Neural Multi-class Classification Approach to Blood Glucose Level Forecasting with Prediction Uncertainty Visualisation

Michael Mayo[1] and **Tomas Koutny**[2]

Abstract. A machine learning-based method for blood glucose level prediction thirty and sixty minutes in advance based on highly multiclass classification (as opposed to the more traditional regression approach) is proposed. An advantage of this approach is the possibility of modelling and visualising the uncertainty of a prediction across the entire range of blood glucose levels without parametric assumptions such as normality. To demonstrate the approach, a long-short term memory-based neural network classifier is used in conjunction with a blood glucose-specific data preprocessing technique (risk domain transform) to train a set of models and generate predictions for the 2018 and 2020 Blood Glucose Level Prediction Competition datasets. Numeric accuracy results are reported along with examples of the uncertainty visualisation possible using this technique.

1 INTRODUCTION AND BACKGROUND

Maintaining blood glucose level (BGL) in the normoglycemic range is a significant challenge for patients with type 1 diabetes (T1D). Traditionally, patient BGL self-management is achieved using finger stick blood samples, testing strips and glucose meters (see [13] for an overview), combined with bolus insulin dosing to approximate proper insulin delivery in the body of non-diabetic person. However, with the recent development of continuous glucose monitors (CGMs) and semi- and fully closed-loop artificial pancreas (AP) systems [14], much finer grained control of patient BGL is now possible. Additionally, significantly greater volumes of BGL data is also available when these devices are used. AP technology has been show to improve patient outcomes [5].

In this paper, the problem of forecasting BGL thirty and sixty minutes in advance is considered using the 2020 BGL Prediction Challenge [3] as a testbed. Although several past systems have considered machine learning techniques for BGL forecasting (see [16] for a comprehensive survey), most approaches take a regression approach to solving the problem. In other words, each "forecast" is a numeric point prediction (such as BGL at some point in the future), and overall system accuracy is a measurement of the error between the forecast and the actual future BGL. Accuracy metrics may be statistical (e.g. mean absolute error) or clinical (e.g. Clarke error grid analysis [12]). Regardless, the focus is usually on point predictions.

Here, an alternative approach is taken: instead of treating BGL forecasting as a regression problem, it is instead viewed as a classification problem. This is achieved by dividing the range of possible BGL values into 100 bins equally spaced in the risk domain [9]. Each bin is mapped to a class, and therefore a given forecast is generated by predicting the probability of each class and computing the expected value across all of the classes. An advantage of this approach is that the probabilities associated with the forecast can be visualised across the BGL range. This could be useful for patients, since it enables the patient to take the reliability of the forecast into consideration when making a decision. Additionally, the probability distribution can be used to estimate the chance of significant events such as hypoglycemic episode. Although a similar idea was explored in the context of regression recently [11], the underlying assumption there was that the uncertainty distribution was Gaussian, whereas in the classification approach presented here, no assumptions need be made about the distribution.

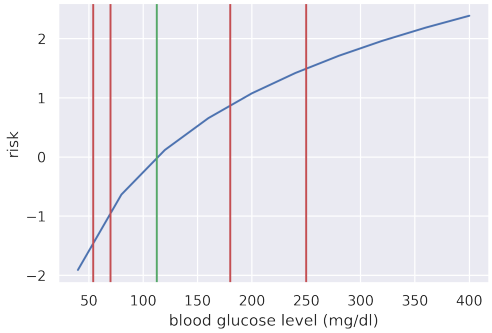

Figure 1. Risk function. The vertical green line at 112.5 mg/dl is the point of least risk while vertical red lines represent (from left to right) the thresholds [6] for level 2 and level 1 hypoglycemia, normoglycemia, and level 1 and level 2 hyperglycemia respectively.

In order to transform the problem of BGL forecasting into a classification problem, a method of breaking the BGL range into sensible classes is required. This is not trivial because the range of BGL values is continuous, and the sizes of clinically-relevant subranges varies non-linearly. For example, a small change in the hypoglycemic part of the BGL range may be highly significant clinically but an equivalent change in the hyperglycemic part of the range may be considered insignificant.

One option is to use the five ranges proposed by Danne et al. [6] as the classes. These ranges are: levels I and II hypoglycemic, normoglycemia, and levels I and II hyperglycemia. In this case, the size and split points of each range are defined. However, this would amount

[1] School of Computing and Mathematical Sciences, University of Waikato, New Zealand, email: michael.mayo@waikato.ac.nz
[2] NTIS – New Technologies for the Information Society, University of West Bohemia, Czech Republic, email: txkoutny@kiv.zcu.cz

to only five classes, and predictions in such a case may lack accuracy within a class. Another option is to arbitrarily divide the BGL range into a much larger number of bins (e.g. 100) which both increases the number of classes considerably (making the machine learning more challenging) but simultaneously increases the granularity of the predictions so that better probability distributions can be produced. In this paper, the latter approach is taken, however this in turn leads to the necessity to decide how the bins should be defined/split across the range of BGL values.

Because of the inherent non-linearity of the BGL range, an approach called the risk domain transformation, first proposed by Kovatchev et al. [9], is utilised. The idea is to define a non-linear transformation function (and by implication, its inverse) that shifts a CGM sensor reading from the blood glucose domain to a new "risk" domain that is better suited for subsequent analysis. This transformation function is illustrated by Figure 1. Also shown by the figure are the breakpoints for the five ranges defined by Danne et al. [6].

As can be observed in the figure, risk domain values typically spans a range from approximately -2 to just over 2, and most normoglycemic readings lie more or less in the range range $[-0.9, 0.9]$. A risk value of 0.0 corresponds to the BGL of 112.5 mg/dl, which is considered the point of least risk. An advantage of the risk domain is that the hypo- and hyperglycemic ranges now have equal size and significance, which reduces the chance of bias in statistical analysis (e.g. due to larger absolute error sizes in the hyperglycemic range).

$$r(x_t) = y_t = 1.509 \left(\texttt{log}(x_t)^{1.084} - 5.381 \right) \tag{1}$$

$$r^{-1}(y_t) = x_t = \texttt{exp} \left(\left(\frac{y_t}{1.509} + 5.381 \right)^{\frac{1}{1.084}} \right) \tag{2}$$

The exact definitions of the risk domain transformation and its inverse are given in Equations 1 and 2 where x_t is a CGM reading at time t and y_t is its corresponding risk value.

2 METHOD

For dividing the BGL range into classes, the following exact procedure is used. The risk domain range $[-2, 2]$ is considered and 100 bin midpoints are placed on it. The bin midpoints are denoted $y_1^*, y_3^*, y_3^* \ldots y_{100}^*$ with $y_1^* = -2$ and $y_{100}^* = 2$. The remaining bin midpoints are equally spaced along the risk range between y_1^* and y_{100}^*. This ensures that for the smaller hypoglycemic range, the bin sizes will be scaled properly in size and that there will be a proportionate number of bins (and therefore classes) in each subrange of the BGL scale.

Next, in order to a assign a new CGM reading x (in mg/dl) to a bin, its corresponding risk value $y = r(x)$ is computed using Equation 1. The reading is then assigned the bin with the closest midpoint. This means that the split points between bins do not need to be calculated explicitly, and if a reading is outside the risk range (either $x < -2$ or $x > 2$) then it will be assigned to one of the bins at the ends of range. However, this tends not happen often since the significant majority of readings in the competition datasets lie on the $[-2, 2]$ range.

Finally, once its class is determined, the reading x is transformed into a one-hot encoded vector $(0, 0 \ldots 0, 1, 0 \ldots 0, 0)$ of length 100 where the single 1 in the vector corresponds to the bin that x is assigned to.

To summarise the process, the predictors X for the model comprise a time series of risk-transformed CGM readings, and the target Y is a one-hot encoded class vector of dimension 100. Predictions are therefore numeric vectors, e.g. $(0, 0 \ldots 0.25, 0.6, 0.3 \ldots 0, 0)$. Note that whichever type of model is used, the values should be positive and sum to unity so that they can be interpreted as normal probabilities.

To evaluate this idea, 100-class classification experiments using a neural network as a predictive model were performed. Figure 2 depicts the particular neural network architecture used here.

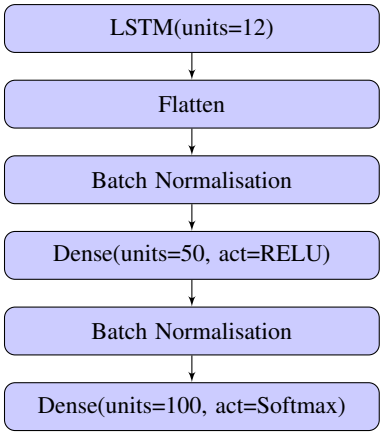

Figure 2. Architecture of the neural network used in the experiments.

The inputs to the neural network are twelve risk value readings, representing the past sixty minutes of BGL variation as sensed by the CGM (assuming that readings occur every five minutes). Since this is a time series, a long-short term memory (LTSM) layer with twelve units is used for initial input processing. Next, the LSTM output is flattened and passed through two dense fully connected layers for further processing. Both of these layers apply batch normalisation first, which ensures faster convergence times and stability during training. Finally, the last dense layer predicts the class and applies a softmax activation function to ensure that the output is a probability vector.

To train each instance (one per patient) of this neural network, the ADAM optimiser [7] was used with a learning rate of 0.0001, 10k epochs, batch size of 32, and validation data set to a random 15% subset of training data. The loss function utilised was categorical cross-entropy, which is commonly used for multiclass classification problems. Early stopping during training was permitted if no improvement in loss was observed for 100 epochs. All other settings were identical to those used in `keras` v. 2.3.1 [4] with `tensorflow` v. 2.1.0 [2] as a base neural network system. The neural network parameters and architecture decisions were made as a result of single-run experiments using data from the first patient in the 2018 competition dataset [10].

To generate data for training the neural network, a strict approach was taken towards missing data and all examples with gaps or time discrepancies (e.g. readings not exactly five minutes apart) exist were excluded. Therefore, to generate one example for the thirty minute $(t+30)$ forecasting problem, it is required that nine consecutive readings (from the start of the example up to and including the prediction target) exist, and for the sixty minute problem $(t + 60)$, twelve consecutive readings were required. No missing value imputation was performed. As a result, the number of test examples varies slightly depending on the forecasting horizon: i.e. the number of 2020 dataset test examples is 2,743, 2,579, 2,177, 2,185, 2,393 and 2,624 for the

30-minute horizon and 2,689, 2,531, 2,111, 2,113, 2,297 and 2,582 for the 60 minute horizon respectively.

With the neural network architecture and training data construction approach described, the final aspect of methodology to be described is the way that numeric point predictions were generated for the competition purposes (which require point predictions). A simplistic approach is allow only bin midpoints (i.e. $y_1^* \dots y_{100}^*$) to be predictions, and select the bin/class with the highest probability. Initial tests showed that this technique had low accuracy. Instead, a more sophisticated approach is to calculate the expected BGL value, as described by the following equation:

$$\begin{aligned} \hat{x}_{t+n} &= f(m_n, y_{t-55}, y_{t-50} \dots y_{t-5}, y_t) \\ &= \sum_{i=1}^{100} p(y_i^* | m_n, y_{t-55} \dots y_t) r^{-1}(y_i^*) \end{aligned} \tag{3}$$

where $n \in \{30, 60\}$, m_n is the neural network for the current patient with forecasting horizon n, y_t is the risk-transformed BGL level at time t ($y_t = r(x_t)$), $y_1^* \dots y_{100}^*$ are risk bin midpoints, $f()$ represents the application of m_n to the observed CGM values, and $\hat{x}_{t+n}$ is the expected value or prediction at time $t + n$. Since the probabilities across the 100 bins sum to 1, the resulting point prediction will be scaled correctly. Initial experiments showed that this expected value approach produced accurate estimates. Source code is available [1].

3 RESULTS

Two rounds of experiments were performed. In the first round, neural networks models were independently trained and tested for each of the twelve patients from both the 2018 and 2020 competition datasets. There was no sharing of information between models. In the second round, models for the 2020 patients only were trained, and the training data for each patient included all of the training data from the 2018 competition in addition to the specific 2020 patient's training data. An individual patient's own training data was therefore a small subset of his/her full training dataset. To account for this imbalance in the second round, samples from the target patient were re-weighted by a factor of six compared to the sample weights from the other patients.

Results for the first round of experiments are given in Tables 1 and 2. The first table is mean absolute error (MAE) results, and the second table gives root mean squared error (RMSE) results. Units are mg/dl. Using both metrics, predictive performance is comparable between the two datasets, with patient 540 from the 2020 dataset being the most "difficult" patient to predict. Conversely, the patient with lowest forecast error is patient 570 from the 2018 dataset.

Results for the second round of experiments are given in Tables 3 and 4. It can be observed that the additional training data leads to a very slight improvement in accuracy. Performing a paired t-test across the six 2020 patients reveals that the improvement in MAE is significant even albeit small (average improvement for thirty minute forecasting is 0.45 with significance $p = 0.000056$, and for sixty minute forecasting it is 0.82 with significance $p = 0.008096$).

More interesting are the prediction plots that can be generated when making a forecast using the classification approach. Figures 3-6 depict some probability densities produced by the model when making four different predictions. For each figure, the point prediction generated using the expected value computation (Equation 3) is shown as red line.

The tidiest example is Figure 3, which depicts a single-peaked distribution with the forecast coinciding with the peak of the distribu-

Patient ID	MAE $t+30$	std	MAE $t+60$	std
559	14.7	0.2	26.2	0.1
570	12.2	0.4	21.0	0.4
588	14.0	0.1	23.4	0.2
563	13.6	0.1	22.5	0.2
575	15.0	0.1	25.9	0.2
591	16.0	0.1	26.3	0.1
Avg.	14.3		24.2	
540	16.8	0.1	30.8	0.1
544	12.9	0.2	23.3	0.2
552	12.5	0.1	23.0	0.1
567	14.9	0.1	27.9	0.2
584	16.7	0.1	28.0	0.2
596	12.6	0.1	21.7	0.0
Avg.	14.4		25.8	

Table 1. Mean absolute error (MAE, mg/dl) results by patient and prediction horizon, averaged over five runs.

Patient ID	RMSE $t+30$	std	RMSE $t+60$	std
559	21.6	0.2	35.5	0.2
570	17.2	0.5	28.3	0.3
588	19.1	0.1	31.8	0.2
563	20.6	0.3	31.2	0.4
575	23.8	0.4	36.4	0.3
591	21.8	0.2	33.7	0.1
Avg.	20.7		32.8	
540	23.0	0.2	40.6	0.14
544	17.4	0.2	30.5	0.17
552	16.9	0.0	30.2	0.10
567	20.9	0.2	36.9	0.22
584	23.0	0.1	36.6	0.16
596	17.8	0.1	29.5	0.03
Avg.	19.8		34.0	

Table 2. Root mean squared error (RMSE, mg/dl) results by patient and prediction horizon, averaged over five runs.

Patient ID	MAE $t+30$	std	MAE $t+60$	std
540	16.4	0.1	29.8	0.1
544	12.3	0.1	22.9	0.2
552	12.2	0.1	22.2	0.2
567	14.5	0.2	27.4	0.4
584	16.1	0.2	26.4	0.1
596	12.2	0.3	21.3	0.1
Avg.	13.9		25.0	

Table 3. Mean absolute error (MAE, mg/dl) results by patient and prediction horizon, averaged over five runs for the 2020 dataset only. Training data includes the entire 2018 dataset.

Patient ID	RMSE $t+30$	std	RMSE $t+60$	std
540	22.4	0.1	39.5	0.1
544	17.0	0.1	30.1	0.2
552	16.5	0.1	29.3	0.1
567	20.8	0.2	36.9	0.4
584	22.4	0.2	35.9	0.2
596	17.2	0.3	29.0	0.2
Avg.	19.4		33.4	

Table 4. Root mean squared error (RMSE, mg/dl) results by patient and prediction horizon, averaged over five runs for the 2020 dataset only. Training data includes the entire 2018 dataset.

tion. While this class of forecast is common, it is not the only type of distribution that is output from the model.

Figure 4 shows a two-peaked distribution with an expected value between the peaks. While the expected value is a useful point prediction, the dual peaks are also useful information since it can be clearly observed that the two most likely outcomes are normoglycemia vs. a state most likely in level I hyperglycemica, although the model is not certain.

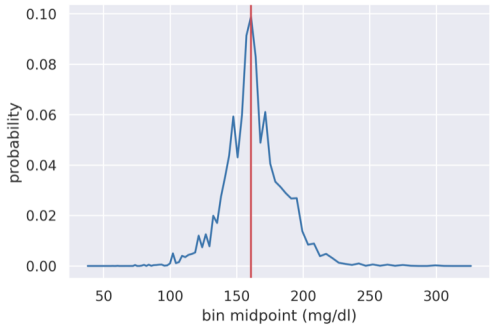

Figure 3. Probability distribution over BGLs for one prediction. In this example, the expected BGL coincides with the distribution peak.

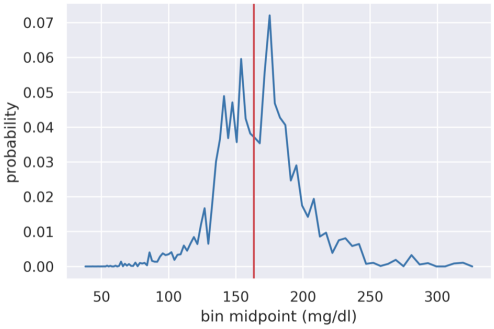

Figure 4. In this probability distribution over BGL levels, two peaks exist in the distribution and the expected BGL lies between the peaks.

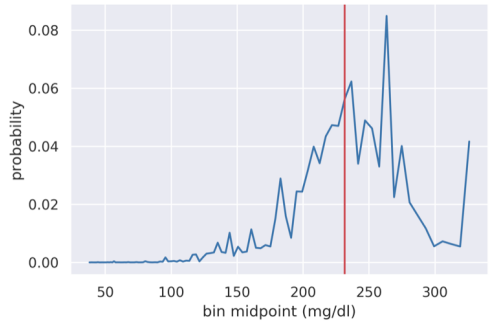

Figure 5. In this example, the distribution is skewed to to the right with the expected BGL quite a distance to the left of the peak.

Figure 5 shows a skewed distribution a significant mass of the probability distribution is at the upper end of the range, but the expected value is closer to the middle of the range (albeit still in the hyperglycemic range). In this case, the expected value underestimates

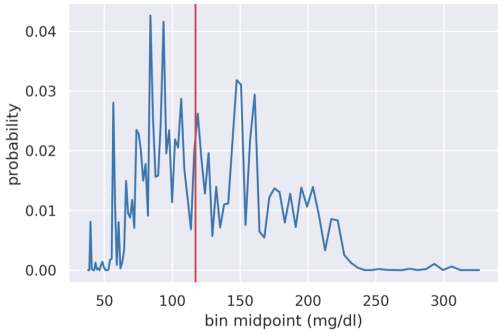

Figure 6. An example of a prediction with considerable uncertainty over the hypoglycemic and normoglycemic ranges.

the true risk to the patient at the current time. Again, this is clearly noticeable.

Finally, Figure 6 depicts a prediction with considerable noise in the distribution. While the forecast is normoglycemic, there is significant mass in the hypoglycemic range. Therefore it could be concluded that although the point prediction is reasonably, there is still significant risk of hypoglycemia.

Further analysis was performed with respect to the variance of the prediction distributions. It was found that for most patients, the distribution of variances is skewed to the left indicating that on average most predictions are more certain (more like Figures 3 and 4) than uncertain. However, more analysis needs to be done on this point.

4 CONCLUSION

This paper describes a system for forecasting BGL at thirty and sixty minutes in advance. This main distinctiveness of this approach is the adoption of a highly multi-class classification-based technique and use of a domain-specific transform for normalising BGL values (opposed to more traditional min/max scaling or standardisation). The ability to visualise non-parametric probability distributions accompanying predictions as a meaningful context is a clear advantage.

To test the proposed method with real patients, we will use a system known as SmartCGMS [8]. SmartCGMS is a continuous glucose monitoring and controlling software framework. It provides infrastructure to connect and develop "building blocks" for an insulin-pump software stack. Principally, the pump developer connects CGM-sensor blocks to computing blocks, which predict BGL and subsequently schedule insulin boluses or adjust the insulin basal rate. Next, another block transforms the results of these computations into insulin-pump control commands. With SmartCGMS, we can close the loop *in-silico*[15] first, before conducting an *in-vivo* experiment to ensure maximum safety.

Our specific approach will be to transform the best trained keras/tensorflow-based neural network into a hard-coded and constant feed-forward neural network in C++. This will enable efficient deployment and computation on low-power devices such as insulin-pump controllers, while we can still train the original neural network using high-performance computers. As a result, a flow in which a neural network is continuously learned from patient BGL measurements, providing personalised BGL predictions, can be established.

Acknowledgment

This publication was partially supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

REFERENCES

[1] https://www.cms.waikato.ac.nz/%7Emmayo/kdhcompsource.py

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Marling C. and Bunescu R. The OhioT1DM dataset for blood glucose level prediction: Update 2020. http://smarthealth.cs.ohio.edu/bglp/OhioT1DM-dataset-paper.pdf, 2020.

[4] François Chollet et al. Keras. https://keras.io, 2015.

[5] Xia Dai, Zu-Chun Luo, Lu Zhai, Wen-Piao Zhao, and Feng Huang, 'Artificial pancreas as an effective and safe alternative in patients with type 1 diabetes mellitus: A systematic review and meta-analysis', *Diabetes therapy : research, treatment and education of diabetes and related disorders*, **9**(3), 1269–1277, (06 2018).

[6] Thomas Danne, Revital Nimri, Tadej Battelino, Richard M. Bergenstal, Kelly L. Close, J. Hans DeVries, Satish Garg, Lutz Heinemann, Irl Hirsch, Stephanie A. Amiel, Roy Beck, Emanuele Bosi, Bruce Buckingham, Claudio Cobelli, Eyal Dassau, Francis J. Doyle, Simon Heller, Roman Hovorka, Weiping Jia, Tim Jones, Olga Kordonouri, Boris Kovatchev, Aaron Kowalski, Lori Laffel, David Maahs, Helen R. Murphy, Kirsten Nørgaard, Christopher G. Parkin, Eric Renard, Banshi Saboo, Mauro Scharf, William V. Tamborlane, Stuart A. Weinzimer, and Moshe Phillip, 'International consensus on use of continuous glucose monitoring', *Diabetes Care*, **40**(12), 1631–1640, (2017).

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[8] Tomas Koutny and Martin Ubl, 'Parallel software architecture for the next generation of glucose monitoring', *Procedia Computer Science*, **141**, 279–286, (2018).

[9] Boris P. Kovatchev, Martin Straume, Daniel J. Cox, and Leon S. Farhy, 'Risk analysis of blood glucose data: A quantitative approach to optimizing the control of insulin dependent diabetes', *Journal of Theoretical Medicine*, **3**(1), (2000).

[10] Cindy Marling and Razvan Bunescu, 'The OhioT1DM dataset for blood glucose level prediction', in *Proceedings of the 3rd International Workshop on Knowledge Discovery in Healthcare Data*, 60–63, CEUR Workshop Proceedings (CEUR-WS.org), (2018).

[11] John Martinsson, Alexander Schliep, Bjorn Eliasson, Christian Meijner, Simon Persson, and Olof Mogren, 'Automatic blood glucose prediction with confidence using recurrent neural networks', in *Proceedings of the 3rd International Workshop on Knowledge Discovery in Healthcare Data*, 64–68, CEUR Workshop Proceedings (CEUR-WS.org), (2018).

[12] Himel Mondal and Shaikat Mondal, 'Clarke error grid analysis on graph paper and microsoft excel', *Journal of Diabetes Science and Technology*, **14**(2), 499–499, (2020). PMID: 31777281.

[13] Leann Olansky and Laurence Kennedy, 'Finger-stick glucose monitoring', *Diabetes Care*, **33**(4), 948–949, (2010).

[14] Dawei Shi, Sunil Deshpande, Eyal Dassau, and Francis J. Doyle, 'Chapter 1 - feedback control algorithms for automated glucose management in t1dm: the state of the art', in *The Artificial Pancreas*, eds., Ricardo S. Sánchez-Peña and Daniel R. Cherñavvsky, 1 – 27, Academic Press, (2019).

[15] Martin Ubl and Tomas Koutny, 'SmartCGMS as an environment for an insulin-pump development with FDA-accepted in-silico pre-clinical trials', *Procedia Computer Science*, **160**, 322–329, (2019).

[16] Ashenafi Zebene Woldaregay, Eirik Årsand, Ståle Walderhaug, David Albers, Lena Mamykina, Taxiarchis Botsis, and Gunnar Hartvigsen, 'Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes', *Artificial Intelligence in Medicine*, **98**, 109 – 134, (2019).