

Convolutional Recurrent Neural Networks for Glucose Prediction

Kezhi Li, John Daniels, Pau Herrero-Viñas, Chengyuan Liu, Pantelis Georgiou

Department of Electronic and Electrical Engineering, Imperial College London, London SW7 2AZ, UK

Abstract—Glucose control serves as the primary method of diabetes management. Current digital therapeutic approaches for subjects with Type 1 diabetes mellitus (T1DM) such as the artificial pancreas and bolus calculators leverage machine learning techniques for predicting subcutaneous glucose for improved control. Deep learning has recently been applied in healthcare and medical research to achieve state-of-the-art results in a range of tasks including disease diagnosis, and patient state prediction among others. In this work, we present a deep learning model that is capable of predicting glucose levels over a 30-minute horizon with leading accuracy for simulated patient cases (RMSE = 10.02 ± 1.28 [mg/dl] and MARD = $5.95 \pm 0.64\%$) and real patient cases (RMSE = 21.23 ± 1.15 [mg/dl] and MARD = $10.53 \pm 1.28\%$). In addition, the model also provides competitive performance in forecasting adverse glycaemic events with minimal time lag both in a simulated patient dataset ($MCC_{hyperglycaemia} = 0.82 \pm 0.06$ and $MCC_{hypoglycaemia} = 0.76 \pm 0.13$) and in a real patient dataset ($MCC_{hyperglycaemia} = 0.79 \pm 0.04$ and $MCC_{hypoglycaemia} = 0.28 \pm 0.11$). This approach is evaluated on a dataset of 10 simulated cases generated from the UVa/Padova simulator and a clinical dataset of 5 real cases each containing glucose readings, insulin bolus, and meal (carbohydrate) data. Performance of the recurrent convolutional neural network is benchmarked against four state-of-the-art algorithms: support vector regression (SVR), latent variable (LVX) model, autoregressive model (ARX), and neural network for predicting glucose algorithm (NNPG).

Index Terms—Type 1 diabetes, continuous glucose monitor (CGM), deep learning, long short term memory (LSTM).

I. INTRODUCTION

Diabetes is a chronic illness characterised by a lack of glucose homeostasis. A healthy pancreas dynamically controls the release of insulin and glucagon hormones through the α -cells and β -cells respectively, in order to maintain euglycaemia [1]–[3]. In Type 1 diabetes, an autoimmune disease, the β -cells are compromised and therefore suffer from impaired production of insulin. This leads to increased periods of hyperglycaemia (blood glucose concentration $> 180\text{mg/dL}$) and hypoglycaemia (blood glucose concentration $< 70\text{mg/dL}$) [4], [5]. Insulin therapy is needed to manage blood glucose (BG) levels to maximise the time in (advised) target [2], [6].

The standard approach to diabetes management requires the subject to actively undertake blood glucose measurements a handful of times throughout the day with a finger prick test - self monitoring of blood glucose (SMBG). The recent development and uptake of continuous glucose monitoring (CGM) devices allow for improved sampling (5 minutes) of glucose measurements. This approach has proven to be effective in enabling glucose control and thus improving the

outcome of subjects in clinical trials relative to standard approaches [6]–[8]. Further improvement of glucose control can be realised through prediction, allowing users to take action ahead of time to minimise the occurrence of adverse glycaemic events. The challenge lies in the multiple factors such as insulin variability, ingested meals, stress and other physical activities that influence glucose variability [9]. In addition, the glycaemic responses of different people to the same conditions can be very different, which is conditioned on high inter- and intra- subject variability.

Machine learning covers a range of techniques that allow intelligent systems to tackle problems by learning and extracting patterns in data to build appropriate models. Consequently, the techniques involve discovering a suitable mapping from the representation of input data to the output. The performance of traditional machine learning algorithms such as logistic regression, k-nearest neighbors [10], or support vector regression [11] heavily depend on the representation of the data they are given. Typically, the features - information the representation comprises - are engineered with prior knowledge and statistical features (mean, variance) [12], principal component analysis (PCA) [13] or linear discriminant analysis [14]. Artificial neural networks are also investigated widely in diabetes management [15]–[18]. However, convolutional neural networks are mostly used with less than 3 layers, which limits the power of neural networks with latest training techniques.

Deep learning offers an approach to automatically learn both the features and consequent nonlinear mapping to the output. This approach, which incorporates multi-layer neural networks has been effective in solutions that have improved on previous upper bounds of performance in computer vision [19], board games [20], diseases diagnosis [21], [22], and healthcare [23]–[25]. Deep learning typically shows superior performance to traditional techniques due to this ability to automatically learn features with physical meanings [26], [27], hence potentially encoding features that may not be previously known to researchers.

In this paper, we propose a deep learning algorithm for glucose prediction using a multi-layer convolutional recurrent neural network (CRNN) architecture. The model is primarily trained on data comprising CGM, carbohydrate and insulin data. After preprocessing, the time-aligned multi-dimensional time series data of BG, carbohydrate and insulin (other factors are optional) are fed to the deep neural network (DNN) for training. The architecture of the DNN is composed of three parts: a multi-layer convolutional neural network that extracts the data features using convolution and pooling, followed

by a recurrent neural network (RNN) layer with long short term memory (LSTM) cells and fully-connected layers. In the LSTM cell, an input gate, an output gate and a forget gate are incorporated to remember the previous model state and update the model state simultaneously when current data are fed seamlessly. The convolutional layer comprises a 1D Gaussian kernel filter to perform the temporal convolution, and pooling layers are used for reducing the feature set. LSTM are the variant of gated RNN used since it shows good performance in predicting time series given long time dependencies [28]. In addition, it deals well with the vanishing gradient problem for training traditional RNNs. The final output is a regression output by the fully connected layer. The DNN is realized using the open-source software library Tensorflow [29], and it can be easily implemented to portable or wearable devices with its simplified version Tensorflow Lite. The performance of the proposed method is evaluated on datasets of simulated cases as well as clinical cases of T1DM subjects, and compared against the support vector regression (SVR) [11], the latent variable model (LVX) [30], the autoregressive model (ARX) [31], and neural network for predicting glucose algorithm (NNPG) [15], which show the competitive performance of the method.

The paper is organized as follows. Section II briefly discusses the setup and specification of the system, the pipeline for preprocessing the data. Section III details the architecture of the proposed multi-layer convolutional recurrent neural network. The performance of the proposed method is evaluated and discussed in Section IV. Finally, Section V concludes the paper with a discussion of challenges in this approach and avenues for future research.

II. DATASET ACQUISITION AND HARDWARE

A. Data Acquisition

The data used in this paper include two datasets, simulated data and clinical data. Simulated data of T1DM subjects was generated using the UVA/Padova T1D, which is the only simulator for glucose level simulation approved by the Food and Drug Administration (FDA). Clinical data was obtained from a 6-month clinical dataset corresponding to 10 adult subjects with T1D undergoing a clinical trial evaluating the benefits of an advanced insulin bolus calculator was employed [32].

B. Software and Hardware

After the model has successfully undergone training and validation, we also implement our algorithm on mobile phones through Tensorflow Lite due to its efficiency running on portable devices. The model is converted to a Lite model file and installed on an Android or iOS system. It needs the associated application programming interface (API) and interpreter to carry out the inference. Fig. 1 shows how the Tensorflow Lite model file [29] is wrapped and loaded in a mobile-friendly format.

Besides the CGM sensors used in data acquisition, we utilize computers equipped with graphics processing unit (GPU) to carry out the training and inference processes. In practice an Intel i7-7700K CPU with 4.20 GHz, 32.0 GB memory

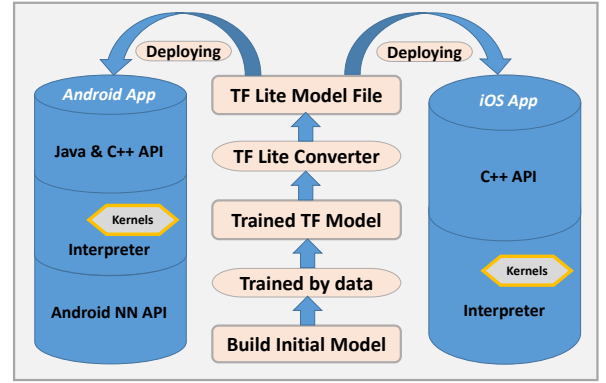


Fig. 1. A brief flow diagram explains the procedure to implement deep learning models to portable devices with the help of Tensorflow Lite. In the figure, the yellow square frames denote the models or files obtained after each operation, and the yellow round frames denote the associated operations applied. The file can be installed in Android or iOS app, with slightly different settings.

with GPU NVIDIA GeForce GTX 1080 Ti were used in the experiments. The program was written in Python 3.6, using CUDA 9.0 to perform the parallel computing. Tensorflow architecture was implemented, because its compatibility and merits for large-scale distributed training and inference. Keras was run on top of Tensorflow to build the CNN and LSTM, because of its simplicity of use.

III. METHODS

The approach consists of several components: preprocessing, feature extraction using CNN, time series prediction using LSTM and signal converter. The architecture of the proposed CRNN is shown in Fig. 2. In the diagram, the input of the algorithm is time series of glycemic data from CGM, carbohydrate and insulin information (time and amount); other related information are optional (exercises, alcohol, fat, stress, etc.). The output of preprocessing is cleaned as time-aligned glycemic, carbohydrate and insulin data, which are then fed to the CNN. The output of CNN is the input of RNN, which is a multi-dimensional time series data, representing the concatenation of features of the original signals. The output of the RNN is the predictive blood glucose level 30-min later, while hidden states are inherited and updated continuously internally inside of the RNN component. 30-min prediction horizon (PH) is applied because it is widely used in glucose prediction software, and it is easier to compare results with other works. Here each component will be explained in the following step by step.

A. Outlier Detection and Filtering

In the preprocessing component, the main purpose is to clean the data, filter the unusual points and make it suitable as the input to the CNN. Besides the normal steps including time stamp alignment, normalization and missing data imputation for time series data with trend, the most important operation to improve the data quality is the outlier detection, interpolation and filtering, in particular for clinic data. Because in the clinical data of glucose time series, there are many missing or

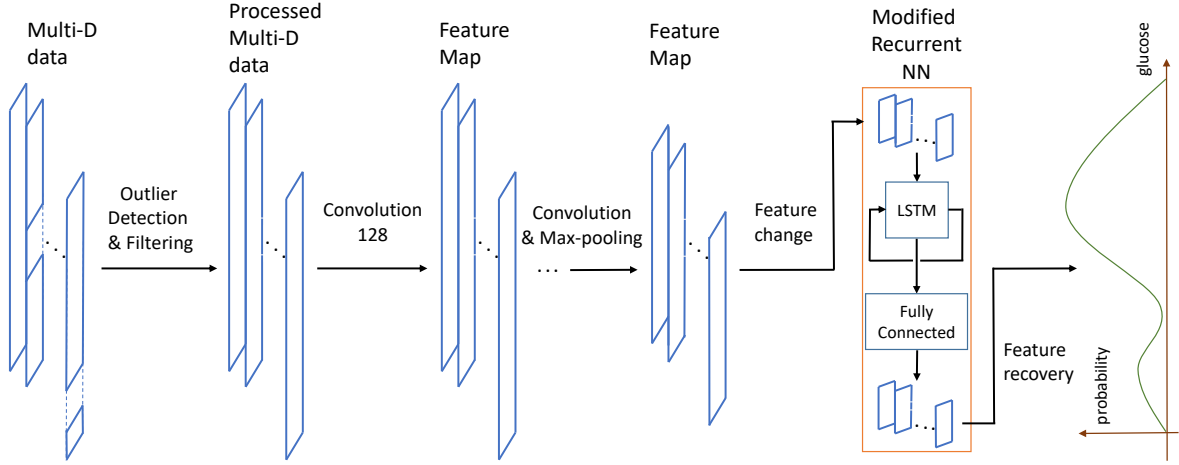


Fig. 2. The architecture of the proposed convolutional recurrent neural network for blood glucose prediction. The data at the left are the concatenated time series data including glucose level, carbohydrate, insulin and other factors. After outlier filtering step, the multi-dimensional data can be sent to the multi-layer convolutional component. Then the resultant is sent to the modified recurrent neural network component presented in a red frame, which includes LSTM cells and fully connected cells. Finally, the resultant is converted back from “change of the glucose value” to “absolute glucose value”. The output is the probability distribution of the future glucose values.

outlier data points due to errors in measurements, calibration or mistakes in the process of collecting or transmitting the data. Here, several methods are introduced to handle these scenarios [33].

- **Dimension Reduction Model:** the time series can be projected into lower dimensions using linear correlations such as principle component analysis (PCA) [34], and data with large residual errors can be considered as outliers.
- **Proximity-based Model:** the data are determined by nearest neighbour analysis [35], cluster or density [36]. Thus the data instances that are isolated from the majority are considered as outliers.
- **Probabilistic Stochastic Filters:** different filters for the 1D signals, such as gaussian mixture models optimized using expectation-maximization. In [37] a similar Gaussian Filter is implemented as a kernel to filter the 1D signal in the CNN. While in our case the filter can be implemented before the CNN, due to the continuous characteristic of the input glycemic time series data.

For some cases when the data fluctuates with high frequency, 1D Gaussian kernel filter is implemented on the glucose time series to get rid of the noise. The result of this process are cleaned 3 time series of glycemic data from CGM, and the time-aligned carbohydrate and the insulin information.

B. A Multi-layer Convolutional Neural Network

The filtered time series signal goes through the multi-layer convolutions, which transforms the input data into a set of feature vectors. The convolution operation follows the 2D convolution definition in which

$$z[m, n] = \sum_{j=-k}^k \sum_{i=-l}^l x[i, j] \cdot \delta[m-i, n-j], \quad (1)$$

where x represents the input signal, δ denotes the kernel, and z is the result of the convolution. The input signal can

be fed block by block, using a sliding window setting. The windows can be overlapped or non-overlapped, determined by the allowed CNN size and computations. The CNN will automatically learn the associated weights and recognize particular patterns and features in the input signal that can best represent or classify the data for the following steps, using the back-propagation technique. It avoids the complicated manual feature selections and extracts the features from raw data in the training process. This technique in DL has been verified useful and efficient in many applications [19], [21]–[23], [38], [39].

Specifically, in our model the proposed CNN has 4 hidden layers, consisting alternatively 2 convolutional layers and 2 pooling layers to down-sample the feature map obtained from the previous convolutional layer. It is common to periodically insert a pooling layer in-between successive convolutional layers. Because this structure can progressively reduce the spatial size of the representation, and reduce the amount of parameters as well as the computation in the CNN. It also can control the overfitting problem. For instance, if the accepted size is $L1 \times D1$, and the down-sampled parameters are spatial extent F and stride S , and it results in a max-pooling vector Y of size $L2 \times D2$ where

$$\begin{aligned} L2 &= (L1 - F)/S + 1 \\ D2 &= D1 \\ Y_i &= \max(y_i^*) \end{aligned} \quad (2)$$

where y_i^* is the vectors after down-sampled, Y_i is the feature map and $\max()$ is the operator that computes the maximum value. Please note that this operation let the depth dimension remain unchanged, and zero-padding is not common in the pooling layer. Finally, the last layer of CNN is a fully connected layer, which is followed by RNNs in the next component. The fully connected layer produces the output with

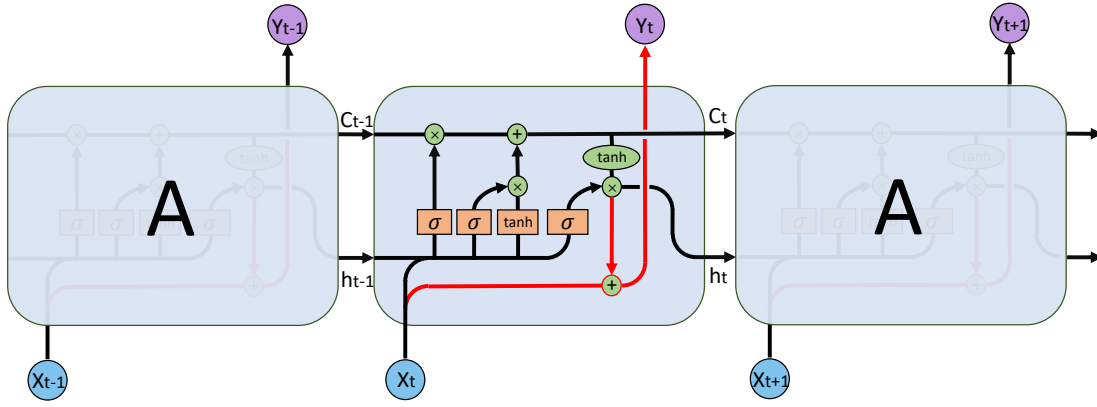


Fig. 3. An illustration of the proposed modified recurrent neural network in the inference. It originates from the conventional LSTM cell, including an input gate, an output gate and a forget gate. The difference lies in the output part, which indicates in red colour in the figure. The output of h_t is still the internal parameter which is transmitted to the next cell. However, the output value signal is y_t instead of h_t . It is because in the training process, the target of RNN is the change of the glucose value between current time and 30 mins later. Thus the predictive glucose level after 30 mins should be the inference value plus the current glucose value.

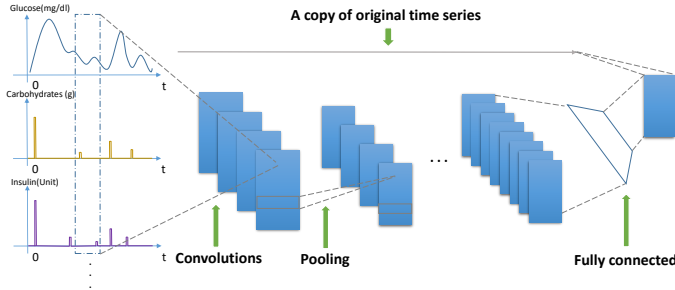


Fig. 4. An illustration of the proposed CNN architecture. The multi-dimensional aligned time series data are concatenated, and then sent to a multi-layer CNN composed of convolutional layers and pooling layers. Finally, after going through a fully connected layer, the resultant is the concatenation of the CNN output and a copy of the original input time series.

an activation function

$$Z_i = \text{act}\left(\sum_{i=1}^N Y_i w_i + b_i\right), \quad (3)$$

where Z_i is the multi-dimensional output, $\text{act}()$ is an activation function, w_i and b_i are weights of the CNN that can be trained. Particularly, $\text{act}()$ can be chosen from a set of activation functions such as sigmoid function $\text{act}(a) = 1/(1 + e^{-a})$, rectifier $\text{act}(a) = \log(1 + \exp(a))$ or simple linearly $\text{act}(a) = a$. In this paper we choose the simple linear function $\text{act}(a) = a$ as the activation function for its simplicity. During the training process, the CNN is trained by back-propagation and the stochastic gradient descent method. The initial weights of the network are set randomly, and the mean-square-error is set as the cost function to be minimized in the training. Partial derivatives of the error in terms of the weights w_i and bias b_i are computed, and the associated w_i and b_i are updated accordingly.

C. An Modified Recurrent Neural Network

The RNN implemented in our model is a multi-layer LSTM network, which is composed of many LSTM cells and

dropouts [40]. Each LSTM cell consists of an input gate, an output gate and a forget gate. Each of the three gates can be thought of as a neuron, and each gate achieves a particular function in the cell. The LSTM network is good at building predictive models for time series and sequential data [41]. It is because the cell remembers the previous data patterns over arbitrary time intervals, thus the internal “memory” can predict the future output according to the previous experience. Its memory is updated sedtiously when new data are fed to the model seamlessly.

The output of the CNN, a 11-dimensional time series, is used to train the LSTM network. We generated an RNN with 6 hidden layers, consisting of 3 LSTM layers and 3 dropout layers alternately. Each LSTM layer has 32 cells, followed by a dropout layer. The dropout layer refers to ignoring neurons randomly during the training phase of certain set of neurons. In many cases it has been verified that dropout layers can effectively avoid the overfitting problem and increase the accuracy [42], [43].

The last layer is a fully connected layer with the glucose change as output. We feed the model last 24 points before the current time in the time series, which implies that last $24 \times 5 = 120\text{mins} = 2\text{ hours}$ glucose/carbohydrate/insulin are considered. A sliding window of size 24 is determined to train the model, because we found it is an appropriate setting to balance the tradeoff between the prediction accuracy and the computation complexity.

The main difference between a normal LSTM and the proposed LSTM is that the proposed model has the transform and recover steps to modify the BG values before going through and after coming out the conventional LSTM model. Because in the training, instead of using the BG values directly, we use the change of BG between the current BG $x(t)$ and the future BG $x(t+6)$ as the target label. It is called the transform step. In the training, the input sliding window matrices are multi-dimensional time series (including BG values, meal, insulin). After the model has been trained, the inference output is the change of BG $\Delta x(t)$ between $x(t)$ and $x(t+6)$. So we need

the BG value $x(t)$ to obtain the prediction of BG at time $t+6$ with $x(t+6) = x(t) + \Delta x(t)$. It is called the recover step.

Specifically, a modified LSTM RNN estimates the conditional probability $p(y_1, \dots, y_{T'} | x_1, \dots, x_T)$ given a sequence of data, where x_1, \dots, x_T denotes the input sequence and $y_1, \dots, y_{T'}$ is the corresponding output sequence, with size T and T' respectively. If x_t, h_t, c_t are used to denote the input vector, output vector and memory cell vector respectively; W, U , and b are the parameter matrices/vectors that can be learned in the network. f_t, i_t , and o_t denote the forget gate, input gate, and output gate vectors. Then mathematical form of the update process can be explicitly written as

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) & i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) & g_t &= \sigma_t(W_g x_t + U_g h_{t-1} + b_g) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_t(g_t) & h_t &= o_t \odot \sigma_t(c_t), \\ y_t &= o_t \odot \sigma_t(c_t) + x_t, \end{aligned} \quad (4)$$

where $\sigma_g, \sigma_t, \odot$ is the sigmoid function, hyperbolic tangent and entrywise product, respectively. In 4, the 1st to 5th equations are the same to the equations of LSTM. However the last 2 equations of h_t and y_t are modified according to the proposed glucose model. It has been shown in Figure 3. In the figure, the difference between the proposed modified LSTM and the normal LSTM is indicated using the red colour. The output of the LSTM cell is not h_t , but y_t . The h_t is used as a remembered state that goes to the next time stage. However the output y_t is calculated from h_t plus the original input x_t . That is because the output (and targets) of the neural network is the change of the blood glucose level. The real value of the predictive glucose level needs to be recovered from the glucose change by adding the baseline glucose value.

In the training, mean-square-error between the target and the predictive value are being minimized. The optimizer we use is *RMSprop*, because it is usually a good choice for recurrent neural networks. It maintains a moving (discounted) average of the square of gradients, and divides gradient by the root of this average.

IV. RESULTS

In this section we test the proposed CRNN algorithm for glucose level prediction using both *in silico* dataset and a clinical dataset, and compare with state-of-the-art algorithms developed in recent years.

A. Verification Set-up

For all methods, the datasets include CGM data recorded every 5 minutes, meal data indicating meal time and amount of carbohydrates, as well as insulin data with each bolus quantity and the associated time. Exercise, stress and alcohol consumption were included in the datasets but as optional variables in the model. In the data preprocessing step, incidences of meal and insulin intake are aligned to CGM data with the same timestamps.

The *in silico* data considered in this section is generated via UVA/Padova T1D [44]. This simulator is approved by the FDA

and therefore, serves as robust and validated framework for generating simulated cases. The cohort of T1D cases generated can be configured with varying meal and insulin information such that each case sufficiently differs. We generate a dataset of 10 unique adult cases and each has 360 days of data for each case. There are 3 meals per day. Insulin entries vary in each day, from 1 to 5. The insulin entry can be with a meal (meal and insulin at almost the same time), or without a meal (correction bolus). Exercise is considered at certain points in the data as well, which occur at any time except for nights. The training data accounts for 50% of the dataset, and the testing set is the rest of the data.

The clinical data was collected from T1DM subjects in a 5 month clinical trial. The CGM data was measured using Dexcom G4 Platinum CGM sensors, with measurements received every 5 minutes. The CGM sensors were inserted from the first day of the study, and calibrated according to the manufacturer instructions. Other information available in the dataset such as meal, insulin, exercises was logged by the diabetic subjects themselves. Though the data has good quality, there still exist many periods of missing data, bad points or unexpected fluctuations. The diabetic subjects behaved normally during the study as they would in their daily lives. Similar to the *in silico* experiment, each subject's clinical data is halved for training and testing data.

B. Criteria for Assessment

Several criteria are used to quantitatively assess the performance of the proposed prediction algorithm. The root-mean-square error (RMSE) and mean absolute relative difference (MARD) between the predicted and reference glucose readings - for simulated or clinical CGM data - serve as the primary indicator to evaluate the general predictive performance of the algorithm.

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k|k - PH))^2}, \quad (5)$$

where $\hat{y}(k|k - PH)$ denotes the prediction results provided the historical data and y denotes the reference glucose measurement, and N refers to the data size.

$$MARD = \frac{1}{N} \sum_{k=1}^N \frac{|\hat{y}_k(k|k - PH) - y(k)|}{y(k)}, \quad (6)$$

The RMSE and MARD provide an overall indication of the predictive performance of the proposed algorithm. As mentioned earlier, the benefit of glucose prediction is in avoiding adverse glycaemic events. In this clinical context, these metrics are limited in the insight they provide. Additional metrics are needed to assess the proposed algorithm in the following perspective:

- Capability of the forecasting algorithm in differentiating between adverse glycaemic events and non-adverse glycaemic events.
- Time delay in the predicted glucose readings and the reference values to evaluate the response time

provided to deal with the potential adverse glycaemic event.

The Matthews Correlation Coefficient (MCC), Sensitivity and Specificity are used to evaluate the performance of the algorithms for detecting either adverse glycaemic event (hypoglycaemia or hyperglycaemia).

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},$$

$$Sensitivity = \frac{TP}{(TP + FN)},$$

$$Specificity = \frac{TN}{(FP + TN)},$$
(7)

where TP, FP, FN, TN stand for true positive, false positive, false negative, and true negative respectively. Here, positive indicates a hypoglycaemia (< 70 mg/dl)/hyperglycaemia (> 180 mg/dl) event in the next or previous 30 minutes, and true means that the classification is correct. A standard confusion matrix typically includes the Accuracy as opposed to Matthews Correlation Coefficient (MCC). This modification addresses the imbalance in classes inherent in this situation - non-adverse events far outweigh adverse events. Consequently, the result is not heavily skewed towards the specificity. Analysis discussed further showed that True Negative occurrences outweighed the occurrences of other classes and thus heavily skewing the accuracy towards the specificity.

The effective prediction horizon is defined as the prediction horizon, taking into account delays due to the responsiveness of the algorithm for a predicted value relative to its reference value. Cross correlation of the predicted and actual readings is employed in performing a time delay analysis of the proposed algorithm to determine the effective prediction horizon.

$$PH_{eff} = PH - \tau_{delay}$$

$$= PH - \arg \max_{\tau} (\hat{y}_k(k|k - PH) \star y(k)) \quad (8)$$

A singular quantitative metric is not sufficient in evaluating performance of the proposed algorithm. Consequently, the set of metrics collectively give a comprehensive description of the quality of the prediction algorithm performance.

C. In Silico Data

The performance of the proposed algorithm is contrasted with that of three baseline methods: SVR, LVX and 3rd order ARX. The performance of the algorithms are compared based on the accuracy over a 30-minute prediction horizon. Different algorithms were tested on the *in silico* data generated in a way described previously. The parameters involved in these algorithms are tuned carefully for optimal result. In SVR, the SVR function in Python is applied with the optimal parameters ($C = 1e2, \gamma = 0.01, cache_size = 1000$). The LVX method are applied based on the MATLAB code provided in [30], the optimal predictor length and the number of LVs are $J_x =$

4 and $N_{LV} = 4$ respectively, that is a 20-minute historical data of glucose measurement, insulin and meal information are used for prediction. The three order AXR model are optimized by MATLAB function *arx()* for every specific subject. The results of RMSE and MARD are summarized in the Table I and the results of confusion matrix are summarized in the Table II.

In Table I, we compare the predictive error of the algorithms to measure the accuracy of the algorithms. The CRNN algorithm exhibits the best overall RMSE and MARD for the 10 simulated cases. This is also evident in the hyperglycaemia region. In the hyperglycaemia region the CRNN shows the best result which is much better than other algorithms, although in the hypoglycaemia region CRNN achieves the second best from RMSE and MARD, slightly worse than LVX (+0.77mg/dl). This could be attributed to the dataset containing relatively fewer hypoglycaemia events for training. However, Table II shows the CRNN report a marginally better MCC than LVX for hypoglycaemia(+0.02).

The CRNN model also reports the lowest standard deviations from which we infer a benefit in building individualized models.

An illustration of a comparison of various algorithms are shown in Fig. 5 for a virtual adult 4. As seen in the Figure 5, the CRNN exhibits the best responsiveness as the predictive curve responds rapidly towards the sharp glycemic uptrend. The algorithm learns representations that appropriately account for both sharp slopes and gradual increments in the glycemic curve. Consequently, at a glycemic peak, CRNN yields a predictive curve with even higher slope to compensate the time lag aiming at reducing the gap between the prediction and real measurements. This feature helps CRNN to decrease the RMSE and MARD as well as maximising the effective prediction horizon, as shown in Figure 5.

D. Clinical Data

As mentioned in the previous section, the data obtained in the clinical trial exhibits missing data, and erroneous data. This results in non-physiological discontinuities that would affect the training process. To mitigate these occurrences, the data is processed with interpolations for gaps in data, yet the interpolation points are not included in all calculations.

Table III shows the RMSE and MARD of the performance of the algorithms for the 5 cases of real data. The CRNN still maintains the best results for both the RMSE and MARD relative to the baseline methods. Relative to the performance achieved in the simulated datasets, the performance with real clinical shows worse average RMSE and MARD. This can be attributed to added external factors leading to discontinuity in data - missing data and sensor errors.

Table IV shows that the CRNN maintains the best performance in regard to detection of hyperglycaemic events. Given results in the previous section, it was observed that the RMSE and MARD of the various algorithms pointed to reduced performance between simulated cases and real cases. This observation is also made in regard to the performance in accuracy in detecting adverse glycemic events. The reduction in performance in detecting hyperglycaemia is not

TABLE I
RMSE AND MARD COMPARISON OF DIFFERENT PREDICTION METHODS FOR 10 VIRTUAL ADULT DIABETIC SUBJECTS

Metric	CRNN	NNPG	SVR	LVX	ARX
Overall					
RMSE (mg/dL)	10.02 \pm 1.28	12.91 \pm 1.19 [‡]	12.48 \pm 1.94 [†]	11.32 \pm 1.34 [*]	13.27 \pm 1.19 [†]
MARD (%)	5.95 \pm 0.64	7.05 \pm 0.94 [*]	6.40 \pm 1.36	6.59 \pm 0.80	7.46 \pm 1.02 [†]
Hyperglycaemia					
RMSE (mg/dL)	15.11 \pm 2.43	21.68 \pm 2.36 [‡]	21.74 \pm 4.41 [‡]	17.95 \pm 4.69	19.91 \pm 1.80 [‡]
MARD (%)	5.46 \pm 0.74	6.69 \pm 0.73 [†]	6.67 \pm 1.49 [†]	6.03 \pm 1.77	6.80 \pm 0.63 [‡]
Hypoglycaemia					
RMSE (mg/dL)	9.85 \pm 1.36	12.72 \pm 2.65 [†]	13.13 \pm 8.39	8.72 \pm 0.55 [*]	9.93 \pm 0.74
MARD (%)	14.55 \pm 2.28	20.27 \pm 4.01 [‡]	18.81 \pm 13.75	12.55 \pm 1.70 [*]	14.13 \pm 1.94

^{*} p -value \leq 0.05

[†] p -value \leq 0.01

[‡] p -value \leq 0.005

TABLE II
COMPARISON OF ACCURACY MEASURES IN DIFFERENT PREDICTION METHODS FOR 10 VIRTUAL ADULT DIABETIC SUBJECTS

Metric	CRNN	NNPG	SVR	LVX	ARX
Hypoglycaemia					
Sensitivity (%)	68.62 \pm 16.66	54.93 \pm 23.30	75.23 \pm 8.80	84.63 \pm 4.41 [†]	78.73 \pm 8.22
Specificity (%)	99.53 \pm 0.66	99.57 \pm 0.61	99.25 \pm 0.86	98.82 \pm 0.97	98.72 \pm 1.06
MCC	0.76 \pm 0.13	0.67 \pm 0.20	0.77 \pm 0.07	0.77 \pm 0.08	0.74 \pm 0.10
Hyperglycaemia					
Sensitivity (%)	85.08 \pm 4.70	78.97 \pm 6.06 [*]	79.81 \pm 6.34	84.56 \pm 5.20	80.94 \pm 5.66
Specificity (%)	97.87 \pm 0.74	98.54 \pm 0.59 [*]	98.65 \pm 0.52 [*]	97.81 \pm 0.58	97.74 \pm 0.61
MCC	0.82 \pm 0.05	0.81 \pm 0.04	0.82 \pm 0.05	0.82 \pm 0.05	0.79 \pm 0.05
Effective Prediction Horizon					
Time (min)	29.1 \pm 1.6	20.8 \pm 1.8 [‡]	23.3 \pm 1.6 [‡]	27.5 \pm 1.3 [‡]	20.5 \pm 1.7 [‡]

^{*} p -value \leq 0.05

[†] p -value \leq 0.01

[‡] p -value \leq 0.005

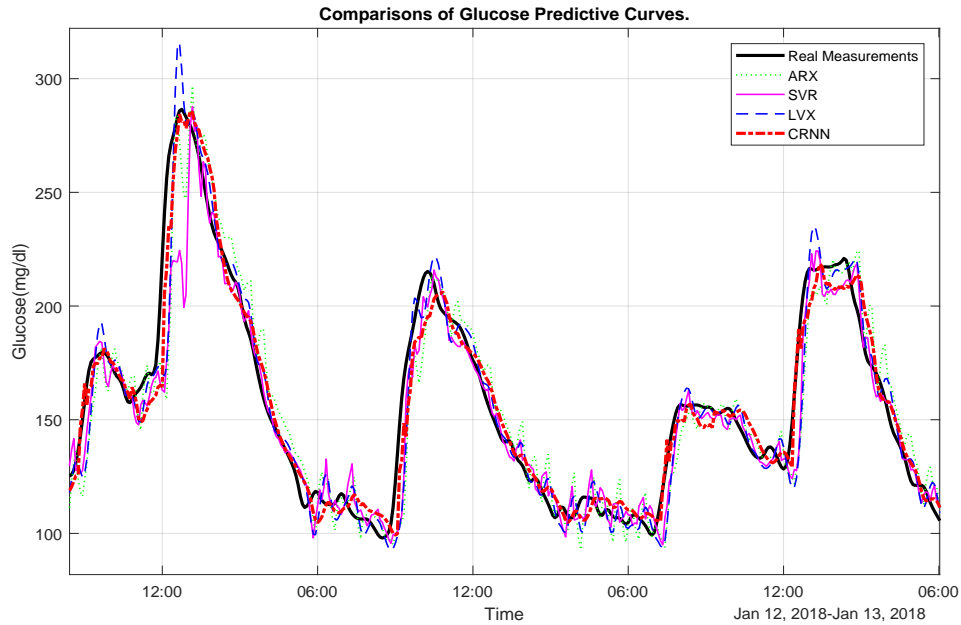


Fig. 5. One-day period prediction results for virtual adult 4. The solid black line, dotted green line, solid magenta line, dashed blue line, dash-dotted red line indicate the simulated glucose measurements, the prediction results of the 3rd order ARX method, the prediction results of the SVR method, the prediction results of the LVX algorithm, the prediction results of the CRNN method, respectively.

significant statistically. CRNN still reports the best result for hyperglycaemia accuracy. However, the performance of LVX

in hypoglycaemia detection is slightly better.

Regarding the effective prediction horizon, CRNN main-

TABLE III

DIFFERENT PREDICTION METHODS COMPARISON FOR THE 5 REAL ADULT DIABETIC SUBJECTS CONSIDERED IN TERMS OF RMSE AND MARD

Metric	CRNN	NNPG	SVR	LVX	ARX
Overall					
RMSE (mg/dL)	21.24 \pm 1.01	23.05 \pm 1.24*	27.23 \pm 1.93 [‡]	25.21 \pm 0.83 [‡]	25.36 \pm 0.96 [‡]
MARD (%)	10.58 \pm 0.81	11.79 \pm 0.86	12.44 \pm 1.02 [†]	11.46 \pm 1.24	11.75 \pm 1.19
Hyperglycaemia					
RMSE (mg/dL)	24.81 \pm 5.16	28.39 \pm 1.32 [‡]	35.21 \pm 2.24 [‡]	29.07 \pm 0.70 [‡]	29.62 \pm 0.80 [‡]
MARD (%)	8.12 \pm 0.82	9.35 \pm 0.81	10.52 \pm 1.05 [†]	8.73 \pm 0.45	9.07 \pm 0.44
Hypoglycaemia					
RMSE (mg/dL)	27.81 \pm 5.16	31.10 \pm 6.01	34.51 \pm 8.56	24.73 \pm 3.14	25.55 \pm 2.27
MARD (%)	36.55 \pm 8.40	46.81 \pm 10.18	43.29 \pm 13.31	31.03 \pm 9.48	29.20 \pm 3.95

* p -value \leq 0.05† p -value \leq 0.01‡ p -value \leq 0.005

TABLE IV

COMPARISON OF DIFFERENT METHODS FOR THE ACCURACY MEASURES OF 5 DIABETIC SUBJECTS IN CLINICAL TRIALS CONSIDERED IN TERMS OF HYPOGLYCAEMIA/HYPERGLYCAEMIA PREDICTION.

Metric	CRNN	NNPG	SVR	LVX	ARX
Hypoglycaemia					
Sensitivity(%)	19.28 \pm 14.97	4.81 \pm 7.13	15.90 \pm 16.38	58.28 \pm 20.65*	41.25 \pm 7.97*
Specificity(%)	99.08 \pm 1.00	98.64 \pm 0.52	99.02 \pm 1.22	96.93 \pm 1.63*	97.74 \pm 1.45
MCC	0.26 \pm 0.13	0.09 \pm 0.36	0.20 \pm 0.16	0.39 \pm 0.07	0.36 \pm 0.07
Hyperglycaemia					
Sensitivity(%)	82.51 \pm 7.69	78.65 \pm 7.98	79.34 \pm 8.79	83.41 \pm 4.84	82.33 \pm 8.30
Specificity(%)	94.19 \pm 2.46	94.04 \pm 2.89	92.75 \pm 3.64	93.60 \pm 1.57	93.38 \pm 1.76
MCC	0.78 \pm 0.04	0.78 \pm 0.04	0.74 \pm 0.04	0.78 \pm 0.04	0.76 \pm 0.03
Effective Prediction Horizon					
Time (min)	19.8 \pm 2.0	13.9 \pm 1.6 [†]	18.0 \pm 1.7	12.4 \pm 1.8 [‡]	10.4 \pm 1.6 [‡]

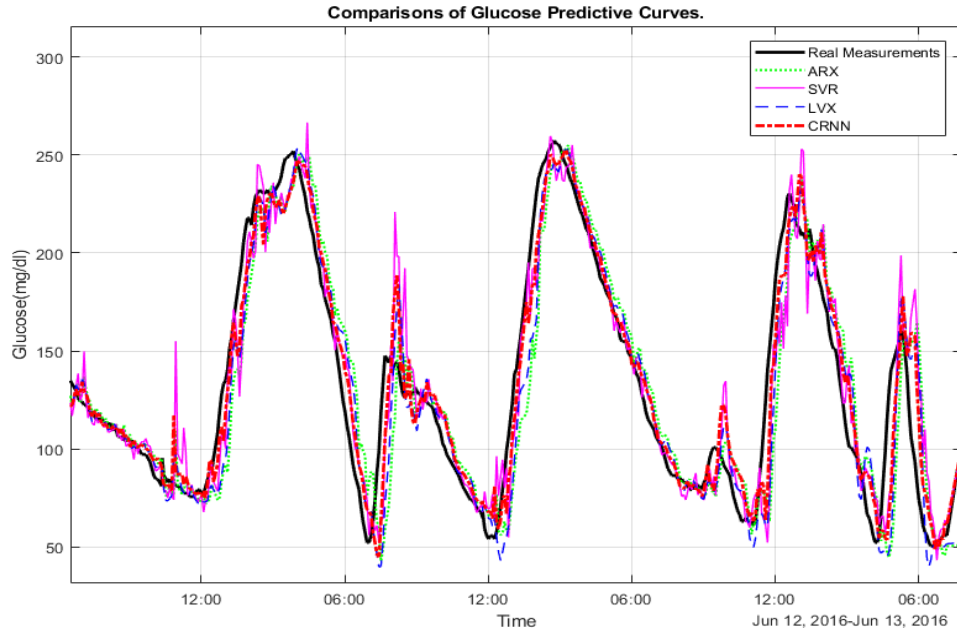
* p -value \leq 0.05† p -value \leq 0.01‡ p -value \leq 0.005

Fig. 6. One-day period prediction results for clinical adult 17. The solid black line, dotted green line, solid magenta line, dashed blue line, dash-dotted red line indicate the simulated glucose measurements, the prediction results of the ARX method, the prediction results of the SVR method, the prediction results of the LVX algorithm, the prediction results of the CRNN method, respectively.

tains a better effective prediction horizon (+7.26 min) than LVX which would give a potential subject more time to avert

an adverse glycemic event. Although the LVX has better hypoglycaemic event prediction, CRNN would be more useful due to the effective prediction horizon. This is a significant improvement over the other baseline methods.

As seen in Fig. 6, the CRNN and LVX both achieve good predictive curves compared to the ground truth measurements. Specifically, at the inflection periods during peaks and troughs, the LVX tends to have a higher and lower predictions, respectively. The CRNN follows the trend at both local and global peak points closely, which increases its overall accuracy.

V. DISCUSSION

A. Performance in Simulated and Clinical Data

In the previous section we noted a discrepancy in the performance of the proposed algorithm and baseline algorithms in simulated cases and the real patient cases. In our opinion, the drop in performance can be primarily attributed to the increased complexity of real data generated from a patient relative to the simulated data generated from a physiological model. In addition the gaps in data and method of interpolation may contribute to the further reduction in performance of the model. The evidence however points that relative to the baseline algorithms, the CRNN is better at capturing the features, because the model of each subject is trained individually based on the subject's own data. This would explain the relatively lower variance in metrics for the performance of the CRNN in different cases relative to baseline models.

B. Results Comparisons

We achieved a mean $RMSE = 10.02$ using the proposed method, and it is the best amongst other advanced algorithms, including SVR, LVX and 3rd order ARX. In addition, we want to compare our algorithm with other approaches in the literature. Also working on glucose dataset generated from simulators, our algorithm is also better than $RMSE = 18.78$ [45] and $RMSE = 13.65$ in [46].

For several other works, it is difficult to compare the RMSE directly. Because the papers' codes are not available publicly, and the performances were computed on various dataset based on different settings. However, we may compare the results using widely used methods as benchmarks, such as SVR or NNPG. For instance, for $PH = 30$ mins as in [17] Table 3 the algorithm is 0.1 better than the result of SVR in terms of RMSE on the real dataset, and our algorithm is 6.0 better than the SVR in terms of RMSE on the real dataset. In [16], for $PH = 30$ mins their RMSEs are 1.3 better than NNPG for the simulated data and 0.2 better than NNPG for the real datasets. Our RMSEs are xxx better than NNPG for the simulated data and xxx better for the real datasets. As far as we are aware, the proposed algorithm achieves a performance state-of-the-art.

C. Parameters in ML and Hyperparameters in DL

For conventional machine learning tasks, feature selection is crucial to achieving a good performance. The quality of the engineered features or patterns the system learns from the data determines the accuracy of prediction or classification.

Moreover, it involves tedious parameter tunings, which tends to an entirely heuristic approach. In deep learning, the features can be learnt optimally from the data automatically, and these features can be interpretable by visualization [47]. Although, several hyperparameters need to be set, including the number of layers, number neurons, activation functions, etc. A grid search approach makes this much easier to implement and tune in programming, which saves time and resources in large dataset problems.

D. Applied to App on Mobiles

The proposed CRNN is an adaptive algorithm for different diabetic subjects. Because it is data driven, different subjects will have their own model. The model is saved as a trained neural network. We use keras sequential model with Tensorflow backend to train the neural network, and the result model can be easily saved as a single file. This file can be compiled as a ".tflite" file to the app on mobiles, by using a Tensorflow Lite converter. Then the app may use it to provide the predictive glycemic curve and show it the screen. A demonstration on the Android system is shown in Figure 7.

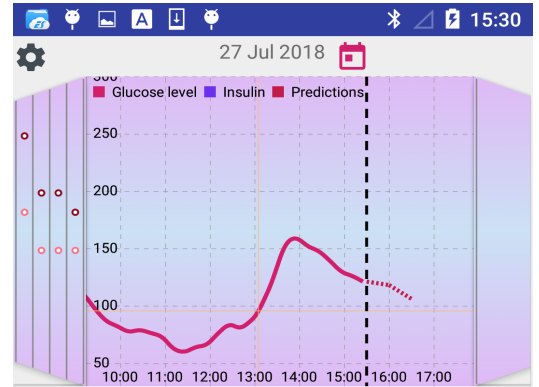


Fig. 7. An illustration of the glucose level shown in an app interface on an Android system, where the red curve is the historic blood glucose, black dash line is the current time, and the red dot curve is the prediction provided by the model.

E. Limitations

Though the CRNN has achieved a state-of-the-art accuracy in prediction, some challenges exist that can form the basis of further research. The underlying assumption of the deep learning approach, and indeed in the supervised learning paradigm, is that data is independent and identically distributed. Given that learning is solely based on historical data, unexpected predictions may occur given that correlations learned in the data may not imply causation. Thus hybrid approaches whereby the deep learning model is used to make an accurate prediction, and rules of meal/bolus supported by physiological model avoid apparent errors that might result. Based on the CRNN approach proposed in this paper, it is possible to develop the hybrid method, which may have the advantages of both conventional and DL algorithms.

VI. CONCLUSION

In this paper a convolutional recurrent neural network was proposed as an effective method for plasma glucose prediction. The architecture of the neural network includes a multi-layer CNN followed by a modified RNN, where the CNN can capture the features or patterns of the multi-dimensional time series data properly, and the modified RNN is capable of analyzing the previous sequential data and provide the predictive glucose level further. The method can train models for each diabetic subject using their own data, thus the model for the prediction is adaptive and specialized for each person. After obtaining the train neural network, it can easily be applied locally or on portable devices, which can potentially be implemented widely in the daily life. It has been found that the proposed method reduces the RMSE, MARD in the *in silico* experiments. While in the test using the clinical data, the proposed method performs quite well in most cases comparing to other latest approaches, and shows competitive performance in forecasting hypoglycaemia and hyperglycaemia.

VII. APPENDIX

The authors thank EPSRC, ARISES project for support and providing the clinical data. We would also like to thank Prof. R. Spence, T. Zhu and C. Demasson's for the contribution to the mobile app interface design.

TABLE V
COMPARISON OF DIFFERENT PREDICTION METHODS FOR THE 10-ADULT VIRTUAL DIABETIC SUBJECTS CONSIDERED IN TERMS OF RMSE(mg/dl)

Case	CRNN	NNPG	SVR	LVX	ARX
1	9.46	12.93	11.80	11.15	13.33
2	8.90	12.12	10.90	14.61	12.40
3	12.70	14.75	13.49	12.07	15.48
4	9.64	12.45	11.12	10.21	12.71
5	8.90	11.60	10.84	10.03	11.80
6	10.13	13.72	12.20	11.95	14.29
7	11.91	15.02	14.86	12.00	14.79
8	10.72	11.54	12.50	10.25	12.00
9	8.90	11.78	10.92	10.25	12.28
10	8.97	13.16	12.46	10.08	13.57
Mean	10.02	12.91	12.48	11.32	13.27
SD	1.28	1.19	1.94	1.34	1.19

REFERENCES

- [1] S. Oviedo, J. Veh, R. Calm, and J. Armengol, "A review of personalized blood glucose prediction strategies for t1dm patients," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 33, no. 6, p. 2833, 2017. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.2833>
- [2] M. Vettoretti, A. Facchinetti, G. Sparacino, and C. Cobelli, "Type 1 diabetes patient decision simulator for in silico testing safety and effectiveness of insulin treatments," *IEEE Transactions on Biomedical Engineering*, pp. 1–1, 2018.
- [3] P. Pesl, P. Herrero, M. Reddy, M. Xenou, N. Oliver, D. Johnston, C. Toumazou, and P. Georgiou, "An advanced bolus calculator for type 1 diabetes: System architecture and usability results," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 1, pp. 11–17, Jan 2016.
- [4] A. Facchinetti, S. Favero, G. Sparacino, and C. Cobelli, "An online failure detection method of the glucose sensor-insulin pump system: Improved overnight safety of type-1 diabetic subjects," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 406–416, Feb. 2013.

TABLE VI
COMPARISON OF DIFFERENT PREDICTION METHODS FOR THE 10-ADULT VIRTUAL DIABETIC SUBJECTS CONSIDERED IN TERMS OF MARD(%)

Case	CRNN	NNPG	SVR	LVX	ARX
1	5.50	6.77	6.84	6.45	8.45
2	4.94	6.22	6.06	7.09	7.30
3	6.34	7.47	6.52	6.46	8.92
4	5.32	7.66	6.43	6.78	9.09
5	5.63	6.64	5.90	6.28	7.95
6	6.29	7.44	6.76	8.02	9.18
7	5.50	6.24	6.23	6.65	7.60
8	6.36	6.23	7.44	6.57	7.51
9	5.43	6.42	6.50	6.83	7.73
10	7.27	9.38	7.98	10.04	11.56
Mean	5.95	7.05	6.40	6.59	7.46
SD	0.64	0.94	1.36	0.80	1.02

TABLE VII
DIFFERENT PREDICTION METHODS COMPARISON FOR THE CLINICAL 5 DIABETIC SUBJECTS CONSIDERED IN TERMS OF RMSE (mg/dL)

Case	CRNN	NNPG	SVR	LVX	ARX
1	20.76	22.15	27.35	25.71	25.82
2	19.49	21.27	23.68	23.77	23.79
3	21.89	23.98	27.56	25.80	26.12
4	21.79	23.15	27.96	24.77	24.73
5	22.26	24.72	29.57	25.99	25.99
Mean	21.24	23.05	27.23	25.21	25.36
SD	1.01	1.24	1.93	0.83	0.96

TABLE VIII
DIFFERENT PREDICTION METHODS COMPARISON FOR THE CLINICAL 5 DIABETIC SUBJECTS CONSIDERED IN TERMS OF MARD (%)

Case	CRNN	NNPG	SVR	LVX	ARX
1	12.82	13.42	14.02	13.58	13.88
2	10.41	11.49	11.85	11.67	11.86
3	9.78	10.92	11.01	10.12	10.56
4	10.82	11.81	13.00	11.66	11.76
5	9.87	11.33	12.29	10.29	10.69
Mean	10.58	11.79	12.44	11.46	11.75
SD	0.81	0.86	1.02	1.24	1.19

- [5] S. Zavitsanou, A. Mantalaris, M. C. Georgiadis, and E. N. Pistikopoulos, "In silico closed-loop control validation studies for optimal insulin delivery in type 1 diabetes," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 10, pp. 2369–2378, Oct. 2015.
- [6] K. van Heusden, E. Dassau, H. C. Zisser, D. E. Seborg, and F. J. D. III, "Control-relevant models for glucose control using a priori patient characteristics," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 7, pp. 1839–1849, Jul. 2012.
- [7] M. M. Ahmadi and G. A. Jullien, "A wireless-implantable microsystem for continuous blood glucose monitoring," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 3, pp. 169–180, Jun. 2009.
- [8] A. Facchinetti, "Continuous glucose monitoring sensors: Past, present and future algorithmic challenges," *Sensors*, vol. 16, no. 12, 2016.
- [9] A. Revert, F. Garelli, J. Pic, H. D. Battista, P. Rossetti, J. Vehi, and J. Bondia, "Safety auxiliary feedback element for the artificial pancreas in type 1 diabetes," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 8, pp. 2113–2122, Aug. 2013.
- [10] Gutierrez-Osuna and N. HT., "A method for evaluating data-preprocessing techniques for odour classification with an array of gas sensors," *IEEE Trans Syst Man Cybern B Cybern.*, vol. 29, no. 5, pp. 626 – 632, 1999.
- [11] E. I. Georga, V. C. Protopappas, D. Ardig, M. Marina, I. Zavaroni, D. Polyzos, and D. I. Fotiadis, "Multivariate prediction of subcutaneous

- glucose concentration in type 1 diabetes patients based on support vector regression," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 1, pp. 71–81, Jan 2013.
- [12] K. Yan and D. Zhang, "Blood glucose prediction by breath analysis system with feature selection and model fusion," in *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 6406–6409.
- [13] S. Bedoui, R. Faleh, H. Samet, and A. Kachouri, "Electronic nose system and principal component analysis technique for gases identification," in *10th International Multi-Conferences on Systems, Signals Devices 2013 (SSD13)*, Mar. 2013, pp. 1–6.
- [14] A. Loutfi, S. Coradeschi, G. K. Mani, P. Shankar, and J. B. B. Rayappan, "Electronic noses for food quality: A review," *Journal of Food Engineering*, vol. 144, pp. 103 – 111, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0260877414003276>
- [15] C. Pérez-Gandía, A. Facchinetti, G. Sparacino, C. Cobelli, E. Gómez, M. Rigla, A. de Leiva, and M. Hernando, "Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring," *Diabetes technology & therapeutics*, vol. 12, no. 1, pp. 81–88, 2010.
- [16] C. Zecchin, A. Facchinetti, G. Sparacino, G. D. Nicolao, and C. Cobelli, "Neural network incorporating meal information improves accuracy of short-time prediction of glucose concentration," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1550–1560, Jun. 2012.
- [17] K. Plis, R. Bunesco, C. Marling, J. Shubrook, and F. Schwartz, "A machine learning approach to predicting blood glucose levels for diabetes management," in *Modern Artificial Intelligence for Health Analytics Papers from the AAAI-14*.
- [18] M. v. d. W. H.N. Mhaskar, S.V. Pereverzyev, "A deep learning approach to diabetic blood glucose prediction," <https://arxiv.org/abs/1707.05828>, 2017.
- [19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: ACM, 2014, pp. 675–678. [Online]. Available: <http://doi.acm.org/10.1145/2647868.2654889>
- [20] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, I. A. Julian Schrittwieser, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, p. 484489, Jan. 2016.
- [21] G. Litjens, C. I. Snchez, N. Timofeeva, M. Hermesen, I. Nagtegaal, I. Kovacs, C. H. van de Kaa, P. Bult, B. van Ginneken, and J. van der Laak, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," *Scientific Reports*, vol. 6, p. 26286, May 2016.
- [22] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608014002135>
- [23] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, p. 111, 2017. [Online]. Available: <http://dx.doi.org/10.1093/bib/bbx044>
- [24] K. Li and P. H. Y. G. P. G. F. Liu, H. Dong, "A deep learning platform for diabetes big data analysis," in *11th Advanced Technologies and Treatments for Diabetes*. Mary Ann Liabery, 2018, pp. A116–A116.
- [25] T. Zhu, K. Li, P. Herrero, J. Chen, and P. Georgiou, "A deep learning algorithm for personalized blood glucose prediction."
- [26] Y. Bengio, "Deep learning of representations: Looking forward," in *Statistical Language and Speech Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–37.
- [27] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018, pp. 8827–8836.
- [28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [30] C. Zhao, E. Dassau, L. Jovanovic, H. C. Zisser, I. Francis J. Doyle, and D. E. Seborg, "Predicting subcutaneous glucose concentration using a latent-variable-based statistical method for type 1 diabetes mellitus," *Journal of Diabetes Science and Technology*, vol. 6, no. 3, pp. 617–633, 2012, pMID: 22768893.
- [31] D. A. Finan, F. J. Doyle III, C. C. Palerm, W. C. Bevier, H. C. Zisser, L. Jovanović, and D. E. Seborg, "Experimental evaluation of a recursive model identification technique for type 1 diabetes," *Journal of diabetes science and technology*, vol. 3, no. 5, pp. 1192–1202, 2009.
- [32] M. Reddy, P. Pesl, M. Xenou, C. Toumazou, D. Johnston, P. Georgiou, P. Herrero, and N. Oliver, "Clinical safety and feasibility of the advanced bolus calculator for type 1 diabetes based on case-based reasoning: A 6-week nonrandomized single-arm pilot study," *Diabetes Technology & Therapeutics*, vol. 18, no. 8, pp. 487–493, 2016, pMID: 27196358. [Online]. Available: <https://doi.org/10.1089/dia.2015.0413>
- [33] R. Atanassov, P. Bose, M. Couture, A. Maheshwari, P. Morin, M. Paquette, M. Smid, and S. Wuhrer, "Algorithms for optimal outlier removal," *Journal of Discrete Algorithms*, vol. 7, no. 2, pp. 239 – 248, 2009, selected papers from the 2nd Algorithms and Complexity in Durham Workshop ACiD 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570866709000021>
- [34] H. Shum, K. Ikeuchi, and R. Reddy, *Principal Component Analysis with Missing Data and Its Application to Polyhedral Object Modeling*. Boston, MA: Springer US, 2001, pp. 3–39. [Online]. Available: https://doi.org/10.1007/978-1-4615-0797-0_1
- [35] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003. [Online]. Available: <https://doi.org/10.1080/713827181>
- [36] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: A study of fuzzy k-means clustering method," in *Rough Sets and Current Trends in Computing*, S. Tsumoto, R. Słowiński, J. Komorowski, and J. W. Grzymała-Busse, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 573–579.
- [37] L. Lekha and S. M., "Real-time non-invasive detection and classification of diabetes using modified convolution neural network," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2018.
- [38] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [39] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Computer Methods and Programs in Biomedicine*, vol. 161, pp. 1 – 13, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0169260718301226>
- [40] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [41] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE Int. Conf. on Acou., Spe. and Sig. Proc.*, 2013, p. 6645.
- [42] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, cite arxiv:1409.2329. [Online]. Available: <http://arxiv.org/abs/1409.2329>
- [43] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [44] "The uva/padova type 1 diabetes simulator," *Journal of Diabetes Sci Technol.*, vol. 8, no. 1.
- [45] G. Sparacino, F. Zanderigo, S. Corazza, A. Maran, A. Facchinetti, and C. Cobelli, "Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 5, pp. 931–937, May 2007.
- [46] S. G. Mougakakou, A. Prountzou, D. Iliopoulou, K. S. Nikita, A. Vazeou, and C. S. Bartsocas, "Neural network based glucose - insulin metabolism models for children with type 1 diabetes," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2006, pp. 3545–3548.
- [47] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1 – 15, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1051200417302385>