Biocybernetics and Biomedical Engineering 44 (2024) 481–500

Contents lists available at ScienceDirect

Biocybernetics and Biomedical Engineering

journal homepage: www.elsevier.com/locate/bbe

Advancing blood glucose prediction with neural architecture search and deep reinforcement learning for type 1 diabetics

Peter Domanski [a,1], Aritra Ray [b,*,1], Kyle Lafata [b], Farshad Firouzi [b], Krishnendu Chakrabarty [c], Dirk Pflüger [a]

[a] University of Stuttgart, Universitätsstraße 38, Stuttgart, 70569, Baden-Württemberg, Germany
[b] Duke University, 534 Research Drive, Durham, 27705, NC, United States
[c] Arizona State University, 699 S Mill Avenue, Tempe, 85281, AZ, United States

ARTICLE INFO

Keywords:
Deep reinforcement learning
Neural architecture search
Blood glucose prediction
Type-1 diabetes

ABSTRACT

For individuals with Type-1 diabetes mellitus, accurate prediction of future blood glucose values is crucial to aid its regulation with insulin administration, tailored to the individual's specific needs. The authors propose a novel approach for the integration of a neural architecture search framework with deep reinforcement learning to autonomously generate and train architectures, optimized for each subject over model size and analytical prediction performance, for the blood glucose prediction task in individuals with Type-1 diabetes. The authors evaluate the proposed approach on the OhioT1DM dataset, which includes blood glucose monitoring records at 5-min intervals over 8 weeks for 12 patients with Type-1 diabetes mellitus. Prior work focused on predicting blood glucose levels in 30 and 45-min prediction horizons, equivalent to 6 and 9 data points, respectively. Compared to the previously achieved best error, the proposed method demonstrates improvements of 18.4 % and 22.5 % on average for mean absolute error in the 30-min and 45-min prediction horizons, respectively, through the proposed deep reinforcement learning framework. Using the deep reinforcement learning framework, the best-case and worst-case analytical performance measured over root mean square error and mean absolute error was obtained for subject ID 570 and subject ID 584, respectively. Models optimized for performance on the prediction task and model size were obtained after implementing neural architecture search in conjunction with deep reinforcement learning on these two extreme cases. The authors demonstrate improvements of 4.8 % using Long Short Term Memory-based architectures and 5.7 % with Gated Recurrent Units-based architectures for patient ID 570 on the analytical prediction performance by integrating neural architecture search with deep reinforcement learning framework. The patient with the lowest performance (ID 584) on the deep reinforcement learning method had an even greater performance boost, with improvements of 10.0 % and 12.6 % observed for the Long Short-Term Memory and Gated Recurrent Units, respectively. The subject-specific optimized models over performance and model size from the neural architecture search in conjunction with deep reinforcement learning had a reduction in model size which ranged from 20 to 150 times compared to the model obtained using only the deep reinforcement learning method. The smaller size, indicating a reduction in model complexity in terms of the number of trainable network parameters, was achieved without a loss in the prediction performance.

1. Introduction

Diabetes mellitus is a medical condition marked by high Blood Glucose (BG) levels [1]. If left untreated, individuals experiencing uncontrolled high BG levels are susceptible to severe health complications, escalating the risk of heart disease, diabetic retinopathy, or kidney damage [2]. The International Diabetes Federation (IDF) estimates that approximately 537 million adults aged 20 to 79 are affected by diabetes, with the year 2021 alone witnessing 6.7 million deaths attributed to diabetes [3]. Diabetes manifests in various forms, including Type-1, Type-2, gestational, pre-diabetes, monogenic, and cystic fibrosis-related diabetes [4]. The authors in this paper specifically delve into predicting future BG values based on historical BG values for individuals with Type-1 Diabetes Mellitus (T1DM). T1DM is characterized as an organ-specific autoimmune disease affecting pancreatic

* Corresponding author.
E-mail addresses: peter.domanski@ipvs.uni-stuttgart.de (P. Domanski), aritra.ray@duke.edu (A. Ray).
[1] Equal Contribution

https://doi.org/10.1016/j.bbe.2024.07.006
Received 8 February 2024; Received in revised form 22 July 2024; Accepted 29 July 2024
Available online 12 August 2024

Table 1
Performance analysis of simple recurrent networks on OhioT1DM dataset, while predicting all BG values throughout the 30 min prediction horizon, with respect to RMSE and MAE metrics.

Reference	Subjects from OhioT1DM dataset	Features	Methods	Analytical performance RMSE (PH = 30 min)
[6]	Testing: 584, 567, 596, 552, 544, 540. Training: All subjects	CGM, finger stick, basal rate, galvanic skin response, skin temp., bolus dose	LSTM	25.0
			BiLSTM	24.4
			Convolutional LSTMs	23.2
			Seq-2-Seq CNN-LSTM	21.0
			Seq-2-Seq LSTM	20.8
			Seq-2-Seq BiLSTM	20.6
[5]	All subjects	CGM	LSTM	22.13
			GRU	22.00
			WaveNet	22.49
			WaveNet+GRU	22.21
			WaveNet+LSTM	22.35
			GRU+LSTM	21.98
[14]	All subjects	CGM	Regression	19.85
			Vanilla LSTM	19.83
			BiLSTM	20.05
			Ensemble via stacking	19.63
			Ensemble via Multi-variate	19.64
			Ensemble via subsequence	19.62

β cells [1], wherein the β cells eventually lose their capacity for insulin synthesis.

The application of Recurrent Neural Networks (RNNs) for BG prediction has been extensively explored in the literature [5–15]. In the healthcare domain, where insulin delivery relies on predicted BG values, surpassing the state-of-the-art in analytical and clinical performance is of paramount importance for the advancement of T1DM care. However, RNNs such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) exhibit inherent numerical instability in time-series prediction tasks. Table 1 provides insights into the performance of simple RNNs on the OhioT1DM dataset [16] for a Prediction Horizon (PH) of 30-mins. OhioT1DM [16] includes Continuous Glucose Monitoring (CGM) data for 12 patients with T1DM, recorded at 5-min intervals over an 8-week duration. Bhimireddy et al. [6] evaluated LSTM, BiLSTM, Convolutional LSTM, Sequence to Sequence (Seq-2-Seq) LSTM, Seq-2-Seq BiLSTM, and Seq-2-Seq CNN-LSTM, considering six features for testing on a portion of the OhioT1DM dataset. Consequently, Dudukcu et al. [5] considered one feature and trained the simple LSTM to achieve a better performance than Bhimireddy et al. [6]. Thus, this paper also defined the BG prediction task as a univariate time series forecasting problem. Performance on other simple RNNs like GRUs, and WaveNets were also evaluated by Dudukcu et al. [5], wherein combining simple RNNs like GRU with LSTM led to a performance boost. Nemat et al. [14] trained a simple regression model to have comparable performance to Vanilla LSTM and BiLSTM models, while ensembling the regression, Vanilla LSTM, and BiLSTM models led to a performance boost. While the use of RNNs in time-series prediction is predominantly studied [17], the role of RNNs in combination with feature extractors [18,19] has also shown promising results, for time-series forecasting tasks. Additionally, Fu [20] discusses the application of various Reinforcement Learning (RL) algorithms, including actor-critic methods, in controlling energy consumption in buildings, highlighting their relevance to time-series prediction problems.

The authors propose a network architecture that combines RNNs with densely connected encoding-decoding networks as feature encoders within a Deep Reinforcement Learning (DRL) framework. Combining RNNs with encoding-decoding networks offers several advantages, such as improving the handling of long-term dependencies or allowing continuous, real-time learning with customization for specific individuals. Additionally, RL is robust to noise and anomalies in the data as it learns from reward signals rather than direct supervision. Furthermore, the proposed design can effectively manage varying sequence lengths, maintaining performance over extended periods where simpler or smaller models might fail. The DRL framework, introduced in [21], trains an architecture with a default number of densely connected encoding-decoding layers, LSTM layers, and RL, as outlined in Fig. 1. To elaborate, the input encoding network consists of densely connected layers that harness their representational capabilities to extract informative features for the subsequent LSTM network followed by an output encoding network. The output of this encoding network, in conjunction with the LSTM, is fed into a projection layer that generates a probability distribution across prediction values, ensuring a non-zero mean and bounded output to improve numerical stability. The authors formulated BG prediction as a DRL learning task and integrated the Soft Actor-Critic (SAC) in the DRL framework. The proposed DRL method surpassed the state-of-the-art BG prediction approach in both Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) within 30-min and 45-min PHs. Compared to the previous best error, when the proposed method was evaluated over the entire OhioT1DM dataset [16] and used only one feature for the prediction task, the proposed DRL method demonstrated improvements of 18.4% and 22.5% on average and for MAE in the 30-min and 45-min PH respectively, without additional optimization of the network's architectural structure. In the 30-min PH, patient ID 570 exhibited the best-case prediction performance. In contrast, patient ID 584 represents the worst-case scenario regarding the analytical performance measured over RMSE on the prediction task using only the DRL method.

The default setting of the DRL framework is derived from human intuition, which might lead to non-optimized network architectures. In this paper, the authors propose a Neural Architecture Search (NAS) framework, integrated with the DRL framework, which is designed to autonomously generate optimal or highly efficient neural network structures for the given BG prediction task. NAS with DRL eliminates the necessity for manual intervention by human experts in constructing network architectures based on domain knowledge and intuition. The authors propose the introduction of the NAS framework with Bayesian Optimization (BO). This provides a systemic approach for (a) designing networks optimized for analytical prediction performance on the BG prediction task, and model size for each patient, and (b) searching through a large search space of network hyperparameters to design optimized subject-specific models in terms of model complexity or the number of trainable network parameters. The authors iteratively generate candidate architectures using the NAS framework, train the architectures using DRL, and assess their performance in the BG prediction task. To accomplish this, the authors employ Optuna, a BO-based framework [22], to pinpoint architectures with low complexity that demonstrate high performance in BG prediction. In addition to the NAS, the authors incorporate GRU into the NAS space, an extension

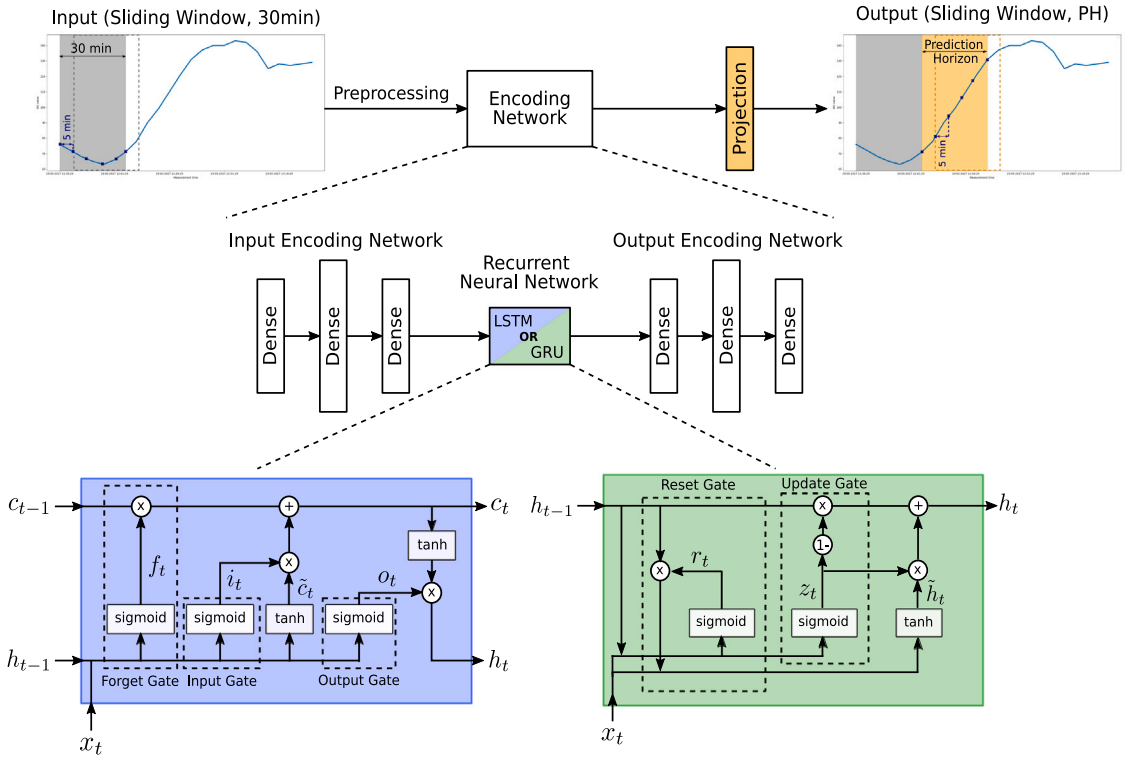


Fig. 1. Overview of the proposed RNNs with densely connected encoding networks as feature extractors and the projection layer. The BG prediction task is formulated as a DRL learning task. The DRL-based approach requires the need of human experts to manually craft network architectures based on domain knowledge and intuition.

of prior work [21] which only utilized LSTM over the DRL framework. The efficacy of the proposed method is evaluated using the OhioT1DM dataset [16]. The candidate architecture, generated by the NAS framework, is evaluated on the BG prediction task. The performance of the candidate architecture, or the prediction error, on the BG prediction task for the particular subject, is used as a reward signal for the NAS controller. The reward signal helps the NAS controller decide on how well the generated candidate architecture performed on the prediction task, thereby helping it generate a new candidate architecture to meet the specific objective. Using a subject's past six BG values as inputs, this paper predicts future BG values in the forecasting prediction horizon of 30 and 45 min, corresponding to six and nine BG values respectively.

Evaluation results prove that the method achieved additional improvements of 4.8% using LSTM-based architectures and 5.7% with GRU-based architectures for patient ID 570, by integrating BO-based NAS to the DRL framework. In the 30-min prediction horizon, patient ID 570 exhibited the best-case prediction performance, while patient ID 584 represents the worst-case scenario in terms of the analytical performance measured over RMSE on the prediction task using only the DRL method. Patient ID 584 had an even greater performance boost, with improvements of 10.0% and 12.6% observed for the LSTM and GRU, respectively with the NAS combined with DRL approach. Moreover, the BO approach excels in identifying more suitable hyperparameter sets. With these sets, the authors achieve competitive results with recent methods [21] while utilizing a fraction of trainable network parameters. Through the NAS approach, the authors achieved a decrease in model size compared to solely employing the DRL method. The subject-specific optimized models over performance and model size from the NAS in conjunction DRL had a reduction in model size which ranged from 20 to 150 times compared to the model obtained using only the DRL method. The size reduction, indicating a reduction in model complexity in terms of the number of trainable network parameters, was achieved without a loss in the prediction performance compared to the state-of-the-art [21]. The reduction varied depending on the specific patient and architecture. Compared with the state-of-the-art results [5,14,21], wherein the same number of features, and all subjects were considered for training and testing, the proposed NAS based on Bayesian optimization framework, built on top of DRL, expressed a very comparable performance in clinical settings, as elucidated in Section 5. While evaluating clinically using Surveillance Error Grid (SEG) [23], the NAS with DRL expressed only marginal changes in the none and slight risk zones, within a range of $\pm 3\%$, compared to the state-of-the-art [5]. The key contributions of the paper are as follows:

- The paper proposes a novel approach for the integration of a NAS framework with DRL to autonomously generate and train architectures for predicting BG values in individuals with Type-1 diabetes.
- The authors train the proposed method on the OhioT1DM dataset. In comparison to the baseline, the proposed Deep Reinforcement Learning method outperforms their best results by 18.5% on Mean Absolute Error and 16.3% on Root Mean Square Error for 30-min prediction horizon, and by 22.5% Mean Absolute Error and 19.5% on Root Mean Square Error for 45-min prediction horizon.
- The prediction results on best (patient ID 570) and worst case (patient ID 584) patients using only the Deep Reinforcement Learning method, were further enhanced by integrating the Bayesian optimization-based Neural Architecture Search to the Deep Reinforcement Learning framework.

The rest of the paper is organized as follows: Section 2 talks about the state-of-the-art approaches proposed in the literature. In Section 3, the authors delve deeper into our proposed methodology. The experimental settings are detailed in Section 4. The evaluation results are discussed in Section 5, while Section 6 concludes the paper. The experimental results can be reproduced from the GitHub repository.[2]

[2] https://github.com/SC-SGS/Optimization-of-DRL-based-BG-Prediction.git

2. Related prior work

Numerous studies have been conducted to predict BG for patients diagnosed with T1DM [5–15,24]. Several external factors like insulin and carbohydrate intake, exhaustion from physical activity, and stress influence the fluctuation of BG levels in individuals with T1DM [25]. Accurate prediction of future BG levels helps in regulating insulin intake and thus helps to avoid serious health complications. To predict BG in T1DM, several techniques have been proposed in the literature that involve both Machine Learning (ML) and Deep Learning (DL) approaches [6,26,27]. The effectiveness of the devised techniques is evaluated using datasets obtained from individuals with T1DM, as well as through dataset from simulators. Development of simulators for T1DM is also an active area of research. The subsequent subsections discuss simulators, datasets, and concentrate on pertinent literature concerning the OhioT1DM [16] dataset.

2.1. Simulators and datasets for Type-1 diabetes

Simulation environments such as the Diabetes Mellitus Metabolic Simulator for Research (DMMS.R) [28] are designed for diabetes research. DMMS.R facilitates clinical studies on virtual subjects with Type- 1, 2, or pre-diabetes, providing in silico environments for testing interventions in diabetes treatment and monitoring, including modeling devices and exploring treatment protocols. It can be leveraged to create device models, apply comprehensive test protocols, analyze subject responses to drug treatments, and evaluate sensor and pump performance in the in silico environment before clinical trials. Moving on to the next, the Padova T1DM Simulator [29] was developed to model the glucose-insulin system during a meal based on closed-loop control [30]. It relates plasma concentrations of glucose and insulin to various glucose and insulin fluxes, incorporating a 2-compartment model for the glucose subsystem. This simulator has been further enhanced to account for nonlinear increases in insulin-dependent utilization during hypoglycemia and includes models for glucagon kinetics, and secretion [29]. Additional simulation tools for modeling T1DM BG levels include the Web-based Simulation Tool [31], Simglucose [32], and GluCoEnv [33].

Predictions on *data from T1DM* like [16,34–40] also contribute to a significant portion of the relevant literature. OhioT1DM [16] includes an eight-week dataset capturing the CGM values, insulin records, physiological sensor data, and self-reported life events for each of 12 individuals with T1DM. The OAPS [34] data collection, initiated in 2015, strives to improve accessibility and transparency of Artificial Pancreas System technology for individuals with T1DM [12]. Participants voluntarily share their data, encompassing CGM-recorded glucose levels, insulin rates, carbohydrate intake, physical activity, and other physiological details. Tidepool [35,41] is another substantial collection of CGM values and insulin pump data, encompassing around 100 participants with a total of 152 million data instances [42].

2.2. Relevant literature on the OhioT1DM dataset

In 2018, Zhu et al. [27] explored the combination of causal dilated CNN layers with fast WaveNet for predicting BG on the OhioT1DM dataset [16], for six subjects, using four different features. In 2020, online Auto-Regressive Integrated Moving Average (ARIMA) with residual compensation network was proposed by Ma et al. [43]. Daniels et al. [44] proposed the use of deep multitask networks and convolutional RNNs, while dilated RNNs in the context of BG prediction for T1DM individuals were explored by Zhu et al. [7]. In the same year, deep residual networks, latent variable-based statistical modeling, and shallow neural networks was explored by Rubin-Falcone et al. [42], Sun et al. [45], and Pavan et al. [46] respectively, on the OhioT1DM dataset [16]. Multi-scale LSTM with multi-lag structures, Convolutional RNN (CRNN), Neural Networks (NN) and RNN in conjunction with attention mechanisms were proposed by Yang et al. [8], Freiburghaus et al. [9], and Bevan et al. [10] respectively. In the same year, a neural physiological encoder with LSTM, Seq-2-Seq models with RNNs, knowledge distillation via RNN and Generative Adversarial Networks (GAN) for time-series prediction on the OhioT1DM [16] were explored by Gu et al. [11], Bhimireddy et al. [6], Hameed et al. [12], and Zhu et al. [47] respectively.

In the following year, combinations of LSTM, GRU, WaveNets combined in a weighted decision-level fusion, GRU-based consolidated versus training models for individual subjects of the OhioT1DM dataset [16], and recurrent self-attention network was explored by Dudukcu et al. [5], Dudukcu et al. [24], and Cui et al. [13] respectively. In 2022, Nemat et al. [14] explored ensembling of simple regression model, vanilla LSTM and Bi-directional LSTM (BiLSTM). In 2023, Shuvo et al. [15] explored deep multi-task learning with stacked LSTM for BG prediction task. Chronologically listed in Table 2 is a comprehensive overview of the prediction methods, preprocessing steps, features utilized, and performance estimation techniques proposed in the literature for the OhioT1DM dataset [16].

In general, throughout the literature, the popular metrics for analytical performance estimation on the prediction task were RMSE and MAE, while SEG [23] and Clarkes' Error Grid (CEG) [48] were the most commonly used metrics for evaluating clinical accuracy. The preprocessing was quite diverse, with some prior work considering interpolation methods [7,11,14,15,27,43–45,47] and imputation methods [6,12] to account for missing data. Depending on the research timeline (6 more subjects were incorporated into the OhioT1DM dataset [16] in 2020), the literature considers either 6 [7–11,27,42–46] or 12 subjects [5,13–15,24]. Some work [7,11,12,42] opted to harness additional datasets like Padova T1DM simulator [29], TidePool [35], and OAPS [34] dataset to bolster the prediction performance. Additionally, commonly explored preprocessing techniques involves median [11,12] and Gaussian [15] filtering, as well as data normalization [5,8,46]. Regarding features employed in the prediction task, aside from CGM, carbohydrate intake, basal, and bolus insulin are frequently taken into account [7–9,11,12,42,44]. While many studies used multiple features for the prediction task, this paper consider only past CGM values to predict future BG values, and evaluate over the entire OhioT1DM [16] dataset, consistent with [5,14].

In this work, the authors propose a novel method of BO-based NAS in conjunction with DRL, encompassing a dense encoder–decoder network in combination with LSTM or GRU cells. The authors propose z-score normalization on the training set and apply the normalization score obtained from the training set to the test dataset, as the only pre-processing step while considering only CGM values. The authors evaluate the proposed method analytically using RMSE and MAE metrics, and clinically via SEG and CEG. The results are compared against the state-of-the-art, which involves weighted decision level fusion across LSTM, WaveNet, and GRU [5] for the prediction task.

3. Methodology

The authors propose the design of a DRL architecture in conjunction with a NAS framework to automatically derive architectures for predicting BG values for individuals affected with T1DM, as presented in Fig. 2. The input encoding network consists of densely connected layers that harness their representational capabilities to extract informative features for the subsequent LSTM network. This is followed by an output encoding network. The output of this encoding network is fed into a projection layer that generates a probability distribution across prediction values, ensuring a non-zero mean and bounded output to improve numerical stability. The authors formulated BG prediction as a DRL learning task, and integrated the Soft Actor-Critic (SAC) to the DRL algorithm. A major drawback of using only DRL based approach is the need of human experts to manually craft network architectures based on domain knowledge and intuition. This indicates that the architecture for the BG prediction models is built with a standard configuration

Table 2
Summary of related research on Type-1 diabetes prediction for OhioT1DM Dataset.

Publication year	Reference	Additional dataset	Subjects from OhioT1DM dataset	Features	Methods	Preprocessing	Performance metrics
2018	[27]	×	559, 563, 570, 575, 588, 591	CGM, insulin, carbohydrate intake, normalized time index	casual dilated CNN layers & fast WaveNet	Interpolation, extrapolation combination, Filtering	RMSE
2020	[43]	×	540, 544, 552, 567, 584, 596	CGM	Online ARMA & Residual Compensation Network	Extrapolation, Backward induction	RMSE, MAE
2020	[44]	×	540, 544, 552, 567, 584, 596	CGM, insulin bolus, carbohydrate intake, exercise	Deep Multitask Networks & Convolutional RNN	Interpolation, Extrapolation, Standardization	RMSE, MAE
2020	[7]	UVA/Padova T1DM Simulator [29]	591, 570, 563, 559, 588, 575	CGM, insulin doses, carbohydrate intake	Dilated RNN	Interpolation, Extrapolation, Filtering, Combination	RMSE
2020	[42]	TidePool [35]	567, 544, 552, 596, 540, 584	CGM, bolus insulin, Finger Stick glucose, carbohydrate intake, sine & cosine of time, CGM missingness indicator	Deep Residual Time-Series forecasting	encode time to sine, cosine embedding, variable of missingness, resampling, missing values as zero	RMSE, MAE Clarke error grid
2020	[45]	×	540, 544, 552, 567, 584, 596	CGM, basal insulin, bolus insulin, insulin on board	latent variable based statistical modeling	Interpolation, statistical modeling of glucose dynamics for missing values	RMSE, MAE Clarke error grid
2020	[46]	×	540, 544, 552, 567, 584, 596	CGM, Insulin on board, carbohydrate on board, slope of CGM	Shallow Neural Network Error imputation module	Normalization by mean & standard deviation	RMSE, MAE, COD, delay
2020	[8]	×	540, 544, 552, 567, 584, 596	CGM, basal insulin, bolus insulin, carbohydrate intake, timestamp	multi-scale LSTM with multi-lag structure	Data alignment, Outlier detection & reconciliation, extrapolation, data normalization	RMSE, MAE
2020	[9]	×	540, 544, 552, 567, 584, 596	CGM, basal insulin, bolus insulin, carbohydrate intake	CRNN	Data alignment, imputation, interpolation, resampling	RMSE, MAE
2020	[10]	×	540, 544, 552, 567, 584, 596	CGM	linear model, NN, RNN, RNN + attention mechanisms	Standardization, replace missing values with zero	RMSE, MAE
2020	[11]	TidePool [35]	559, 563, 570 575, 588, 591	CGM, basal insulin, bolus insulin, carbohydrate intake	Neural Physiological Encoder, LSTM	interpolation, extrapolation median filtering	RMSE
2020	[6]	×	Testing: 584, 567, 596, 552, 544, 540, Training: 12 subjects	CGM, finger stick, basal rate, galvanic skin response, skin temp., bolus dose	LSTM, BiLSTM, Convolutional LSTMs, TCN, & sequence-to-sequence models	Imputation, Resampling, Parameter merging	RMSE, MAE
2020	[12]	OAPS [34]	Training: All 12 subjects, Testing: 540, 544, 552, 567, 584, 596 subjects. Validation: 559, 563, 570, 575, 588, 591 subjects	CGM values, insulin basal rate, bolus amount, carbs intake & difference between consecutive CGM	Knowledge Distillation: RNN	Imputation, Median Filtering	RMSE, MAE
2020	[47]	×	Pre-train: 559, 563, 570, 575, 588, 591 subjects. Testing: 540, 544, 552, 567, 584, 596 subjects	CGM, insulin, meal, work, sleep, psychological stress, physical exercise	GAN, Gated RNN, CNN	Interpolation, Extrapolation, Data resolution matching	RMSE, MAE Clarke error grid
2021	[5]	×	All 12 subjects	CGM	LSTM, WaveNet, GRU & weighted decision-level fusion	Normalization, missing data intervals dropped	SEG, RMSE, MSA, MAPE, RMSPE
2021	[24]	×	All 12 subjects	CGM	GRU-based consolidated & individual training	Missing data intervals dropped	RMSE, MAE, MAPE, RMSPE
2021	[13]	×	All 12 subjects	basal rate, bolus intake, carbohydrate intake, CGM	Recurrent Self-Attention Network	Missing data intervals dropped	RMSE, T-test
2022	[14]	×	All 12 subjects	CGM	Regression, Vanilla LSTM, BiLSTM, Stacking, Multivariate, Subsequence ensemble	Interpolation, data scaling, reframing for supervised learning task	RMSE, MAE, MCC, SEG, Friedman test, Wilcoxon test, CDD
2023	[15]	×	All 12 subjects	CGM, Finger stick glucose, bolus insulin, carbohydrate intake	Deep Multi-task learning, Stacked LSTM	Interpolation, Missing data intervals dropped, Extrapolation, Gaussian filtering	RMSE, MAE, Clarke error grid
2024	Our work	×	All 12 subjects	CGM	NAS, DRL, Pruning, LSTM, GRU	z-score normalization	RMSE, MAE, SEG

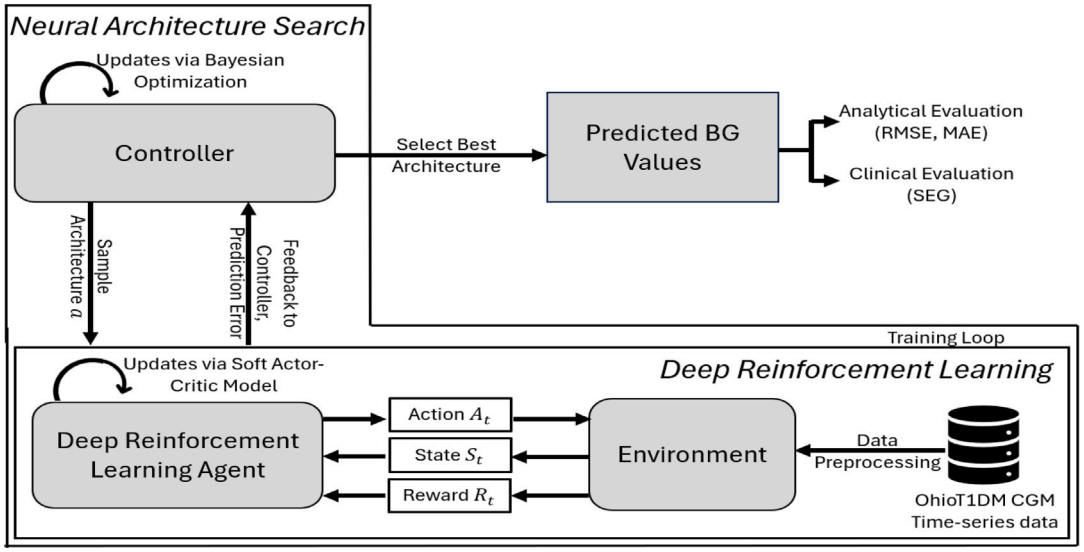


Fig. 2. Overview of the proposed approach. The Neural Architecture Search samples an architecture from the search space, trains it using the Deep Reinforcement Learning loop, and receives the prediction error on the train dataset as feedback. The Deep Reinforcement Learning agent and Neural Architecture Search are updated using the Soft-actor critic model and Bayesian optimization framework, respectively.

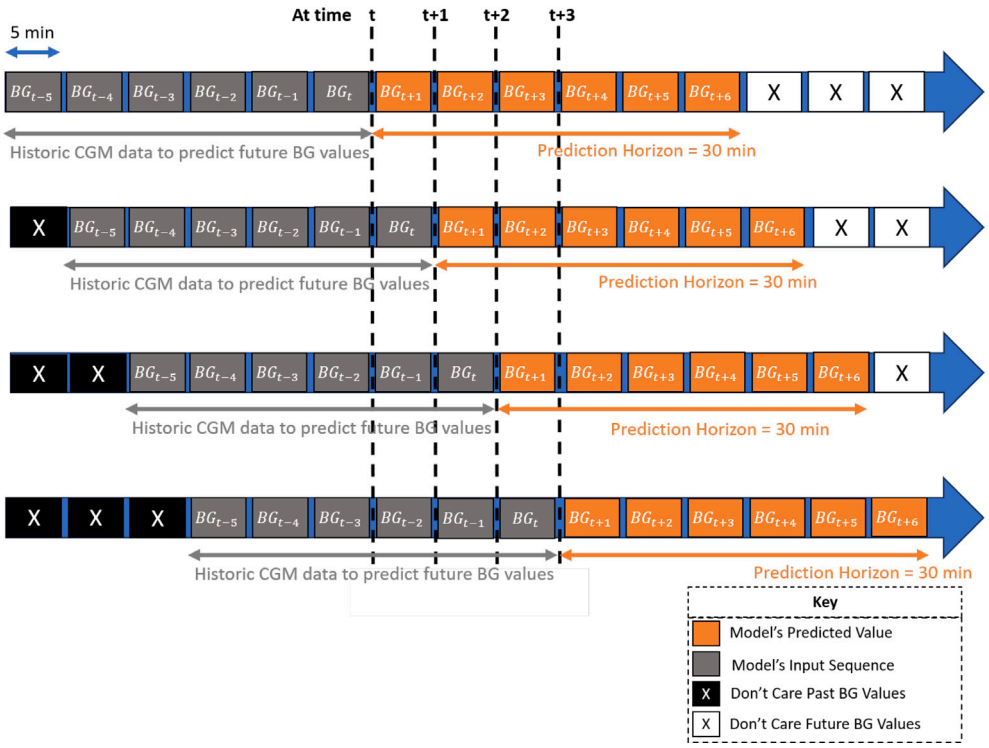


Fig. 3. Using a subject's past six BG values as inputs, the proposed method predicts future BG values in the forecasting prediction horizon of 30 and 45 min, corresponding to six (as shown in this fig.) and nine BG values respectively, in a moving window fashion.

of densely connected encoding-decoding layers, LSTM layers, and RL parameter settings. However, relying solely on human intuition for this default setup may result in sub-optimal network architecture. In this work, the introduction of the NAS framework alongside BO offers a systematic method for (a) crafting tailored networks optimized for predicting future BG levels for individual patients, and (b) efficiently navigating through a vast search space to converge on a patient-specific optimal model. The NAS framework starts with the architecture proposed in prior work [21] iteratively generating candidate architectures that are optimized in network architecture, prediction performance, and model size. The candidate architecture, generated by the NAS framework, is evaluated on the BG prediction task. The performance of the candidate architecture, or the prediction error, on the BG prediction task for the particular subject, is used as a reward signal for the NAS controller. The feedback signal helps the NAS controller decide on how well the generated candidate architecture performed on the prediction task, thereby helping it generate a new candidate architecture to meet the specific objective. This work uses a BO-based framework called Optuna [22] to find architectures with a low complexity that achieve high performance in the BG prediction task, optimized in a subject-specific manner. In addition to the prior work [21] which only involved DRL with LSTM as the sole variant of the RNN, this paper includes GRUs as an additional variant of the RNNs, alongside the major contribution to the design of NAS with Bayesian optimization framework on top of the DRL. Fig. 3 elucidates the model predictions in a moving window fashion.

Consistent with the state-of-the-art baseline [5], this work uses only CGM values for the prediction task. Using a patient's past six BG values as inputs, we predict future BG values in the forecasting prediction horizon of 30 and 45 min, corresponding to six and nine BG values respectively. The models are trained and evaluated on all 12 subjects. In the 30-min prediction horizon, patient ID 570 exhibited the best-case prediction performance, while patient ID 584 represents the worst-case scenario in terms of the analytical performance measured over RMSE on the prediction task using only the DRL method. Therefore, the selection of patient ID 570 and ID 584 for the NAS with DRL experiments is based on their representation of extreme cases. By focusing on these two scenarios, the authors aim to comprehensively assess the effectiveness of the NAS with DRL approach in improving prediction accuracy across a wide spectrum of patient scenarios. Comprehensive details regarding the network architecture are provided in the following sections.

3.1. Deep neural networks

Prior work identified DNNs as promising models for BG prediction [27,44] on the OhioT1DM dataset [16]. Variants of the same including CNN [27] and CRNN [9] were also studied. In our work, we combine densely connected neural networks with LSTM or GRU, within a DRL framework. Diverging from other studies, we propose a DRL and NAS approach over supervised learning, thereby adapting both the architecture search space and strategy to aid the training process.

3.1.1. Densely connected encoding networks

Densely connected networks are a type of neural network architecture where each neuron in one layer is linked to every neuron in the subsequent layer. Reasons for utilizing these networks include their high representational power, flexibility, and effective feature learning, incorporating multiple levels of abstraction. In several applications, they have proven to be effective when integrated with convolutional networks, serving as feature extractors, as exemplified in architectures like DenseNet [49] and similar other models.

Encoding networks: The input encoding network comprises a densely connected neural network with three layers (256, 512, and 256 neurons per layer with ReLU as the activation function). This architecture leverages their representational power to extract informative features for the subsequent LSTM or GRU network. Likewise, the output encoding networks capture features from the LSTM or GRU for the final BG prediction through the projection layer.

Projection layer: It performs the prediction of the BG values. The size of the (densely connected) layer aligns with the number of time

steps in the prediction horizon. Employing a tanh normal projection, the projection layer transforms the output to create a normal distribution with a mean and standard deviation, which represents the probability distribution across prediction values (referred to as actions) in the DRL framework, as discussed in Section 3.2. The use of tanh normal projection ensures an efficient distribution with a non-zero mean, promoting bounded output and enhancing numerical stability.

3.1.2. Long short-term memory

An RNN variant specifically crafted for handling sequential data, such as time-series data, employs a feedback mechanism (referred to as memory) to incorporate prior network outputs for current predictions. Illustrated in Fig. 1, the LSTM structure features multiple gates (input, forget, and output gates) regulating information flow through time within the memory [50]. The LSTM cell processes the input x_t at time step t, the previous cell memory c_{t-1}, and the previous output h_{t-1} of the cell to produce the outputs c_t and h_t respectively. The input gate in (1) and the forget gate in (2) govern the cell state c_t, as seen in (4), wherein the operator $\odot$ denotes the Hadamard product. Along with the outcome of the output gate in (3), the hidden state h_t of the cell is determined in (5).

$$i_t = \text{sigmoid}\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{1}$$

$$f_t = \text{sigmoid}\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2}$$

$$o_t = \text{sigmoid}\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{4}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{5}$$

The presence of these gates and the recurrent structure within the network enables us to mitigate the vanishing gradients problem and enhance the learning of long-term dependencies, as discussed in [51].

3.1.3. Gated recurrent unit

A GRU is similar in structure to LSTMs but without a forget gate, thus consisting only of a reset and an update gate [52]. Therefore, GRUs have fewer trainable parameters than LSTMs and are widely used in tasks involving natural language processing, speech recognition, and time series prediction. The general structure of a GRU cell is given in Fig. 1. As shown, the input at a given time t is x_t, and the output of the previous time step is h_{t-1}. The resulting hidden state h_t is calculated as given in Eqs. (8)–(9), including the calculations of the reset and update gates in Eqs. (6)–(7). In the following equations W_z, U_z, and b_z are (trainable) parameter matrices or vectors, respectively.

$$z_t = \text{sigmoid}\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{6}$$

$$r_t = \text{sigmoid}\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{7}$$

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{8}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{9}$$

Despite the widespread adoption of GRUs, the effectiveness of GRUs and LSTMs depends on the specific application [53].

3.2. Deep reinforcement learning

DRL merges the representational capabilities of DL with the decision-making proficiency of RL. It constitutes a subset of ML tailored for complex, high-dimensional decision-making tasks, where an agent learns optimal behavior by interacting with its environment, receiving rewards or penalties based on the outcomes of its actions [54]. DRL tasks are conceptualized as Markov Decision Processes, defining the spaces of states $\mathcal{S}$, agent actions $\mathcal{A}$, and a scalar training signal $\mathcal{R}$ (reward). At each time step t, the agent gets a state $s_t \in S$ and chooses an action $a_t \in \mathcal{A}$ following a policy $\pi\left(a_t|s_t\right)$. Consequently, the agent receives a reward $r_t \in \mathcal{R}$ and transitions to the next state $s_{t+1} \in S$. We parameterize the policy π using a DNN and aim to maximize the cumulative reward (return) during training. Analogous to the approach in [55], we adopt Actor-Critic training algorithms. These algorithms employ a value function to predict future rewards, where the value function in Eq. (10) denotes the anticipated total (discounted) reward originating from state s, with γ representing the discount factor.

$$V(s) = \mathrm{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right] \tag{10}$$

Hereafter, our emphasis is on the Soft Actor-Critic (SAC) algorithm [56], identified as the most effective among the algorithms studied in [55].

Soft Actor-Critic: The DRL algorithm designed for continuous control tasks adheres to the Actor-Critic framework [57]. This algorithm incorporates two networks – the actor and the critic network – to learn the policy and estimate a soft value function akin to Eq. (10). The soft value function is employed to update the policy using Temporal Difference (TD) error. SAC's value function and entropy regularization enhance exploration and prevent the policy from becoming overly deterministic. Additionally, the automatic temperature tuning simplifies the hyperparameter adjustment process, promoting stability and robustness in training [56]. Nevertheless, the training process may demand more computational resources, potentially extending the training time compared to conventional supervised approaches.

In the subsequent experiments, we employ an actor and a critic network with a structure illustrated in Fig. 1. To frame the BG prediction as a DRL task, we define the state S at time t as a vector:

$$S(t) = \left[BG_{t-5}, BG_{t-4}, BG_{t-3}, BG_{t-2}, BG_{t-1}, BG_t\right]^T. \tag{11}$$

The reward is determined as described in Eq. (12), and actions are represented by continuous BG values within the range of 35 to 500 mg/dL for a given prediction horizon. Throughout the training process, the agent generates continuous BG values from 35 to 500 based on the observed state of the environment.

$$R_{t+1} = -|BG_t - a_t| \tag{12}$$

The reward is formulated in a way that it approaches zero when the output action a_t closely aligns with the ground truth BG value B_t and deviates significantly from zero (in the negative direction) otherwise.

3.3. Automated machine learning

Automated Machine Learning (AutoML) is a subfield of ML that focuses on algorithms, techniques, and systems to automate and optimize ML pipelines [58]. AutoML encompasses various areas and aspects of ML pipelines to make AI more accessible, efficient, and accountable in different applications, e.g., healthcare [59]. In this work, we focus on hyperparameter optimization, NAS, and resource efficiency, e.g., for resource-constrained environments like edge devices. In the following experiments, we use model-based Bayesian optimization for NAS to optimize resource efficiency.

3.3.1. Neural architecture search

NAS is used to automatically design optimal or highly efficient neural network structures for a given task. It aims to optimize the network parameters and topology, such as the number of layers or units per layer, to maximize the performance on the given task. The process of NAS involves multiple components where key considerations are the definition of the search space, the scoring metric to evaluate the quality of different architectures, and the search algorithm to explore the search space. Whereas the search space encodes a set of task-specific architectural hyperparameters and the scoring metric a task-specific quality measure, different search algorithms exist [60]. In this work, the author opt for BO, which is a probabilistic, model-based optimization technique that uses a surrogate model (usually a Gaussian

Table 3
Description of the OhioT1DM dataset properties [16].

Patient ID	Gender	Age range	Pump model	Sensor band	Training samples	Test samples
540	Male	20–40	630G	Empetica	11 947	2884
544	Male	40–60	630G	Empetica	10 623	2704
552	Male	20–40	630G	Empetica	9080	2352
567	Female	20–40	630G	Empetica	10 858	2377
584	Male	40–60	530G	Empetica	12 150	2653
596	Male	60–80	530G	Empetica	10 877	2731
559	Female	40–60	530G	Basis	10 796	2514
563	Male	40–60	530G	Basis	12 124	2570
570	Male	40–60	530G	Basis	10 982	2745
575	Female	40–60	530G	Basis	11 866	2590
588	Female	40–60	530G	Basis	12 640	2791
591	Female	40–60	530G	Basis	18 847	2760

process) of the objective function and a corresponding acquisition function iteratively to guide the selection of new samples in the search space. After each iteration, the surrogate model and acquisition function are updated to select the next configuration. BO is commonly used in AutoML frameworks and other applications, e.g., drug discovery. Thus, many software packages for BO are available. The authors use Optuna [22] with a Tree-structured Parzen Estimator (TPE) to perform NAS. Among other methods such as RL or genetic algorithms, BO is especially appealing to the authors as:

- BO is particularly efficient in scenarios where the evaluation of the objective function is time-consuming or expensive. This is true in our case because the DRL training needs to be performed in every iteration.
- BO is designed to find global optimum, making it less likely to get stuck in local optima.
- BO can adaptively balance exploration and exploitation in hyperparameter optimization. Additionally, BO provides room for iterative refinement.
- BO can also handle black-box and noisy objective functions.

4. Experimental setup

4.1. Dataset

This paper utilizes the widely used OhioT1DM dataset [16], a key resource in Blood Glucose Level Prediction (BGLP) research. After the 2020 BGLP Challenge, the dataset now includes data from 12 subjects managing T1DM with insulin pump therapy. Collected over eight weeks per individual, the dataset incorporates CGM, insulin, physiological sensors, and self-reported life-event data. Participants used Medtronic 530G or 630G insulin pumps and Medtronic Enlite CGM sensors, ensuring complete anonymization with randomly assigned ID numbers. Six subjects used Empetica sensor band, while the other six used Basis peak sensor to monitor real-time physiological signals. The dataset comprises 19 features, including CGM values monitored at 5-min intervals. Notably among those features are BG values from periodic self-monitoring of BG via finger sticks, insulin doses involving both bolus and basal separately, self-reported meal times with estimated carbohydrate intake, self-reported times of exercise, sleep, work, stress, and illness. Demographically, it involves 7 male and 5 female subjects, aged between 20 to 80 years. Table 3 provides a detailed overview of the dataset characteristics. Consistent with the state-of-the-art baseline [5], the authors use only CGM values for the prediction task.

4.2. Preprocessing

To standardize the data, the authors employ z-score normalization on the training set and apply the normalization score obtained from the training set to the test dataset. This method rescales the values of each feature within the data to achieve zero mean and unit variance. The computation of z-score normalization on the considered CGM feature is outlined in Eq. (13). It requires the determination of the distribution mean (μ_x) and standard deviation (σ_x) for the specific feature (x) on the training dataset.

$$x' = \frac{x - \mu_x}{\sigma_x} \tag{13}$$

The rationale behind employing z-score normalization lies in the correlation between average BG values and key indicators such as HbA1C (A1C) a metric to measure the amount of blood glucose attached to haemoglobin and conditions like hyperglycemia. In this paper, the z-score normalization is done on the training set, and the normalization score obtained from the training set is applied to the test dataset, as the only pre-processing step while considering CGM values. Z-score normalization centrally revolves around the mean, and clinically speaking, differences in A1C and mean glucose levels have been noted among different racial groups, and these variations are even more pronounced among individuals belonging to the same racial group [61, 62]. Additionally, this normalization method prioritizes identifying potentially risky fluctuations in BG values while disregarding statistical aspects in the data that do not contribute meaningful information for accurate CGM prediction [63].

4.3. Evaluation metrics

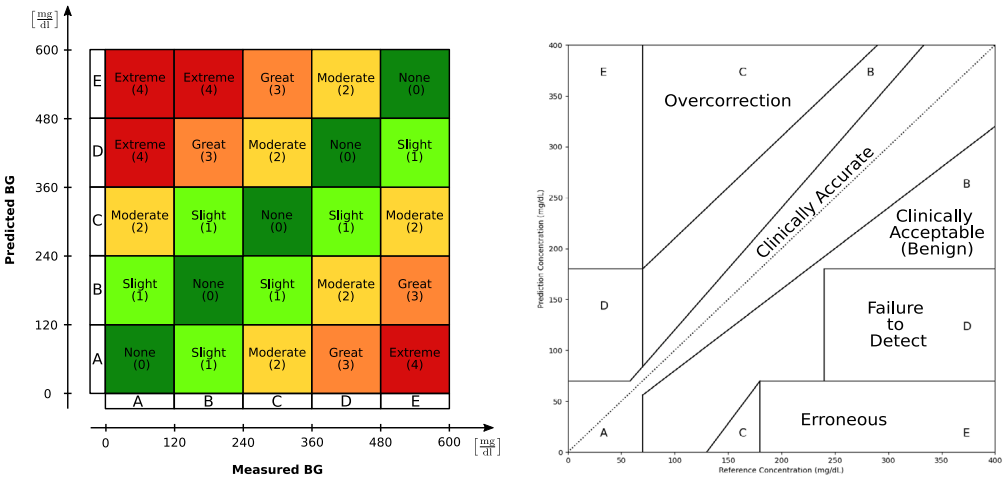
Evaluation metrics serve as tools for quantifying the error of a system. Numerous metrics are available to assess the error of BG prediction tasks. Analytical error evaluation involves quantitative approaches to depict the proximity of predictions to the ground truth. In contrast, clinical error serves as a qualitative measure, assessing the clinical outcome of the prediction results. As such, its definition encompasses statistical metrics and the expertise of clinicians.

Analytical error: The primary numerical metric employed is the RMSE, as outlined in Eq. (14). Furthermore, for assessing the analytical error of BG prediction in this study, the MAE, detailed out in Eq. (15) is also used.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{\mathrm{predicted}} - y_{\mathrm{measured}}\right)^2} \tag{14}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_{\mathrm{predicted}} - y_{\mathrm{measured}}\right| \tag{15}$$

Clinical error: The SEG [23] is a recently adopted grid-based visualization method for assessing clinical error. It serves as a metric for both error and clinical risk evaluation in BG measurements [23]. Generally, the error grid displays a series of risk zones with assigned scores (risk levels) reflecting clinical impact, ranging from 0 (none) to 4 (extreme). In Fig. 4(a), a simplified SEG [23] is presented, discretely structured with limits spanning from 0 to 600 mg/dL and risk zones separated by 120 mg/dL intervals. The predictions are overlaid on a continuously



(a) Visualization of SEG risk zones. (b) Visualization of CEG risk zones.

Fig. 4. Visualization of risk zones for two different metrics to evaluate the performance of the predicted BG values in clinical settings.

color-coded SEG. The color indicates the average risk rating determined by clinician respondents in a survey [23]. It represents the collective decision of experts, reflecting the mean group consensus. Furthermore, CEG [48], which was developed in 1987 to assess the clinical risk, is used by the authors to assess the clinical risk for the predicted BG values to the ground truth in the test set. The visualization of the CEG risk zones is presented in Fig. 4(b).

4.4. Automated machine learning

The authors implement a fully automated AutoML pipeline using model-based Bayesian optimization in Optuna. It focuses on optimizing the hyperparameter and topological structures of model architectures. Still, the proposed framework supports the manual optimization of hyperparameters or to apply random search as in previous works [21].

Neural Architecture Search: Hyperparameter optimization is a fundamental component of AutoML and usually plays a crucial role in the context of (automatic) training and fine-tuning ML models. Therefore, applying models successfully to a given task requires a suitable selection of hyperparameters. However, the hyperparameters are not learned but set prior to training. They control various aspects of the model's behavior, including the structure and properties of architectures or algorithms, and thus are suitable for NAS. The search space of hyperparameters for NAS is given in Table 4. The hyperparameters can be categorized in two different ways. First, in the proposed implementation, the authors differ the parameters for the actor and critic network, which are both part of the SAC algorithm. Second, this work categorizes the parameters as those that affect the model's topology (network structures) and those that influence the training process.

The authors define the hyperparameter such that the symmetrical network structure of the encoding networks shown in Fig. 1 is preserved. In this work, the authors extend the approach of manually applying a random search for hyperparameter optimization and replace it with an automatized, model-based Bayesian methodology using Optuna. Therefore, the authors define the objective metric as given in Eq. (16).

$$f_{\text{NAS}} = \frac{\text{RMSE}_{\text{train, NAS}}}{\text{RMSE}_{\text{train, initial}}} + \frac{\text{model size}}{\text{initial model size}} \tag{16}$$

The objective function f_{NAS} is defined in a multi-objective manner where the first term aims to minimize the RMSE on the training data, and the second aims to reduce the overall model size. To balance the optimization of both terms and thus find the best models with low error and small model size, the authors normalize the terms with the initial values of the baseline. Initially, the proposed architecture (actor network) has a size of around 1.39M trainable parameters and a test RMSE error of approximately 20, depending on the patient. The authors chose the value ranges of individual hyperparameters in alignment with existing literature [21]. In the following, if not mentioned otherwise, the authors generate 100 candidate architectures for every NAS iteration, considering the time and computing expenses associated with conducting experiments. The individual candidate architectures are evaluated concerning the performance of the training data and model size to determine the best hyperparameters.

5. Results

5.1. Model performance

The proposed DRL and NAS algorithms are trained on the OhioT1DM dataset [16], which includes continuous BG level monitoring at 5-min intervals for 12 patients. During the training phase, the authors utilize only the BG recordings of each patient from the OhioT1DM dataset, in an 80–20 train-test split ratio. For each patient, training is conducted over 30-min intervals (equivalent to 6 consecutive BG level recordings), followed by a multi-step prediction for the BG level in the subsequent 30 or 45 min. The obtained results are clinically evaluated using SEG and CEG, and compared analytically to the baseline [5] in terms of RMSE and MAE. The rest of the section analyzes the prediction results for the entire dataset spanning 250 h, focusing on multi-step predictions.

Multi-Step Predictions for 30 min using DRL framework: The SAC model proposed in this paper predicts the BG level for 6 consecutive steps using a batch size of 1024 and undergoes 50,000 training steps, considering input data over a 30-min duration. The evaluation of RMSE and MAE for all 12 patients is presented in Table 5 along with the standard deviation, for predicting all values throughout the 30 min prediction interval, corresponding to 6 data points. Comparative results with prior work which uses the same number of features, and all subjects for train and test cases, including LSTM, WaveNet, GRU, ensemble methods, and a potential combination of these approaches, are highlighted in Table 6. It is evident that our proposed method consistently demonstrates significant improvement over the baseline [5] across all patients, as indicated by both metrics. On average, for all 12 patients, our proposed method enhances RMSE by 16.34% and MAE by 18.4%. Additionally, the BG prediction regression task was translated to a classification task using data labeling approach [64], for hypoglycemia, normoglycemia, and hyper-glycemia for BG values

Table 4
Description of hyperparameter search space for NAS.

Parameter		Category	Values
Cell type		Actor/Critic structure	{LSTM, GRU}
Cell size		Actor/Critic structure	{4, 8, 16, 32, 64, 128, 256}
Observation FCN	Layers	Critic structure	{1, 2, 3}
	Neurons	Critic structure	{4, 8, 16, 32, 64, 128, 256, 512}
Input FCN	Layers	Actor structure	{1, 2, 3}
	Neurons	Actor structure	{4, 8, 16, 32, 64, 128, 256, 512}
Action FCN		Critic structure	{4, 8, 16, 32, 64, 128, 256, 512}
Joint FCN	Layers	Critic structure	{1, 2, 3}
	Neurons	Critic structure	{4, 8, 16, 32, 64, 128, 256, 512}
Output FCN	Layers	Actor/Critic structure	{1, 2, 3}
	Neurons	Actor/Critic structure	{4, 8, 16, 32, 64, 128, 256, 512}
Activation function		Actor/Critic structure	{ReLU, Tanh, Sigmoid}
Target update period		Training	[1, 100]
Target update τ		Training	[0.001, 1.0]

Table 5
Evaluation of the proposed SAC-based DRL approach for all twelve subjects, while predicting BG values throughout the 30 min prediction horizon, with respect to RMSE and MAE metrics.

Patient ID	Root mean square error			Mean absolute error		
	Our proposed method	Baseline [5] (LSTM + WaveNet + GRU)	% Improvement over baseline	Our proposed method	Baseline [5] (LSTM + WaveNet + GRU)	% Improvement over baseline
540	19.25 ± 10.49	25.28	23.83	13.80 ± 9.58	18.77	12.99
544	15.28 ± 7.33	19.76	22.63	11.34 ± 6.45	14.36	12.99
552	14.49 ± 7.14	19.43	25.37	10.58 ± 6.42	14.66	27.83
559	19.37 ± 11.02	21.78	19.77	13.47 ± 9.83	15.35	12.22
563	17.77 ± 10.09	20.43	12.97	12.30 ± 9.34	14.39	14.46
567	22.20 ± 14.17	23.96	7.32	14.68 ± 12.74	17.41	12.62
570	15.80 ± 7.69	18.06	12.49	11.71 ± 6.97	12.85	8.83
575	20.37 ± 12.21	25.02	18.58	14.01 ± 11.16	16.77	16.45
584	24.30 ± 14.33	24.84	3.23	16.60 ± 12.76	18.57	10.57
588	15.02 ± 7.55	21.26	29.33	11.04 ± 6.77	15.55	29.00
591	18.09 ± 9.74	23.76	23.83	12.99 ± 8.82	18.61	28.43
596	17.31 ± 9.28	19.23	9.93	12.62 ± 8.47	13.58	7.03

BG ≤ 70 mg/dL, 70 mg/dL < CGM < 180 mg/dL, and BG ≥ 180 mg/dL respectively. For the predictions of all the BG values throughout the 30-min prediction interval, corresponding to 6 data points, the classification results are presented in Table 7. Table 8 highlights our prediction results in terms of RMSE and MAE for all patients for predicting only BG values at 30-min prediction interval.

Multi-Step Predictions for 45 min using DRL framework: The proposed SAC model predicts the BG level for 9 consecutive steps, utilizing a batch size of 1024 and undergoing 50,000 training steps, considering input data over a 30-min interval. The evaluation of RMSE and MAE for all 12 patients is detailed in Table 9. Table 10 provides a comparative analysis of the proposed method against the baseline [5], which is a combination of LSTM, WaveNet, and GRU. Notably, the proposed method exhibits significant improvement over the baseline [5], as reflected in both metrics. On average, across all 12 patients, the proposed method enhances RMSE by 19.53% and MAE by 22.5%. For a visual representation of the proposed model's fitting to the data, Fig. 5 depicts the ground truth compared to the prediction results for a representative test dataset spanning 30 h for patient IDs 570 and 584. Gaussian filtering with varying σ is used to reduce the fluctuations arising from sensor faults, connectivity problems, electrical interference, and manufacturing variability [15,65,66], see Fig. 5. Additionally, the BG prediction regression task was translated to a classification task using data labeling approach [64], for hypoglycemia, normoglycemia, and hyper-glycemia for BG values BG ≤ 70 mg/dL, 70 mg/dL < CGM < 180 mg/dL, and BG ≥ 180 mg/dL respectively. For the predictions of all the BG values throughout the 45-min prediction interval, corresponding to 9 data points, the classification results are presented in Table 11. Also, a comparative analysis of the proposed method against prior work is highlighted in Table 12, many of which use additional datasets to

Table 6
Comparison of the proposed SAC-based DRL approach with prior work which uses only historical CGM values to predict future BG values for all twelve subjects, while predicting all values throughout the 30 min prediction horizon.

Method	Avg. root mean square error	Avg. mean absolute error
LSTM [5]	22.13	16.02
BiLSTM [14]	20.05	14.29
Vanilla LSTM [14]	19.83	14.09
Ensemble via stacking [14]	19.63	13.88
Ensemble via multi-variate [14]	19.64	13.92
Ensemble via subsequences [14]	19.62	13.88
GRU-NN [24]	21.54	15.39
WaveNet [5]	22.49	16.47
GRU [5]	22.00	15.91
WaveNet + LSTM [5]	22.35	16.29
WaveNet + GRU [5]	22.21	16.15
LSTM + GRU [5]	21.98	15.86
LSTM + WaveNet + GRU [5]	21.90	15.87
Deep RL (**Proposed method**)	**18.32**	**12.93**

Table 7
Classification results of the proposed SAC-based DRL approach in terms of accuracy, precision, recall, and F1-score for hypoglycemic, normoglycemic, and hyperglycemic range, for all twelve subjects, corresponding to all predicted BG values throughout the 30 min prediction horizon.

Patient	Accuracy	Hypoglycemia			Normoglycemia			Hyperglycemia		
ID		Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score
540	0.885	0.546	0.528	0.545	0.915	0.896	0.906	0.875	0.916	0.895
544	0.918	0.243	0.273	0.257	0.920	0.963	0.941	0.946	0.845	0.893
552	0.923	0.581	0.507	0.541	0.950	0.951	0.951	0.861	0.872	0.867
559	0.909	0.671	0.662	0.667	0.921	0.922	0.922	0.909	0.908	0.908
563	0.885	0.300	0.167	0.214	0.912	0.891	0.902	0.851	0.887	0.869
567	0.869	0.879	0.437	0.584	0.870	0.946	0.906	0.864	0.807	0.834
570	0.947	0.333	0.091	0.143	0.894	0.928	0.911	0.971	0.959	0.965
575	0.896	0.704	0.496	0.582	0.914	0.922	0.918	0.884	0.913	0.898
584	0.888	0.667	0.074	0.133	0.903	0.917	0.910	0.864	0.864	0.864
588	0.895	1.000	0.250	0.400	0.876	0.936	0.905	0.920	0.849	0.883
591	0.881	0.575	0.615	0.595	0.894	0.931	0.912	0.912	0.812	0.859
596	0.900	0.635	0.446	0.524	0.897	0.976	0.935	0.942	0.718	0.815
Average	0.899	0.594	0.378	0.432	0.905	0.931	0.918	0.899	0.862	0.879

Table 8
Evaluation of the proposed SAC-based DRL approach for all twelve subjects, while predicting BG values at the 30 min prediction mark, with respect to RMSE and MAE metrics.

Patient ID	Root mean square error our proposed method	Mean absolute error our proposed method
540	27.48	21.19
544	20.23	14.89
552	19.72	14.82
559	25.38	18.61
563	22.61	16.79
567	28.85	20.41
570	20.56	15.64
575	26.03	18.76
584	32.04	22.53
588	21.11	15.84
591	24.41	17.69
596	23.06	17.23

Table 9
Evaluation of the proposed SAC-based DRL approach for all twelve subjects, while predicting BG values throughout the 45 min prediction horizon, with respect to RMSE and MAE metrics.

Patient ID	Root mean square error our proposed method	Mean absolute error our proposed method
540	24.99 ± 13.21	18.02 ± 11.67
544	20.52 ± 10.90	14.84 ± 9.87
552	19.82 ± 10.29	14.42 ± 9.28
559	24.83 ± 13.72	17.60 ± 12.23
563	20.88 ± 12.40	14.35 ± 11.29
567	27.87 ± 17.54	18.49 ± 15.72
570	19.32 ± 9.50	14.15 ± 8.53
575	25.15 ± 14.97	17.24 ± 13.66
584	31.51 ± 16.10	22.68 ± 14.70
588	20.45 ± 10.92	14.70 ± 9.91
591	23.08 ± 12.78	16.38 ± 11.50
596	22.72 ± 10.35	16.94 ± 9.35
Average	23.43 ± 12.72	16.65 ± 11.47

train the model, perform testing on six out of the twelve OhioT1DM subjects, and harness more than one feature to predict the future BG values.

Clinical Evaluation for Multi-Step Prediction: Utilizing SEG, the authors evaluate the clinical error between predicted and ground truth BG levels for all 12 patients. SEG calculates the percentage of predictions within predefined risk zones, as outlined by clinical practices. Table 13 presents the SEG results for all the subjects of the OhioT1DM dataset, followed by the comparison to prior work in Table 14, for the 30-min PH, wherein six BG values are predicted. Additionally, Fig. 6 reports the clinical error using the CEG metric for patient ID 570 and patient ID 584 corresponding to the best case and worst case, respectively. Table 15 presents the SEG results for all the subjects of the OhioT1DM dataset, followed by the comparison to prior work in Table 16, for the 45-min PH, wherein nine BG values are predicted. Fig. 7 visualizes the clinical error using the SEG metric for patient ID 570 and patient ID 584 corresponding to the best case and worst case, respectively. In general, through SEG analysis, it is evident that a substantial proportion of the prediction results fall within the no risk to slight risk zone, with a few outliers. Furthermore, the predictions consistently either outperform or align with the baseline.

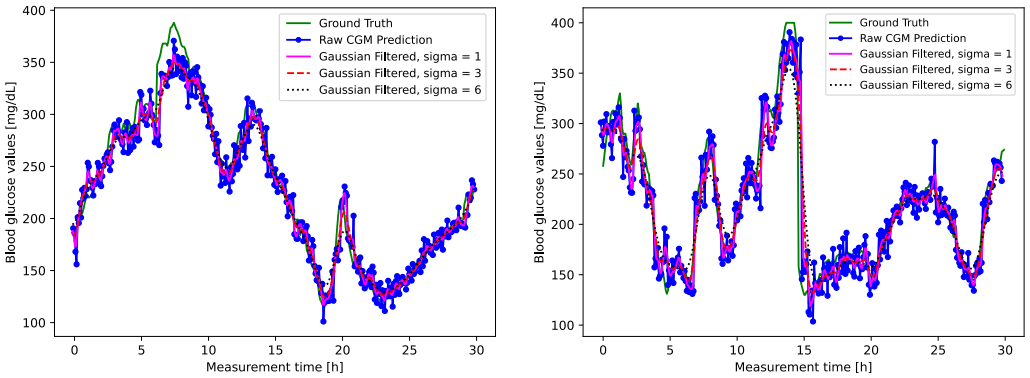
Table 10
Comparison of the proposed SAC-based DRL approach with prior work which uses only historical CGM values to predict future BG values for all twelve subjects, while predicting all values throughout the 45 min prediction horizon.

Method	Root mean square error	Mean absolute error
LSTM [5]	29.28	21.61
WaveNet [5]	29.68	22.19
GRU [5]	29.22	21.50
WaveNet + LSTM [5]	29.46	21.87
WaveNet + GRU [5]	29.44	21.83
LSTM + GRU [5]	29.26	21.56
LSTM + WaveNet + GRU [5]	29.12	21.52
Deep RL (**Proposed method**)	**23.43**	**16.66**

To prove the effectiveness of the proposed method over other datasets, the authors accessed S1 dataset [37], and D1NAMO dataset [38], and used the historical CGM values to predict BG values throughout the prediction horizon of 30 mins, corresponding to six BG values. Both the S1 dataset [37] and the D1NAMO dataset [38] have CGM values recorded from individuals with Type-2 Diabetes Mellitus (T2DM) and T1DM individuals respectively, at an interval of 5 min. Table 17 reports RMSE and MAE for five selected patients from both datasets. A low RMSE and MAE proves the efficacy of the proposed method.

5.2. Performance optimization

In all prior work [5,21], hyperparameters and network architectures were manually defined which often necessitated a significant depth of domain knowledge. Moreover, these models were not optimized for performance on analytical and clinical accuracy, network architecture, and model size, in a subject specific manner. This paper proposes a NAS approach using sequential, model-based BO. The authors implement

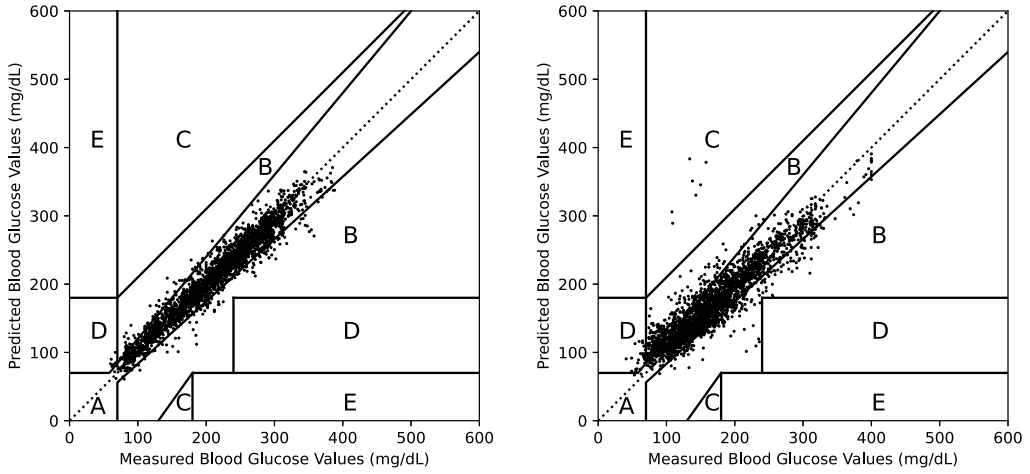


(a) Patient ID 570 (b) Patient ID 584

Fig. 5. BG ground truth in comparison to the predictions and Gaussian smoothing for a representative sample of the testing dataset of 30 h for patient ID 570 and patient ID 584.

Table 11
Classification results of the proposed SAC-based DRL approach in terms of accuracy, precision, recall, and F1-score for hypoglycemic, normoglycemic, and hyperglycemic range, for all twelve subjects, corresponding to all predicted BG values throughout the 45 min prediction horizon.

Patient ID	Accuracy	Hypoglycemia			Normoglycemia			Hyperglycemia		
		Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score
540	0.868	0.614	0.370	0.462	0.850	0.955	0.899	0.940	0.779	0.852
544	0.888	0.222	0.077	0.114	0.879	0.969	0.922	0.929	0.745	0.827
552	0.901	0.411	0.550	0.471	0.940	0.933	0.936	0.850	0.825	0.837
559	0.870	0.522	0.541	0.532	0.896	0.879	0.887	0.859	0.881	0.870
563	0.874	0.342	0.52	0.413	0.913	0.871	0.892	0.834	0.884	0.858
567	0.835	0.798	0.553	0.654	0.861	0.900	0.880	0.767	0.753	0.760
570	0.934	0.182	0.125	0.148	0.883	0.894	0.888	0.959	0.956	0.957
575	0.875	0.669	0.573	0.617	0.872	0.940	0.905	0.918	0.800	0.855
584	0.857	0.294	0.143	0.192	0.889	0.878	0.883	0.814	0.841	0.827
588	0.866	0.125	0.167	0.143	0.849	0.911	0.879	0.891	0.815	0.852
591	0.851	0.615	0.417	0.497	0.866	0.918	0.891	0.845	0.776	0.809
596	0.877	0.525	0.294	0.376	0.891	0.949	0.919	0.849	0.722	0.781
Average	0.875	0.443	0.361	0.385	0.882	0.916	0.898	0.871	0.815	0.840



(a) CEG for Patient ID 570 (b) CEG for Patient ID 584

Fig. 6. CEG for BG predictions in comparison to measurements for a representative sample of the testing dataset of 30 h for patient ID 570 and patient ID 584.

Optuna as a BO framework and optimize the model obtained from the DRL framework with respect to performance and model size respectively. The authors perform 100 trials using the defined hyperparameter search space in 4.4. In each trial, the NAS generates candidate architectures, which are trained using the DRL framework. The parameters *Target Update Period* and *Target Update* τ are used as the feedback to the NAS controller in each trial since they strongly affect the training of SAC. The remaining hyperparameters are used to optimize the model's structure and performance.

5.2.1. *Neural architecture search*

Fig. 8 shows the performance and model size of the individual candidate architectures proposed by the NAS framework. The authors investigate the trade-off between RMSE on the test data and the size

Table 12
Comparison for all predictions made throughout the PH of 30 min and 45 min over average RMSE and average MAE.

Reference	Additional dataset	Subjects from OhioT1DM dataset	Number of features used	PH = 30 min		PH = 45 min	
				Avg. RMSE	Avg. MAE	Avg. RMSE	Avg. MAE
[12]	Yes	Train: 12 subjects Test: 6 subjects	CGM, insulin bolus rate, bolus amount, carbohydrate intake, difference between consecutive glucose values	20.10	13.63	N/A	N/A
[6]	No	Train: 12 subjects Test: 6 subjects	finger stick value, basal rate value, galvanic skin response value, skin temperature value, bolus dose value, CGM	20.6	18.3	N/A	N/A
[27]	No	Train: 6 subjects, Test: 6 subjects	CGM, bolus amount, carbohydrate intake, time index normalized to the unit for each day	21.72	N/A	N/A	N/A
[14]	No	Train: 12 subjects, Test: 6 subjects	CGM	19.63	13.88	N/A	N/A
[5]	No	Train: 12 subjects, Test: 12 subjects	CGM	21.90	15.86	29.12	21.50
[24]	No	Train: 12 subjects, Test: 12 subjects	CGM	21.54	15.39	N/A	N/A
[11]	Yes	Train: 6 subjects Test: 6 subjects	CGM, basal insulin dose, bolus insulin dose, carbohydrate intake	17.80	N/A	N/A	N/A
[13]	No	Train: 12 subjects, Test: 6 subjects	interstitial fluid glucose concentration from CGM recordings, basal insulin infusion rate, insulin bolus delivery, carbohydrate oral intake	18.90	N/A	N/A	N/A
[47]	No	Train: 6 subjects Test: 6 subjects	CGM, carbohydrate intake, bolus insulin	18.34	13.37	N/A	N/A
[15]	No	Train: 12 subjects, Test: 12 subjects	CGM, carbohydrate intake, bolus insulin, finger stick glucose	16.06	10.64	N/A	N/A
[43]	No	Train: 6 subjects, Test: 6 subjects	CGM	20.03	14.52	N/A	N/A
[44]	No	Train: 6 subjects, Test: 6 subjects	CGM, insulin, carbohydrate intake, exercise	19.79	13.62	N/A	N/A
[7]	Yes	Train: 6 subjects, Test: 6 subjects	sampling time, CGM values, meal intake and insulin dose	18.90	N/A	N/A	N/A
[42]	Yes	Train: 6 subjects, Test: 6 subjects	CGM, finger stick, bolus dose, carbohydrate input, time of day encoded using sine and cosine embeddings	18.22	12.83	N/A	N/A
[45]	No	Train: 6 subjects, Test: 6 subjects	CGM, basal insulin, bolus insulin, insulin on board	19.37	13.76	N/A	N/A
[46]	No	Train: 6 subjects, Test: 6 subjects	CGM, insulin on board, carbohydrates on board, exercise on board, slope of CGM, time of day	18.69	10.08	N/A	N/A
[8]	No	Train: 6 subjects, Test: 6 subjects	CGM, basal insulin dosage, bolus insulin dosage, carbohydrate intake, time stamp	19.05	13.50	N/A	N/A
[9]	No	Train: 6 subjects, Test: 6 subjects	basal insulin, bolus insulin, carbohydrates, CGM	17.45	23.25	N/A	N/A
[10]	No	Train: 6 subjects, Test: 6 subjects	CGM	18.23	14.37	N/A	N/A
Our work	No	Train: 12 subjects, Test: 12 subjects	CGM	18.32	12.93	23.43	16.66

Table 13
Evaluation of the proposed SAC-based DRL approach for all twelve subjects, while predicting BG values throughout the 30 min prediction horizon, with respect to SEG metric.

Patient ID	None (0)	Slight (1)	Moderate (2)	Great (3)	Extreme (4)
540	85.104	14.896	0.000	0.000	0.000
544	90.815	9.185	0.000	0.000	0.000
552	89.541	10.459	0.000	0.000	0.000
559	89.928	9.952	0.120	0.000	0.000
563	89.085	10.915	0.000	0.000	0.000
567	86.869	12.963	0.168	0.000	0.000
570	90.367	9.560	0.073	0.000	0.000
575	89.161	10.801	0.039	0.000	0.000
584	85.332	14.517	0.075	0.075	0.000
588	91.361	8.531	0.108	0.000	0.000
591	89.592	10.408	0.000	0.000	0.000
596	89.011	10.989	0.000	0.000	0.000
Average	88.847	11.098	0.048	0.000	0.000

Table 14
Comparison of the proposed SAC-based DRL approach with prior work which uses only historical CGM values to predict future BG values for all twelve subjects, while predicting all values throughout the 30 min prediction horizon, with respect to the SEG metric.

Method	None (0)	Slight (1)	Moderate (2)	Great (3)	Extreme (4)
LSTM [5]	86.42	13.56	0.02	0.00	0.00
WaveNet [5]	85.91	14.06	0.03	0.00	0.00
GRU [5]	86.41	13.55	0.04	0.00	0.00
WaveNet + LSTM [5]	86.43	13.54	0.03	0.00	0.00
WaveNet + GRU [5]	86.39	13.57	0.04	0.00	0.00
LSTM + GRU [5]	86.52	13.44	0.04	0.00	0.00
LSTM + WaveNet+ GRU [5]	86.53	13.45	**0.02**	0.00	0.00
Deep RL (**Proposed method**)	**88.85**	**11.10**	0.05	**0.00**	**0.00**

Table 15
Evaluation of the proposed SAC-based DRL approach for all twelve subjects, while predicting BG values throughout the 45 min prediction horizon, with respect to SEG metric.

Patient ID	None (0)	Slight (1)	Moderate (2)	Great (3)	Extreme (4)
540	86.343	13.542	0.116	0.000	0.000
544	89.605	10.395	0.000	0.000	0.000
552	89.344	10.656	0.000	0.000	0.000
559	89.022	10.951	0.027	0.000	0.000
563	91.111	8.863	0.0261	0.000	0.000
567	85.035	14.515	0.366	0.084	0.000
570	92.015	7.790	0.195	0.000	0.000
575	86.817	13.157	0.0259	0.000	0.000
584	82.212	17.611	0.151	0.025	0.000
586	90.380	9.620	0.000	0.000	0.000
591	86.847	13.153	0.000	0.000	0.000
596	87.151	12.849	0.000	0.000	0.000

Table 16
Comparison of the proposed SAC-based DRL approach with prior work which uses only historical CGM values to predict future BG values for all twelve subjects, while predicting all values throughout the 45 min prediction horizon, with respect to the SEG metric averaged for all twelve subjects.

Method	None (0)	Slight (1)	Moderate (2)	Great (3)	Extreme (4)
LSTM [5]	80.95	18.98	0.07	0.00	0.00
WaveNet [5]	79.88	20.07	0.05	0.00	0.00
GRU [5]	81.08	18.85	0.07	0.00	0.00
WaveNet + LSTM [5]	81.04	18.90	0.06	0.00	0.00
WaveNet + GRU [5]	81.01	18.92	0.07	0.00	0.00
LSTM + GRU [5]	81.10	18.83	0.07	0.00	0.00
LSTM + WaveNet+ GRU [5]	81.14	18.80	**0.06**	**0.00**	0.00
Deep RL (**Proposed method**)	**87.99**	**11.93**	0.08	0.01	**0.00**

Table 17
Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) evaluation of the proposed method for BG predictions in T2DM individuals [37] and BG predictions on T1DM for D1NAMO dataset [38] for a prediction horizon of 30 min.

Dataset	Patient ID	RMSE our proposed method	MAE our proposed method
S1 Dataset [37]	1636 - 69 - 001	11.01	7.98
	1636 - 69 - 026	12.62	8.57
	1636 - 69 - 028	17.41	12.62
	1636 - 69 - 048	8.88	6.37
	1636 - 69 - 053	18.57	14.37
D1NAMO [38]	001	22.71	17.31
	002	27.54	21.21
	003	14.91	12.00
	005	18.86	14.59
	007	15.93	11.710

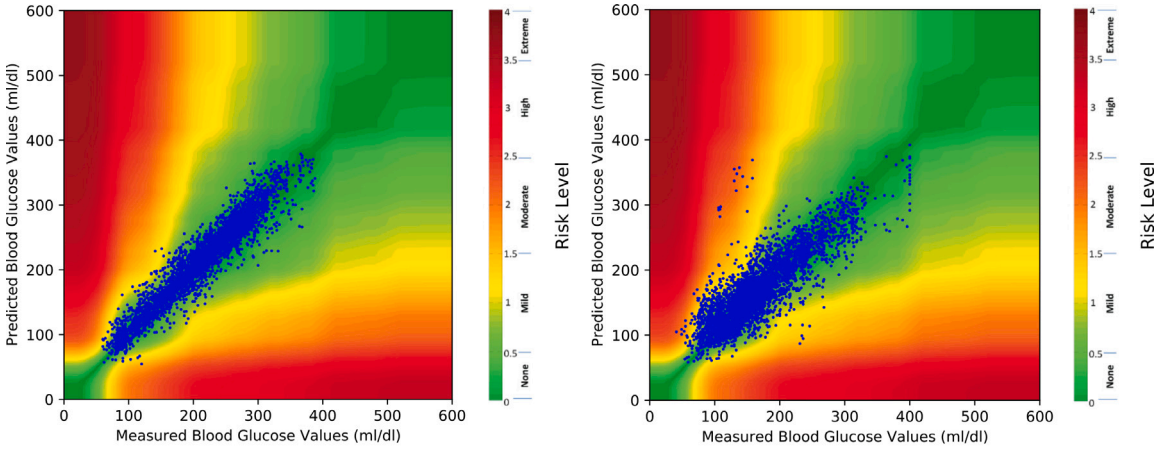
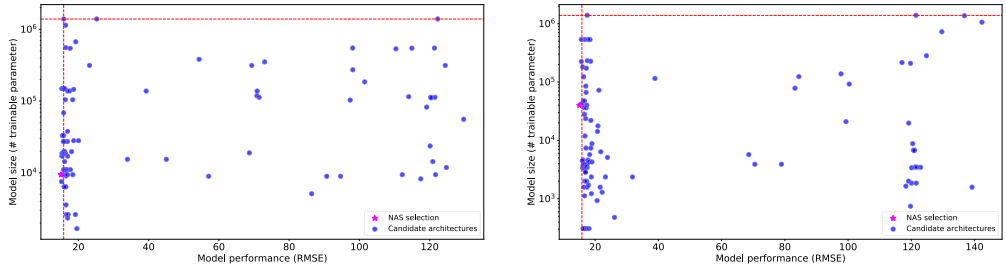
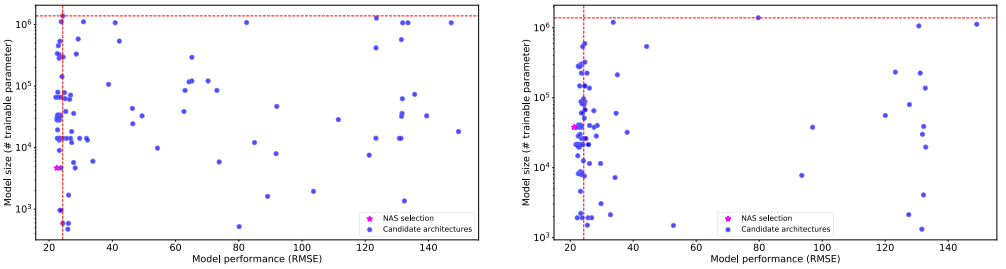


Fig. 7. SEG for BG predictions in comparison to measurements for a representative sample of the testing dataset of 30 hours for patient ID 570 and patient ID 584, while predicting all values throughout the 45-minute prediction horizon..



(a) LSTM-based NAS candidate architectures for patient ID 570

(b) GRU-based NAS candidate architectures for patient ID 570



(c) LSTM-based NAS candidate architectures for patient ID 584

(d) GRU-based NAS candidate architectures for patient ID 584

Fig. 8. Model size vs. test performance (RMSE) for candidate architectures proposed by our NAS framework. The scatter plots 8(a) and 8(b) show the result on patient ID 570 for different architecture types. Similarly, the plots 8(c) and 8(d) show results on patient ID 584, respectively.

(number of trainable parameters) of the actor network (critic network is only necessary for training). For both patients, comparative results of LSTM-based and GRU-based candidate architectures with prior work in [21] indicated by the red dashed lines. Both architecture types show a high sensitivity to different hyperparameter settings, including outliers with respect to test performance with an RMSE up to 140 (training is no longer converging). However, Fig. 8 visualizes that the BO based approach can identify higher amounts of suitable hyperparameter sets, as shown by the clusters in the scatter plots. With these hyperparameter sets, the authors achieve competitive results but using a fraction of network parameters. Depending on the specific patient and type of architecture, the proposed approach reduces the model size by a factor between 20 and 150 without sacrificing performance w.r.t. RMSE. A more detailed analysis with respect to test performance and model size is given in Table 18 and Table 19 respectively. In Table 18, the authors show the top 5 candidate architectures with respect to their performance on the test data. In the given experimental setting, our framework prefers network architectures with high performance over a small model size. Therefore, the authors select architectures that improve previous results, e.g., in [21], between 5% and 12% while sacrificing a possible reduction in model size by a factor of up to 4500. Depending on the patient, GRU-based architectures outperform LSTM-based structures but do not always show better scaling behavior with respect to test performance and small model sizes, see Fig. 8 and Table 19.

Additionally, the BG prediction regression task was translated to a classification task using data labeling approach [64], for hypoglycemia, normoglycemia, and hyper-glycemia for BG values BG $\leq$ 70 mg/dL, 70 mg/dL < CGM < 180 mg/dL, and BG $\geq$ 180 mg/dL respectively. For the predictions of all the BG values throughout the 30-min prediction interval, corresponding to 6 data points, the classification results are presented in Table 20.

Fig. 9 visualizes the distribution of the model parameters across the individual architectural components. Interestingly, the input encoding seems less relevant and thus typically requires less trainable parameters. The remaining parameters are almost equally distributed across the output encoder and the RNN, with a slight preference for one of the two depending on the network type and patient in the dataset.

Table 18
Overview of Top 5 candidate architectures sorted by test performance (here: RMSE). Additionally, we specify the number of parameters to show the different scaling behavior of the architecture types.

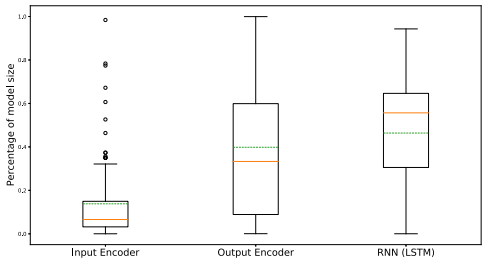
Top 5 architectures (RMSE)	LSTM-based architecture		GRU-based architecture	
	Parameters	Error (RMSE)	Parameters	Error (RMSE)
Patient ID 570	**9428**	**15.050**	**19 212**	**14.898**
	7552	15.175	1 192 460	15.594
	148 980	15.267	279 436	15.625
	17 076	15.291	101 516	15.909
	18 716	15.335	224 524	15.926
Patient ID 584	**65 884**	**21.876**	**37 920**	**21.246**
	4668	22.384	21 428	21.809
	30 556	22.479	37 920	21.839
	28 096	22.503	21 716	22.220
	338 316	22.580	21 408	22.224

Table 19
Overview of Top 5 candidate architectures sorted by the number of parameters. Additionally, we specify the test RMSE to show the different scaling behavior of the architecture types.

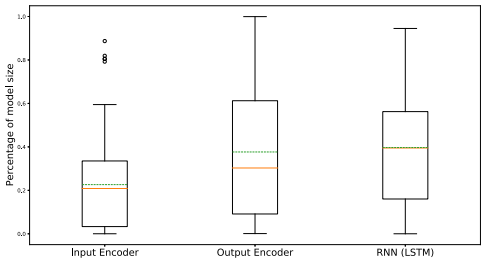
Top 5 architectures (complexity)	LSTM-based architecture		GRU-based architecture	
	Parameters	Error (RMSE)	Parameters	Error (RMSE)
Patient ID 570	1672	19.557	308	16.176
	2348	16.922	308	16.179
	2620	19.141	308	17.257
	2620	16.556	308	16.680
	2620	17.060	308	16.517
Patient ID 584	472	25.940	1320	131.642
	520	80.117	1492	52.776
	588	24.336	1508	25.458
	588	26.125	1920	22.226
	956	23.756	1920	26.814

Table 20
Classification results on patient ID 570 and patient ID 584 for the proposed NAS in conjunction to DRL approach in terms of accuracy, precision, recall, and F1-score for hypoglycemic, normoglycemic, and hyperglycemic range, for all twelve subjects, corresponding to all predicted BG values throughout the 30 min prediction horizon.

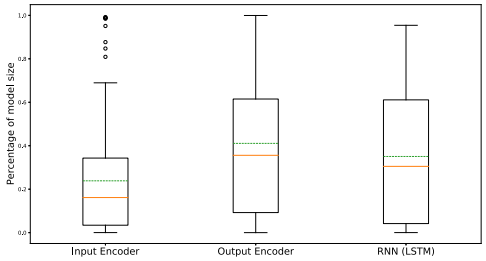
Patient ID	Accuracy	Hypoglycemia			Normoglycemia			Hyperglycemia		
		Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score
570 (LSTM architecture)	0.949	0.235	0.364	0.286	0.925	0.898	0.912	0.965	0.973	0.969
570 (GRU architecture)	0.949	0.364	0.364	0.364	0.912	0.913	0.913	0.968	0.967	0.967
584 (LSTM architecture)	0.891	0.5	0.074	0.129	0.912	0.911	0.911	0.86	0.881	0.87
584 (GRU architecture)	0.922	0.583	0.259	0.359	0.931	0.942	0.937	0.91	0.906	0.908



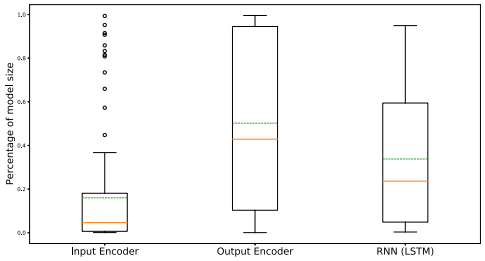
(a) Distribution of model size for LSTM-based candidates, patient ID 570



(c) Distribution of model size for GRU-based candidates, patient ID 570



(b) Distribution of model size for LSTM-based candidates, patient ID 584



(d) Distribution of model size for GRU-based candidates, patient ID 584

Fig. 9. The box plots show the distribution of (trainable) model parameters across the individual model components: the input encoder, the output encoder, and the RNN.

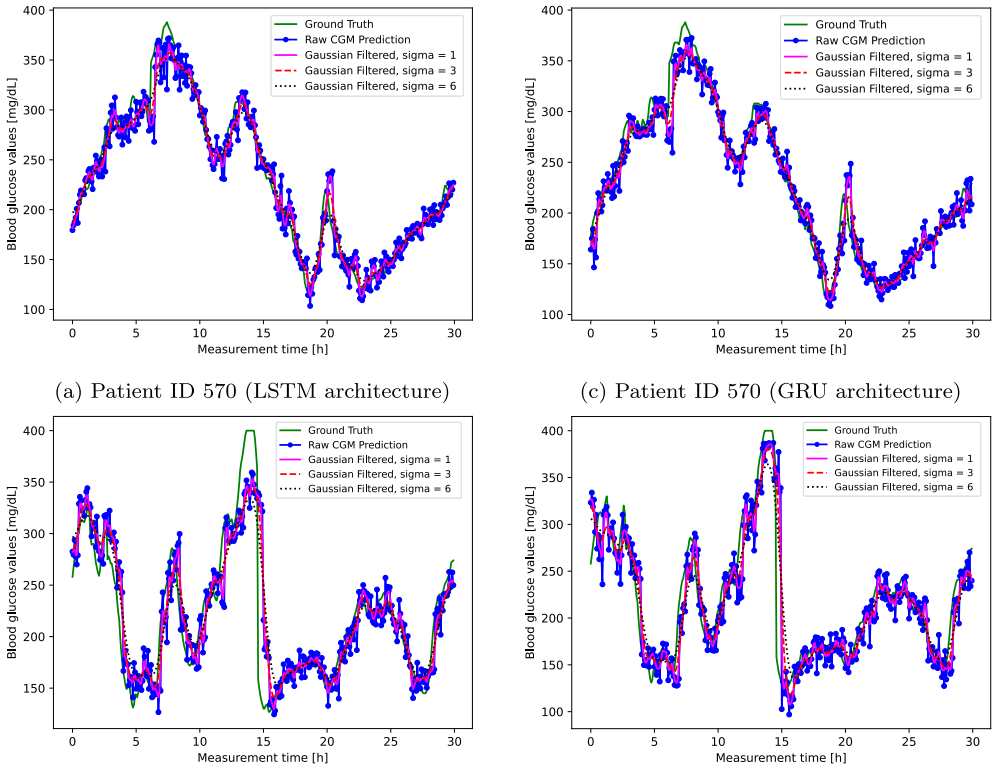
Multi-Step Predictions for 30 minutes: In the following, the authors predict BG values for six consecutive time steps while considering input data with a time frame of 30 min using the best model of the NAS run for LSTM-based and GRU-based architectures and patient ID 570 as well as 584. For patient ID 570, the proposed method achieved improvements of 4.8% (RMSE: 15.050) using LSTM-based architectures and 5.7% (RMSE: 14.898) with GRU-based architectures, respectively. Furthermore, increasing improvement for patient ID 584 with 10.0% (RMSE: 21.876) and 12.6% (RMSE: 21.246), respectively was observed. Fig. 10 shows how well the selected candidate architectures fit the test data. Therefore, it visualizes the ground truth BG values compared to the model predictions for a representative test data of 30 h for both patients. Similar to Fig. 5, we observe that both architecture types fit the data well but show small imprecision predictions at sudden changes in BG values. Gaussian filtering with varying σ is used to reduce the fluctuations arising from sensor faults, connectivity problems, electrical interference, and manufacturing variability [15,65,66], see Fig. 10.

SEG for Multi-Step Prediction: The authors assess the risk between using the prediction of the selected candidate architectures and the ground truth BG values. The SEG visualizations of patient ID 570 and ID 584 for the different architecture types are shown in Fig. 11. Using the SEG metric to evaluate clinical risks, the authors conclude that most predictions are in no risk to slight risk zones. Moreover, the results remain consistent with state-of-the-art [5,21]. Thus, the proposed architectures effectively reduce the model size without increasing clinical risks severely.

In Table 21, the authors present the average SEG values in multi-step prediction of 30 min using the proposed architectures of the NAS framework. Compared with the latest results in [21], the authors encounter only small changes in the none and slight risk zones in a range of $\pm 3\%$ (see Table 21).

6. Conclusion

The authors proposed a novel methodology for time series modeling of BG levels in T1DM using a densely connected encoder–decoder network and an LSTM or GRU formulated as a DRL problem. The authors have evaluated the results for the OhioT1DM dataset benchmark. Furthermore, the authors proposed an automatized optimization workflow, effectively introducing NAS in BG prediction tasks. Compared to the prior work that achieved the best prediction error, on average and for the MAE the proposed method have improved by 18.4% and 22.5% in 30-min and 45-min prediction horizons without applying additional optimization of the network's architectural structure. The prediction



(a) Patient ID 570 (LSTM architecture)

(c) Patient ID 570 (GRU architecture)

(b) Patient ID 584 (LSTM architecture)

(d) Patient ID 584 (GRU architecture)

Fig. 10. BG ground truth in comparison to the predictions and Gaussian smoothing for a representative sample of the testing dataset of 30 h for patient ID 570 and patient ID 584, for our proposed method of DRL in conjunction with NAS.

Table 21
Average SEG in NAS-based multi-step prediction results (PH = 30 min).

Patient ID	None (0)			Slight (1)			Moderate (2)			Great (3)			Extreme (4)		
	[21]	NAS		[21]	NAS		[21]	NAS		[21]	NAS		[21]	NAS	
		LSTM	GRU		LSTM	GRU		LSTM	GRU		LSTM	GRU		LSTM	GRU
570	90.367	89.267	91.575	9.560	10.659	8.425	0.073	0.073	0.000	0.000	0.000	0.000	0.000	0.000	0.000
584	85.332	88.311	86.953	14.517	11.614	12.971	0.075	0.075	0.075	0.075	0.000	0.000	0.000	0.000	0.000

results on the best (ID 570) and worst case (ID 584) patients using only the DRL method were further enhanced by integrating the BO-based NAS to the DRL framework. The proposed method achieved additional improvements of 4.8% using LSTM-based architectures and 5.7% with GRU-based architectures for patient ID 570 by integrating NAS. The patient with the lowest performance (ID 584) on the DRL method shows an even greater performance boost, with improvements of 10.0% and 12.6% observed for the LSTM or GRU, respectively. Moreover, for risk assessment in our predictions, the authors have visualized the error and evaluated clinical risk through an SEG and CEG approach. The authors have shown that we outperform or remain consistent with the results of the previous risk assessments and that NAS can reduce the model size by a significant factor without increasing clinical risks to unacceptable levels.

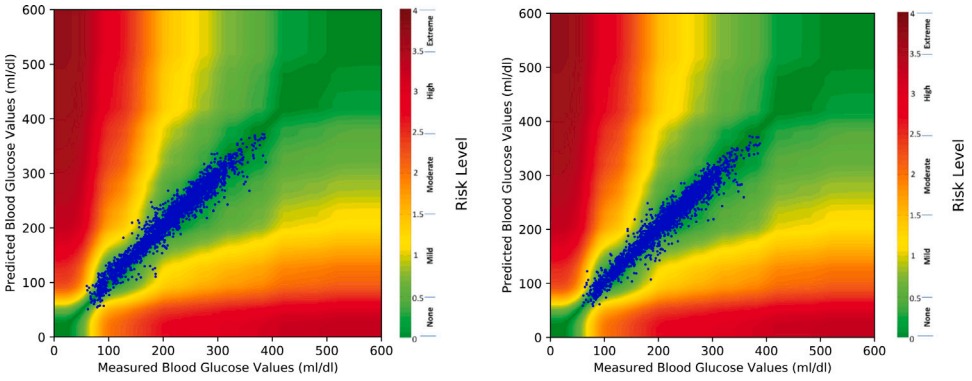
For future work, the successful compression of the model sizes paves the way for applying selected candidate architectures on resource-constrained devices, e.g., edge or smart devices. Therefore, deploying the model efficiently on edge devices will be another goal of the future work.

CRediT authorship contribution statement

Peter Domanski: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Conceptualization. **Aritra Ray:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Conceptualization. **Kyle Lafata:** Supervision, Project administration. **Farshad Firouzi:** Supervision, Project administration. **Krishnendu Chakrabarty:** Writing – original draft, Project administration, Funding acquisition. **Dirk Pflüger:** Writing – review & editing, Writing – original draft, Resources.

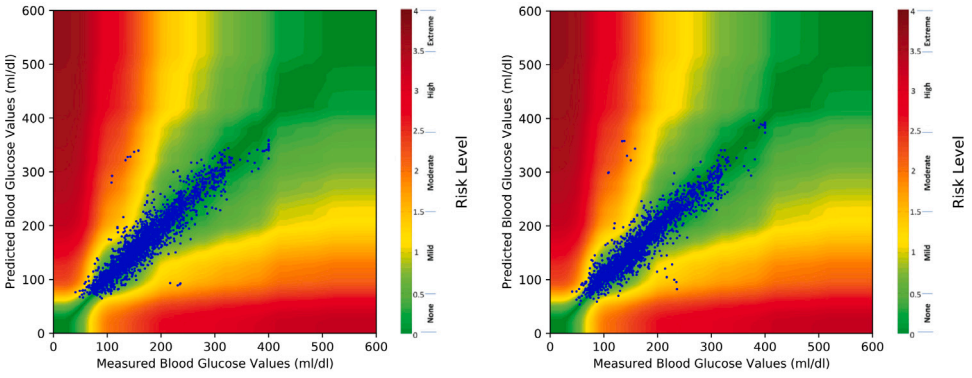
Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Peter Domanski reports financial support was provided by Advantest. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) SEG for LSTM-based architectures on patient ID 570

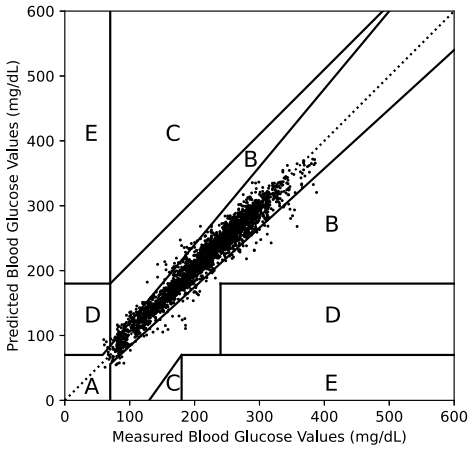
(b) SEG for GRU-based architectures on patient ID 570



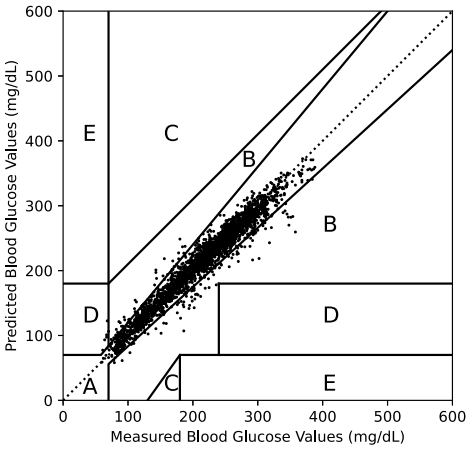
(c) SEG for LSTM-based architectures on patient ID 584

(d) SEG for GRU-based architectures on patient ID 584

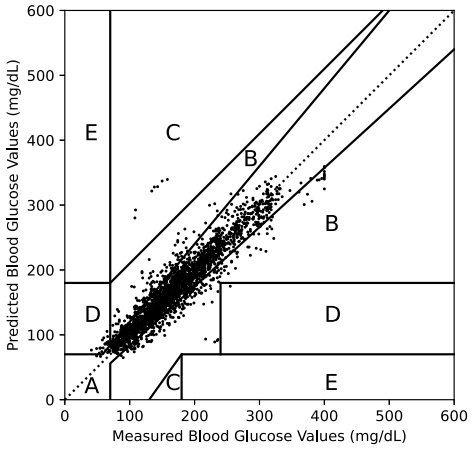
Fig. 11. SEG for NAS-based BG predictions in comparison to the measurements for a representative sample of the testing dataset of 30 h for patient ID 570 (11(a), 11(b)) and patient ID 584 (11(c), 11(d)).



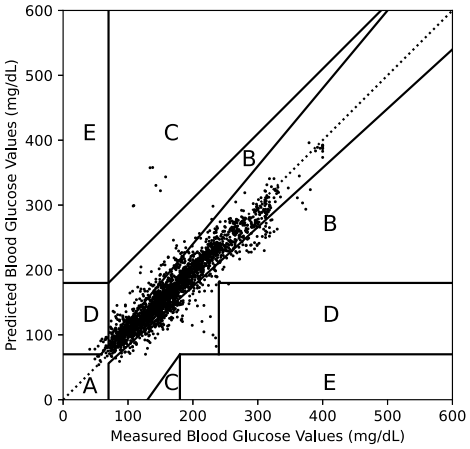
(a) CEG for LSTM-based architectures on patient ID 570.



(c) CEG for GRU-based architectures on patient ID 570.



(b) CEG for LSTM-based architectures on patient ID 584.



(d) CEG for GRU-based architectures on patient ID 584.

Fig. 12. SEG for NAS-based BG predictions in comparison to the measurements for a representative sample of the testing dataset of 30 h for patient ID 570 (12(a), 12(c)) and patient ID 584 (12(b), 12(d)).

References

[1] American Diabetes Association. Diagnosis and classification of diabetes mellitus. Diabetes Care 2014;37(Supplement_1):S81–90.

[2] Deshpande AD, Harris-Hayes M, Schootman M. Epidemiology of diabetes and diabetes-related complications. Phys Ther 2008;88(11):1254–64.

[3] Ogurtsova K, Guariguata L, Barengo NC, Ruiz PL, Sacre JW, Karuranga S, et al. IDF diabetes atlas: Global estimates of undiagnosed diabetes in adults for 2021. Diabetes Res Clin Pract 2022;183:109118.

[4] Elsayed N, ElSayed Z, Ozer M. Early stage diabetes prediction via extreme learning machine. In: SoutheastCon 2022. IEEE; 2022, p. 374–9.

[5] Dudukcu HV, Taskiran M, Yildirim T. Blood glucose prediction with deep neural networks using weighted decision level fusion. Biocybern Biomed Eng 2021;41(3):1208–23.

[6] Bhimireddy A, Sinha P, Oluwalade B, Gichoya JW, Purkayastha S. Blood glucose level prediction as time-series modeling using sequence-to-sequence neural networks. In: CEUR workshop proceedings.

[7] Zhu T, Li K, Chen J, Herrero P, Georgiou P. Dilated recurrent neural networks for glucose forecasting in type 1 diabetes. J Healthc Inform Res 2020;4:308–24.

[8] Yang T, Wu R, Tao R, Wen S, Ma N, Zhao Y, et al. Multi-scale long short-term memory network with multi-lag structure for blood glucose prediction. In: KDH@ ECAI, vol. 45, 2020, p. 136–40.

[9] Freiburghaus J, Rizzotti A, Albertetti F. A deep learning approach for blood glucose prediction of type 1 diabetes. In: Proceedings of the proceedings of the 5th international workshop on knowledge discovery in healthcare data co-located with 24th European conference on artificial intelligence. 2020.

[10] Bevan R, Coenen F. Experiments in non-personalized future blood glucose level prediction. In: CEUR workshop proceedings, vol. 2675, 2020, p. 100–4.

[11] Gu K, Dang R, Prioleau T. Neural physiological model: A simple module for blood glucose prediction. In: 2020 42nd annual international conference of the IEEE engineering in medicine & biology society. IEEE; 2020, p. 5476–81.

[12] Hameed H, Kleinberg S. Investigating potentials and pitfalls of knowledge distillation across datasets for blood glucose forecasting. In: Proceedings of the 5th annual workshop on knowledge discovery in healthcare data. 2020.

[13] Cui R, Hettiarachchi C, Nolan CJ, Daskalaki E, Suominen H. Personalised short-term glucose prediction via recurrent self-attention network. In: 2021 IEEE 34th International Symposium on Computer-Based Medical Systems. IEEE; 2021, p. 154–9.

[14] Nemat H, Khadem H, Eissa MR, Elliott J, Benaissa M. Blood glucose level prediction: advanced deep-ensemble learning approach. IEEE J Biomed Health Inform 2022;26(6):2758–69.

[15] Shuvo MM, Islam SK. Deep multitask learning by stacked long short-term memory for predicting personalized blood glucose concentration. IEEE J Biomed Health Inform 2023;27(3):1612–23.

[16] Marling C, Bunescu R. The OhioT1DM dataset for blood glucose level prediction: update 2020. In: CEUR workshop proceedings, vol. 2675, NIH Public Access; 2020, p. 71.

[17] Benidis K, Rangapuram SS, Flunkert V, Wang Y, Maddix D, Turkmen C, et al. Deep learning for time series forecasting: Tutorial and literature survey. ACM Comput Surv 2022;55(6):1–36.

[18] Tang S, Pan Y. Feature extraction via recurrent random deep ensembles and its application in gruop-level happiness estimation. 2017, arXiv preprint arXiv:1707.09871.

[19] Keren G, Schuller B. Convolutional RNN: An enhanced model for extracting features from sequential data. In: 2016 International joint conference on neural networks. (IJCNN), IEEE; 2016, p. 3412–9.

[20] Fu Q, Han Z, Chen J, Lu Y, Wu H, Wang Y. Applications of reinforcement learning for building energy efficiency control: A review. J Build Eng 2022;50:104165.
[21] Domanski P, Ray A, Firouzi F, Lafata K, Chakrabarty K, Pflüger D. Blood glucose prediction for type-1 diabetics using deep reinforcement learning. In: 2023 IEEE international conference on digital health. IEEE; 2023, p. 339–47.
[22] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019, p. 2623–31.
[23] Klonoff DC, Lias C, Vigersky R, Clarke W, Parkes JL, Sacks DB, et al. The surveillance error grid. J Diabetes Sci Technol 2014;8(4):658–72.
[24] Dudukcu HV, Taskiran M, Yildirim T. Consolidated or individual training: which one is better for blood glucose prediction?. In: 2021 International conference on innovations in intelligent systems and applications. IEEE; 2021, p. 1–6.
[25] Francescato MP, Geat M, Fusi S, Stupar G, Noacco C, Cattin L. Carbohydrate requirement and insulin concentration during moderate exercise in type 1 diabetic patients. Metabolism 2004;53(9):1126–30.
[26] Zhu T, Li K, Kuang L, Herrero P, Georgiou P. An insulin bolus advisor for type 1 diabetes using deep reinforcement learning. Sensors 2020;20(18):5058.
[27] Zhu T, Li K, Herrero P, Chen J, Georgiou P. A deep learning algorithm for personalized blood glucose prediction. In: KHD@ IJCAI. 2018, p. 64–78.
[28] Ubl M, Koutny T, Della Cioppa A, De Falco I, Tarantino E, Scafuri U. Distributed assessment of virtual insulin-pump settings using smartcgms and dmms. r for diabetes treatment. Sensors 2022;22(23):9445.
[29] Man CD, Micheletto F, Lv D, Breton M, Kovatchev B, Cobelli C. The UVA/PADOVA type 1 diabetes simulator: new features. J Diabetes Sci Technol 2014;8(1):26–34.
[30] Kovatchev BP, Breton M, Man CDalla, Cobelli C. In silico preclinical trials: A proof of concept in closed-loop control of type 1 diabetes.
[31] Colmegna P, Bisio A, McFadden R, Wakeman C, Oliveri MC, Nass R, et al. Evaluation of a web-based simulation tool for self-management support in type 1 diabetes: A pilot study. IEEE J Biomed Health Inform 2022;27(1):515–25.
[32] Xie J. [dataset] jxx123/simglucose. GitHub. 2021. Available from: https://github.com/jxx123/simglucose.
[33] Hettiarachchi C. [dataset] chirathyh/GluCoEnv. GitHub. 2024 [cited 2024 Feb 1]. Available from: https://github.com/chirathyh/GluCoEnv.
[34] [dataset] What is OpenAPS? — OpenAPS.org. Openaps.org. 2018. Available from: https://openaps.org/what-is-openaps/.
[35] [dataset] Tidepool. www.tidepool.org. Available from: https://www.tidepool.org/.
[36] Broll S, Urbanek J, Buchanan D, Chun E, Muschelli J, Punjabi NM, et al. Interpreting blood glucose data with R package iglu. PLoS One 2021;16(4):e0248560.
[37] Colás A, Vigil L, Vargas B, Cuesta–Frau D, Varela M. Detrended fluctuation analysis in the prediction of type 2 diabetes mellitus in patients at risk: Model optimization and comparison with other metrics. PLoS One 2019;14(12):e0225817.
[38] Dubosson F, Ranvier JE, Bromuri S, Calbimonte JP, Ruiz J, Schumacher M. The open D1NAMO dataset: A multi-modal dataset for research on non-invasive type 1 diabetes management. Inform Med Unlocked 2018;13:92–100.
[39] Hall H, Perelman D, Breschi A, Limcaoco P, Kellogg R, McLaughlin T, et al. Glucotypes reveal new patterns of glucose dysregulation. PLoS Biol 2018;16(7):e2005143.
[40] Weinstock RS, DuBose SN, Bergenstal RM, Chaytor NS, Peterson C, Olson BA, et al. Risk factors associated with severe hypoglycemia in older adults with type 1 diabetes. Diabetes Care 2016;39(4):603–10.
[41] Neinstein A, Wong J, Look H, Arbiter B, Quirk K, McCanne S, et al. A case study in open source innovation: developing the tidepool platform for interoperability in type 1 diabetes management. J Am Med Inform Assoc 2016;23(2):324–32.
[42] Rubin-Falcone H, Fox I, Wiens J. Deep residual time-series forecasting: application to blood glucose prediction. In: KDH@ ECAI. 2020, p. 105–9.
[43] Ma N, Zhao Y, Wen S, Yang T, Wu R, Tao R, et al. Online blood glucose prediction using autoregressive moving average model with residual compensation network. In: KDH@ ECAI. 2020, p. 151–5.
[44] Daniels J, Herrero P, Georgiou P. Personalised glucose prediction via deep multitask networks. In: KDH@ ECAI. 2020, p. 110–4.
[45] Sun X, Rashid MM, Sevil M, Hobbs N, Brandt R, Askari MR, et al. Prediction of blood glucose levels for people with type 1 diabetes using latent-variable-based model. In: KDH@ ECAI. 2020, p. 115–9.
[46] Pavan J, Prendin F, Meneghetti L, Cappon G, Sparacino G, Facchinetti A, et al. Personalized machine learning algorithm based on shallow network and error imputation module for an improved blood glucose prediction. In: KDH@ ECAI. 2020, p. 95–9.
[47] Zhu T, Yao X, Li K, Herrero P, Georgiou P. Blood glucose prediction for type 1 diabetes using generative adversarial networks. In: CEUR workshop proceedings. vol. 2675, 2020, p. 90–4.
[48] Clarke WL. The original clarke error grid analysis (EGA). Diabetes Technol Therapeutics 2005;7(5):776–9.
[49] Zhou T, Ye X, Lu H, Zheng X, Qiu S, Liu Y. Dense convolutional network and its application in medical image analysis. BioMed Res Int 2022;2022.
[50] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[51] Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
[52] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. 2014, arXiv preprint arXiv:1406.1078.
[53] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014, arXiv preprint arXiv:1412.3555.
[54] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT Press; 2018.
[55] Liu T, Tan Z, Xu C, Chen H, Li Z. Study on deep reinforcement learning techniques for building energy consumption forecasting. Energy Build 2020;208:109675.
[56] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. PMLR; 2018, p. 1861–70.
[57] Konda V, Tsitsiklis J. Actor-critic algorithms. In: Advances in neural information processing systems, 1999, p. 12.
[58] He X, Zhao K, Chu X. AutoML: A survey of the state-of-the-art. Knowl-Based Syst 2021;212:106622.
[59] Waring J, Lindvall C, Umeton R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. Artif Intell Med 2020;104:101822.
[60] Elsken T, Metzen JH, Hutter F. Neural architecture search: A survey. J Mach Learn Res 2019;20(1):1997–2017.
[61] Bergenstal RM, Gal RL, Connor CG, Gubitosi-Klug R, Kruger D, Olson BA, et al. Racial differences in the relationship of glucose concentrations and hemoglobin A1c levels. Ann Internal Med 2017;167(2):95–102.
[62] Beck RW, Connor CG, Mullen DM, Wesley DM, Bergenstal RM. The fallacy of average: how using HbA1c alone to assess glycemic control can be misleading. Diabetes Care 2017;40(8):994–9.
[63] Peters AL. Role of continuous glucose monitoring in diabetes treatment. Arlington (VA): American Diabetes Association; 2018.
[64] D'Antoni F, Petrosino L, Marchetti A, Bacco L, Pieralice S, Vollero L, et al. Layered meta-learning algorithm for predicting adverse events in type 1 diabetes. IEEE Access 2023;11:9074–94.
[65] Staal OM, Sælid S, Fougner A, Stavdahl Ø. Kalman smoothing for objective and automatic preprocessing of glucose data. IEEE J Biomed Health Inform 2018;23(1):218–26.
[66] Facchinetti A, Del Favero S, Sparacino G, Cobelli C. Modeling transient disconnections and compression artifacts of continuous glucose sensors. Diabetes Technol Therapeutics 2016;18(4):264–72.