Blood Glucose Prediction in Type 1 Diabetes Using Deep Learning on the Edge

Taiyu Zhu[†], Lei Kuang[†], Kezhi Li[†⋆], Junming Zeng[†], Pau Herrero[†] and Pantelis Georgiou[†]
† Centre for Bio-inspired Technology, Imperial College London, London, United Kingdom
Email: {taiyu.zhu17, lei.kuang18, junming.zeng16, pherrero, pantelis}@imperial.ac.uk
⋆ Institute of Health Informatics, University College London, London, United Kingdom
Email: ken.li@ucl.ac.uk

Abstract—**Real-time blood glucose (BG) prediction can enhance decision support systems for insulin dosing such as bolus calculators and closed-loop systems for insulin delivery. Deep learning has been proven to achieve state-of-the-art performance in BG prediction. However, it is usually seen as a very computationally expensive approach, hence difficult to implement in wearable medical devices such as transmitters in continuous glucose monitoring (CGM) systems. In this work, we introduce a novel deep learning framework to predict BG levels with the edge inference on a microcontroller unit embedded in a low-power system. By using glucose measurements from a CGM sensor and a recurrent neural network that builds on long-short term memory, the personalized models achieves state-of-the-art performance on a clinical data set obtained from 12 subjects with T1D. In particular, the proposed method achieves an average root mean square error of 19.10 ± 2.04 for a 30-minute prediction horizon (PH) and 32.61 ± 3.45 for a 60-minute PH with high clinical accuracy. Notably, the framework has been optimized to achieve a minimum use of hardware resources (34KB FLASH and 1KB SRAM) as well as an execution time of 22 ms for low power operations (8 µW). The presented system has the potential to be implemented in wearable medical devices for diabetes care (CGM and insulin pumps) and to be integrated within an Internet of Things platform.**

Index Terms—**Diabetes, deep learning, artificial pancreas, Internet of things, microcontroller, edge inference, LSTM.**

I. Introduction

Diabetes is a global disease with an estimated prevalence of over 400 million worldwide [1]. Due to the absolute deficiency of endogenous insulin secretion, people living with type 1 diabetes (T1D) require long-term glucose management by delivering exogenous insulin and monitoring blood glucose (BG) levels. The aim of this therapy is to maintain BG levels in a therapeutic range and minimize the risk of potentially life-threaten events (e.g. hypo- and hyperglycemia). Accurate BG prediction is highly desirable, since it enables proactive actions and early interventions to mitigate potential adverse glucose events, such as hyperglycemia and hypoglycemia. However, due to high inter-subject variability, sensor errors. and various external factors that affect glucose dynamics (e.g., meal ingestion, exercise), the challenge of personalized and accurate BG prediction still remains.

In the era of embedded systems and Internet of things (IoT), the rapid development of wearable medical devices (e.g. glucose sensors and insulin pumps) and closed-loop systems for automatic insulin delivery, i.e., artificial pancreas (AP), has significantly improved the treatment of people with T1D [2]. An AP system consists of, at least, a continuous glucose monitoring (CGM) sensor, an insulin pump, and a control algorithm that modulates insulin delivery [2]. CGM measures BG levels in real-time at a fixed frequency (e.g. 5 minutes) and generates a large amount of rich clinical data. Leveraging on these data, various data-driven algorithms for BG prediction have been proposed in the literature, and feed-forward neural networks are the most used approaches, as indicated by a recent review [3]. Moreover, deep learning algorithms have become new paradigms in BG prediction [4] and achieved state-of-the-art performance in the Blood Glucose Level Prediction Challenge in 2018 and 2020 [5]–[8]. Deep neural networks (DNNs), i.e. neural networks with a stack of non-linear hidden layers, are powerful tools for representation learning and feature extraction [9].

II. Related Work

The architectures based on recurrent neural networks (RNNs) have been widely used to model sequential CGM data, especially with the recurrent cells of long short-term memory (LSTM) [10]–[12]. Aiming at improving the accuracy of BG prediction, numerous efforts continue to push the boundaries of deep learning models using the latest techniques, such as dilation [13], attention mechanism [14], and residual connections [8]. It is noted that, although the models become increasingly complex, there is a lack of research on the implementation of these models to bring actual therapeutic benefits to people with T1D. Pioneering works have deployed deep learning algorithms on smartphones to provide real-time glucose predictions for AP systems [13], [15]. However, owing to its power and data transmission capabilities, smartphones cannot be considered as the edge of inference in deep learning. In addition, the stringent data privacy and security requirements in the medical device industry make smartphones not the ideal platforms [16]. Therefore, packing the front-end sensors together with the back-end deep learning algorithms [17] delivers an intuitive instruction on the edge inference with microcontroller units (MCUs), which stands out as a more resource-efficient solution within an IoT network. Fig. 1 depicts the overall system and methodology.

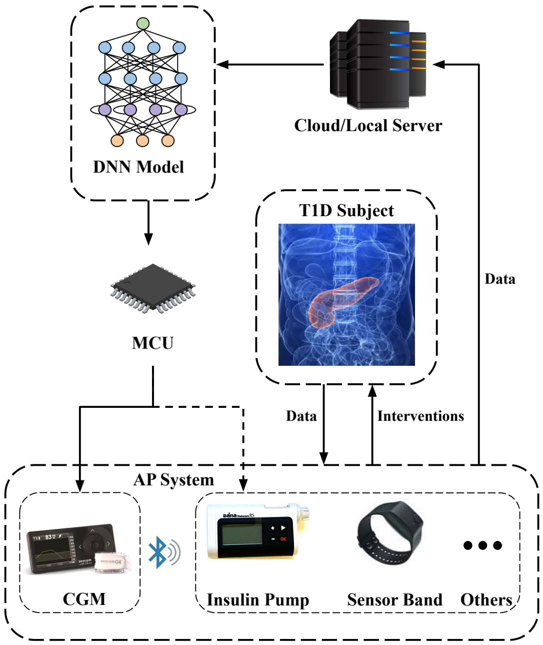


Fig. 1: System architecture of the proposed framework for BG prediction by means of edge deep learning. A T1D subject uses CGM, insulin pumps, wrist bands and other wearable devices to collect real-time measurement data, which are upload to servers for backup and model training. Then the well-trained DNNs are embedded on an MCU, and the MCU can be further implemented in the wearable devices to assist decision support.

Besides, being able to fetch data in real-time without wireless transmission provides another main advantage, since it allows faster decision support and more reliable communications. In this work, we propose a new BG prediction framework based on personalized LSTM models and a 32-bit MCU.

III. Methods

A. BG Prediction by Deep Learning

In CGM, BG levels are constantly sampled with a resolution r, (where $r = 5$ minutes in this work) forming a time series $\mathbf{G}$. Given a prediction horizon (PH) p, the aim at time step t is to predict the future BG levels $G_{t+\tau}$, where $\tau = p/r$. After pre-processing, the scaled sensor measurements form a time series as the input data $\mathbf{X}$. Here, we use a univariate input to perform local inference without data fusion, which only contains the scaled BG values with a normalization function f, i.e., $\mathbf{X} \triangleq f(\mathbf{G})$. A sliding window is employed to slice the input data with sequence length w for input batches $\mathbf{X}_{t-w:t} = [\mathbf{x}_{t-w+1}, \dots, \mathbf{x}_t]$. To mitigate the bias and trend, we used the signed BG change as the target labels, which is defined as:

$$y_t = f(G_{t+\tau}) - f(G_t). \tag{1}$$

Using the previous output as an input at the current time step, RNNs can model the temporal dynamics. However, it is difficult for vanilla RNNs to learn long-term dependencies, due to gradient vanishing. LSTM solves this problem by introducing gated operations, i.e, forget gate ($\mathbf{f}_t$), input gate ($\mathbf{i}_t$) and output gate ($\mathbf{o}_t$), to control the information flow and enable the memory of cell states ($\mathbf{C}_t$) [18]. It is noted that the number of parameters and operations in an LSTM-based RNN is much larger than that of a vanilla RNN or a dense layer, which rapidly increases with the dimension of input features and hidden units [5]. Considering the limited resource on MCUs, we instantiate a single light-weight LSTM layer with 32 hidden units to extract temporal dependencies of the input BG sequences. Then the output of the final state is processed by two dense layers with 64 and 32 hidden units and ReLU activation [19], which is denoted as:

$$\mathbf{h}_t^d = \text{ReLU}(\mathbf{W}_d * \mathbf{h}_t + \mathbf{b}_d), \tag{2}$$

where $\mathbf{h}_t$ is the hidden state of the LSTM, and $\mathbf{h}_t^d$ is the output of the dense layer. Finally, the model output $\hat{y}$ is computed with the linear transformation

$$\hat{y}_t = \mathbf{W}_l * \mathbf{h}_t^d + b_l, \tag{3}$$

The BG prediction $\hat{G}_{t+\tau}$ is obtained by the inverse transform of label encoder and normalization, which is written as

$$\hat{G}_{t+\tau} = G_t + f^{-1}(\hat{y}_t). \tag{4}$$

B. Model Training and Validation

The weights and bias in DNN models are trainable parameters, which are iteratively updated by the back-propagation during training. To this end, the mean square error (MSE) is employed as the loss function, as a reliable estimator that measures average errors in regression tasks. The mini-batches of input data with the batch size of 64 are fed into the model to compute the MSE between a group of $\hat{y}_t$ and y_t. Adam optimizer [20] is applied with a learning rate of 0.001. Considering the temporal dependencies of the CGM sequences and the limited size of clinical datasets, we split the last 20% training data as the validation set. We set the total epochs to 500 and introduce the early stopping technique with the patience of 50. This setting allows for the automatic termination of experiments when validation MSE is not improved for 50 consecutive epochs, in order not to over-fit the training data. All the hyper-parameters are determined by the MSE scores of the grid search during validation.

The model is constructed with the framework of TensorFlow 2.0.0 with Keras 2.2.4 and Python 3.7.7. The training is accelerated by a GPU (NVIDIA GTX 1080 Ti).

C. Embedded System Implementation

With the models trained and validated, we then load the personalized models onto embedded systems for validation. As we aim to deliver a low power solution, the model size has been constrained to less than 50 KB while the dimension of the hidden units on each layer is chosen to be minimum, without loosing much accuracy. Regarding data transmission, we simulated the front-end sensor on the PC with the general serial port data transmission protocol (UART) and provide the latest clinical data from our local/cloud servers.

In this work, considering the requirements and specifications of healthcare platforms, we have implemented the models on a well established development board (Nucleo-F303RE) based on a mainstream MCU with float-point unit (STM32F303RE, an ARM Cortex-M4 processor running at 72MHz clock frequency, integrated with 80KB SRAM and 512KB Flash) [21]. The amount of resources in STM32 can be further chosen for resource-optimization, however, our main focus in this work is the validation of the proposed approach. An expansion package of STM32CubeMX, namely X-CUBE-AI, is utilized to deploy the floating-point (32-bit) and fixed-point (8-bit) deep learning models onto the target MCU [22]. With the STM32 optimizer and C-code generator, it is capable to convert the pre-trained Keras h5 models into executable STM32 C models. Specifically, we use the STM32CubeIDE 1.3.0 and X-CUBE-AI 5.1.2 to construct the local network, validate the converted model, and run real-time experiments.

Table I summarizes the technical details of the layer implementations on the MCU, including the output shape, the number of multiply-and-accumulate (MACC) operations, the percentage of FLASH storage, the number of layer parameters, and the duration time, which are provided by X-CUBE-AI. In order to imitate real-life situation, only one testing input is sent at one timestamp, i.e., the batch size equals to one.

TABLE I: Implementation of the network layers on the MCU.

Layer	**Shape**	**MACC**	**Flash (%)**	**Param**	**Time (ms)**
LSTM	(1, 32)	52,608	51.3	4,352	21.359
Dense	(1, 64)	2,048	24.4	2,112	0.410
ReLU	(1, 64)	64	0	0	0.027
Dense	(1, 32)	2,048	24.0	2,080	0.390
ReLU	(1, 32)	32	0	0	0.017
Dense	(1, 1)	32	0.4	33	0.016

It is noted that the LSTM layer contributes to the largest portion of the FLASH memory and computational time. Thus, for optimized power/resource efficiency, it is recommended not to stack too many LSTM layers as it is likely to result in occupying a significant portion of memory as well as introducing a considerable delay comparing to the rest. In this implementation, 34.69KB out of 512KB is used. On the other hand, STM32 AI engine has a built-in function which enables minimum usage of working memory (SRAM). The principle of optimising SRAM relies on reusing temporal memory chunks. Therefore, operations of different layers under one signal pipeline can claim the same area within the SRAM, where the maximum size of the activation buffer is bound to the maximum memory requirements of two consecutive layers which could be trade-off with its computation accuracy if required. In this case, 1KB out of 80KB of SRAM is used.

IV. Experiments and Results

A. Experimental Configuration

1) Clinical Dataset: To allow reproducibility, we use the OhioT1DM dataset to develop the proposed model, which is publicly available in the research community [23]. It contains the data from 12 anonymous subjects with T1D over an eight-week clinical trial. The subjects were equipped with various wearables, including Medtronic Enlite CGM sensors that measure BG levels every 5 minutes. The training (~ 40 days) and testing sets (~ 10 days) are provided separately. There are some missing gaps in the CGM data, due to many reasons, such as sensor calibration and signal loss. To avoid current predictions using future information, we perform extrapolation to estimate the missing values.

2) Evaluation Metrics: As a regression problem, the most common statistical metrics in BG prediction are the root mean square error (RMSE) and mean absolute error (MAE). Besides, to analyze the outcomes with the clinical significance, we also use glucose-specified RMSE (gRMSE) [24] based on the Clark error grid (CEG) [25]. This metric penalizes the predictions that would lead to clinical issues, such as underestimations in hyperglycemia and overestimations in hypoglycemia.

B. Prediction Results

In the experiments, we validated the proposed model with 30-minute and 60-minute prediction horizons (PHs). Three traditional BG prediction algorithms based on machine learning are employed as baseline methods, including support vector regression (SVR) [26], random forests (RF) regression [27], and the artificial neural network (ANN) [28]. For the ANN model, it contains two dense layers with 64 and 32 hidden units and an output layer. All the baseline methods were tested in the local server.

Table II and III shows the prediction performance of the considered methods for 30-minute and 60-minute PHs, respectively. The results are presented as mean values and standard deviation. A a paired t-test is used to compute the P values when comparing the baseline methods with respect to the edge-LSTM, where the statistical significance is indicated as † for $P < 0.005$ and * for $P < 0.05$. Notably, compared to the RF, SVR, and ANN baseline methods, the edge-LSTM model achieved the best prediction accuracy with a significant improvement for both the two PHs, in terms of RMSE, MAE, and gRMSE. In particular, it obtained the mean RMSE of 19.10 with the MAE of 13.59 and the gRMSE of 22.08 for the 30-minute PH and the RMSE of 32.61 with the MAE of 24.25 and the gRMSE of 38.04 for the 60-minute PH.

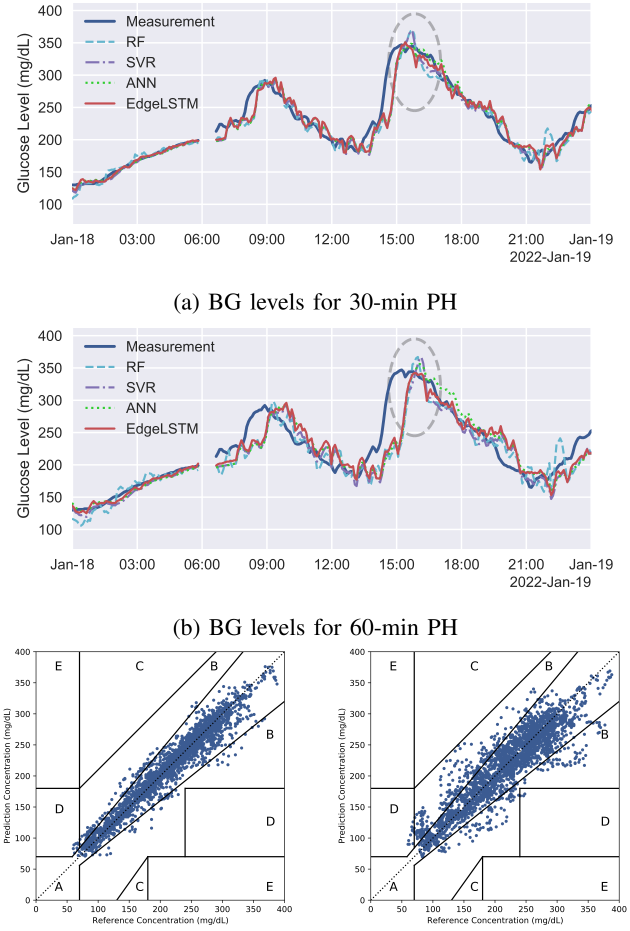
TABLE II: Prediction performance of the considered methods with 30-minute PH on 12 T1D subjects.

Method	**RMSE (mg/dL)**	**MAE (mg/dL)**	**gRMSE (mg/dL)**
RF	$21.23 \pm 2.17^\dagger$	$15.35 \pm 1.56^\dagger$	$24.64 \pm 2.73^\dagger$
SVR	$19.99 \pm 2.21^\dagger$	$14.11 \pm 1.46^\dagger$	$22.86 \pm 2.67^\dagger$
ANN	$19.81 \pm 2.13^\dagger$	$14.06 \pm 1.43^\dagger$	$22.99 \pm 2.63^\dagger$
Edge-LSTM	$\mathbf{19.10 \pm 2.04}$	$\mathbf{13.59 \pm 1.47}$	$\mathbf{22.08 \pm 2.41}$

Fig. 2a and 2b show the prediction curves together with the actual CGM measurements. Note that the dates in the dataset have been shifted for a random number of months to protect the privacy of the data contributors. The peaks on the plots indicate the increase of BG levels caused by meal intake, where minor delays can be observed between the prediction

TABLE III: Prediction performance of the considered methods with 60-minute PH on 12 T1D subjects.

Method	RMSE (mg/dL)	MAE (mg/dL)	gRMSE (mg/dL)
RF	$35.36 \pm 3.63^{\dagger}$	$26.47 \pm 2.92^{\dagger}$	$41.45 \pm 4.71^{\dagger}$
SVR	$33.75 \pm 3.52^{\dagger}$	$24.75 \pm 2.78^{\dagger}$	$38.56 \pm 4.38^{*}$
ANN	$33.58 \pm 3.61^{\dagger}$	$25.07 \pm 2.90^{\dagger}$	$39.31 \pm 4.53^{\dagger}$
Edge-LSTM	$\mathbf{32.61 \pm 3.45}$	$\mathbf{24.25 \pm 2.84}$	$\mathbf{38.04 \pm 4.17}$



(a) BG levels for 30-min PH

(b) BG levels for 60-min PH

(c) CEG for 30-min PH

(d) CEG for 60-min PH

Fig. 2: One-day prediction performance of the considered methods and the CEG plots of edge-LSTM for subject 570.

algorithms and the actual BG levels. The dashed ellipse in each subplot highlights the postprandial hyperglycemia region at 16:00, where the edge-LSTM model achieved the smallest overestimation, compared to the baseline methods. To illustrate the clinical performance, we employ the CEG plot (Fig. 2c and 2d), where each data point represents a pair of prediction and ground truth. For the chosen subject (#570), we observe that most of the points concentrate on A and B zones (99.67% for 30-minute PH and 99.49% for 60-minute PH). The treatment decisions based on the predictions in these zones are clinically acceptable for people with T1D, which indicates high clinical accuracy of the edge-LSTM model.

Table IV shows the overall inference performance on the embedded system. It is worth noting that the optimized edge-LSTM model consumed only 34.69 KB FLASH and 1.00 KB SRAM, accounting for 6.78% and 1.19% of the total memory of the MCU, respectively. It implies that the proposed framework can use lower-end MCUs with smaller volume or DNN models with higher complexity. The tiny RMSE, MAE, and relative error in L_2 norm demonstrate that differences between the inference on the edge and the local server are well below consideration. Note that the power is measured by averaging based on CGM resolution, i.e, the model would only be executed when there is a new CGM measurement data and stays asleep otherwise. An execution time of 22.219 ms as well as the 8 μW consumed power proves that the edge-LSTM model is feasible for real-time inference on the wearables with small-scale batteries.

TABLE IV: Performance of the edge inference.

Parameter	Value	Inference Error	Result
FLASH	34.69 KB	RMSE	0.0029
SRAM	1.00 KB	MAE	0.0025
Execution Time	22.219 ms	L_2 Relative Error	1.46×10^{-6}
CPU Cycles	1.60×10^{7}		
MACC	56,832		
Cycles/MACC	28.2		
Average Power	8 μW		

V. Discussion and Conclusion

To the best of our knowledge, this is the first work implementing an edge deep learning algorithm on an MCU for BG prediction. We have validated the outcomes of the edge-LSTM model by means of standard performance metrics (RMSE and MAE) and clinical metrics (gRMSE and CEG). Empowered by the LSTM in the embedded system to extract the temporal dependencies of BG trajectories, the proposed model outperforms the selected machine learning baseline algorithms in terms of all the metrics and shows good BG prediction performance. Future work includes integrating with other useful modules, such as WiFi and Bluetooth, into the embedded system, designing a printed circuit board for prototyping, and evaluating the system in an actual clinical setting. The proposed framework is agnostic to the types of neural network employed and learning targets. Therefore, it can be applied to realize a variety of tasks on the wearables, such as the detection of events (e.g. meals, exercise, illness, faults) and glucose regulation via deep reinforcement learning [29], [30]. We envision that the edge intelligence on IoT platforms will have profound implications for the next generation of decision support systems and AP systems and will significantly relieve the daily burden of the self-management for people with T1D.

VI. Acknowledgement

This research has been funded by Engineering and Physical Sciences Research Council (EPSRC EP/P00993X/1) and the President's PhD Scholarship at Imperial College London (UK).

References

[1] P. Saeedi, I. Petersohn, P. Salpea, B. Malanda, S. Karuranga, N. Unwin, S. Colagiuri, L. Guariguata, A. A. Motala, K. Ogurtsova *et al.*, "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas," *Diabetes research and clinical practice*, vol. 157, p. 107843, 2019.

[2] P. Herrero, M. El-Sharkawy, J. Daniels, N. Jugnee, C. N. Uduku, M. Reddy, N. Oliver, and P. Georgiou, "The bio-inspired artificial pancreas for type 1 diabetes control in the home: system architecture and preliminary results," *Journal of diabetes science and technology*, vol. 13, no. 6, pp. 1017–1025, 2019.

[3] A. Z. Woldaregay, E. Årsand, S. Walderhaug, D. Albers, L. Mamykina, T. Botsis, and G. Hartvigsen, "Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes," *Artificial intelligence in medicine*, vol. 98, pp. 109–134, 2019.

[4] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Deep Learning for Diabetes: A Systematic Review," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2020.

[5] T. Zhu, K. Li, P. Herrero, J. Chen, and P. Georgiou, "A deep learning algorithm for personalized blood glucose prediction," in *The 3rd International Workshop on Knowledge Discovery in Healthcare Data, IJCAI-ECAI*, 2018, pp. 64–78.

[6] J. Chen, K. Li, P. Herrero, T. Zhu, and P. Georgiou, "Dilated recurrent neural network for short-time prediction of glucose concentration." in *The 3rd KDH workshop, IJCAI-ECAI 2018*, 2018, pp. 69–73.

[7] T. Zhu, X. Yao, K. Li, P. Herrero, and P. Georgiou, "Blood glucose prediction for type 1 diabetes using generative adversarial networks," in *The 5th KDH workshop, ECAI 2020*, 2020, pp. 90–94.

[8] H. Rubin-Falcone, I. Fox, and J. Wiens, "Deep residual time-series forecasting: Application to blood glucose prediction," in *The 5th KDH workshop, ECAI 2020*, 2020, pp. 105–109.

[9] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[10] T. Zhu, K. Li, J. Chen, P. Herrero, and P. Georgiou, "Dilated recurrent neural networks for glucose forecasting in type 1 diabetes," *Journal of Healthcare Informatics Research*, pp. 1–17, 2020.

[11] Q. Sun, M. V. Jankovic, L. Bally, and S. G. Mougiakakou, "Predicting blood glucose with an lstm and bi-lstm based deep neural network," in *2018 14th Symposium on Neural Networks and Applications (NEUREL)*. IEEE, 2018, pp. 1–5.

[12] A. Aliberti, I. Pupillo, S. Terna, E. Macii, S. Di Cataldo, E. Patti, and A. Acquaviva, "A multi-patient data-driven approach to blood glucose prediction," *IEEE Access*, vol. 7, pp. 69 311–69 325, 2019.

[13] K. Li, C. Liu, T. Zhu, P. Herrero, and P. Georgiou, "GluNet: A deep learning framework for accurate glucose forecasting," *IEEE journal of biomedical and health informatics*, 2019.

[14] S. Mirshekarian, H. Shen, R. Bunescu, and C. Marling, "LSTMs and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 706–712.

[15] M. He, W. Gu, Y. Kong, L. Zhang, C. J. Spanos, and K. M. Mosalam, "Causalbg: Causal recurrent neural network for the blood glucose inference with IoT platform," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 598–610, 2019.

[16] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, pp. 216–222, 2018.

[17] L. Lai and N. Suda, "Enabling deep learning at the LoT edge," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–6.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML'10*. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." *International Conference on Learning Representations 2015*, pp. 1–15, 2015.

[21] "Datasheet of TM32F303xD STM32F303xE," [Online]. Available:https://www.st.com/resource/en/datasheet/stm32f303re.pdf. Accessed: September 3, 2020.

[22] "X-Cube-AI: AI expansion pack for STM32CubeMX, STMicroelectronics," [Online]. Available:https://www.st.com/en/embedded-software/x-cube-ai.html. Accessed: September 3, 2020.

[23] C. Marling and R. Bunescu, "The OhioT1DM dataset for blood glucose level prediction: Update 2020," in *The 5th KDH workshop, ECAI 2020*, 2020, pp. 71–74.

[24] S. Del Favero, A. Facchinetti, and C. Cobelli, "A glucose-specific metric to assess predictors and identify models," *IEEE transactions on biomedical engineering*, vol. 59, no. 5, pp. 1281–1290, 2012.

[25] W. L. Clarke, D. Cox, L. A. Gonder-Frederick, W. Carter, and S. L. Pohl, "Evaluating clinical accuracy of systems for self-monitoring of blood glucose," *Diabetes care*, vol. 10, no. 5, pp. 622–628, 1987.

[26] E. I. Georga, V. C. Protopappas, D. Polyzos, and D. I. Fotiadis, "A predictive model of subcutaneous glucose concentration in type 1 diabetes based on random forests," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 2889–2892.

[27] E. I. Georga, V. C. Protopappas, D. Ardigò, M. Marina, I. Zavaroni, D. Polyzos, and D. I. Fotiadis, "Multivariate prediction of subcutaneous glucose concentration in type 1 diabetes patients based on support vector regression," *IEEE journal of biomedical and health informatics*, vol. 17, no. 1, pp. 71–81, 2012.

[28] C. Pérez-Gandía, A. Facchinetti, G. Sparacino, C. Cobelli, E. Gómez, M. Rigla, A. de Leiva, and M. Hernando, "Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring," *Diabetes technology & therapeutics*, vol. 12, no. 1, pp. 81–88, 2010.

[29] T. Zhu, K. Li, L. Kuang, P. Herrero, and P. Georgiou, "An insulin bolus advisor for type 1 diabetes using deep reinforcement learning," *Sensors*, vol. 20, no. 18, p. 5058, 2020.

[30] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2020.