia in patients with type 1 diabetes. Diabet Technol Therap 16(3):172–179. https://doi.org/10.1089/dia.2013.0229
49. Li Z, Zhou F, Chen F, Li H (2017) Meta-SGD: learning to learn quickly for few-shot learning. arXiv:1707.09835 [cs]
50. Zakharov E, Shysheya A, Burkov E, Lempitsky V (2019) Few-shot adversarial learning of realistic neural talking head models. arXiv:1905.08233 [cs]