Automatic blood glucose prediction with confidence using recurrent neural networks

John Martinsson[1], **Alexander Schliep**[2], **Björn Eliasson**[3], **Christian Meijner**[1], **Simon Persson**[1], **Olof Mogren**[4]
[1] Chalmers University of Technology
[2] Gothenburg University
[3] Sahlgrenska University Hospital
[4] RISE AI
john.martinsson@gmail.com, alexander@schlieplab.org, bjorn.eliasson@gu.se, olof@mogren.one

Abstract

Low-cost sensors continuously measuring blood glucose levels in intervals of a few minutes and mobile platforms combined with machine-learning (ML) solutions enable personalized precision health and disease management. ML solutions must be adapted to different sensor technologies, analysis tasks and individuals. This raises the issue of scale for creating such adapted ML solutions. We present an approach for predicting blood glucose levels for diabetics up to one hour into the future. The approach is based on recurrent neural networks trained in an end-to-end fashion, requiring nothing but the glucose level history for the patient. The model outputs the prediction along with an estimate of its certainty, helping users to interpret the predicted levels. The approach needs no feature engineering or data pre-processing, and is computationally inexpensive.

1 Introduction

Our future will be recorded and quantified in unprecedented temporal resolution and with a rapidly increasing variety of variables describing activities we engage in as well as physiologically and medically relevant phenomena. One example is the increasingly wide adoption of continuous blood glucose monitoring systems (CGM) which has given type-1 diabetics (T1D) a valuable tool for closely monitoring and reacting to their current blood glucose levels and trends. Blood glucose levels adhere to complex dynamics that depend on many different variables (such as carbohydrate intake, recent insulin injections, physical activity, stress levels, the presence of an infection in the body, sleeping patterns, hormonal patterns, etc) [Bremer and Gough, 1999; Cryer *et al.*, 2003]. This makes predicting the short term blood glucose changes (up to a few hours) a challenging task, and developing machine learning (ML) approaches an obvious approach for improving patient care. Variations in sensor technologies must be reflected in the ML method. However, acquiring domain expertise, understanding sensors, and handcrafting features is expensive and not easy to scale up. Sometimes natural, obviously important and well-studied variables (e.g. caloric intake for diabetics) might be too inconvenient to measure for end-users. On the other hand deep learning approaches are a step towards automated machine learning, as features, classifiers and predictors are simultaneously learned. Thus they present a possibly more scalable solution to the myriad of machine learning problems in precision health management resulting from technology changes alone.

The hypothesis underlying our approach are:

- It is feasible to predict glucose levels from glucose levels alone.
- Appropriate models can be trained by non-experts without feature engineering or complicated training procedures.
- Models can quantify uncertainty in their predictions to alert users to the need for extra caution or additional input.
- Physiologically motivated loss functions improve the quality of predictions.

We trained and evaluated our method on the Ohio T1DM Dataset for Blood Glucose Level Prediction; see [Marling and Bunescu, 2018] for details.

2 Methodology

A recurrent neural network (RNN) is a feed forward artificial neural network that can model a sequence of arbitrary length, using weight sharing between each position in the sequence. In the basic RNN variant, the transition function is a linear transformation of the hidden state and the input, followed by a pointwise nonlinearity:

$$\mathbf{h}_t = \tanh(W\mathbf{x}_t + U\mathbf{h}_{t-1} + \mathbf{b}),$$

where W and U are weight matrices, $\mathbf{b}$ is a bias vector, and tanh is the selected nonlinearity. W, U, and $\mathbf{b}$ are typically trained using some variant of stochastic gradient descent (SGD).

Basic RNNs struggle with learning long dependencies and suffer from the vanishing gradient problem. This makes them difficult to train [Hochreiter, 1998; Bengio *et al.*, 1994], and has motivated the development of the Long Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997], that to

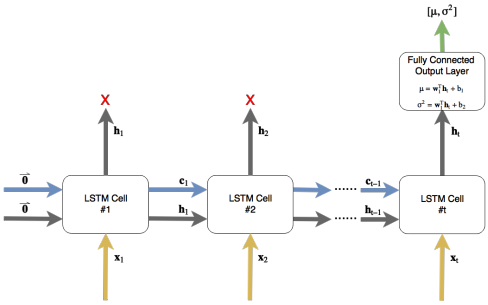

Figure 1: High-level illustration of the LSTM network used in this work. Each cell updates the internal memory vector c_i with information from the current input, and outputs a vector h_i. c_i and h_i is passed on to the next cell, and finally h_t is used as input to a fully connected output layer which applies a linear transformation and outputs the predicted μ, σ^2.

some extent solves these shortcomings. An LSTM is an RNN where the cell at each step t contains an internal memory vector $\mathbf{c}_t$, and three gates controlling what parts of the internal memory will be kept (the forget gate $\mathbf{f}_t$), what parts of the input that will be stored in the internal memory (the input gate $\mathbf{i}_t$), as well as what will be included in the output (the output gate $\mathbf{o}_t$). In essence, this means that the following expressions are evaluated at each step in the sequence, to compute the new internal memory $\mathbf{c}_t$ and the cell output $\mathbf{h}_t$. Here "$\odot$" represents element-wise multiplication.

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(W_i\mathbf{x}_t + U_i\mathbf{h}_{t-1} + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + \mathbf{b}_f),\\
\mathbf{o}_t &= \sigma(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + \mathbf{b}_o),\\
\mathbf{u}_t &= \tanh(W_u\mathbf{x}_t + U_u\mathbf{h}_{t-1} + \mathbf{b}_u),\\
\mathbf{c}_t &= \mathbf{i}_t \odot \mathbf{u}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1},\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t).
\end{aligned}
$$

We model the blood glucose levels using a recurrent neural network (see Fig. 1), working on the sequence of input data provided by the CGM sensor system. The network consists of Long short-term memory (LSTM) cells [Hochreiter and Schmidhuber, 1997]. The whole model takes as input a sequence of blood glucose measurements from the CGM system and outputs one prediction regarding the blood glucose level after time T (we present experimental evaluation for $T \in \{30, 60\}$ minutes). An RNN is designed to take a vector of inputs at each timestep, but in the case of feeding the network with blood glucose measurements only, the input vectors are one-dimensional (effectively scalar valued).

The output vector from the final LSTM cell (see $\mathbf{h}_t$ in Fig. 1) in the sequence is fed through a fully connected output layer having two outputs with a linear activation function,

$$[\mu, \sigma^2] = W_l\mathbf{h}_t + \mathbf{b}_l.$$

The output is modeled as a univariate Gaussian distribution [Graves, 2013], using one value for the mean, μ, and one value for the variance, σ^2. This gives us an estimate of the confidence in the models' predictions.

The negative log-likelihood (NLL) loss function is based on the Gaussian probability density function,

$$\mathcal{L} = \frac{1}{k}\sum_{i=0}^{k} -\log\left(\mathcal{N}(y_i|\mu_i, \sigma_i^2)\right),$$

where y_i is the target value from the data, and μ_i, σ_i are the network's output given the input sequence $\boldsymbol{x}_i$. This way of modeling the prediction facilitates basing decisions on the predictions.

Physiological loss function: We also trained the model with a glucose-specific loss function [Favero *et al.*, 2012], which is a metric that combines the mean squared error with a penalty term for predictions that would lead to clinically dangerous treatments.

2.1 Experimental setup

The only preprocessing done on the glucose values are scaling by 0.01 as in [Mirshekarian *et al.*, 2017] to get the glucose values into a range fit for training.

Hyperparmeter selection was performed by selecting patient 559 and 591 in the Ohio T1DM Dataset for Blood Glucose Level Prediction [Marling and Bunescu, 2018] and train on the first 60% of the training data for each patient, using the next 20% of the data for early stopping, selecting the hyperparameters by the performance on the last 20% of the data. We then proceeded to train five models, with different random initializations, on a set of different configurations using 30, 120 and 240 minutes of history in combination with an LSTM state size of 8, 32, 96 and 128. Each model was allowed a maximum of 200 epochs and early stopping with a patience of 8. The configuration which generalized best for the two patients was using 30 minutes of glucose level history and 128 LSTM states. This can be seen in Fig. 2; note the blue line. Using 30 minutes of history in combination with few LSTM states results in a high RMSE score for both patients, but 30 minutes of history in combination with 128 LSTM states works well both patients. The problem of selecting the proper model and the amount of glucose level history that the model should use to make the future prediction is something that warrants further research, and which should be addressed in future work.

Final models: The final models were trained using 30 minutes of glucose level history for predictions 30 and 60 minutes into the future, respectively. The setup for the final training was to train on the first 80% of the glucose level training data for each patient, and validate on the last 20%. The final models were given a low learning rate of 10^{-5}, a maximum number of $10,000$ epochs, and an early stopping patience of 256 to allow them more time to converge. These final models were then the only models run on the supplied test data. Note that the there are values in the test data for which no predictions have been made.

Missing data: The number of missing predictions depends on the number of gaps in the data, i.e., the number of pairwise consecutive measurements in the glucose level data

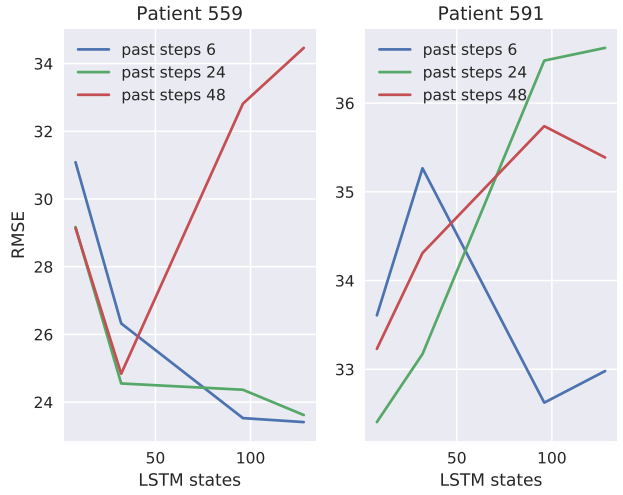

Figure 2: We display the RMSE score on the validation data to select the number of LSTM states and the number of previous time-steps (history) of the blood glucose signal that should be used to predict the future value.

where the time-step is not exactly five minutes. We do not interpolate to fill the missing values since it is unclear how much bias this would introduce, and instead only use data for which it is possible to create the (x, y) pairs of glucose history and regression targets at the given horizon. The greatest number of gaps in the test data is 11 for patient 559. Using 30 minutes of history (6 time-steps) and predicting 30 minutes into the future (6 time-steps) results in $12 * 11 = 132$ values which have no predictions, since we need at least 12 consecutive measurements to create a (x, y) pair. The test portion of the dataset contains 2514, 2570, 2745, 2590, 2791 and 2760 test points, which gives us a upper-bound of roughly 5% of missing predictions for each patient. See the discussion of missing data for further explanation.

Computational requirements: In our experimental setup training of the model could be performed on a commodity laptop. The model is small enough to fit in, and be used on mobile devices (e.g. mobile phones, blood glucose monitoring devices, etc). Training could initially be performed offline and then incremental training would be light enough to allow for training either on the devices or offline.

3 Results

The results presented in Table 1 are the root mean squared error (RMSE) for the model when trained with the mean squared error (MSE) loss function and the negative log-likelihood (NLL) loss function. The results indicate that the model performs comparably when trained with NLL and MSE, but with the added benefit of estimating the variance of the prediction.

The glucose level of patient 591 is harder to predict than the glucose level for patient 570, which can be seen in the Table 1 where the RMSE for patient 570 is 16.3 and the RMSE for

Table 1: We show results individually per patient and averages in predicting glucose levels with a 30 respectively 60 min interval. The table show the root mean squared error (RMSE) of the predictions when the LSTM is trained with the negative log-likelihood (NLL) loss function and the mean squared error (MSE) loss function respectively. t_0 refers to the naive baseline of predicting the last value.

	30 min horizon			**60 min horizon**		
Patient ID	NLL	MSE	t_0	NLL	MSE	t_0
559	19.5	19.5	23.4	34.8	34.4	39.7
570	16.4	16.5	19.0	28.8	28.6	31.9
588	19.3	19.2	21.8	32.5	33.1	35.8
563	19.0	19.0	20.8	30.8	29.9	34.0
575	24.8	24.2	25.6	38.4	37.3	39.7
591	25.4	22.0	24.4	36.0	36.0	38.6
μ	20.7	20.1	22.5	33.6	33.2	36.6
σ	$\pm$3.2	$\pm$2.5	$\pm$2.2	$\pm$3.2	$\pm$3.1	$\pm$3.0

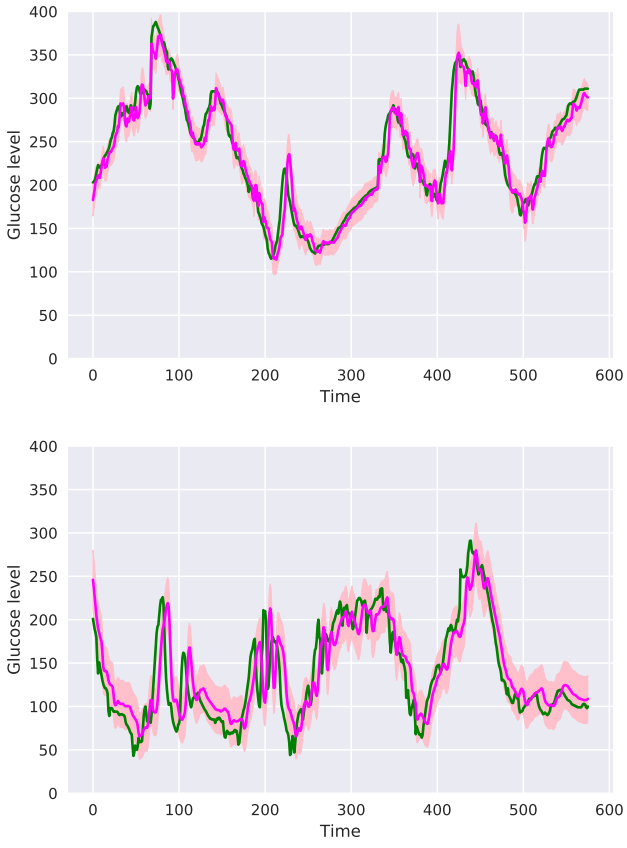

Figure 3: We display the prediction (purple) and standard deviation (shaded pink) compared to the ground truth (green) for patient 570 (top) and 591 (bottom). Note the much larger uncertainty for patient 591.

patient 591 is 24.6. Fig. 3 indicate that the model is able to learn this by assigning a higher variance to the predictions for patient 591 than for patient 570. The standard deviation is illustrated by the pink shaded region in the figure. This is further illustrated in the Clarke error grid plots in Fig. 4 where we can see that for patient 570 most of the predictions are in region A, which is considered as a clinically safe region, but for patient 591 we can see that more predictions are in the B region, which is still considered non-critical, but also in the more critical D region. That is, the variance of the error in the predictions is higher for patient 591 than for patient 570. In particular, the model has a hard time predicting hypoglycemic events.

4 Discussion

As the competition will provide the benchmarking we focus on particular insights we have gained during the development of the method.

Minimalistic ML: Compared to results in the literature for other datasets our system based on recurrent neural networks can predict blood glucose levels in type-1 diabetes for horizons of up to 60 minutes into the future using only blood glucose level as inputs. Generally, the minor improvement over a naive baseline algorithm demonstrate that the prediction problem is a rather difficult one, partly due to large intra and inter patient variation. Nevertheless, our results suggest that a substantially reduced human effort—avoiding labor-intensive prior work by experts hand-crafting features based on extensive domain knowledge —in designing and training machine learning methods for precision health management can be feasible.

Quantifying uncertainty: Our model also outputs an estimate of the variance of the prediction, thus measuring uncertainty in prediction. This is a useful aspect for a system which will be used by continuous glucose monitoring users for making decisions about administration of insulin and/or caloric intake. We expect that large-scale data collection of data from many users will further improve results. The results in Fig. 3 show the two ends of the spectrum in this uncertainty quantification.

One principle problem is that disambiguating between intra-patient variation and sensor errors is unlikely to be feasible. An interesting research question concerns methods which can detect sensor degradation over time or identify defects by comparing sensors for the same patient in long-term physiological; it is unclear if the often smoothed data supplied by sensors is sufficient for that.

Physiological loss function: To our surprise we did not see improvements when using a physiologically motivated loss function [Favero *et al.*, 2012] (results not shown), essentially a smoothed version of the Clarke error grid [Clarke *et al.*, 1987]. Of course our findings are not proof that such loss functions cannot improve results. Possibly a larger-scale investigation, exploring in particular a larger area of the param-

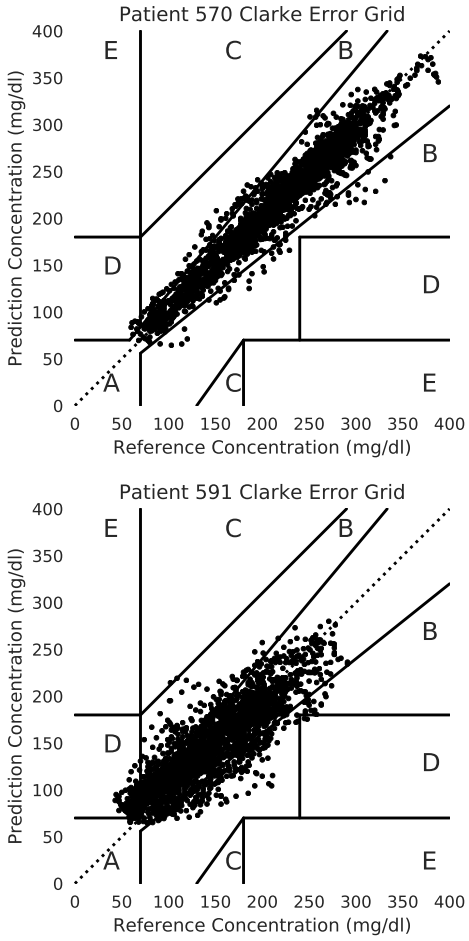

Figure 4: We show the Clarke error grid plots for patient 570 (top) and patient 591 (bottom). Note that the variance of the error in the predictions is higher for patient 591 than for patient 570.

eter space and different training regimes might provide further insights. Penalizing errors for hypo- or hyper-glycemic states should lead to better real-world performance, as we observed comparatively larger deviations in minima and maxima. One explanation for that is the relative class imbalance, as extrema are rare. This could be countered with data augmentation techniques.

Model selection: The large inter-patient variation also suggest that selecting one model for all patients might yield suboptimal results, see Fig. 1. Consequently, precision health apps should not only adapt parameters to individuals, but also entertain increasing or decreasing model complexity. While this is clearly undesirable from a regulatory point-of-view (e.g., how to show efficacy in a trial), the differences we observed seemed to suggest that adaption of complexity improves quality of care.

Missing data: There are gaps in the training data with missing values. Most of the gaps are less than 10 hours, but some of the gaps are more than 24 hours. The number of missing data points account for roughly 23 out of 263 days, or 9% of the data. The gaps could be filled using interpolation, but it is not immediately clear how this would affect either the training of the models, or the evaluation of the models, since this would introduce artificial values. Filling a gap of 24 hours using interpolation would not result in realistic data. Instead we have chosen not to fill the gaps with artifical values and limit our models to be trained and evaluated only on real data. This has its own limitations since we can not predict the initial values after a gap, but the advantage is that model training and evaluation is not biased by the introduction of artificial values.

Conclusion: The field is certainly in desperate need of larger data sets and standards for the evaluation. Crowd sourcing from patient associations would be one possibility, but differences in sensor types and sensor revisions, life styles, and genetic markup are all obvious confounding factors. Understanding sensor errors by measuring glucose level in vivo, for example in diabetes animal models, with several sensors simultaneously would be very insightful, and likely improve prediction quality. Another question concerns preprocessing in the sensors, which might be another confounding factor in the prediction. While protection of proprietary intellectual property is necessary, there has been examples, e.g. DNA microarray technology, where only a completely open analysis process from the initial steps usually performed with vendor's software tools to the final result helped to realize the full potential of the technology.

Software

The software including all scripts to reproduce the computational experiments is released under an open-source license and available from `https://github.com/johnmartinsson/blood-glucose-prediction`. We have used Googles TensorFlow framework, in particular the Keras API of TensorFlow which allows for rapid prototyping of deep learning models, to implement our model and loss functions.

References

[Bengio *et al.*, 1994] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[Bremer and Gough, 1999] Troy Bremer and David A Gough. Is blood glucose predictable from previous values? a solicitation for data. *Diabetes*, 48(3):445–451, 1999.

[Clarke *et al.*, 1987] William L Clarke, Daniel Cox, Linda A Gonder-Frederick, William Carter, and Stephen L Pohl. Evaluating clinical accuracy of systems for self-monitoring of blood glucose. *Diabetes care*, 10(5):622–628, 1987.

[Cryer *et al.*, 2003] Philip E Cryer, Stephen N Davis, and Harry Shamoon. Hypoglycemia in diabetes. *Diabetes care*, 26(6):1902–1912, 2003.

[Favero *et al.*, 2012] Simone Del Favero, Andrea Facchinetti, and Claudio Cobelli. A Glucose-Specific Metric to Assess Predictors and Identify Models. 59(5):1281–1290, 2012.

[Graves, 2013] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hochreiter, 1998] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[Marling and Bunescu, 2018] Cindy Marling and Razvan Bunescu. The ohiot1dm dataset for blood glucose level prediction. *Glucose Prediction News*, 2018.

[Mirshekarian *et al.*, 2017] Sadegh Mirshekarian, Razvan Bunescu, Cindy Marling, and Frank Schwartz. Using LSTMs to learn physiological models of blood glucose behavior. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 2887–2891, 2017.