JSCDM

JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 7 No. 2 (2026) 163-183
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Edge-Based Deep Learning for Population-Specific Blood Glucose Level Prediction in Type 1 Diabetes Mellitus

Ade Anggian Hakim[1], Farhanahani Mahmud[1,2]*, Kazuki Nakajima[3], Marlia Morsin[1,2], Chin Fhong Soon[1,2]

[1] *Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA*
[2] *Microelectronics and Nanotechnology Shamsuddin Research Centre (MiNT-SRC), Institute of Integrated Engineering (I2E), Universiti Tun Hussein Onn Malaysia (UTHM), 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA*
[3] *Division of Bio-information Eng., University of Toyama, Toyama, JAPAN*

*Corresponding Author: farhanah@uthm.edu.my
DOI: https://doi.org/10.30880/jscdm.2026.07.02.011

Article Info

Received: 7 February 2026
Accepted: 7 June 2026
Available online: 30 June 2026

Keywords

Blood glucose level prediction, Type 1 Diabetes Mellitus, deep learning, edge computing, NVIDIA Jetson Orin Nano

Abstract

This study evaluates a population-based deep learning (DL) approach to predict blood glucose levels (BGL) in patients with Type 1 Diabetes Mellitus (T1DM) using edge computing. Eight DL architectures, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Neural Hierarchical Interpolation for Time Series Forecasting (NHITS), Kolmogorov-Arnold Network (KAN), Temporal Convolutional Network (TCN), Bidirectional Temporal Convolutional Network (BiTCN), Temporal Fusion Transformer (TFT), and Deep Non-Parametric Time Series Model (DeepNPTS), were optimized using Optuna Auto-Tuning and tested on the ShanghaiT1DM dataset, which consists of 16 subjects with a continuous glucose monitoring (CGM) sampling interval of 15 minutes. Two input configurations (univariate and multivariate) were assessed at prediction horizons (PHs) of 30 and 60 minutes through cloud-based simulations in Google Colab and deployment on an NVIDIA Jetson Orin Nano edge device. Performance is evaluated using MAE, MAPE, and RMSE for predictive performance, along with inference time, latency, throughput, and power consumption for deployment efficiency. In both PHs, KAN and TFT achieve superior predictive performance compared to the LSTM baseline and other models, with minimal differences observed between univariate and multivariate configurations. Crucially, KAN uniquely achieves the highest throughput (4.05 preds./s) at the lowest power consumption (7.64 W), making it the first architecture jointly characterized for population-based BGL prediction and uncompressed edge deployment, with improved predictive performance on edge over cloud at the 60-minute horizon, unseen in other architectures. These characteristics position KAN as a strong candidate for efficient on-device inference. Furthermore, with consistent power consumption of approximately 8 W across all eight models, the edge device implementation yielded predictive performance comparable to cloud-based simulations. These

findings confirm that population-specific, edge-based BGL prediction is a feasible and practical solution for contemporary T1DM management.

1. Introduction

According to the International Diabetes Federation, the global number of patients with diabetes has exceeded half a billion [1]. Type 1 Diabetes Mellitus (T1DM), which accounts for about 10% of all cases, is a chronic autoimmune disease characterized by damage to pancreatic β cells, leading to insulin deficiency and necessitating strict management and monitoring [2],[3]. Insulin serves a crucial function in maintaining blood glucose levels (BGLs) within the normal range of 70–180 mg/dL. BGL instability can reduce the quality of life of patients and increase the risk of serious complications, such as cardiovascular disease, kidney damage, neuropathy, and visual impairment [2]. For T1DM patients, the most important preventive management is to avoid hypoglycemia (BGL < 70 mg/dL) and hyperglycemia (BGL > 180 mg/dL) with careful management of meal timing, and they need lifelong insulin therapy through repeated injections or an insulin pump [3].

The introduction of continuous glucose monitoring (CGM) technology has changed the management of diabetes by providing continuous BGL recordings, usually at 5- or 15-minute intervals. This technology not only allows patient monitoring but also the implementation of automated insulin delivery systems [4],[5]. This has led to significant advances in predicting BGL using time-series CGM data. Accurate BGL prediction is clinically important as it enables proactive decision-making regarding insulin dosing and dietary intake, thus reducing the risk of prolonged hypoglycemia and hyperglycemia [6],[7].

BGL prediction models can be broadly classified into two categories: physiology-based models and data-driven models [8]. Physiology-based models are based on metabolic mechanisms and are implemented in complex mathematical calculations, providing reasonable interpretability. However, the mathematical complexity of these models creates challenges for scalability and deployment. In contrast, data-driven models do not require detailed physiological parameters and can extract patterns directly from CGM data, possibly integrating exogenous variables such as insulin dosage and meal intake. Deep learning (DL) architectures, which leverage their multi-layered structure to process massive datasets with high fidelity [9], have been driving rapid progress in artificial intelligence (AI).

Common sequence-based DL models, such as Recurrent Neural Networks (RNNs) [11], Long Short-Term Memory (LSTM) [12], and Gated Recurrent Unit (GRU) [13], may represent complex non-linear patterns and long-range temporal dependencies [14],[15]. On these foundations, transformer-based models are more advanced sequence models. The Temporal Fusion Transformer (TFT), designed specifically for time series forecasting, retains LSTM units for local sequence processing and combines them with multi-head attention to capture long-range dependencies [16]. Later, convolutional architectures were introduced, e.g. Temporal Convolutional Networks [17] and the Bidirectional Temporal Convolutional Network (BiTCN) [18]. State-of-the-art models based on Multilayer Perceptron (MLP) are Neural Hierarchical Interpolation for Time Series Forecasting (NHITS) [19], Deep Non-Parametric Time Series Model (DeepNPTS) [20], and the Kolmogorov-Arnold Network (KAN) [21], which is an alternative to the traditional MLP [22], where activation functions are learned on edges (as splines) rather than fixed at nodes [23]. By generating transparent symbolic expressions for clinical decision-making, KAN can achieve higher predictive performance, interpretability, and robustness at the expense of increased computational cost [24]. It has strong potential for time-series prediction tasks. However, all DL-based models encounter issues at longer prediction horizons (PHs), which is a key limitation for practical BGL prediction [25],[26].

In addition to architecture selection, the model training strategy is also an important consideration. Previous studies have focused on individual-specific approaches, training the models with CGM data from a single patient only [27],[28]. While this yields personalized predictions, the approach is less efficient because it requires retraining for each new patient. On the other hand, a population-based approach is a promising solution for scalable model optimization. Joint training of models on data from multiple patients leads to a joint representation that provides generalization across patients, while being flexible enough to capture patient-specific patterns. This approach is more efficient and scalable, and is superior in generating a single, robust and generalizable model, particularly when per-patient data are sparse, but suffers from an inherent bias towards the population mean [10].

From an implementation point of view, most of the research is still based on cloud-based computing or mobile devices [29], [30]. However, this method is highly dependent on a stable internet connection at all times, is susceptible to system updates that can degrade performance, is power-hungry, and poses data privacy risks [31], [32]. The introduction of edge computing is a more secure and efficient solution, as processing can be done locally on the device itself, reducing latency and improving the security of sensitive data [32],[33]. Edge computing enables reliable real-time glucose forecasting without internet connectivity and is critical for timely intervention, improved quality of life, and clinical decision support in T1DM patients [34]. Edge computing offers several advantages over the cloud for diabetes management, including faster data transmission, reduced bandwidth

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 163-183

dependency, enhanced privacy and security, greater control over data, and lower operational costs [35]. Local inference on edge devices also conserves battery power and memory and enhances data privacy, particularly in environments with poor or no network connectivity[36]. Here, the cutting-edge computer platform is the NVIDIA Jetson Orin Nano, which provides robust GPU computing, low power consumption, small size and an ecosystem that is well-suited to deploying complex DL models [37],[38].

Based on these observations, this study focuses on evaluating time-series DL-based models for BGL prediction in T1DM using edge computing, accounting for several modeling strategies. This has been achieved by firstly optimizing the time-series DL models for BGL prediction, utilizing the latest ShanghaiT1DM dataset with characteristics relevant to T1DM[39],[40]. Optimization was performed using a population-specific approach that included the following steps: data preprocessing, selecting DL model architectures, using an auto-tuning function to obtain optimal hyperparameter values, and conducting simulation tests to evaluate the models' predictive performance. The resulting models were then implemented on an edge device to evaluate their predictive performance and deployment efficiency. The distinctive contribution of this work is threefold. First, to the best of the authors' knowledge, this is the first study to evaluate KAN for population-based BGL prediction and explicitly characterize its edge-deployment behavior, a dimension absent from prior KAN applications in healthcare, which have been confined to cloud or simulation settings [51]. Second, unlike existing edge-based glucose prediction studies that rely on model compression or reduced-capacity architectures to fit resource-constrained hardware [10], this study demonstrates that full, uncompressed DL models, including the computationally intensive KAN, can be executed directly on the NVIDIA Jetson Orin Nano while maintaining competitive predictive performance and favorable power efficiency. Third, this study reveals a practically significant and previously unreported insight: KAN exhibits a reversal of the typical cloud-to-edge performance trend at the 60-minute prediction horizon, achieving lower prediction error in edge deployment than in cloud simulation, which suggests that KAN's spline-based activation mechanism may be particularly well-suited to the stable, GPU-optimized runtime of edge inference environments. This research is expected to contribute to T1DM management applications and bridge the gap between advances in DL models and the needs of practical implementation in modern diabetes management.

2. Methodology

The workflow in Figure 1 illustrates the research methodology for evaluating eight time-series DL-based models for BGL prediction in T1DM with edge computing. The first step of the research is to preprocess CGM, which is composed of several series of historical BGLs (mg/dL) with their timestamps, insulin and meal intake data for the 16 subjects from the ShanghaiT1DM dataset, and to convert it to a population-based dataset. This includes dealing with missing values in the CGM, building a population-based dataset with an 80:20 split between training and test data, and linear interpolation with clipping to deal with outliers. Preceding the primary DL modeling phase, the hyperparameter optimization procedure was carried out in order to identify the best configurations using the Optuna Auto-Tuning framework from NeuralForecast for automatic hyperparameter tuning, which was a crucial component of the training procedure[41]. Time-based cross-validation for cloud-based simulation on Google Colab, followed by a regular time-based train-test split for deployment on the NVIDIA Jetson Orin Nano edge device, are the two approaches used later to evaluate the DL models. The two approaches differ not only in their computational environments but also in their training configurations, as described later in this section, and this distinction is the primary explanation for the systematic differences in reported error scores between simulation and deployment conditions in Section 4. The evaluation was conducted in univariate and multivariate input configurations across both the 30-minute and 60-minute PH conditions.

In this study, the models' performance evaluation encompasses predictive performance of error metrics (MAE, MAPE, and RMSE) in simulation on Google Colab, deployment efficiency of latency, throughput, inference time, and power consumption, and predictive performance using these error metrics on the edge device. The edge-based deployment was carried out on the NVIDIA Jetson Orin Nano with GPU acceleration (8.0 GB RAM) and Python 3.8 running in the Visual Studio Code (VS Code) environment, version 1.105.1 and the cloud-based simulation was carried out on Google Colab using Python 3.8 with GPU acceleration enabled (15.0 GB RAM). NeuralForecast version 2.0.1 and PyTorch Lightning version 2.4.0 are the software frameworks used, and these versions are consistently employed across both cloud and edge environments to ensure operational consistency.

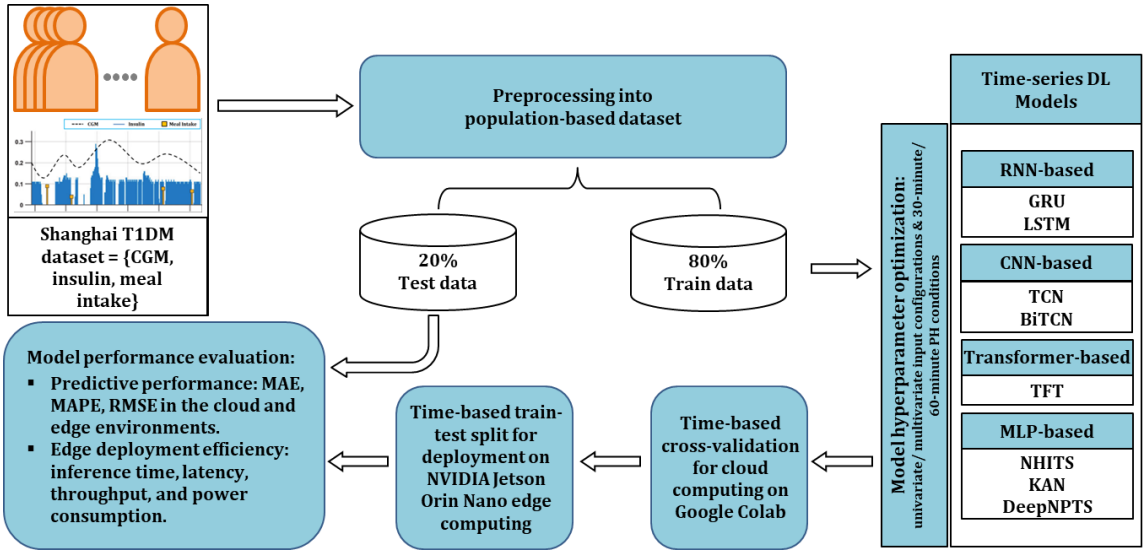


Fig. 1 *A workflow of the research methodology for time-series DL-based BGL predictions in T1DM with edge computing*

2.1 Dataset Description and Preprocessing

The dataset utilized is the ShanghaiT1DM dataset, consisting of time-series CGM recordings with 15-minute sampling intervals, capillary blood glucose (CBG) readings, detailed insulin dosage, dietary intake, physical activity, clinical demographics and extensive laboratory blood markers. The data set included 16 patients with T1DM. This dataset was chosen as it is recent, of high quality, low noise and complete for other variables such as food intake and insulin input [39]. The cohort size of 16 subjects is relatively small, but is consistent with the size of benchmark datasets that are widely quoted in the BGL prediction literature. For example, the OhioT1DM dataset, comprising 12 subjects, has served as a common evaluation benchmark in numerous DL-based glucose forecasting studies [15],[47]. The high temporal resolution of CGM recording in the ShanghaiT1DM dataset (15-min intervals over long inpatient and outpatient monitoring periods) also supports the suitability of the dataset for a population-based modeling approach, by providing a significantly larger number of time-series observations per subject compared to datasets with less frequent sampling. The population-based training strategy, which merges data from all 16 subjects into a shared model, further reduces the statistical risk of small cohort sizes by exploiting the intra- and inter-subject variability available during training.

Before the model hyperparameters optimization process, a preprocessing phase was performed. The process included imputing missing values in the CGM data, making sure timestamps were continuous at 15-minute intervals, and dealing with outlier BGL values. The preferred approach was to use linear interpolation and clipping value within the functional range of the glucose sensor (40–400 mg/dL) [10]. We performed linear interpolation because the CGM signals are a physiologically continuous process (blood glucose changes gradually between measurements) and linear transition between adjacent valid measurements preserves the temporal dynamics of the signal better than a step-wise or a global-mean substitution. Extremes that were physiologically implausible due to sensor saturation, calibration artifacts, or transcription errors rather than true glycemic events were clipped to the blood glucose sensor operational range (40–400 mg/dL) [27]. Retaining such values would inflate error metrics and bias model training toward rare, unrealistic extremes. Importantly, the clipping boundary was defined in the functional range of the sensor, not in the clinical normoglycemic range (70–180 mg/dL), so as to preserve the events of true hypoglycemic and hyperglycemic events, which are clinically critical events for the model to learn to predict. This preprocessing step was necessitated by the impact of sensor calibration errors and the presence of signal artifacts within the dataset.

Another special technique used in the model optimization effort was the application of binary conversion to the insulin and food intake variables, where 0 means no event and 1 means an event. It deliberately chose to use binary encodings for insulin and meal intake rather than raw continuous dosage or raw caloric values. As seen from the raw insulin and meal magnitudes of the ShanghaiT1DM dataset, there is high inter-subject variability and inconsistent logging precision which introduces noise in the feature space. Binary encoding reduces this noise by distilling the clinically most actionable information, whether an intervention event occurred at a given timestamp, into a form that is consistent and comparable across subjects and avoids the risk of the model over-weighting dosage values that may not generalize across patients. Apart from the dataset considerations, binary encoding has the pragmatic advantage of improving the predictive performance of models in real-world edge deployment [43]. It significantly reduces the input requirements of the multivariate configuration, as tracking the

occurrence of an insulin or meal event is much less demanding than detailed recording of dosage amounts and caloric values in a continuous monitoring context. This reduction in complexity reduces the load on data acquisition pipelines and makes the multivariate model more feasible to be implemented on resource-constrained devices. However, the trade-off is that by limiting the input to event indicators rather than event magnitudes, the multivariate configuration has less physiological specificity than a fully detailed input would provide, which in turn creates the possibility that predictive performance does not improve significantly over the univariate configuration, as observed in this study. This outcome should therefore be understood in the context of a deliberate design choice that prioritizes deployment practicality and does not reflect a general limitation of multivariate modeling for BGL prediction. Finally, to train the model collectively, all data were aggregated into a population-based dataset, enabling a dual-configuration analysis with univariate and multivariate inputs. The univariate input vector consists of a two-dimensional feature space, while the multivariate input vector consists of a four-dimensional feature space, as shown in Equations (1) and (2), respectively.

$$\text{Univariate input} = \{\text{BGL, timestamp}\} \tag{1}$$

$$\text{Multivariate input} = \{\text{BGL, insulin, meal, timestamp}\} \tag{2}$$

2.2 DL Models

The eight DL models for time series forecasting from various architectural families were evaluated and optimized to maximize predictive performance in BGL prediction. They are the LSTM baseline, GRU, TCN, BiTCN, TFT, NHITS, DeepNPTS, and KAN models. LSTM [12] and GRU [13] are members of the recurrent family. Dilated convolutions are used by both the TCN [17] and its bidirectional variant, BiTCN [18], to effectively capture long-range temporal dependencies. Meanwhile, the transformer-based TFT [16] uses self-attention mechanisms to improve interpretability and capture global dependencies. Lastly, the new MLP-based models include the hierarchical feed-forward model, NHITS [19], the probabilistic neural forecasting framework, DeepNPTS [20], and the innovative architecture of the MLP, KAN [21] which is based on the principles of functional decomposition. When collectively analyzed, these models provide a comprehensive foundation for enhancing the predictive performance, efficiency, and scalability of BGL prediction.

2.3 Hyperparameter Optimization

In the model hyperparameter optimization, the Optuna Auto-Tuning function in the NeuralForecast library [44] was used, integrated with the time-based cross-validation scheme. The hyperparameters tested included combinations of input size, learning rate, hidden size, batch size, and other model-specific hyperparameters over specific ranges of values [42],[45]. Each hyperparameter configuration was tested using a sliding-window approach with a step size of 1 to cover all temporal intervals at each sampling time. The tuning process was repeated three times for each model, yielding three distinct hyperparameter values. The optimal hyperparameter configuration was determined based on the lowest MAE value [42]. The result of this stage was the selection of the optimal hyperparameters with the lowest MAE, which were then used during the primary training phase.

2.4 Time-based Cross-Validation

The model was trained in Google Colab using 80% of the training data, with 25% reserved for the validation set. Cross-validation applied a combination of hyperparameter values, and tuning was performed using various combinations of relevant hyperparameter values. This process yielded optimal hyperparameter values and used them to generate the optimal model, allowing it to be directly applied to the 20% test data. This approach yielded faster and more robust performance estimates, but required higher computational requirements, making it more suitable for cloud-based scenarios [10].

2.5 Time-based Train-Test Split

The models at this stage were trained in Google Colab using 80% of the training data without a validation set allocation (train-test split), and the trained models were saved for deployment later. By removing the validation set allocation, the model in this configuration gains access to a larger effective training window, the full 80% of data, compared to the cross-validation scheme, where 25% of training data is withheld for validation at each fold. This expanded training exposure is the primary mechanistic reason why the deployment results in Section 4 frequently report lower prediction errors than their simulation counterparts, particularly at the 60-minute PH, where longer historical context is most beneficial. This is not an artifact of the edge hardware or deployment pipeline, but a direct consequence of the difference in training data utilization between the two schemes. Model training at this stage used the best hyperparameter values from the cross-validation stage, allowing the trained

model to be evaluated on 20% of the test data separately during deployment on the NVIDIA Jetson Orin Nano using a rolling-forecasting approach that mimics real-time prediction. An identical 20% test subset was utilized across both the cross-validation and train-test split to maintain comparability. This approach focuses on generating computationally light predictions from trained models, specifically those ready for edge computing deployment [37],[46].

2.6 Performance Evaluation

The predictive performance of each approach in each model was evaluated using 20% of the test data from each subject in the dataset. This evaluation involves three statistical error metrics, namely MAE, MAPE, and RMSE [47], which are defined in Equations (3), (4), and (5), respectively.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|a_i - b_i| \tag{3}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|a_i - b_i|}{a_i} * 100\% \tag{4}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(a_i - b_i)^2}{n}} \tag{5}$$

In these equations, n denotes the number of data points, a_i represents the actual value at the i-th observation, and b_i represents the predicted value at the same observation.

The analysis was conducted at two levels: (1) Analysis of the eight time-series DL models' performance for the BGL prediction that have gone through the optimization process, carried out by comparing the predictive performance results between simulations in Google Colab and edge computing on NVIDIA Jetson Orin Nano, and (2) Specific analysis on evaluating deployment efficiency on the NVIDIA Jetson Orin Nano edge device, which includes total inference time, latency, throughput, and power consumption[37]. In this analysis, the study assessed the models' BGL predictive performance and deployment efficiency in edge computing. For the statistical comparison of deployment efficiency metrics, normality was first assessed using the Anderson-Darling test (α = 0.001) with 128 total samples across all models and measurements treated as independent by design; each subject's inference session was conducted separately with no shared state. Based on normality results, normally distributed metrics were compared using one-way ANOVA, and non-normally distributed metrics were compared using the Kruskal-Wallis H-test, both at α = 0.05.

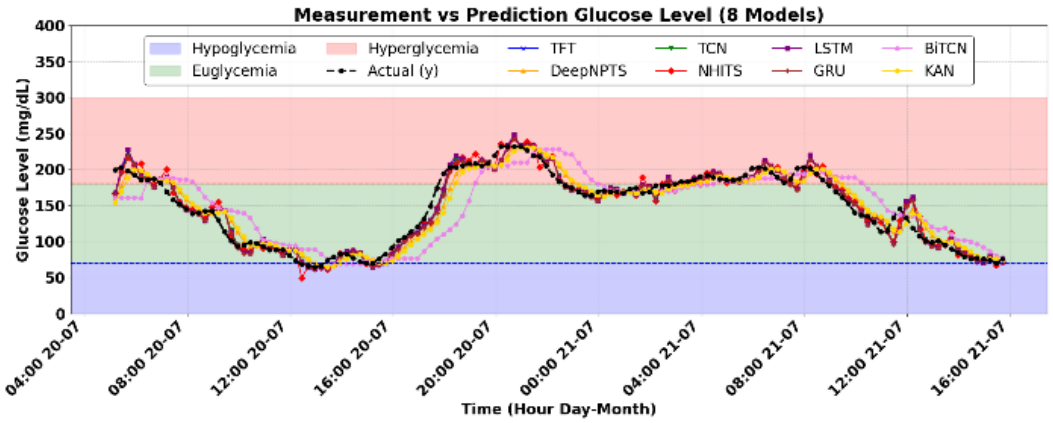
3. Results

The following outlines the findings for eight optimized DL models predicting BGL at 30- and 60-minute PHs, including analyses of their predictive performance in both edge compute and cloud-based simulation settings and of deployment efficiency in the edge compute setting.

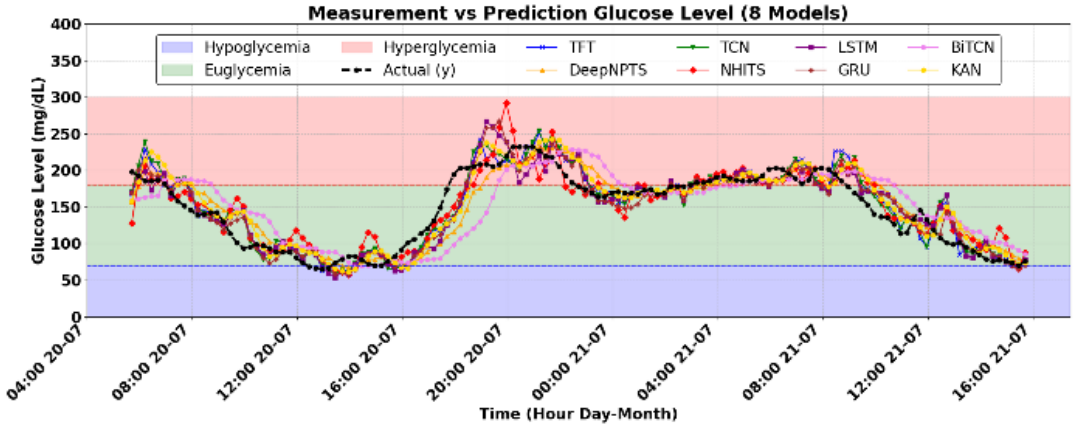
3.1 Predictive Performance

Figure 2 (simulation) and Figure 3 (deployment) provide a visual comparison of the prediction results from 8 models (univariate and multivariate) at 30- and 60-minute PHs against the actual CGM data. For these results, the test data of one subject (Unique_id 12A) in the ShanghaiT1DM cohort are used. The prediction visually was a better fit for the 30 min PH than the 60 min PH. The graphs from univariate models were smoother and more stable, while some of the multivariate configurations presented more fluctuations. The edge deployment showed smaller deviations from the actual data than the cloud simulations, for a comparison of the environments.

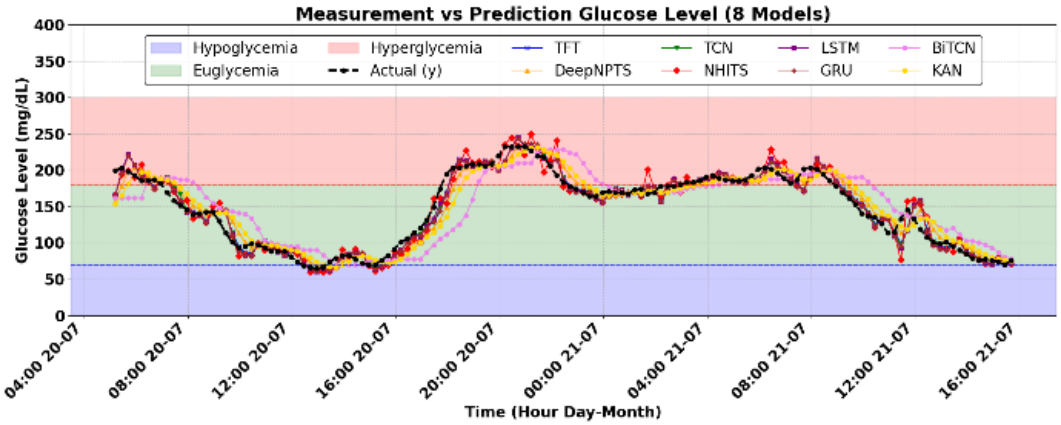
J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 163-183



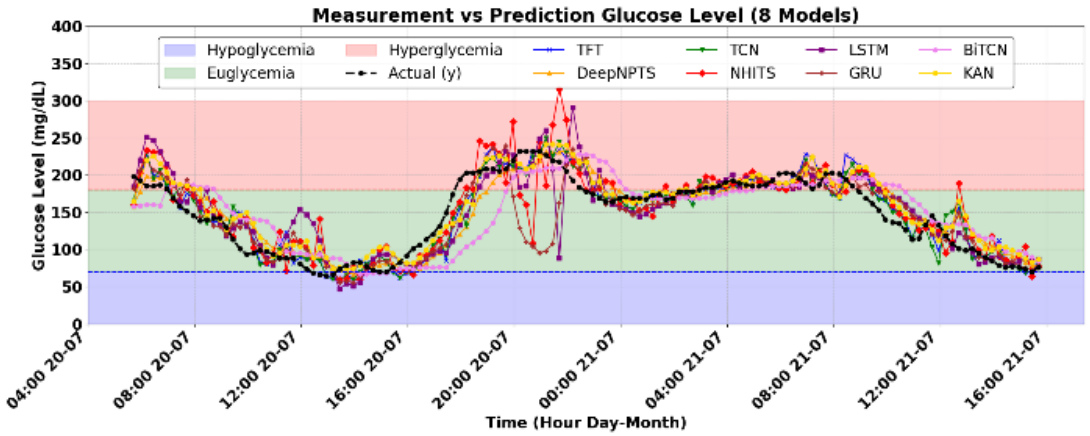
(a)



(b)

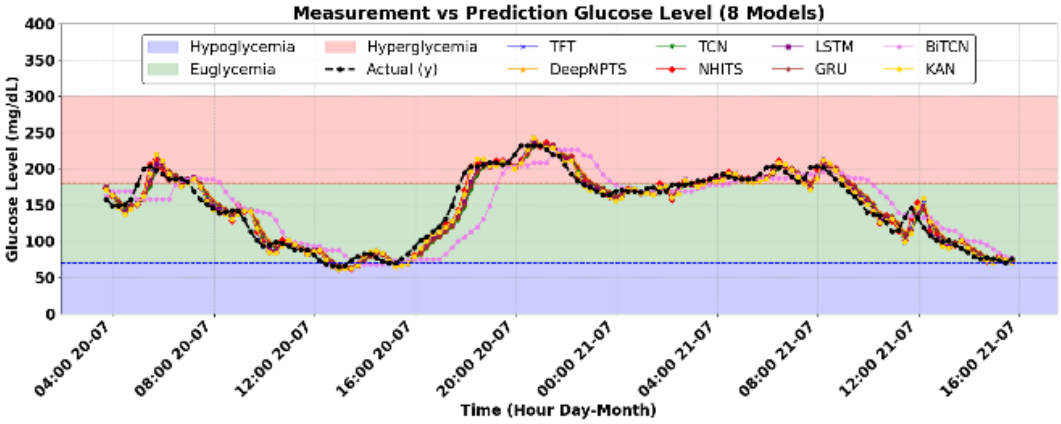


(c)

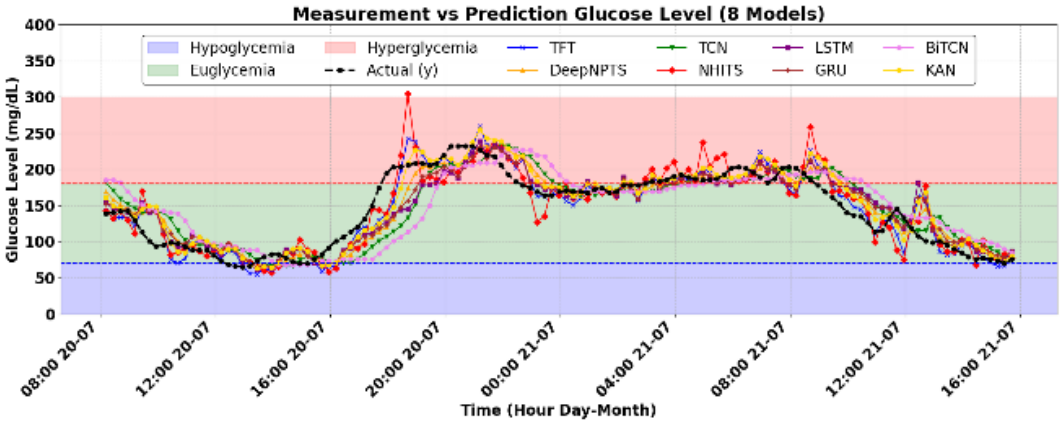


(d)

Fig. 2 *Time plots of BGL predictions from the simulation in Google Colab versus actual data of one subject (Unique_id 12A) in the ShanghaiT1DM dataset (a) Univariate for 30-minute PH; (b) Univariate for 60-minute PH; (c) Multivariate for 30-minute PH; (d) Multivariate for 60-minute PH*

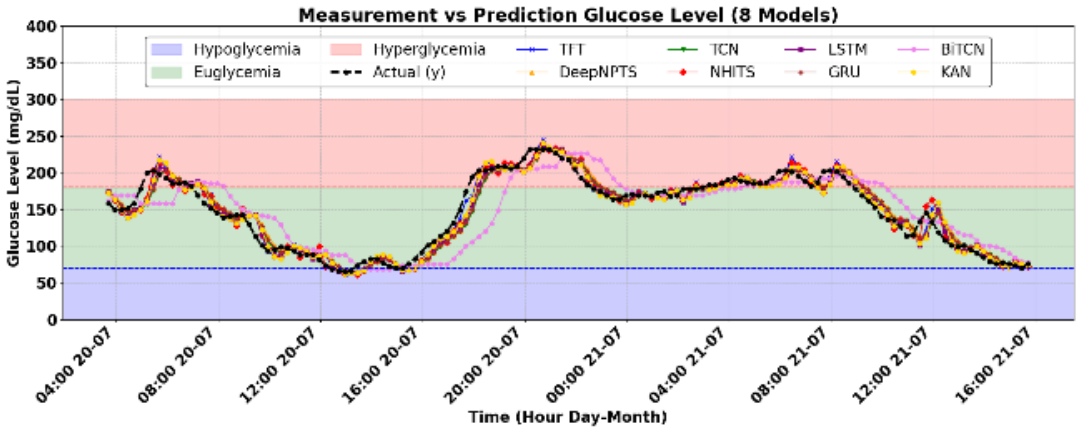


(a)

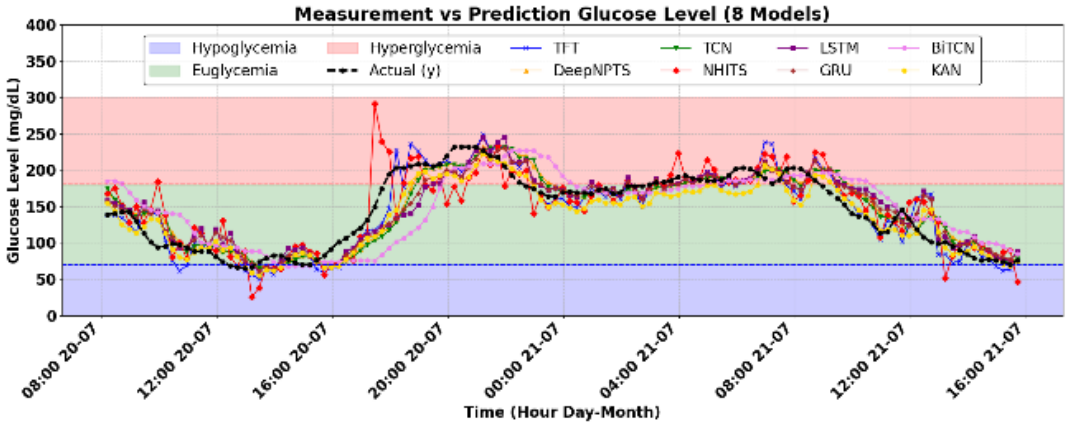


(b)

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 163-183



(c)



(d)

Fig. 3 *Time plots of BGL predictions data versus actual data of one subject (Unique_id 12A) from the deployment on NVIDIA Jetson Orin Nano in the ShanghaiT1DM dataset (a) Univariate for 30-minute PH; (b) Univariate for 60-minute PH; (c) Multivariate for 30-minute PH; (d) Multivariate for 60-minute PH*

The quantitative analysis in Table 1 confirms the visual observations. In the simulation setting with 30-minute PH and univariate inputs, TFT achieved the lowest MAE (8.77 mg/dL), and NHITS the second lowest (9.38 mg/dL). The LSTM baseline achieved an MAE of 13.45 mg/dL, and both models outperformed it significantly. During deployment, NHITS outperformed the other models in the 30-minute PH with univariate configuration with an MAE of 8.44 mg/dL, compared to TFT's 8.88 mg/dL. The MAE of the LSTM model improved from 13.45 mg/dL in the simulation to 9.07 mg/dL in the deployment.

On the other hand, BiTCN had the worst performance among all the models with an MAE of ~20.67 mg/dL. All models showed greater error on the 60-minute PH, but the relative ordering of the models remained essentially unchanged. TFT was the best model, with MAEs of 16.63 mg/dL (simulation/univariate) and 16.02 mg/dL (deployment/univariate), outperforming LSTM (22.53 mg/dL and 18.44 mg/dL, respectively). Again, the BiTCN model showed the lowest MAE of 26.76 mg/dL. Meanwhile, KAN demonstrated competitive performance, especially in the deployment environment with univariate inputs, with an MAE of 16.85 mg/dL. The error values for both the 30-minute and 60-minute PH conditions were very similar between the univariate and multivariate configurations. However, the multivariate configuration showed a higher standard deviation (STD) of the absolute errors. The deployment results showed smaller error scores and smaller STD of absolute errors in both conditions than in the simulation.

Table 1 shows the full per-subject error metrics for a representative subject (Unique_id 12A). For ease of interpretation of the dense full table, Table 2 summarizes the best-performing model per condition based on the lowest MAE for a quick comparison of top performers.

Table 1 *Complete performance error metrics (MAE ± STD, MAPE (%), RMSE) of one subject (Unique_id 12A) in the ShanghaiT1DM dataset from the simulation and deployment results for univariate and multivariate inputs at 30- and 60-minute PHs. Results are grouped by PH for easier cross-model comparison*

30-minute PH							
Models	Input	Univariate			Multivariate		
		MAE ± STD (mg/dL)	MAPE (%)	RMSE (mg/dL)	MAE ± STD (mg/dL)	MAPE (%)	RMSE (mg/dL)
LSTM	Simulation	13.45 ± 9.56	10.59	16.50	13.27 ± 11.44	9.91	17.52
	Deployment	9.07 ± 8.27	6.56	12.28	9.01 ± 8.17	6.52	12.17
GRU	Simulation	11.64 ± 8.75	9.53	14.56	13.12 ± 9.33	10.53	16.10
	Deployment	10.22 ± 8.72	7.46	13.44	10.16 ± 8.78	7.40	13.43
NHITS	Simulation	9.38 ± 8.44	7.07	12.62	10.50 ± 9.80	7.60	14.36
	Deployment	8.44 ± 7.09	6.18	11.02	8.79 ± 7.39	6.36	11.49
KAN	Simulation	10.03 ± 8.35	7.37	13.05	10.11 ± 8.66	7.46	13.31
	Deployment	9.43 ± 8.07	6.83	12.42	9.67 ± 8.15	7.04	12.64
TCN	Simulation	11.31 ± 9.48	7.92	14.76	13.24 ± 12.70	9.17	18.35
	Deployment	10.00 ± 8.99	7.21	13.45	10.10 ± 8.97	7.29	13.51
BiTCN	Simulation	20.67 ± 17.55	15.35	27.11	16.79 ± 13.90	12.48	21.80
	Deployment	20.47 ± 18.02	14.91	27.27	20.46 ± 18.00	14.91	27.26
TFT	Simulation	8.77 ± 7.93	6.37	11.82	8.74 ± 7.85	6.34	11.75
	Deployment	8.88 ± 7.81	6.41	11.83	8.77 ± 7.47	6.31	11.52
DeepNPTS	Simulation	10.35 ± 8.86	7.69	13.62	10.56 ± 9.15	7.80	13.97
	Deployment	10.64 ± 8.95	7.81	13.91	10.65 ± 8.97	7.82	13.92
60-minute PH							
Models	Input	Univariate			Multivariate		
		MAE ± STD (mg/dL)	MAPE (%)	RMSE (mg/dL)	MAE ± STD (mg/dL)	MAPE (%)	RMSE (mg/dL)
LSTM	Simulation	22.53 ± 16.49	16.66	27.91	25.23 ± 17.12	19.00	30.50
	Deployment	18.44 ± 15.69	13.85	24.21	19.59 ± 16.29	15.05	25.48
GRU	Simulation	22.07 ± 14.43	19.15	26.37	21.20 ± 14.44	18.47	25.65
	Deployment	18.12 ± 14.38	13.99	23.13	18.35 ± 14.90	14.28	23.64
NHITS	Simulation	17.51 ± 14.23	13.92	22.56	21.18 ± 17.56	15.87	27.52
	Deployment	19.60 ± 16.72	14.78	25.77	22.74 ± 19.97	17.96	30.26
KAN	Simulation	19.21 ± 14.66	14.34	24.16	19.80 ± 14.61	15.38	24.61
	Deployment	16.85 ± 14.04	13.08	21.93	17.91 ± 12.73	12.87	21.97
TCN	Simulation	16.37 ± 13.63	12.40	21.30	18.55 ± 13.88	10.24	23.13
	Deployment	21.04 ± 18.59	15.93	28.08	17.38 ± 15.02	13.24	22.97
BiTCN	Simulation	19.87 ± 14.61	14.91	24.67	20.05 ± 16.29	15.00	25.84
	Deployment	26.76 ± 22.60	20.21	35.02	26.70 ± 22.63	20.15	35.00
TFT	Simulation	16.63 ± 14.03	12.48	21.75	17.68 ± 14.68	13.52	22.98
	Deployment	16.02 ± 13.45	12.36	20.92	17.02 ± 13.57	13.44	21.77
DeepNPTS	Simulation	17.98 ± 14.08	13.56	22.84	17.97 ± 14.11	13.62	22.84
	Deployment	17.33 ± 14.21	13.32	22.41	17.36 ± 14.31	13.36	22.50

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 163-183

Table 2 *Summary of best-performing models by lowest MAE for one subject (Unique_id 12A) across all conditions*

Condition	Best Model	MAE ± STD (mg/dL)	MAPE (%)	RMSE (mg/dL)
30-min PH / Univariate / Simulation	TFT	8.77 ± 7.93	6.37	11.82
30-min PH / Univariate / Deployment	NHITS	8.44 ± 7.09	6.18	11.02
30-min PH / Multivariate / Simulation	TFT	8.74 ± 7.85	6.34	11.75
30-min PH / Multivariate / Deployment	TFT	8.77 ± 7.47	6.31	11.52
60-min PH / Univariate / Simulation	TFT	16.63 ± 14.03	12.48	21.75
60-min PH / Univariate / Deployment	KAN	16.85 ± 14.04	13.08	21.93
60-min PH / Multivariate / Simulation	TFT	17.68 ± 14.68	13.52	22.98
60-min PH / Multivariate / Deployment	TFT	17.02 ± 13.57	13.44	21.77

Note: BiTCN consistently records the highest MAE at all 30-minute PH conditions and at the 60-minute PH deployment conditions. At the 60-minute PH simulation conditions, LSTM surpasses BiTCN as the worst performer. Full results for all eight models are provided in Table 1.

Table 3 summarizes the average of the eight models' predictive performance across the entire 16-subject population, comparing simulation and edge deployment for univariate and multivariate inputs at 30- and 60-minute PHs. In the 30-minute PH simulation/univariate setting, TFT recorded the lowest $\overline{\text{MAE}}$ of 10.35 mg/dL, and KAN recorded an $\overline{\text{MAE}}$ of 10.87 mg/dL, both significantly lower than the LSTM baseline. In the 30-minute PH with the deployment/univariate setting, NHITS performed best with an $\overline{\text{MAE}}$ of 10.03 mg/dL, slightly outperforming TFT (10.29 mg/dL) and KAN (10.72 mg/dL). These three models outperformed the LSTM baseline, which had an average MAE of 10.99 mg/dL. BiTCN remained the lowest performer with an average MAE of 27.25 mg/dL.

For the 60-minute PH, TFT and KAN consistently led the results, both significantly lower than the LSTM baseline. TFT achieved average MAEs of 21.61 mg/dL in the simulation/univariate setting and 20.10 mg/dL in the deployment/univariate setting. In comparison, KAN achieved average MAEs of 20.02 mg/dL and 21.12 mg/dL, in the simulation/univariate and the deployment/univariate settings, respectively. NHITS experienced performance degradation at this longer PH, with its $\overline{\text{MAE}}$ increased noticeably.

Minimal differences in error values were generally observed across the population between the univariate and multivariate configurations. An exception was noted for the TCN model at a 60-minute PH, where its multivariate configuration ($\overline{\text{MAE}}$ of 22.50 mg/dL) outperformed its univariate configuration ($\overline{\text{MAE}}$ of 28.15 mg/dL). Besides, most models deployed on the NVIDIA Jetson Orin Nano showed lower error scores than their cloud-simulated counterparts.

Table 3 reports the population-level average performance across all 16 subjects, which serves as the primary basis for the study's conclusions. To guide interpretation of the full table, Table 4 presents a condensed ranking of all eight models by average MAE under the most clinically relevant condition (univariate input for both PHs), allowing rapid identification of top and bottom performers; consult the complete metrics in Table 3.

Table 3 *Complete average predictive performance of eight population-based DL models across 16 subjects from the ShanghaiT1DM dataset, comparing simulation and edge deployment for univariate and multivariate inputs at 30- and 60-minute PHs*

Models	Input	30-minute PH					
		Univariate			Multivariate		
		$\overline{\text{MAE}} \pm$ STD (mg/dL)	$\overline{\text{MAPE}} \pm$ STD (%)	$\overline{\text{RMSE}} \pm$ STD (mg/dL)	$\overline{\text{MAE}} \pm$ STD (mg/dL)	$\overline{\text{MAPE}} \pm$ STD (%)	$\overline{\text{RMSE}} \pm$ STD (mg/dL)
LSTM	Simulation	11.49±0.86	8.66±0.53	16.11±1.35	12.04±0.96	8.90±0.58	17.34±1.54
	Deployment	10.99±0.94	8.02±0.51	15.21±1.31	10.88±0.92	7.95±0.50	15.03±1.28
GRU	Simulation	10.92±0.68	8.59±0.48	14.79±0.94	12.34±1.07	9.80±1.03	16.86±1.41
	Deployment	12.44±1.16	9.21±0.66	17.03±1.61	12.42±1.16	9.17±0.65	17.03±1.61
NHITS	Simulation	12.36±0.79	9.37±0.50	17.26±1.07	12.84±0.94	9.65±0.55	18.42±1.46
	Deployment	10.03±0.77	7.52±0.46	13.82±1.11	10.14±0.83	7.61±0.51	14.04±1.19
KAN	Simulation	10.87±0.85	7.99±0.50	14.87±1.18	10.77±0.89	8.02±0.53	14.87±1.24
	Deployment	10.72±0.85	7.94±0.49	14.76±1.20	10.84±0.89	8.05±0.53	14.97±1.26
TCN	Simulation	12.61±1.07	9.61±0.77	18.36±1.87	12.06±0.84	9.14±0.54	16.94±1.26

Models	Input	30-minute PH					
		Univariate			Multivariate		
		$\overline{\text{MAE}} \pm$ STD (mg/dL)	$\overline{\text{MAPE}} \pm$ STD (%)	$\overline{\text{RMSE}} \pm$ STD (mg/dL)	$\overline{\text{MAE}} \pm$ STD (mg/dL)	$\overline{\text{MAPE}} \pm$ STD (%)	$\overline{\text{RMSE}} \pm$ STD (mg/dL)
	Deployment	12.33±1.14	8.94±0.61	17.09±1.59	12.43±1.16	9.07±0.63	17.16±1.61
BiTCN	Simulation	27.25±3.24	19.97±1.85	37.36±4.56	27.20±3.22	20.03±1.87	37.23±4.53
	Deployment	26.94±3.09	19.88±1.81	36.88±4.32	26.93±3.09	19.88±1.81	36.87±4.32
TFT	Simulation	10.35±0.81	7.78±0.53	14.13±1.07	10.59±0.87	7.92±0.55	14.92±1.30
	Deployment	10.29±0.78	7.67±0.47	14.14±1.10	10.27±0.83	7.66±0.49	14.17±1.17
DeepNPTS	Simulation	13.09±1.31	9.68±0.72	17.85±1.84	13.42±1.35	10.00±0.77	18.51±1.85
	Deployment	12.94±1.24	9.65±0.70	17.69±1.72	12.94±1.24	9.65±0.70	17.69±1.72
		60-minute PH					
Models	Input	Univariate			Multivariate		
		$\overline{\text{MAE}} \pm$ STD (mg/dL)	$\overline{\text{MAPE}} \pm$ STD (%)	$\overline{\text{RMSE}} \pm$ STD (mg/dL)	$\overline{\text{MAE}} \pm$ STD (mg/dL)	$\overline{\text{MAPE}} \pm$ STD (%)	$\overline{\text{RMSE}} \pm$ STD (mg/dL)
LSTM	Simulation	26.26±2.09	20.30±1.40	36.95±3.13	28.78±2.47	21.81±1.37	40.70±3.48
	Deployment	22.15±2.80	17.92±2.37	31.27±4.72	22.58±2.87	18.45±2.43	31.69±4.81
GRU	Simulation	20.86±1.59	15.80±1.04	28.55±2.17	22.09±1.76	16.58±1.05	30.32±2.43
	Deployment	21.58±2.83	17.65±2.33	30.09±4.72	22.19±2.90	17.79±2.32	30.77±4.69
NHITS	Simulation	24.51±2.33	19.23±1.66	33.93±3.45	27.75±3.11	20.58±1.77	40.33±4.74
	Deployment	22.93±1.85	17.50±1.28	31.83±2.58	23.41±2.30	17.78±1.54	32.54±3.28
KAN	Simulation	20.02±1.87	14.76±1.04	27.24±2.47	21.70±1.79	16.81±1.25	29.54±2.43
	Deployment	21.12±1.86	15.91±1.12	28.38±2.60	21.15±2.22	14.76±1.12	28.08±2.82
TCN	Simulation	21.96±1.76	16.54±1.14	30.00±2.41	23.06±2.07	17.50±1.21	32.44±2.91
	Deployment	28.15±3.11	21.04±1.85	38.10±4.31	22.50±2.31	16.82±1.36	31.58±3.32
BiTCN	Simulation	33.86±4.12	25.29±2.47	45.34±5.66	33.94±4.11	25.33±2.50	45.43±5.63
	Deployment	34.05±4.00	25.60±2.46	45.63±5.49	34.05±4.01	25.58±2.47	45.65±5.50
TFT	Simulation	21.61±1.70	16.34±1.01	29.98±2.47	22.96±1.98	17.58±1.41	33.45±2.93
	Deployment	20.10±1.81	15.10±1.10	28.15±2.60	21.09±2.16	16.03±1.33	30.06±3.13
DeepNPTS	Simulation	22.75±2.42	17.05±1.43	30.70±3.35	22.89±2.50	17.17±1.49	30.89±3.44
	Deployment	22.54±2.34	17.03±1.40	30.63±3.27	22.54±2.34	17.04±1.40	30.64±3.27

Table 4 *Condensed ranking of eight population-based DL models by average MAE for univariate-input configuration at 30- and 60-minute PHs, comparing simulation and edge deployment*

Rank	Model	30-min Sim. $\overline{\text{MAE}}$ (mg/dL)	30-min Deploy. $\overline{\text{MAE}}$ (mg/dL)	60-min Sim. $\overline{\text{MAE}}$ (mg/dL)	60-min Deploy. $\overline{\text{MAE}}$ (mg/dL)
1	TFT	10.35	10.29	21.61	20.10
2	KAN	10.87	10.72	20.02	21.12
3	GRU	10.92	12.44	20.86	21.58
4	LSTM	11.49	10.99	26.26	22.15
5	NHITS	12.36	10.03	24.51	22.93
6	TCN	12.61	12.33	21.96	28.15
7	DeepNPTS	13.09	12.94	22.75	22.54
8 (worst)	BiTCN	27.25	26.94	33.86	34.05

Key observations: (1) TFT and KAN are always one of the top two in all conditions. (2) NHITS achieves the best deployment MAE at 30-min PH (10.03 mg/dL) but deteriorates drastically at 60-min PH. (3) The greatest cloud-to-edge performance degradation of TCN is at 60-min PH (21.96 → 28.15 mg/dL, univariate). (4) In all conditions, BiTCN is clearly the worst performer by a large margin. Table 3: Full metrics including MAPE, RMSE, and multivariate setups.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 163-183

Figure 4 (simulation) and Figure 5 (deployment) visually confirm the error comparisons from Table 3. In Figure 4 (simulation), TFT and KAN were consistently at the top, followed by GRU, NHITS, DeepNPTS, and LSTM, then TCN, with BiTCN consistently ranking last. Figure 5 (deployment) shows a similar pattern for the 60-minute PH. However, at the 30-minute PH in deployment, NHITS showed the smallest error bars, consistent with its lowest average MAE (10.03 mg/dL). Overall, errors were consistently lower at the 30-minute PH than at the 60-minute PH across all models, and BiTCN remained the worst performer. Minor differences were observed between univariate and multivariate configurations, except for the TCN model on edge deployment at a 60-minute PH.

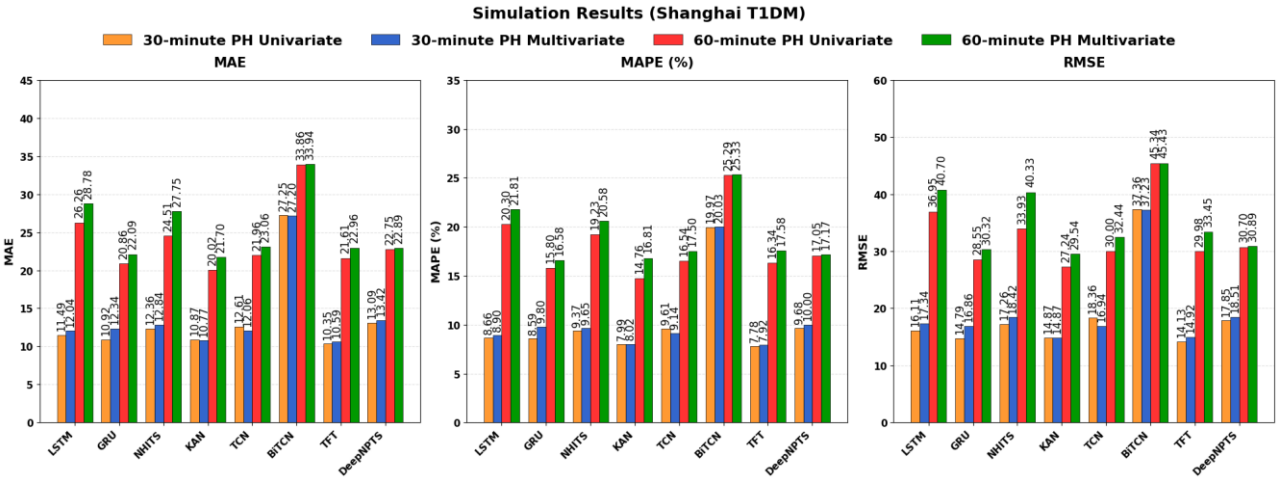


Fig. 4 *Visualization comparison of matrix error scores across models at 30- and 60-minute PHs in univariate- and multivariate-input configurations (simulation approach)*

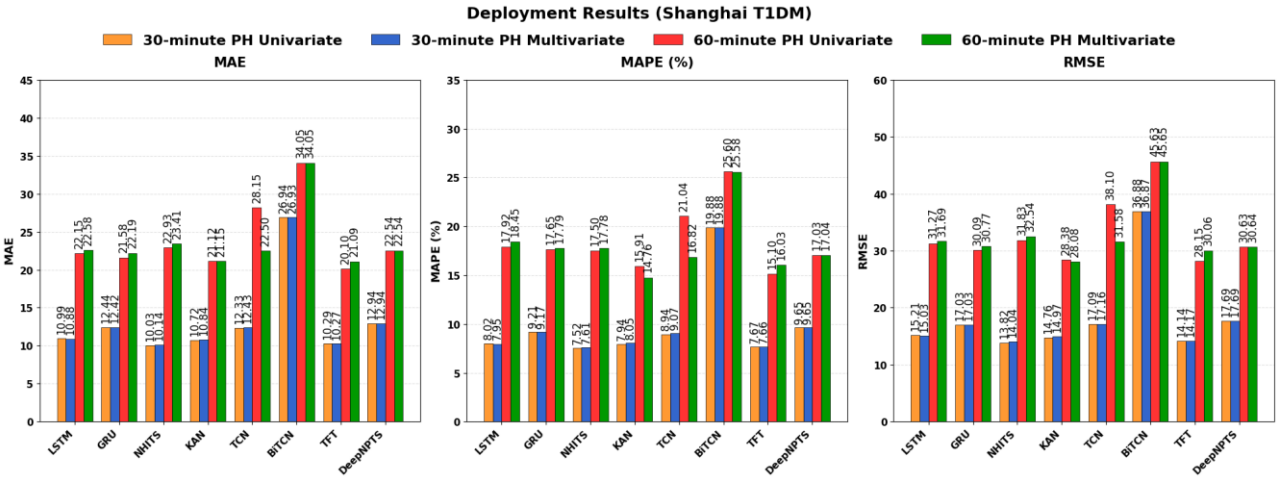


Fig. 5 *Visualization comparison of matrix error scores across models at 30- and 60-minute PHs in univariate and multivariate input configurations (edge deployment approach)*

3.2 Deployment Efficiency at 30- and 60-Minute PHs

Table 5 and Figure 6 present the average values of the eight models' deployment efficiency metrics across the 16-subject population on the NVIDIA Jetson Orin Nano, including total inference time, latency, throughput, and power consumption. For the 30-minute PH, LSTM and GRU models showed high inference efficiency. LSTM, with univariate input, recorded the fastest overall inference time of 50.50 s and the lowest latency of 0.28 s, achieving a throughput of 3.96 preds./s. KAN demonstrated competitive efficiency, attaining the highest throughput of 4.05 preds./s and the lowest power consumption of 7.64 W. In contrast, TFT required the longest inference time of 66.34 s and exhibited the lowest throughput of 3.05 preds./s, with a latency of 0.34 s. BiTCN's inference efficiency was comparable to that of the other models.

Figure 6 shows that deployment efficiency at the 30-minute PH was higher than at the 60-minute PH, primarily because the total number of predictions was higher. A slight increase in deployment efficiency was observed in the multivariate input configuration compared to the univariate configuration. Across all models, the sustained power consumption remained approximately 8 W.

Table 5 presents full deployment efficiency metrics across all models and conditions on the NVIDIA Jetson Orin Nano. Given the density of the full table, Table 6 summarizes the key deployment efficiency characteristics per model under the univariate configuration, the primary condition of interest, at both PHs, to allow direct identification of efficiency trade-offs, consulting the complete data in Table 5.

Table 5 *Complete average deployment efficiency metrics of eight population-based DL models across 16 subjects from the ShanghaiT1DM dataset at 30- and 60-minute PHs for univariate- and multivariate-input configurations on NVIDIA Jetson Orin Nano*

30-minute PH						
Models	Input	Total Predictions	Total Inference Time (s)	Latency (s)	Throughput (preds./s)	Power (W)
LSTM	Univariate	193.69±15.86	50.50±4.30	0.28±0.05	3.96±0.30	7.95 ± 0.06
	Multivariate	193.69±15.86	53.38±6.07	0.29±0.04	3.81±0.20	7.80 ± 0.05
GRU	Univariate	193.69±15.86	53.40±6.03	0.28±0.02	3.97±0.31	7.78 ± 0.03
	Multivariate	193.69±15.86	56.16±6.31	0.29±0.02	3.77±0.30	7.82 ± 0.03
NHITS	Univariate	191.69±15.86	60.00±6.81	0.31±0.02	3.10±0.12	7.87 ± 0.06
	Multivariate	191.69±15.86	62.11±8.96	0.38±0.02	3.05±0.18	7.83 ± 0.06
KAN	Univariate	191.69±15.86	56.24±2.42	0.32±0.02	4.05±0.55	7.64 ± 0.06
	Multivariate	191.69±15.86	57.22±2.22	0.34±0.03	4.00±0.28	7.83 ± 0.08
TCN	Univariate	193.69±15.86	55.11±6.20	0.31±0.02	3.80±0.88	8.11 ± 0.04
	Multivariate	193.69±15.86	56.64±6.30	0.32±0.02	3.75±0.30	7.90 ± 0.07
BiTCN	Univariate	187.69±15.86	57.32±2.22	0.27±0.02	3.75±0.24	7.79 ± 0.04
	Multivariate	187.69±15.86	58.40±5.24	0.28±0.02	3.65±0.20	7.80 ± 0.04
TFT	Univariate	193.69±15.86	66.34±3.96	0.34±0.01	3.05±0.34	7.91±0.06
	Multivariate	193.69±15.86	67.75±5.54	0.36±0.02	2.99±0.47	7.99 ± 0.07
DeepNPTS	Univariate	191.69±15.86	57.44±4.32	0.26±0.02	3.70±0.42	7.85 ± 0.04
	Multivariate	191.69±15.86	59.02±4.05	0.27±0.02	3.65±0.50	7.73 ± 0.04
60-minute PH						
Models	Input	Total Predictions	Total Inference Time (s)	Latency (s)	Throughput (preds./s)	Power (W)
LSTM	Univariate	169.69±15.86	45.38±5.58	0.26±0.02	4.20±0.12	7.82 ± 0.04
	Multivariate	169.69±15.86	48.44±2.68	0.27±0.03	4.05±0.16	7.80±0.03
GRU	Univariate	169.69±15.86	44.76±5.41	0.26±0.02	4.24±0.40	7.79 ± 0.05
	Multivariate	169.69±15.86	45.54±5.30	0.27±0.02	4.12±0.20	7.79 ± 0.05
NHITS	Univariate	173.69±15.86	53.16±6.71	0.30±0.02	3.47±0.17	7.85 ± 0.10
	Multivariate	173.69±15.86	55.62±6.31	0.31±0.20	3.25±0.22	7.84 ± 0.04
KAN	Univariate	185.69±15.86	49.77±5.69	0.28±0.02	4.35±0.46	7.79 ± 0.05
	Multivariate	185.69±15.86	51.12±2.22	0.30±0.03	4.20±0.20	7.79 ± 0.03
TCN	Univariate	189.69±15.86	52.63±5.93	0.29±0.02	3.95±0.42	7.78 ± 0.06
	Multivariate	189.69±15.86	52.95±7.98	0.30±0.03	3.89±0.49	8.07 ± 0.04
BiTCN	Univariate	185.69±15.86	51.34±5.59	0.25±0.02	3.88±0.28	7.86 ± 0.05
	Multivariate	185.69±15.86	54.59±7.07	0.26±0.02	3.80±0.31	7.98 ± 0.08
TFT	Univariate	189.69±15.86	61.10±3.11	0.32±0.02	3.35±0.44	8.14 ± 0.04
	Multivariate	189.69±15.86	63.55±5.22	0.33±0.02	3.30±0.24	8.17 ± 0.04
DeepNPTS	Univariate	189.69±15.86	53.20±6.18	0.24±0.02	3.90±0.29	7.77 ± 0.04
	Multivariate	189.69±15.86	55.40±4.24	0.25±0.02	3.87±0.29	7.79 ± 0.04

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 163-183

Table 6 *Condensed deployment efficiency summary for eight DL models (univariate input) on NVIDIA Jetson Orin Nano at 30- and 60-minute PHs. Values represent averages across 16 subjects*

Model	30-min: Inf. Time (s)	30-min: Latency (s)	30-min: Throughput (preds./s)	30-min: Power (W)	60-min: Inf. Time (s)	60-min: Latency (s)	60-min: Throughput (preds./s)	60-min: Power (W)
LSTM	50.50	0.28	3.96	7.95	45.38	0.26	4.20	7.82
GRU	53.40	0.28	3.97	7.78	44.76	0.26	4.24	7.79
NHITS	60.00	0.31	3.10	7.87	53.16	0.30	3.47	7.85
KAN	56.24	0.32	4.05	7.64	49.77	0.28	4.35	7.79
TCN	55.11	0.31	3.80	8.11	52.63	0.29	3.95	7.78
BiTCN	57.32	0.27	3.75	7.79	51.34	0.25	3.88	7.86
TFT	66.34	0.34	3.05	7.91	61.10	0.32	3.35	8.14
Deep-NPTS	57.44	0.26	3.70	7.85	53.20	0.24	3.90	7.77

Key observations: (1) KAN has the highest throughput (4.05 preds./s) at 30-min PH and the lowest power consumption (7.64 W), with the best efficiency profile among all models. (2) Although TFT has high predictive performance, it has the longest inference time and the lowest throughput at both PHs and the highest power at 60-min PH. (3) LSTM and GRU have the fastest inference time given their simpler recurrent architectures. (4) All models consume power in a narrow range (~7.6–8.2 W) that is well below the device maximum of 15 W, confirming their uniform viability for continuous deployment. Full metrics are shown in Table 5, including multivariate configurations and standard deviations.

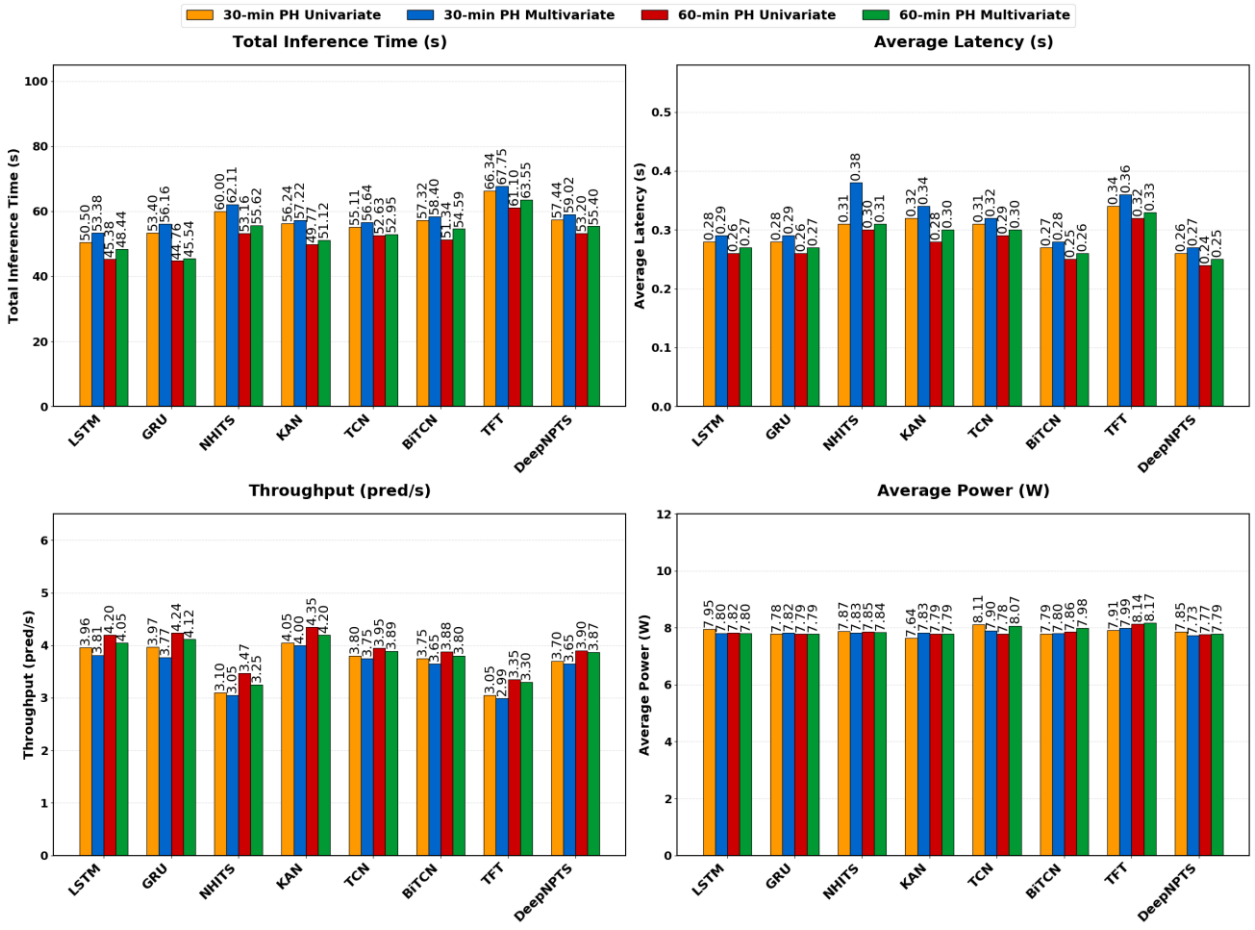


Fig. 6 *Visualization comparison of deployment efficiency metrics across models (30- and 60-minute PHs in univariate and multivariate input configurations) from Table 5*

4. Discussion

In this section, we interpret the experimental results and try to analyze the significance, implications and relevant tradeoffs for BGL prediction in T1DM management. The visual inspection of the prediction performance shows that the predictions are more accurate at 30 minutes rather than at 60 minutes, which indicates the inherent

complexity of predicting BGL over longer time horizons. This degradation is technically expected with the prediction horizon. As the horizon increases, the window of input data includes glucose dynamics increasingly distal from the target time step. Unlogged physiological events, like delayed postprandial glucose absorption, unlogged physical activity, or stress-induced excursions, add increasing cumulative uncertainty, constraining the model's ability to reliably project future values. The fact that this degradation is seen across all eight architectures provides evidence that it is a fundamental signal-theoretic constraint of CGM-based forecasting and not a weakness of any particular model. The larger absolute STD in the multivariate configuration indicates higher variability in the prediction errors, which means higher uncertainty when more input features are considered. But the difference in mean errors is relatively small. This modest multivariate advantage is mechanistically explained by the binary encoding of insulin and meal features, which, without dose-magnitude information, contribute primarily as temporal event markers, providing little extra predictive signal beyond the glucose series itself. This link is further discussed in the context of dataset limitations later in this section. In particular, the multivariate configuration significantly outperforms the univariate configuration for the TCN model at the 60-minute PH. This may be explained by the dilated causal convolution architecture of TCN, which progressively widens its receptive field across layers and may therefore be particularly sensitive to the temporal co-occurrence between binary event indicators and subsequent glucose trajectories at longer horizons, a dependence that architectures relying on global attention or recurrent gating may already encode implicitly from the glucose signal alone.

The consistent trend of lower error scores and smaller STDs in edge deployments compared to cloud simulations is attributable to a well-defined methodological difference rather than a hardware-specific effect. In the simulation (cross-validation) scheme, 25% of the 80% training split is withheld at each fold for validation. In the deployment (train-test split) scheme, the model is trained on the full 80% training partition without a validation holdout, providing greater and more comprehensive exposure to inter-subject glucose patterns in the population-based dataset. This more extensive training leads to a lower generalization error on the common 20% test set, which explains the observation that most models have a lower MAE and STD when deployed on the edge compared to the cloud simulation. This explains what can be named the edge paradox of this study. The clearest example is the LSTM baseline, which obtains an MAE of 9.07 mg/dL in deployment versus 13.45 mg/dL in simulation for the representative subject (Unique_id 12A) at 30-minute PH, a reduction of roughly 32.5%, which can be attributed to the increased training partition in the deployment scheme. It verifies the following to ensure no data leakage: (1) the 20% test split is strictly time-ordered and chronologically follows the 80% training split for both schemes, with no data from the test period available during training; (2) the same 20% test subset is used for both conditions, ensuring that the MAE reduction is due to a real improvement in generalization, rather than composition of the test set; and (3) the rolling-forecasting approach during edge inference updates the input window sequentially, with no look-ahead, precluding any form of future data access. The same directional pattern is evident across almost all eight models and the entire 16-subject population (Table 3), confirming that the effect is methodological. It is also observed that the rolling-forecasting approach adopted during edge deployment may further smooth the prediction errors over the test period. These two factors, combined, the increased training data and rolling inference, account for the advantage in predictive performance on the deployment side and should not be taken as evidence that the edge hardware itself improves the predictive performance of the model.

The consistently better performance of TFT and KAN at both 30- and 60-minute PHs confirms the robustness of their architectural designs across a broad range of prediction scenarios. Both models consistently outperform the LSTM baseline and other models. For the cloud simulations, TFT achieves the lowest MAE. KAN also performs well in edge-deployment scenarios, due to the optimal trade-off between predictive performance and deployment efficiency. NHITS, on the other hand, shows strong performance for short-term prediction (30-minute PH) in edge deployment (MAE: 10.03 mg/dL), but this advantage is lost at longer PHs. Despite its high architectural complexity, BiTCN is unsuitable for BGL prediction, as findings have repeatedly shown low predictive performance across all conditions.

The KAN and TFT architectures offer significant improvements over conventional architectures, such as LSTM, with higher predictive performance and greater interpretability. KAN uses learnable activation functions on weights, resulting in superior predictive performance, faster scaling laws, and visual interpretability that facilitates the discovery of mathematical patterns. KAN achieves competitive BGL predictive performance in T1DM across edge and cloud environments, comparable to state-of-the-art methods [48],[49],[50]. The use of a KAN architecture offers a novel approach to BGL prediction, balancing interpretability, efficiency, and robustness to fluctuations and data noise [51]. KAN differs from other architectures tested in this work, especially TFT, which provides similar or slightly better predictive performance in cloud simulations than in real edge-deployment scenarios. KAN achieved a lower deployment MAE of 21.12 mg/dL at the clinically critical 60-minute prediction horizon than its own simulation MAE (20.02 mg/dL averaged across the population). This cloud-to-edge improvement pattern is not consistently found in other models, suggesting that KAN's parametric spline functions benefit from the deterministic, low-overhead execution environment of the Jetson Orin Nano GPU. Moreover, KAN had the lowest power consumption of all evaluated models (7.64 W at 30-minute PH), and the highest throughput (4.05 preds./s), which were not simultaneously accomplished by any other model in this study. This trade-off

between energy and predictive performance is especially important for CGM wearables or implantables with stringent power budgets. TFT, on the other hand, had the lowest MAE of all models, but the highest inference latency (66.34 s total inference time) and the lowest throughput (3.05 preds./s) and was not suitable for low-latency edge inference. Altogether, these results demonstrate that KAN not only provides a competitive benchmark result but also a uniquely advantageous architecture for real-time, low-power BGL prediction on resource-constrained edge hardware. TFT is designed to be better at multi-horizon forecasting, where it handles heterogeneous data, provides interpretable insights into temporal patterns and important variables through an attention mechanism, and achieves significant performance gains over LSTM-based models [50].

The magnitude of these errors needs to be explicitly framed in terms of their clinical implications. The International Organisation for Standardisation (ISO 15197:2013) recommends that glucose monitoring systems provide results within ±15 mg/dL of the reference value for concentrations below 100 mg/dL and within ±15% for concentrations above 100 mg/dL [52],[53]. At a 30-minute prediction horizon, the best population-averaged models (TFT, MAE: 10.35 mg/dL; KAN, MAE: 10.87 mg/dL) nearly reached the clinical threshold, suggesting that their predictions may be accurate enough to facilitate proactive interventions (e.g., generation of preventative alerts or assisted insulin dosing decisions in clinical decision support systems). The best population mean MAE at the 60-minute horizon was 20.02 mg/dL (KAN simulation), TFT application 20.10 mg/dL) Being above the ISO threshold indicates that predictions at this horizon are better used as a trend indicator or early warning signal for impending hypoglycemia or hyperglycemia rather than as a precise guide to dosing. Furthermore, the consistently poor performance of BiTCN across all conditions (population MAE > 27 mg/dL at 30-minute PH; > 33 mg/dL at 60-minute PH) places it beyond clinically acceptable bounds under any prediction horizon, definitively excluding it from consideration for any glucose management application regardless of its quite favorable deployment efficiency.

The models' deployment efficiency on NVIDIA Jetson Orin Nano is comparable to previous studies[15],[27]. compressed-edge approaches that reduce model capacity [10], Jetson Orin Nano enables the execution of uncompressed models that utilize its integrated AI GPU, resulting in low latency, high throughput and competitive power consumption[38]. Statistical significance tests comparing the eight DL models specific to deployment efficiency of total inference, latency and throughput indicate that there is no significant difference ($p>0.05$) between the models. However, power consumption among models differs significantly ($p<0.05$) across all conditions (30- and 60-minute PHs, univariate and multivariate input configurations) on the NVIDIA Jetson Orin Nano, presumably due to model complexity affecting GPU/CPU utilization[45]. Additionally, the normality test indicates that the power is not normally distributed ($p<0.001$), possibly due to dynamic fluctuations in GPU/CPU usage during inference. In contrast, latency and throughput tend to follow a normal distribution ($p>0.001$) across most conditions, as each model operates within a stable system environment.

Prediction performance metrics (MAE, MAPE, and RMSE) and deployment efficiency metrics (inference time, latency, throughput, and power consumption) were used to thoroughly evaluate the trade-off between predictive performance and power consumption on edge devices. We consider KAN as a model that strikes an ideal balance, with high predictive performance and very favorable deployment efficiency (4.05 preds./s throughput at 7.64 W power consumption), which makes it very suitable for practical applications on low-power devices. On the other hand, BiTCN is not commonly advised because of its notable predictive constraints, despite having similar deployment efficiency. Additional efficiency analysis confirmed the feasibility of implementing edge-based DL models for real-time BGL prediction. The power consumption is low (~8 W) for all the models, which is well below the maximum limit of the NVIDIA Jetson Orin Nano (15 W), eliminating the heavy reliance on cloud connectivity.

The results also demonstrate that the population-based modeling approach obtained predictive performance and generalization similar to previous population-based studies [10] and even comparable to an individual model-based approach [45], thus more scalable for clinical implementation. However, this study has several limitations that should be clearly acknowledged. First, the use of 16 subjects from a single clinical dataset limits generalizability of the findings to more broadly representative ethnically diverse T1DM populations. The ShanghaiT1DM dataset was collected in a specific clinical setting in China, and the glucose dynamics, dietary habits, and insulin regimens may systematically differ in other populations or healthcare systems. Thus, reported model rankings and error magnitudes should be cautiously interpreted when extrapolating to Western cohorts or patients with markedly different glycemic profiles. Second, the small number of subjects in each cohort may cause over-fitting even with a cross-validation scheme. Although Optuna-based hyperparameter tuning with time-based cross-validation reduces this risk by searching optimal configurations over temporal folds rather than random splits, it cannot eliminate the possibility that some models have been tuned to patterns specific to this dataset's 16 subjects. Future work should prioritize cross-dataset validation to establish the clinical reliability of the proposed edge-based population modeling framework at scale. Third, the binary encoding of insulin and carbohydrate features is an acknowledged limitation in the model's capacity to learn dose-response relationships, which are physiologically fundamental to glycemic dynamics in T1DM. The absence of dosage magnitude information in the multivariate feature space is the primary reason why multivariate configurations did not yield statistically meaningful improvements over univariate configurations in this study. This is not an inherent

limitation of the population-based or edge-deployment framework, but a dataset-specific and preprocessing-specific constraint. Future implementations on datasets with standardized, complete insulin dosage and carbohydrate logging, such as those generated by closed-loop insulin delivery systems, should incorporate continuous or dose-normalized feature representations to fully realize the potential of multivariate modeling for BGL prediction.

The research successfully bridges the gap between DL algorithm development and practical application, and the edge computing-based BGL prediction is an attractive option for modern T1DM management, featuring high predictive performance, efficiency, and scalability. These findings further enable integration of these models into clinical decision support systems and low-power wearable devices, with the potential to improve patients' quality of life through more reliable and timely predictions.

5. Conclusion

Since this paper was written, this work is the first systematic evaluation of KAN in a population-based DL setting for BGL prediction and on an uncompressed edge computing platform with seven competing architectures. This work uncovers novel insights on the predictive performance-efficiency trade-off that would not be apparent from a cloud-only benchmarking. In this study, a population-based DL approach for BGL in T1DM patients using edge computing was successfully proposed and evaluated. The tests of eight DL architectures on the ShanghaiT1DM dataset demonstrated that KAN and TFT consistently exhibited the best predictive performance for both 30- and 60-minute PHs, and KAN was the optimal choice in terms of both high predictive performance and superior deployment efficiency. By contrast, BiTCN performed poorly in all conditions. Edge-based predictions led to lower errors than cloud simulations with low power consumption in deployments on the NVIDIA Jetson Orin Nano edge device. The results confirm the feasibility of deploying real-time DL models on low-power edge devices with minimal reliance on cloud connectivity. In summary, the edge-based approach with KAN models offers an accurate, efficient, and scalable solution for modern T1DM management on resource-constrained devices, bridging advances in DL with the practical needs of contemporary diabetes management. However, generalizability of the findings is a recognized limitation of this study since it is based on 16 subjects from a single clinical dataset. Further studies are warranted to test the proposed framework in larger, multi-center and ethnically diverse cohorts to determine broader clinical applicability.

Acknowledgement

The authors acknowledge Universiti Tun Hussein Onn Malaysia through TIER 1 Grant No. Q872. Appreciation is also extended to the Microelectronics & Nanotechnology - Shamsuddin Research Centre (MiNT-SRC) for providing laboratory facilities. Additionally, the authors used Grammarly and Quillbot for grammar checking and language editing, as well as Google NotebookLM to support the literature gap analysis and the determination of the research methodology. All content produced has been reviewed and verified by the authors, who are fully responsible for the final submission.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Ade Anggian Hakim, Farhanahani Mahmud;* ***data collection:*** *Ade Anggian Hakim;* ***analysis and interpretation of results:*** *Ade Anggian Hakim, Farhanahani Mahmud;* ***draft manuscript preparation:*** *Ade Anggian Hakim, Farhanahani Mahmud, Kazuki Nakajima, Marlia Morsin, Chin Fhong Soon. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Sun, H., Saeedi, P., Karuranga, S., Pinkepank, M., Ogurtsova, K., Duncan, B. B., Stein, C., Basit, A., Chan, J. C. N., Mbanya, J. C., Pavkov, M. E., Ramachandaran, A., Wild, S. H., James, S., Herman, W. H., Zhang, P., Bommer, C., Kuo, S., Boyko, E. J., & Magliano, D. J. (2022). IDF Diabetes Atlas: Global, regional and country-level diabetes prevalence estimates for 2021 and projections for 2045. Diabetes Research and Clinical Practice, 183, Article 109119. https://doi.org/10.1016/j.diabres.2021.109119

[2] Rimon, M. T. I., Hasan, M. W., Hassan, M. F., & Cesmeci, S. (2024). Advancements in insulin pumps: a comprehensive exploration of insulin pump systems, technologies, and future directions. *Pharmaceutics*, 16(7), 944. https://doi.org/10.3390/pharmaceutics16070944

[3] Beck, R. W., Bergenstal, R. M., Laffel, L. M., & Pickup, J. C. (2019). Advances in technology for management of type 1 diabetes. *The Lancet, 394*(10205), 1265–1273. https://doi.org/10.1016/s0140-6736(19)31142-0

[4] Woldaregay, A. Z., Årsand, E., Walderhaug, S., Albers, D., Mamykina, L., Botsis, T., & Hartvigsen, G. (2019). Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes. *Artificial Intelligence in Medicine, 98*, 109–134. https://doi.org/10.1016/j.artmed.2019.07.007

[5] Brown, S. A., Kovatchev, B. P., Raghinaru, D., Lum, J. W., Buckingham, B. A., Kudva, Y. C., Laffel, L. M., Levy, C. J., Pinsker, J. E., Wadwa, R. P. and Dassau, E. (2019). Six-month randomized, multicenter trial of closed-loop control in type 1 diabetes. *New England Journal of Medicine*, *381*(18), 1707-1717. https://doi.org/10.1056/nejmoa1907863

[6] Della Cioppa, A., De Falco, I., Koutny, T., Scafuri, U., Ubl, M., & Tarantino, E. (2023). Reducing high-risk glucose forecasting errors by evolving interpretable models for type 1 diabetes. *Applied Soft Computing, 134*, Article 110012. https://doi.org/10.1016/j.asoc.2023.110012

[7] Maahs, D. M., Calhoun, P., Buckingham, B. A., Chase, H. P., Hramiak, I., Lum, J., Cameron, F., Bequette, B. W., Aye, T., Paul, T., & others. (2014). A randomized trial of a home system to reduce nocturnal hypoglycemia in type 1 diabetes. *Diabetes Care, 37*(7), 1885–1891. https://doi.org/10.2337/dc13-2159

[8] Shuvo, M. M. H., & Islam, S. K. (2023). Deep multitask learning by stacked long short-term memory for predicting personalized blood glucose concentration. *IEEE Journal of Biomedical and Health Informatics, 27*(4), 1612–1623. https://doi.org/10.1109/JBHI.2022.3233486

[9] Wardhani, K. D. K. , Shahreen Kasim, Wawan Yunanto, Deshinta Arrova Devi, Rohayanti Hassan, Sheeba Armoogum, & Mohammad Syafwan Arshad. (2026). Multimodal Alignment and Fusion for Diabetic Retinopathy Detection using Deep Learning Approach. *Journal of Soft Computing and Data Mining*, 7(1), 106-126. https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/23986

[10] Zhu, T., Kuang, L., Piao, C., Zeng, J., Li, K., & Georgiou, P. (2024). Population-specific glucose prediction in diabetes care with transformer-based deep learning on the edge. *IEEE Transactions on Biomedical Circuits and Systems, 18*(1), 236–246. https://doi.org/10.1109/TBCAS.2023.3348844

[11] Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179-211. https://doi.org/10.1207/s15516709cog1402_1

[12] Ghojogh, B., & Ghodsi, A. (2023). Recurrent neural networks and long short-term memory networks: Tutorial and survey. arXiv. http://arxiv.org/abs/2304.11461

[13] Dey, R., & Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems* (pp. 1597–1600). https://doi.org/10.1109/mwscas.2017.8053243

[14] Cichosz, S. L., Kronborg, T., Jensen, M. H., & Hejlesen, O. (2021). Penalty weighted glucose prediction models could lead to better clinically usage. *Computers in Biology and Medicine, 138*, Article 104865. https://doi.org/10.1016/j.compbiomed.2021.104865

[15] Zhu, T., Kuang, L., Daniels, J., Herrero, P., Li, K., & Georgiou, P. (2023). IoMT-enabled real-time blood glucose prediction with deep learning and edge computing. *IEEE Internet of Things Journal, 10*(4), 3706–3719. https://doi.org/10.1109/JIOT.2022.3143375

[16] Lim, B., Arık, S., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting, 37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

[17] Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9915 LNCS, pp. 47–54). https://doi.org/10.1007/978-3-319-49409-8_7

[18] Chen, J., Lv, T., Cai, S., Song, L., & Yin, S. (2023). A novel detection model for abnormal network traffic based on bidirectional temporal convolutional network. *Information and Software Technology, 157*, Article 107166. https://doi.org/10.1016/j.infsof.2023.107166

[19] Challu, C. (2023). NHITS: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(6), 6989–6997. https://doi.org/10.1609/aaai.v37i6.25854

[20] Rangapuram, S. S., Gasthaus, J., Stella, L., Flunkert, V., Salinas, D., Wang, Y., & Januschowski, T. (2023). Deep non-parametric time series forecaster. arXiv. http://arxiv.org/abs/2312.14657

[21] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). KAN: Kolmogorov-Arnold networks. arXiv. http://arxiv.org/abs/2404.19756

[22] Amit, M. L. (2026, May). Performance Comparison of Combined LSTM-MLP with Traditional LSTM and MLP in Hand Gesture Recognition. In *2026 22nd IEEE International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 96-101). IEEE. https://doi.org/10.1109/cspa68262.2026.11517878

[23] Yu, R., Yu, W., & Wang, X. (2024). Kan or mlp: A fairer comparison. *arXiv preprint arXiv:2407.16674*. https://doi.org/10.48550/arXiv.2407.16674

[24] Pendyala, V. S., & Venkatachalam, N. (2025). The effectiveness of Kolmogorov–Arnold networks in the healthcare domain (pp. 1–23). https://doi.org/10.3390/app15169023

[25] Sergazinov, R., Chun, E., Rogovchenko, V., Fernandes, N., Kasman, N., & Gaynanova, I. (2024). GlucoBench: Curated list of continuous glucose monitoring datasets with prediction benchmarks. arXiv. http://arxiv.org/abs/2410.05780

[26] Mosquera-Lopez, C., & Jacobs, P. G. (2022). Incorporating glucose variability into glucose forecasting accuracy assessment using the new glucose variability impact index and the prediction consistency index: An LSTM case example. *Journal of Diabetes Science and Technology, 16*(1), 7–18. https://doi.org/10.1177/19322968211042621

[27] Zhu, T., Chen, T., Kuangt, L., Zeng, J., Li, K., & Georgiou, P. (2023, May). Edge-based temporal fusion transformer for multi-horizon blood glucose prediction. In 2023 IEEE International symposium on circuits and systems (ISCAS) (pp. 1-5). IEEE. https://doi.org/10.1109/ISCAS46773.2023.10181448

[28] Lara-Abelenda, F. J., Chushig-Muzo, D., Peiro-Corbacho, P., Wägner, A. M., Granja, C., & Soguero-Ruiz, C. (2025). Personalized glucose forecasting for people with type 1 diabetes using large language models. *Computer Methods and Programs in Biomedicine, 265*, Article 108737. https://doi.org/10.1016/j.cmpb.2025.108737

[29] He, M., Gu, W., Kong, Y., Zhang, L., Spanos, C. J., & Mosalam, K. M. (2020). CausalBG: Causal recurrent neural network for the blood glucose inference with IoT platform. *IEEE Internet of Things Journal, 7*(1), 598–610. https://doi.org/10.1109/JIOT.2019.2946693

[30] Zhu, T., Uduku, C., Li, K., Herrero, P., Oliver, N., & Georgiou, P. (2022). Enhancing self-management in type 1 diabetes with wearables and deep learning. *NPJ Digital Medicine, 5*, Article 1–11. https://doi.org/10.1038/s41746-022-00626-5

[31] Yetisen, A. K., Martinez-Hurtado, J. L., da Cruz Vasconcellos, F., Simsekler, M. C. E., Akram, M. S., & Lowe, C. R. (2014). The regulation of mobile medical applications. *Lab on a Chip, 14*(5), 833–840. https://doi.org/10.1039/c3lc51235e

[32] Trifan, A., Oliveira, M., & Oliveira, J. L. (2019). Passive sensing of health outcomes through smartphones: Systematic review of current solutions and possible limitations. *JMIR mHealth and uHealth, 7*(4), Article e12649. https://doi.org/10.2196/12649

[33] Benhamou, P.-Y., Franc, S., Reznik, Y., Thivolet, C., Schaepelynck, P., Renard, E., Guerci, B., Chaillous, L., Lukas-Croisier, C., Jeandidier, N., & others. (2019). Closed-loop insulin delivery in adults with type 1 diabetes in real-life conditions: A 12-week multicentre, open-label randomised controlled crossover trial. *The Lancet Digital Health, 1*(1), e17–e25. https://doi.org/10.1016/s2589-7500(19)30003-2

[34] D'Antoni, F., Petrosino, L., Sgarro, F., Pagano, A., Vollero, L., Piemonte, V., & Merone, M. (2022). Prediction of glucose concentration in children with type 1 diabetes using neural networks: An edge computing application. *Bioengineering, 9*(4), Article 183. https://doi.org/10.3390/bioengineering9050183

[35] Klonoff, D. C., Edin, F., & Aimbe, F. (2017). Fog computing and edge computing architectures for processing data from diabetes devices connected to the medical internet of things. https://doi.org/10.1177/1932296817717007

[36] Rodríguez-Rodríguez, I., Campo-Valera, M., Rodríguez, J. L. M., & Frisa-Rubio, A. (2023). Constrained IoT-based machine learning for accurate glycemia forecasting in type 1 diabetes patients. *Sensors, 23*(7), Article 3665. https://doi.org/10.3390/s23073665

[37] Wang, W., Miller, D. A., Price, H. B., Yang, X., Brown, W. J., & Wax, A. (2025). High-performance, low-cost optical coherence tomography system using a Jetson Orin Nano for real-time control and image processing (pp. 1–9). https://doi.org/10.1167/tvst.14.3.24

[38] Hakim, A. A., Juanara, E., & Rispandi, R. (2023). Mask detection system with computer vision-based on CNN and YOLO method using Nvidia Jetson Nano. *Journal of Information Systems Exploration and Research, 1*(2), 109–122. https://doi.org/10.52465/joiser.v1i2.175

[39] Zhao, Q., Zhu, J., Shen, X., Lin, C., Zhang, Y., Liang, Y., Cao, B., Li, J., Liu, X., Rao, W., & Wang, C. (2023). Chinese diabetes datasets for data-driven machine learning. *Scientific Data, 10*, Article 1–8. https://doi.org/10.1038/s41597-023-01940-7

[40] Hakim, A. A., Durai, S. S., Mahmud, F., Soon, F., Tukiran, Z., & Setiawan, R. (2025). Exploratory anomaly detection with blood glucose level time series prediction. *International Journal of Innovation in Engineering, 17*, 292–303. https://penerbit.uthm.edu.my/ojs/index.php/ijie/article/view/22993

[41] Nixtla. (2022). NeuralForecast: User friendly state-of-the-art neural forecasting models [Computer software]. GitHub. https://github.com/Nixtla/neuralforecast

[42] Hakim, A. A., Mahmud, F., & Morsin, M. (2024). Significant of blood glucose time-series forecasting with temporal fusion transformer model for type 1 diabetes. *Journal of Advanced Research in Applied Sciences and Engineering Technology, 4*, 312–324. https://doi.org/10.37934/araset.60.2.312324

[43] Kim, D. Y., Choi, D. S., Kang, A. R., Woo, J., Han, Y., Chun, S. W., & Kim, J. (2022). Intelligent ensemble deep learning system for blood glucose prediction using genetic algorithms. *Complexity, 2022*, Article 7902418. https://doi.org/10.1155/2022/7902418

[44] Nguyen, T. A., & Nguyen, T. D. (2026). A Study on the Impact of Hyperparameters on the Temporal Convolutional Network Model for Electric Load Forecasting. International Journal of Robotics and Control Systems, 6(1), 741-761. https://doi.org/10.31763/ijrcs.v6i1.2532

[45] Hakim, A. A., Mahmud, F., & Morsin, M. (2024). Assessment of deep learning model system for blood glucose time-series prediction. *Journal of Science and Technology, 16*, 65–75. https://doi.org/10.30880/jst.2024.16.01.007

[46] Martin, I., Jose, S., Oscar, M. B., & German, V. (2025). Evaluating and accelerating vision transformers on GPU-based embedded edge AI systems. *The Journal of Supercomputing*, 1–21. https://doi.org/10.1007/s11227-024-06807-1

[47] Zhu, T., Li, K., Herrero, P., & Georgiou, P. (2021). Deep learning for diabetes: A systematic review. *IEEE Journal of Biomedical and Health Informatics, 25*(8), 2744–2757. https://doi.org/10.1109/JBHI.2020.3040225

[48] Nemat, H., Khadem, H., Elliott, J., & Benaissa, M. (2024). Data-driven blood glucose level prediction in type 1 diabetes: A comprehensive comparative analysis. *Scientific Reports, 14*, Article 21863. https://doi.org/10.1038/s41598-024-70277-x

[49] Moon, K., Kim, J., Yoo, S., & Cho, J. (2025). Personalized blood glucose prediction in type 1 diabetes using meta-learning with bidirectional long short term memory-transformer hybrid model. *Scientific Reports, 15*, Article 30636. https://doi.org/10.1038/s41598-025-13491-5

[50] Rajagopal, S., & Thangarasu, N. (2023). A novel hybrid deep learning model for blood glucose prediction with extended prediction horizons in type-1 diabetic patients. In *2023 International Conference on Emerging Research in Computer Science* (pp. 1–7). https://doi.org/10.1109/ICERCS57948.2023.10434032

[51] Zhu, B. (2025). SugarNet: Personalized blood glucose forecast with a Fourier Kolmogorov–Arnold network. *Expert Systems with Applications, 280*, Article 127476. https://doi.org/10.1016/j.eswa.2025.127476

[52] International Organization for Standardization. (2015). *In vitro diagnostic test systems: requirements for blood-glucose monitoring systems for self-testing in managing diabetes mellitus*. International Organization for Standardization. https://doi.org/10.3403/30208526u

[53] Yon Hin, B., Bueno, I., Lowe, C. R., & Jones, S. (2017). Clinical accuracy study of an GDH-NAD blood glucose monitoring system using the performance criteria of ISO 15197: 2013. *Journal of Diabetes Science and Technology*, *11*(2), 444-445. https://doi.org/10.1177/1932296816664362