Robust Data-Driven Control of Artificial Pancreas Systems using Neural Networks

Souradeep Dutta, Taisa Kushner and Sriram Sankaranarayanan.

University of Colorado, Boulder, CO, USA,
firstname.lastname@colorado.edu

Abstract. In this paper, we provide an approach to data-driven control for artificial pancreas system by learning neural network models of human insulin-glucose physiology from available patient data and using a mixed integer optimization approach to control blood glucose levels in real-time using the inferred models. First, our approach learns neural networks to predict the future blood glucose values from given data on insulin infusion and their resulting effects on blood glucose levels. However, to provide guarantees on the resulting model, we use quantile regression to fit multiple neural networks that predict upper and lower quantiles of the future blood glucose levels, in addition to the mean.
Using the inferred set of neural networks, we formulate a model-predictive control scheme that adjusts both basal and bolus insulin delivery to ensure that the risk of harmful hypoglycemia and hyperglycemia are bounded using the quantile models while the mean prediction stays as close as possible to the desired target. We discuss how this scheme can handle disturbances from large unannounced meals as well as infeasibilities that result from situations where the uncertainties in future glucose predictions are too high. We experimentally evaluate this approach on data obtained from a set of 17 patients over a course of 40 nights per patient. Furthermore, we also test our approach using neural networks obtained from virtual patient models available through the UVA-Padova simulator for type-1 diabetes.

1 Introduction

This paper investigates the use of neural networks as data-driven models of insulin glucose regulation in the human body. Furthermore, we show how the networks can be used as part of a *robust* control design that can compute optimal insulin infusion for patients with type-1 diabetes based on the predictive models. We test our approach in instances of large disturbances such as unannounced meals consisting of over 130g of carbs. Such a scheme is commonly known as *model-predictive control* (MPC)[8]. Neural networks have emerged as a versatile data-driven approach to modeling numerous processes, including biological processes. They can model nonlinear functions, and be trained from data using the well-known backpropagation algorithm[22]. This has led to numerous machine learning algorithms that use neural networks for regression and classification tasks.

We first consider the process of learning neural network models from data. To do so, we used two different data sets: a *synthetic* dataset using a well-known differential equation model of human insulin regulation was simulated under randomly varying

conditions [15,17]; and a clinical dataset consisting of 17 patients with 40 nights per patient [34]. Both datasets report blood glucose (BG) levels at 5 minute intervals along with the insulin infusions provided. From these datasets, we infer a feedforward neural network that employs 180 minutes of BG and insulin values to predict the likely BG value 30 minutes into the future. However, a key challenge lies in capturing the uncertainty inherent in this value. To do so, we infer multiple networks including a network that is trained to predict the mean BG level, as well as networks that are trained to predict upper and lower quantiles using *quantile regression* [30]. The upper (lower) α quantile network attempts to predict a value that will lie above (below) the actual value with α probability.

Next, we use the prediction models to calculate the optimal insulin infusion for a patient given the initial history of BG values to (a) maintain a target BG level well inside the normal range of $[70, 180]\texttt{mg/dL}$; (b) ensure that the risk of hypoglycemia (BG $\leq$ $70\texttt{mg/dL}$) and high BG levels (BG $\geq$ $210\texttt{mg/dL}$) are bounded. The latter objectives are achieved by enforcing the constraint that the lower and upper quantile networks must predict values that are within the range.

We evaluate our approach by exploring an optimal network structure to improve prediction accuracy. Our approach yields relatively small networks with 16 neurons in two hidden layers that can predict the BG levels with a mean prediction error of about $7\texttt{mg/dL}$. This error is comparable to the measurement error of the sensor used in the prediction.

Next, we evaluate the resulting control scheme on two different types of experiments. In one experiment, a neural network based predictive model is learned offline against a popular ODE-based simulation model proposed by Dalla-Man et al [17,15]. In another set of experiments, two different neural network models are trained against the same data set using stochastic gradient descent. One of the models is used as the predictive model and the other is used as a stand-in for a real patient. In both instances, we simulate the controllers under varying initial conditions and disturbances of sensor drop out and unannounced meals. The simulation results show an average time in range of about 73.2% with hypoglycemic incidence rate of only 2% over 2580 simulations.

1.1 Related Work

There has been a recent upsurge of work on data-driven model inference and control synthesis. Data driven inference techniques are being used in applications ranging from high level demand-response strategies for cyber-physical energy systems [3], to artificial pancreas models [37]. In light of this paper, we focus on applications to the latter. Recent work by Paoletti et al [37], used data-driven methods to learn uncertainty sets from historic meal and exercise patterns in order to eliminate the need for meal announcements by the patient. Griva et al. utilized data-driven ARX models to assist the predictive functional control algorithm used in their artificial pancreas. This model achieved good prediction for a 30 minute look ahead time within $10\ \texttt{mg/dL}$ [23]. Perez et al. used past BG levels and a three-layer feed-forward neural network to predict BG values $30 - 45$min out, with accuracy $18 - 27\ \texttt{mg/dL}$. [39]. In comparison, our models achieve accuracy of about $7 - 10\ \texttt{mg/dL}$. Our previous work investigated a data driven modeling approach using linear ARMAX models for verifying closed-loop PID

controllers [32]. Therein, we focused entirely on the verification of controllers and tuning of gains for specific patients. We employed a nondeterministic model with intervals around the prediction used to model errors. In contrast, the neural networks used here are deterministic. However, we use multiple models to predict the mean, and the quantiles.

Model Predictive Control (MPC) is a well known approach to control synthesis [8]. Numerous MPC schemes for insulin infusion control have been constructed based on ODE-based models (see survey by Bequette [4]). Multiple groups have developed nonlinear MPC strategies using neural networks. Piche et al [40] combined a linear dynamic model with a steady-state model learned on historic data to develop MPC solutions. Other have used neural networks to replace either, or both, the controller and plant models in an MPC framework with varied success rates [41,7,47]. To our knowledge this work is the first to propose MPC using multiple neural networks to construct non-deterministic models, and compute optimal infusion schedules using integer linear optimization solvers.

Another contribution of this work is to use multiple predictive models for the mean and the quantiles of the distribution. In this regard, our work is related to that of Cameron et al, who also use multiple models [9]. However, their models are instantiations of multiple possible meal scenarios weighted by their likelihoods during the prediction horizon. In contrast, we use models for the mean and the upper/lower quantiles, which are able to handle large disturbances from unannounced meals.

Lastly, it's worth mentioning that, although neural networks are popular as data-driven models, their applications to safety-critical domains has been limited by the lack of guarantees. Recent work has sought to provide such guarantees by solving verification problems posed on neural networks [19,33,29].

2 Preliminaries

In this section, we present some preliminary notions involving the "artificial pancreas" that controls insulin delivery for patients with type-1 diabetes, and discuss various modeling approaches, including physiological models. Next, we discuss preliminary concepts about neural network and the encoding of neural networks into mixed integer optimization problems.

2.1 Type-1 Diabetes and Artificial Pancreas

Patients with type-1 diabetes depend on external insulin delivery to maintain their blood glucose (BG) levels within a *euglycemic* range of $[70, 180]$ $\texttt{mg/dL}$. BG levels below $70\texttt{mg/dL}$ lead to hypoglycemia which is characterized by a loss of consciousness, coma or even death [11]. On the other side, levels above $180\texttt{mg/dL}$ constitute hyperglycemia which leads to longer term damage to the eyes, kidneys, peripheral nerves and the heart. Levels above $300\texttt{mg/dL}$ are associated with a condition called ketoacidosis, where, due to insufficient insulin, the body breaks down fat for energy resulting in a buildup of ketones. To treat type-1 diabetes, patients receive artificial insulin externally

either through multiple daily injections, or insulin pumps [11]. The latter allows a continuous basal infusion at a pre-programmed rate through the day along with large insulin boluses delivered to counteract the effect of meals. For the most part, the management of BG levels is performed manually by the patient. This requires careful counting of meal carbohydrates, and almost constant vigilance on the part of the patient, as relatively minor errors result in poor BG control at best, and life threatening hypoglycemia or ketacidosis, in the worst cases.

Artificial pancreas (AP) systems look to ease the burden of BG control by automating the delivery of insulin through an insulin pump, using continuous glucose sensors to periodically measure the BG levels and employing a control algorithm to decide on how much insulin to deliver [14,28,44]. A variety of strategies have been proposed for the artificial pancreas, ranging from relatively simple pump shutoff controllers [35], PID-based algorithms [42,48,43], rule-based systems [2,36], predictive controllers that use a model to forecast future BG trends in the patient against planned future insulin infusions [9,26,4]. Furthermore, the control is classified as either fully closed loop, wherein the user is (in theory) not needed to announce impending meals or exercise versus hybrid closed loops which continue to rely on users to bolus for meals or announce impending exercise [31]. Numerous control algorithms are currently in various stages of clinical trials. The Medtronic 670G device, based on a PID control algorithm, was recently approved by the US FDA and is available as a commercial product [25,21].

A key consideration for many AP devices involves adapting the insulin delivery to the personal characteristics of the patient. Patients display a wide range of variability in their response to insulin [37]. This variability is crudely summarized into numbers such as the daily insulin requirement and the insulin to carbohydrate ratio, in order to calculate insulin requirements for basal insulin and meal boluses. However, in order to achieve safe and effective control, it is important to model many additional aspects of the patient's insulin-glucose response [32]. Thus, the challenge involves how to build mathematical models that capture important aspects of a specific patient's physiology.

2.2 Mathematical Models

Mathematical models of insulin-glucose response have a long history. Bergman proposed a minimal ODE-based model of insulin-glucose physiology [12,6]. His model involves three state-variables, captures the nonlinear insulin-glucose response. At the same time, it does not model aspects such as endogenous glucose production by the liver, insulin dependent vs. insulin independent uptake of glucose by various tissues in the body, and the effect of renal clearance of glucose that happens during hyperglycemia. Finally, the model assumes direct glucose and insulin inputs into the blood stream.

More detailed physiological models have been proposed, notably by Hovorka et al. [27] and Dalla Man et al [18,15]. These models address many of the missing aspects of the original Bergman model. A recently updated version of the Dalla Man model accounts for the effect of fasting and exercise through the counter-regulatory hormone glucagon [16]. The Dalla-Man ODE model has been approved by the US FDA as a replacement for animal trials in testing control algorithms at the pre-clinical stage [38].

A key critique of ODE-based physiological models involves the estimation of model parameters corresponding to the available patient data. The Dalla-Man model involves upwards of 40 patient parameters which need to be identified to model a particular individual. Some of these parameters may in fact be time varying. Furthermore, the model involves state variables that cannot be directly measured without intrusive radiological tracer studies. For most patients, available data consists mainly of noisy measurements of BG levels coupled with insulin infusion logs.

2.3 Neural Network

Neural networks are a *connectionist*, data-driven model that represents functions from a domain of inputs to outputs. There are two types of neural networks: (a) feedforward neural networks which do not have internal memory; and (b) recurrent neural networks that have internal memory in the form of units called *long short term memory* (LSTM). In this paper, we focus exclusively on feedforward neural network models, but briefly discuss recurrent neural networks since they form a viable modeling option that was not chosen for this paper.

A feedforward network $\mathcal{N}$ consists of n inputs, m outputs, and k hidden layers with $N_1, \ldots, N_k$ neurons in each of the hidden layers. The j^{th} neuron in the i^{th} hidden layer is denoted $N_{i,j}$ for $1 \leq i \leq k$ and $1 \leq j \leq N_i$. The inputs are connected to the first hidden layer, each hidden layer i is connected to the subsequent hidden layer $i+1$ for $1 \leq i \leq k-1$, and finally, the last hidden layer is connected to the output layer. The connections have associated weights denoted by matrices W_i and biases denoted by a vector $\mathbf{b}_i$.

Definition 1 (Feedforward Neural Networks). *Formally, a feedforward network is a tuple* $\langle n, m, k, (N_i)_{i=1}^{k}, (W_i, \mathbf{b}_i)_{i=0}^{k} \rangle$*, modeling a function* $F_N : \mathbb{R}^n \mapsto \mathbb{R}^m$ *wherein the weights of the connection from input layer to the first hidden layer are given by* $(W_0, \mathbf{b}_0)$ *with* $W_0 \in \mathbb{R}^{N_0, n}, \mathbf{b}_0 \in \mathbb{R}^{N_0}$*, the connection from layer* i *to* $i+1$ *is given by* $(W_i, \mathbf{b}_i)$ *for* $1 \leq i \leq k-1$*, where* $W_i \in \mathbb{R}^{N_{i+1}, N_i}$ *and* $\mathbf{b}_i \in \mathbb{R}^{N_{i+1}}$*, and finally* $(W_k, \mathbf{b}_k)$ *are the weights connecting the last hidden layer to the output layer.*

Each neuron $N_{i,j}$ is associated with a nonlinear activation function $\sigma_{i,j} : \mathbb{R} \mapsto \mathbb{R}$. Table 1 lists the commonly used activation function, with the ReLU being the most popular, recently. For convenience, we assume that all neurons have the same activation function σ. We lift σ to vectors of variables as $\sigma(\mathbf{x}) : \begin{pmatrix} \sigma(x_1) \\ \vdots \\ \sigma(x_n) \end{pmatrix}$. A neural network N computes a function $F_N : \mathbb{R}^n \mapsto \mathbb{R}^m$. Given a feedforward neural network, the function computed is defined as a composition of two types of functions (a) hidden layer functions $F_i : \sigma(W_i \mathbf{x}_i + \mathbf{b}_i)$, and (b) the output function $F_{out} : W_k \mathbf{x}_k + \mathbf{b}_k$. The overall function computed by the network is $F_N(\mathbf{x}) : F_{out}(F_{k-1}(\cdots(F_0(\mathbf{x}))\cdots))$.

Table 1. Commonly used activation functions.

Name	$\sigma(z)$
RELU	$\max(z, 0)$
SIGMOID	$\frac{1}{1+e^{-z}}$
TANH	$\tanh(z)$
PRELU	$\max(\alpha z, z)$
SOFTPLUS	$\log(1+\exp(z))$

Feedforward vs. Recurrent: Feedforward networks are useful in modeling functions from input to output. They are used in problems such as learning a function from data through regression and classifying between different categories. As such, they do not have internal states. Modeling sequences involves recurrent neural networks that augment feedforward networks with "feedback connections" through a series of memory units which can remember values between successive time steps. In analogy with digital circuits, feedforward networks are analogous to combinational circuits made from logic gates such as and/or/not, whereas recurrent networks correspond to sequential circuits that involve feedback using memory elements such as flip-flops. In this paper, we focus entirely on data-driven models using feedforward neural networks. We justify our choice on the basis of three factors: (a) the networks are easier to model and we encode some of the existing knowledge in terms of the structure of the network to avoid overfitting; (b) the networks are easier to train and (c) finally, we provide simpler approaches to reason about these networks. On the other hand, using a recurrent network can simplify some of the choices made in our model.

2.4 Encoding Networks into Constraints

A core primitive in this paper is to solve control problems involving neural networks as predictive models. We recall how the function computed by a network can be systematically modeled as a set of mixed integer linear constraints [45]. This linear programming (LP) encoding is standard, and has been presented in details elsewhere [19,33].

Let $\mathcal{N}$ be a network with n inputs, $k-1$ hidden layers, and a single output. We restrict our discussion in this section to ReLU activation units. Let $\mathbf{x} \in \mathbb{R}^n$ be the input to a neural network, represented as n (LP) variables, $\{F_1, F_2, \ldots, F_{k-1}\}$ represent the outputs of the hidden layers, and $y \in \mathbb{R}$ be the (LP) variables representing the output of the network. Let, $\{W_0 \ldots W_k\}$, and $\{\mathbf{b}_0 \ldots \mathbf{b}_k\}$, be the weights and biases, as described in Definition 1.

We introduce binary (LP) variable vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}\}$. Such that for each variable in the vector, $v_i[j] \in \{0, 1\}$. The binary variables are introduced in order to model the piecewise linear nature of ReLU units. That is, if $v_i[j]$ is 1, the ReLU is off, and on otherwise.

At a hidden layer i, the network constraints require that $F_{i+1} = ReLU(W_i F_i + b_i)$. We use the binary (LP) variables v_{i+1} to encode the piecewise linear behavior of ReLU units.

$$\mathsf{C}_{i+1} : \begin{cases} F_{i+1} \geq W_i F_i + b_i, \\ F_{i+1} \leq W_i F_i + b_i + \mathsf{M}\mathbf{v}_{i+1}, \\ F_{i+1} \geq \mathbf{0}, \\ F_{i+1} \leq \mathsf{M}(\mathbf{1} - v_{i+1}) \end{cases}$$

Where M is a "large enough" constant. In practice, the number M often decides the performance of the solver, and it is possible to derive tight estimates of M using interval analysis on the network [33]. Thus we can combine these constraints, to form the encoding of the entire network: $C : \mathsf{C}_0 \wedge \ldots \wedge \mathsf{C}_{k+1}$. Next, we can use additional constraints on the inputs and outputs of the neural network $\mathcal{N}$, to find feasible assignments.

3 Data Driven Models with Neural Networks

In order to predict future blood glucose values for each patient, we used the various patient datasets to train neural-network models. In this section, we describe the data sources, followed by a description of our prediction models. Next, we will describe and justify our choice of the neural network structure. Finally, we will describe the training process and the results obtained.

3.1 Data Sources

The paper examines two different sources of data: (a) synthetic data obtained from running the UVA-Padova simulation model over a randomly selected set of meals and outputting the insulin input and BG values encountered over time; and (b) clinical trial data for a pump shutoff control algorithm recorded "longitudinal data" for $n = 17$ patients with 40 nightly sessions for each patient.

Synthetic Data Collection: The synthetic data collection is based on the Dalla-Man model [15,17,18], a 10 state variable nonlinear ODE model with upwards of 40 patient parameters. We selected 6 parameter sets describing virtual adult patients to perform our simulation. Each daytime simulation run for a patient involved two randomly sized simulated meals with the first meal having up to $[20, 35]$ grams of carbohydrates (CHO) and the second meal involving $[35, 135]$ grams of CHO. The insulin infusion was controlled by the "multi-basal" controller reported in our previous work [13]. The output data involved the BG values reported in 5 minute intervals, and the insulin infusion also reported in 5 minute intervals. Overall, the collected data consists of 6 different patients with 500 simulation runs per patient.

PSO3 Clinical Trial Data: The patient data was obtained from a previously held home trial of a predictive pump shutoff algorithm [10]. This pump is not an artificial pancreas and does not adjust insulin delivery rates, however, it does shut off delivery when BG is predicted (via Kalman filter) to drop below a threshold. We use a total of 17 patients with 40 nights of data per patient [34]. The collected data reports the BG levels collected every 5 minutes and the insulin infusion logs reporting basal insulin and boluses delivered. The logs were converted into insulin infusions at 5 minute intervals (units delivered/5min). For each nightly session, we discarded the first three hours worth of data to mitigate the effect of an unknown meal size and insulin bolus delivered just before the start of the session. This provided roughly 8 hours of glucose and insulin data for each session. Entire nightly sessions were discarded from consideration for two main reasons: (a) the patient suffered from a hypoglycemia and were treated using "rescue" carbohydrates; and (b) the sensor readings were incomplete due to dropouts or data recording issues.

3.2 Prediction Model Structure

We now describe the structure of the prediction model. There are two aspects to the prediction model. Let t be the current time in minutes. Our goal is to predict the glucose value at a future "lookahead" time $t + T$, using the history of past N glucose values

with a "past" step size of h: $G(t), G(t-h), \ldots, G(t-Nh)$ and past M insulin values $u(t), u(t-h), \ldots, u(t-Mh)$. The nature of the dataset, described previously, restricts the value of $h = 5$ minutes.

To decide on the value of the future lookahead, we note some known physiological details of insulin action: (a) insulin infused subcutaneously has a delay of $20-30$ minutes before it reaches the patient's blood stream and starts action; (b) insulin has an action profile with peak action time of about 75 minutes and persists in the blood stream up to 4 hours [20]. Therefore, to model the effect of the insulin administered at time t, we set the future lookahead $T = 30$ minutes. Our attempts to infer models for shorter lookahead such as $T = 5$ minutes leads to accurate but ultimately useless models that predict $G(t+5) = G(t)$. Next, we set $Mh = 180$ minutes to account for the entire past history of insulin which may affect the value at $G(t+30)$. Finally, to simplify the structure of the prediction model, we set $N = M$.

Let $\overleftarrow{G}_N(t)$ denote the column vector consisting of the N past glucose values $(G(t), G(t-h), \ldots, G(t-Nh))$ and likewise $\overleftarrow{u}_N(t)$ denote the past N insulin values $(u(t), \ldots, u(t-Nh))$. Thus, the desired prediction model is a distribution-valued function:

$$G(t+T) \sim F(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t)) .$$

wherein $F(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t))$ is a random variable that models the distribution of the predicted values. Unfortunately, modeling and reasoning about distributions is a hard problem. Therefore, we use a simpler approach using finitely many models that predict the mean and quantiles of the distribution.

The mean predictive model captures the expected value at time $t+T$:

$$\mathbb{E}(G(t+T)) = F_m(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t)) .$$

Likewise, we wish to formulate α upper quantile models $\overline{F}_\alpha(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t))$ such that

$$\mathbb{P}\left\{G(t+T) \geq \overline{F}_\alpha(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t))\right\} \leq (1-\alpha) .$$

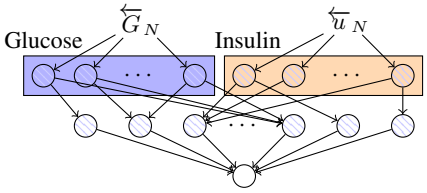
For instance, for $\alpha = 0.95$, we train an upper quantile model that predicts a level $\overline{F}_{0.95}$ which can be exceeded with probability at most 0.05. Similarly, an α lower quantile model $\underline{F}_\alpha(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t))$ seeks a prediction such that

$$\mathbb{P}\left\{G(t+T) \leq \underline{F}_\alpha(\overleftarrow{G}_N(t), \overleftarrow{u}_N(t))\right\} \leq (1-\alpha) .$$

In other words, $\underline{F}_{0.95}$ model predicts a value such that the actual value will lie below the predicted value with probability at most 0.05. The overall distribution is represented by a *mean* predictive model F_m and a set of upper quantile models $\overline{F}_\alpha$ and lower quantile models $\underline{F}_\beta$ for selected values of α and β.

Neural Network Structure:

Figure 1(a) shows the overall structure of the neural network used in our model. Our neural networks involve two layers with the input layer partitioned into two parts: glucose inputs and insulin inputs. Next, based on existing ODE models [5,27,15] we



(a)

# neurons in layer 2	# neurons in layer 1				
	3	**6**	**8**	**20**	**200**
3	1152	-	-	-	-
4	-	623	8558	4518	-
5	-	5.57	6.82	5.98	-
6	-	10.34	3.37	2.15	-
8	-	6.20	1.34	15.85	-
10	-	3.21	7.45	5.99	-
20	-	-	-	-	323.98

(b)

Fig. 1. **(a)** Neural network structure, **(b)** Mean training error (mg/dL) with varying combinations of neurons in each layer. Values presented are averaged over three trials for real patient ID: PSO3-001-0001.

partition the first hidden layer into two parts meant to model *insulin transport* and *glucose transport*. Finally, we add a joint hidden layer to model the insulin action on glucose. Besides this, note that our model does not have internal states. Rather, we assume that this state is well captured by the history of glucose and insulin inputs of the model. The number of neurons in the hidden layer was chosen by training a series of networks with different hidden layer sizes and choosing the best performing model, as will be described subsequently in this section.

Loss Functions: Our goal is to train neural network predictive models F_m for the mean prediction and the quantile models for lower quantiles $\underline{F}_\beta$ and upper quantiles $\overline{F}_\alpha$, respectively. These are achieved using the same regression process that chooses unknown network weights $\mathcal{W} : \{(W_0, \mathbf{b}_0), (W_1, \mathbf{b}_1), (W_2, \mathbf{b}_2)\}$ that minimize a loss function $\mathcal{L}$ over the prediction error $e(t; \mathcal{W}) : G(t+T) - F_{\mathcal{W}}(\overleftarrow{G}_N, \overleftarrow{u}_N)$ at time t, wherein the values of $G(t), u(t)$ are obtained from the training data: $\mathcal{W} :$ argmin $\sum_t \mathcal{L}(e(t; \mathcal{W}))$. It is well known that the mean of a distribution is obtained as the minimizer of a quadratic loss function over the samples $\mathcal{L}_m(e) : ||e||_2^2$. Likewise, the quantiles are also obtained by minimizing loss functions corresponding to α upper quantiles [30].

$$\mathcal{L}_\alpha(e) = \max(-\alpha e, (1-\alpha)e)\,.$$

Similarly, the loss function for the β lower quantile minimizes the upper quantile loss function for $1-\beta$: $\mathcal{L}_{1-\beta}$. It must be remarked that precise quantile regression requires large amounts of data since we are fitting a function at the tails of the distributions we wish to model, and consequently the process can be sensitive to outliers.

Training: The network weights are initialized at random and trained using off the shelf backpropagation algorithms implemented in the popular package TensorFlow [1]. The training process supports user specified loss functions as long as they are differentiable. To this end, a differentiable approximation of $\mathcal{L}_\alpha$ was obtained by replacing the $\max$ with a *softmax* operator: $\mathsf{softmax}(x, y) = \frac{e^x}{e^x + e^y}$. The training is performed by first partitioning the given data into different training and test sets. Next, gradient descent is performed over randomly selected batches of data points. Once a model is trained, its accuracy is evaluated over the test data. The training is performed by selecting loss

functions for the mean model, the $\alpha = 0.95$ upper quantile and $\beta = 0.8$ lower quantile. The reason for the asymmetry is that our training results for $\beta = 0.95$ yielded a model with very poor predictive accuracy over the test data, rendering it useless. Training with $\beta = 0.8$ yielded a tighter lower bound in 94% of tests cases. The approach results in 3 neural network models F_m, $\overline{F}_{0.95}$ and $\underline{F}_{0.8}$.

3.3 Evaluation Results.

In order to select number of neurons in each hidden layer, we tested networks with $4 - 200$ neurons. Figure 1(**b**) shows mean error for various selected network structures, centered around those which gave lowest test error. Choosing 8 neurons per hidden layer produces the optimal result, a relatively small network with low prediction error.

Overall, we have found a two-layer network topology to be sufficient for obtaining fairly high prediction accuracies of 7.2 ± 3.0 mg/dL, which is competitive with the error bounds on current predictive models. [24,23]

We were able to successfully train neural network models to fit all but two real-data patients, wherein success is defined as test error $< 12\texttt{mg/dL}$, and all synthetic data patients. Simple factors such as the amount of data or the variability in the BG levels fail to account for the two failure cases. Further in-depth analysis will be is being completed for future work. For the successful instances, the mean prediction error is 7.1867 ± 3.028 mg/dL. Training a model for each patient required 328 seconds on average, as measured on a MacBook Pro laptop with 3.3GHz Intel Core i7 processor.

In Fig. 2 we present example model predictions, and in Fig. 3, we present histograms of error for the quantile networks.

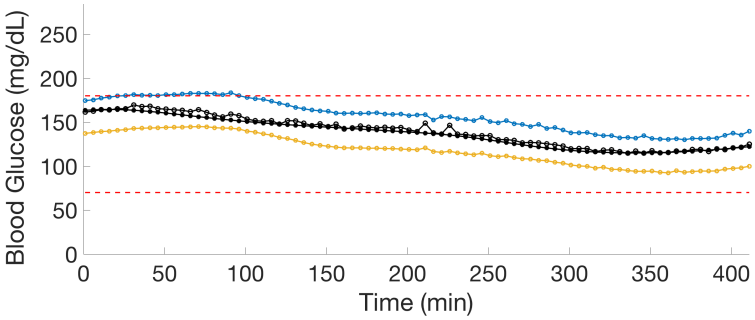


Fig. 2. Sample plot comparing our neural network model predictions to real data. Legend: Filled black circles: actual data, open black circles: mean network prediction, open blue circles: upper 0.95 quantiles, open yellow circles: lower 0.8 quantiles. The dashed red lines show the normal blood glucose ranges.

4 Model Predictive Control

In this section, we present the overall controller design that uses the neural networks inferred in the previous section as a starting point. The predictive controller takes as inputs:

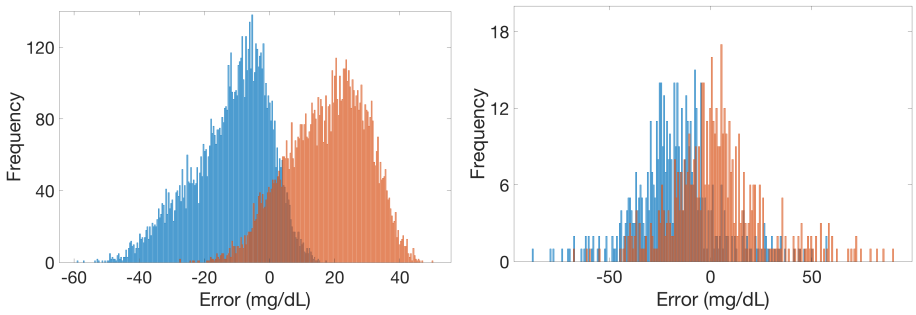


Fig. 3. Histograms of error for the lower quantile regression (orange), and the upper quantile regression (blue) models learned on synthetic data (left) and real data (right). Error is defined as $y_{data} - y_{mdl}$, positive error indicates our model is predicting below the data, and negative error indicates our model prediction is above the data.

1. Neural network models F_m, $\overline{F}_\alpha$ and $\underline{F}_\beta$ for the mean, upper quantile and lower quantile predictions, respectively, for time $t+T$ given historical data for times in the range $[t-Nh, t]$; and
2. Historical data $\overleftarrow{G}_N(t)$ and $\overleftarrow{u}_N(t)$ for glucose and insulin.

We wish to compute an *optimal schedule* of future insulin infusions over some time horizon Kh, where h is the time period for the controller: $u(t+h),\ u(t+2h),\ \ldots, u(t+(K-1)h)$, to achieve the following goals:

1. The mean predicted BG levels at time $\mathcal{T} = Kh$ lie in some goal range $[G^*_{min}, G^*_{max}]$.
2. The lower quantile BG levels as predicted by $\underline{F}_\beta$ must lie above the hypoglycemia limit of $70\texttt{mg/dL}$.
3. The upper quantile BG levels as predicted by $\overline{F}_\alpha$ must lie below an upper hyperglycemia limit taken to be $210\texttt{mg/dL}$.

This control can be computed by solving a mixed integer linear program (MILP). The unknowns for this problem are as follows:

1. The predicted mean BG values for the future $G(t+h), \ldots, G(t+Kh)$. Thus, will assume that $G(s)$ denotes a known data point if $s \leq t$ and an unknown variable in the MILP if $s > t$.
2. The lower quantile BG values $G_l(t+h), \ldots, G_l(t+Kh)$,
3. The upper quantile BG values $G_u(t+h), \ldots, G_u(t+Kh)$.
4. The insulin infusions $u(t+h), \ldots, u(t+Kh)$. Once again $u(s)$ denotes a known data point if $s \leq t$ and an unknown variable in the MILP, otherwise.

Other unknowns will be introduced when we encode networks as MILP, as we will describe subsequently.

We will now describe the constraints that need to be enforced; fixed parameters are shown in blue. Figure 4 shows how the networks for the mean, upper and lower quantiles interact in the system of constraints shown below.

1. The final predicted mean value must be within bounds: $G^*_{min} \leq G(t+Kh) \leq G^*_{max}$.
2. The lower and upper quantiles must lie within hypo and hyper glycemia limits: $70\texttt{mg/dL} \leq G_l(t+Kh)$, and $G_u(t+Kh) \leq 210\texttt{mg/dL}$.

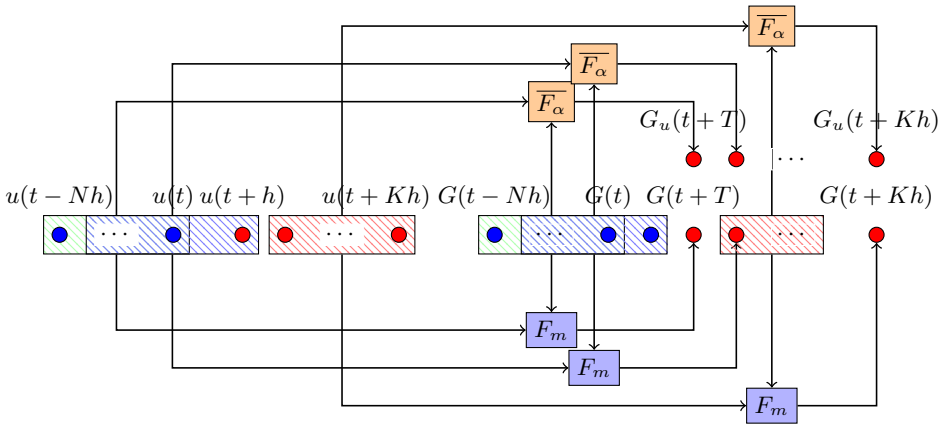


Fig. 4. Schematic illustration of the recursive unfolding setup for the model predictive control optimization problem. The red dots represent unknown variables, whereas the blue dots represent known data at optimization time. The unknown mean glucose values $G_m(t+h), \ldots, G(t+Kh)$ are constrained to be the mean predictions through the network F_m. Similarly, the unknown upper bounds $G_u(t+h), \ldots, G_u(t+Kh)$ are obtained using the network $\overline{F_\alpha}$. The lower bounds $G_l(t)$ are not shown in this figure.

3. The values of the insulin infused must remain within limits: $u_l \leq u(s) \leq u_h$, $s \in \{t+h, \ldots, t+(K-1)h\}$.
4. The total insulin delivered must be within limit: $\sum_{s=t-Nh}^{t+(K-1)h} u(s) \leq U_{tot}$.
5. The glucose level $G(s)$ must be predicted according to the network F_m: $G(t+jh) = F_m(\overleftarrow{G}(t+jh-T), \overleftarrow{u}(t+jh-T))$, $s \in \{t+h, \ldots, t+(K-1)h\}$.
6. The upper limit $G_u(s)$ must be as predicted using the network $\overline{F_\alpha}$: $G_u(t+jh) = \overline{F_\alpha}(\overleftarrow{G}(t+jh-T), \overleftarrow{u}(t+jh-T))$, $s \in \{t+h, \ldots, t+(K-1)h\}$.
7. The lower limit $G_l(s)$ must be predicted using the network $\overline{F_\beta}$: $G_l(t+jh) = \overline{F_\beta}(\overleftarrow{G}(t+jh-T), \overleftarrow{u}(t+jh-T))$, $s \in \{t+h, \ldots, t+(K-1)h\}$.

The overall MILP formulation combines the constraints noted above with the objective that minimizes the deviation at time $t+Kh$ from a desired target value: $|G(t+Kh) - G^*|$. Note that constraints involve equalities written as $z = F(x,y)$, wherein F is the function computed by a neural network. These constraints are converted into MILP constraints through the introduction of fresh binary variables, as described in Section 2.4. The objective involves minimizing the absolute value of a linear expression, and can be converted to a linear objective in a standard manner [45].

Filling-in Prediction Gap: Since our approach uses history up to time t to predict the value at $t+T$, there is a potential "gap" in our predictions for BG values in the set $\{t+h, \ldots, t+T-h\}$ that require historical values past what was assumed to be available. There are two solutions: (a) Rather than use historical values spanning $[t-Nh, t]$, we will extend our history to $[t-Nh-T, t]$ and thus eliminate the gap; (b) Alternatively, we will assume unknown values for the "missing" history in the range

Table 2. Summary of results for clinical trial and synthetic datasets. Legend: #P - number of patients, # Tr - number of random trials per patient, T.H - total time horizon for the simulations, Avg. Hyper - Average number of trials resulting in hyper glycemia, Avg. Hypo - Average number of trials resulting in hypoglycemia, TiR - Percent time in the range, BG $\in [70, 180]$mg/dL.

Dataset	#P	# Tr	T.H.	Avg. Hyper	Avg. Hypo	Avg. TiR
Clin. Trial	15	100	7.5h	15.3	2.7	68.9
Synthetic	6	180	8.5h	1.6	3.6	92.95

$[t - Nh - T, t - (N + 1)h]$ that lie inside a admissible range $[0, 500]$. Furthermore, we can constrain the increase/decrease of successive BG values to a physiologically feasible range.

Deploying the Controller: There are two ways of deploying the control scheme: (a) compute a "single shot" insulin infusion schedule over the time horizon $[t + h, t + Kh]$ and deliver this to the patient; or (b) use some part of the computed insulin infusion schedule and update the schedule in real-time as new BG measurements are obtained. The latter option is called *receding horizon control*, and is preferred since it can act against errors between the predictive model and the actual patient. In practice, we use a receding horizon scheme that computes a control schedule for the entire time horizon $[t + h, t + Kh]$. The first insulin value of the resulting solution is delivered to the patient. At time $t + h$, a new BG measurement is obtained, and the entire computation is restarted over the new horizon $[t + 2h, t + (K + 1)h]$.

Handling Infeasibilities: The optimization problem can be infeasible whenever the quantile models are inconsistent w.r.t the mean model prediction. This can happen for two key reasons: (a) the uncertainty in the future BG values is too large for us to guarantee that both the lower and upper bounds will be safe; (b) the networks are "extrapolating" on inputs that lie far from the training data. Situation (a) can be addressed by dropping the constraints from the problem, starting with the constraint on the upper bound network and hyperglycemia limit. If this does not resolve the infeasibility, we drop the mean value in range constraint. Failing this, we conclude that the predictions are potentially inconsistent with each other. Thus, we revert to a safe insulin infusion, often given by the patient's basal insulin or complete shutoff of insulin.

5 Evaluation

Table 2 summarizes the overall results obtained across the two datasets. For each patient in a given dataset, we inferred predictive models, as described in Section 3. Additionally, for the clinical trial patients, we learned an additional neural network by repeating the backpropagation process. For the synthetic patient, the ODE model was used as the patient model in our simulation. The randomized nature of this process provides us with different results and accuracies for each run. For each patient, we simulated numerous trials with different randomly chosen initial conditions. For all trials, the value of G^* was taken to be 140 mg/dL with the tolerance limit D of 40 mg/dL.

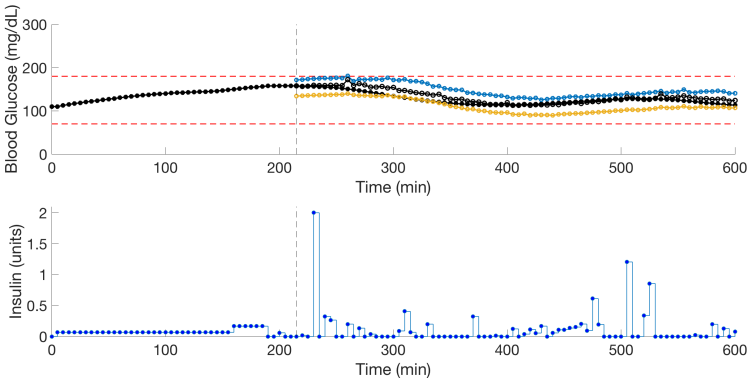


Fig. 5. Example of an instance when the controller gave a "correction bolus" of 2units at $t = 220min$ insulin after the patient ate a meal, then reverted back to injecting only basal insulin of $< 0.5U/5min$ until glucose began to rise again near $t = 500$, where smaller "correction boluses" are given.

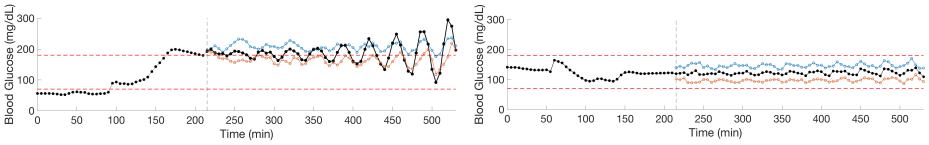
Clinical Trial Data: For this controller, the total insulin allowed was set to $U : 7$ IU over 3.5 hrs, with insulin limits $u_l = 0$ and $u_h = 12$ U/hr. The initial conditions consist of 180 minutes of randomly chosen insulin and BG data in the range $[60, 250]$ $\texttt{mg/dL}$. Whenever the initial condition starts above $180\,\texttt{mg/dL}$, the controller is able to bring it inside euglycemic range 30% of the time. Averaging across patients, 2.7 out of 100 trials resulted in hypoglycemia. In all instances, the controller delivered no insulin whenever BG fell below $90\,\texttt{mg/dL}$. About 6% of the optimization instances resulted in infeasibilities that required us to drop all constraints and revert to 0 insulin delivery. However, in all cases, we were able to recover feasibility and complete the simulation. The controller took a maximum of 2.87s on our hardware platform to execute the control algorithm.

Synthetic Data: The total insulin allowed was set to $U : 5$ IU over 3.5 hrs, with insulin limits $u_l = 0$ and $u_h = 24$ U/hr. For this data, the Dalla Man et al. ODE model was used as the plant model in lieu of a real patient. For each virtual patient, we simulated 180 different combinations of initial BG values and randomly chosen set of two meals over 10 hours. The controller itself was "blind" to these meals, and meals were made particularly large (eg. 135g CHO) in order to maximize disturbances and BG variability, and hence stress the control algorithm. Hyperglycemic events that could likely be attributed by our controller - i.e, events where a patient has $BG < 180\texttt{mg/dL}$ initially, and $BG > 180\texttt{mg/dL}$ at the end occurred in 2.24% of the trials. Infeasibilities were limited to 2.82% on an average. The control computation step took a maximum of 4.65s on our hardware platform. The controller action typically treats high BG by delivering a large insulin bolus and adjusts future insulin delivery down in anticipation of a hypoglycemia (see Fig 5).

Instabilities in Predictive Models: For the clinical trial dataset, certain models exhibited high sensitivity to large jumps in the BG value, leading to apparent instabilities characterized by growing oscillations defined as repeated change in BG values exceeding $20\texttt{mg/dL}$ over each 5 minute period. This was observed in roughly

33% of all the simulation runs involving clinical trial patients. An example of such oscillations is presented in Fig. 6(**a**). The presence of these were correlated to data instances that included "discontinuous" jumps in the initial history, characterized by $|G(t+h) - G(t)| \geq 20\mathtt{mg/dL}$, likely a result of sensor calibration events. Across all initial histories that involved such a discontinuity, 73% of cases led to such unstable oscillations. However, a number of instances involved instabilities although the initial conditions did not have significant jumps.

Despite instabilities in some models, all simulations remained stable for at least for $t = 135$ minutes, and on average models remained stable $t = 221$ minutes, after which growing oscillations began. At the same time, not all cases of discontinuities in the initial history led to such oscillations, as shown in Fig. 6(**b**). Hence, if models are updated with new initial history data every $90 - 120$ minutes, these instabilities should not pose significant issue.



(a) **(b)**

Fig. 6. (**a**) Instance where a calibration error occurring at $t = 95min$, can over time cause large ($> 20\mathtt{mg/dL}$) oscillations in the simulation. **(b)** the calibration error occurring at $t = 55min$, did not cause large ($> 20\mathtt{mg/dL}$) oscillations in the simulation. For each figure, the vertical black line separates the initial data from the simulated data.

Instabilities in the Synthetic Data: We note that these instabilities did not occur in the models learned from the synthetic data. This is consistent with our theory since no discontinuous jumps are possible in the ODE output. This is notable as the synthetic data included large jumps in BG due to meal disturbances, yet no instabilities arose in model prediction.

6 Limitations and Future Work

The first limitation lies in the way the data was collected. Although the PSO3 trial was a home trial of a device, the conditions of the trial could arguably result in extra scrutiny and presumably stricter adherence to clinical norms. This naturally biases the BG values to a tighter range and limits the number of hypoglycemia incidents seen (since it is often a stopping criterion for a participant in a trial). It is also well known that neural networks often perform poorly when tested on data that is "far" away from training data. This is one of the biggest limitations of our approach. To mitigate this, our results must be reproduced on longitudinal data obtained over a longer time periods, so that we may include the vast majority of clinically relevant situations.

Another limitation lies in the use of a neural network as the plant model. Although the network was not identical to the predictive model, it is nevertheless trained from the

same dataset. This can arguably lead to better results than what is possible in reality. However, the inverse is also likely: by using a learned model as the plant, additional noise is introduced, and this could have contributed to instabilities observed in model prediction for the clinical data set. Hence, future work includes improvement to plant model construction.

The Dalla Man model is considered state-of-the-art. Nevertheless, it is hard to fit real patient data to it, due to the numerous parameters involved in the model [46].

Future work will consider alternative network topologies, or varied the history length needed. The use of recurrent neural network is yet another direction for future work. One of the limitations of our approach is the lack of support for meal detection or meal announcements for clinical data, though our controllers were able to bolus for unannounced meals in the case of synthetic data. We will attempt to integrate our work with related and complementary approaches for this problem proposed by Paoletti et al [37], as well as obtain a patient data set which includes daytime data.

7 Conclusions

Thus, we have presented a data-driven approach that infers neural networks for the mean, upper and lower quantiles for predicting future BG levels from data. We formulate a robust control scheme for calculating safe and optimal infusions on this data in the presence of unannounced meals, and sensor errors. Finally, we have evaluated our performance over a variety of datasets, initial histories, patient models, and meal sizes. Our approach shows promising results. However, we also noted instabilities in the model that must be addressed as part of future work.

Acknowledgments

This work was supported by the US National Science Foundation (NSF) through awards 1446900, 1646556, and 1815983. All opinions expressed are those of the authors and not necessarily of the NSF.

References

1. Abadi *et al.*: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR abs/1603.04467 (2016), `http://arxiv.org/abs/1603.04467`
2. Atlas, E., Nimri, R., Miller, S., Grunberg, E.A., Phillip, M.: MD-Logic artificial pancreas system: A pilot study in adults with type 1 diabetes. Diabetes Care 33(5), 1072–1076 (May 2010)
3. Behl, M., Jain, A., Mangharam, R.: Data-driven modeling, control and tools for cyber-physical energy systems. In: Proceedings of the 7th International Conference on Cyber-Physical Systems. pp. 35:1–35:10. ICCPS '16, IEEE Press, Piscataway, NJ, USA (2016)
4. Bequette, B.W.: Algorithms for a closed-loop artificial pancreas: The case for model predictive control. J. Diabetes Science and Technology 7, 1632–1643 (November 2013)
5. Bergman, R.N., Urquhart, J.: The pilot gland approach to the study of insulin secretory dynamics. Recent Progress in Hormone Research 27, 583–605 (1971)

6. Bergman, R.N.: Minimal model: Perspective from 2005. Hormone Research pp. 8–15 (2005)
7. Bhat, N., McAvoy, T.J.: Use of neural nets for dynamic modeling and control of chemical process systems. Computers & Chemical Engineering 14(4-5), 573–582 (1990)
8. Camacho, E., Bordons, C., Alba, C.: Model Predictive Control. Advanced Textbooks in Control and Signal Processing, Springer London (2004)
9. Cameron, F., Niemeyer, G., Bequette, B.W.: Extended multiple model prediction with application to blood glucose regulation. J. Process Control 22(8), 1422–1432 (Sep 2012)
10. Cameron, F., *others*: Inpatient studies of a kalman-filter-based predictive pump shutoff algorithm. J. Diabetes Science and Technology 6(5), 1142–1147 (2012)
11. Chase, H.P., Maahs, D.: Understanding Diabetes (Pink Panther Book). Children's Diabetes Foundation, 12 edn. (2011), available online through CU Denver Barbara Davis Center for Diabetes
12. Chee, F., Fernando, T.: Closed-Loop Control of Blood Glucose. Springer (2007)
13. Chen, X., Dutta, S., Sankaranarayanan, S.: Formal verification of a multi-basal insulin infusion control model. In: Workshop on Applied Verification of Hybrid Systems (ARCH). p. 16. Easychair (2017)
14. Cobelli, C., Dalla Man, C., Sparacino, G., Magni, L., Nicolao, G.D., Kovatchev, B.P.: Diabetes: Models, signals and control (methodological review). IEEE reviews in biomedical engineering 2, 54–95 (2009)
15. Dalla Man, C., Camilleri, M., Cobelli, C.: A system model of oral glucose absorption: validation on gold standard data. IEEE Trans. on Biomedical Engg. 53(12), 2472–2478 (2006)
16. Dalla Man, C., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., Cobelli, C.: The UVa/Padova type I diabetes simulator: New features. J. Diabetes Science and Technology 8(1), 26–34 (2014)
17. Dalla Man, C., Raimondo, D.M., Rizza, R.A., Cobelli, C.: Gim, simulation software of meal glucoseinsulin model (2007)
18. Dalla Man, C., Rizza, R.A., Cobelli, C.: Meal simulation model of the glucose-insulin system. IEEE Trans. on Biomedical Engg. 1(10), 1740–1749 (2006)
19. Dutta, S., Jha, S., Sankaranarayanan, S., Tiwari, A.: Output range analysis for deep feedforward neural networks. In: Dutle, A., Muñoz, C., Narkawicz, A. (eds.) NASA Formal Methods. pp. 121–138. Springer International Publishing, Cham (2018)
20. Freeman, J.S.: Insulin analog therapy: improving the match with physiologic insulin secretion. The Journal of the American Osteopathic Association 109(1), 26–36 (2009)
21. Garg, S.K., , *others*: Glucose outcomes with the in-home use of a hybrid closed-loop insulin delivery system in adolescents and adults with type 1 diabetes. Diabetes Technology and Therapeutics 19(3), 1–9 (2017)
22. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
23. Griva, L., Breton, M., Chernavvsky, D., Basualdo, M.: Commissioning procedure for predictive control based on arx models of type 1 diabetes mellitus patients. IFAC-PapersOnLine 50(1), 11023–11028 (2017)
24. van Heusden, K., Dassau, E., Zisser, H.C., Seborg, D.E., III, F.J.D.: Control-relevant models for glucose control using a priori patient characteristics. IEEE Trans. on Biomedical Engg. 59(7), 1839–1849 (2012)
25. Hooman Hakami (Medtronic Inc.): FDA approves MINIMED 670G system - world's first hybrid closed loop system
26. Hovorka, R., Canonico, V., Chassin, L., *others*: Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. Physiological Measurement 25, 905–920 (2004)
27. Hovorka, R., Shojaee-Moradie, F., *others*: Partitioning glucose distribution/transport, disposal and endogenous production during IVGTT. Am. J. Physiol. Endocrinol. Metab. 282, 992–1007 (2002)

28. Hovorka, R.: Continuous glucose monitoring and closed-loop systems. Diabetic Medicine 23(1), 1–12 (2005)
29. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, pp. 97–117. Springer (2017)
30. Koenker, R.: Quantile regression. Econometric Society Monographs (38), 342 (2005)
31. Kowalski, A.: Pathway to artificial pancreas revisited: Moving downstream. Diabetes Care 38, 1036–1043 (June 2015)
32. Kushner, T., Bortz, D., Maahs, D., Sankaranarayanan, S.: A data-driven approach to artificial pancreas verification and synthesis. In: Intl. Conference on Cyber-Physical Systems (ICCPS'18). IEEE Press (2018)
33. Lomuscio, A., Maganti, L.: An approach to reachability analysis for feed-forward relu neural networks. CoRR abs/1706.07351 (2017), http://arxiv.org/abs/1706.07351
34. Maahs, D.M., Others: A randomized trial of a home system to reduce nocturnal hypoglycemia in type 1 diabetes. Diabetes Care 37(7), 1885–1891 (July 2014)
35. Medtronic Inc.: "paradigm" insulin pump with low glucose suspend system (2012), cf. http://www.medtronicdiabetes.ca/en/paradigm_veo_glucose.html
36. Nimri, R., Muller, I., Atlas, E., Miller, S., Kordonouri, O., Bratina, N., Tsioli, C., Stefanija, M., Danne, T., Battelino, T., M, P.: Night glucose control with md-logic artificial pancreas in home setting: a single blind, randomized crossover trial-interim analysis. Pediatric Diabetes 15(2), 91–100 (March 2014)
37. Paoletti, N., Liu, K.S., Smolka, S.A., Lin, S.: Data-driven robust control for type 1 diabetes under meal and exercise uncertainties. In: Computational Methods in Systems Biology (CMSB). LNCS, vol. 10545, pp. 214–232. Springer (2017)
38. Patek, S., Bequette, B., Breton, M., Buckingham, B., Dassau, E., Doyle III, F., Lum, J., Magni, L., Zisser, H.: In silico preclinical trials: methodology and engineering guide to closed-loop control in type 1 diabetes mellitus. J Diabetes Sci Technol. 3(2), 269–82 (2009)
39. Pérez-Gandía, C., Facchinetti, A., Sparacino, G., Cobelli, C., Gómez, E., Rigla, M., de Leiva, A., Hernando, M.: Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring. Diabetes technology & therapeutics 12(1), 81–88 (2010)
40. Piche, S., Sayyar-Rodsari, B., Johnson, D., Gerules, M.: Nonlinear model predictive control using neural networks. IEEE Control Systems 20(3), 53–62 (2000)
41. Psichogios, D.C., Ungar, L.H.: Direct and indirect model based control using artificial neural networks. Industrial & engineering chemistry research 30(12), 2564–2573 (1991)
42. Ruiz, J.L., Sherr, J.L., *others*: Effect of insulin feedback on closed-loop glucose control: a crossover study. J. Diabetes Science and Technology 6(5), 1123–1130 (2012)
43. Steil, G.M., Rebrin, K., Darwin, C., Hariri, F., Saad, M.F.: Feasibility of automating insulin delivery for the treatment of type 1 diabetes. Diabetes 55, 3344–3350 (2006)
44. Teixeira, R.E., Malin, S.: The next generation of artificial pancreas control algorithms. J. Diabetes Sci. and Tech. 2, 105–112 (Jan 2008)
45. Vanderbei, R.J.: Linear Programming: Foundations & Extensions (Second Edition). Springer (2001), cf. http://www.princeton.edu/~rvdb/LPbook/
46. Visentin, R., Dalla Man, C., Cobelli, C.: One-day bayesian cloning of type 1 diabetes subjects: Toward a single-day UVa/Padova type 1 diabetes simulator. IEEE Transactions on Biomedical Engineering 63(11), 2416–2424 (2016)
47. Wang, T., Gao, H., Qiu, J.: A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. IEEE Transactions on Neural Networks and Learning Systems 27(2), 416–425 (2016)
48. Weinzimer, S., Steil, G., Swan, K., Dziura, J., Kurtz, N., Tamborlane, W.: Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas. Diabetes Care 31, 934–939 (2008)