



# Directional Representation Encoder-Decoder for Personalized Blood Glucose Forecasting

Yu Chen<sup>1</sup>, Zhijin Wang<sup>1(✉)</sup>, Jinmo Tang<sup>2</sup>, Henghong Lin<sup>4</sup>, Senzhen Wu<sup>1</sup>,  
and Yaohui Huang<sup>3</sup>

<sup>1</sup> College of Computer Engineering, Jimei University, Xiamen 361021, China  
zhijinecnu@gmail.com

<sup>2</sup> Xiamen Hospital of Traditional Chinese Medicine, Xiamen 361015, China

<sup>3</sup> Heshan District Health Service Center, Xiamen 361015, China

<sup>4</sup> School of Automation, Central South University, 410083 Changsha, China

**Abstract.** Diabetes affects approximately 14% of the global population. Accurate blood glucose forecasting (BGF) is crucial for diabetes management and helps prevent dangerous glycemic fluctuations. Nonetheless, significant challenges arise due to the inherent heterogeneity among patients and the disparities in data distributions, which hinder model personalization and reduce training efficiency. This study proposes the Directional Representation Encoder-Decoder (DRED), a novel personalized time series model, to address the challenge of personalized blood glucose forecasting (PBGF). The DRED framework leverages directional representations (DR) to capture individual glycemic dynamics. The model leverages an advanced encoder-decoder (ED) framework that incorporates a DR module, capturing complex temporal dependencies and patient-specific traits. Experiments on two datasets show that DRED outperforms twenty recent methods. It achieves a mean absolute error (MAE) of 6.79 mg/dl and RMSE of 11.74 mg/dl on KDD18, and an MAE of 6.74 mg/dl and RMSE of 10.18 mg/dl on CDD.

**Keywords:** Blood Glucose · Forecasting · Heterogeneity · Personalization · Diabetes

## 1 Introduction

Diabetes, a chronic disease affecting over 830 million people globally, poses significant risks of complications such as cardiovascular disease, kidney disease, and retinopathy [5]. These complications severely affect patients' quality of life and place a heavy burden on individuals and society. However, diabetes can be effectively managed through comprehensive measures such as diet, medication, and

---

This work was supported by the FAST community. The correspondence is addressed at Zhijin Wang.

timely screening. Blood glucose levels serve as key indicators for disease progression and management. Accurate blood glucose forecasting (BGF) is crucial for preventing extreme glucose events and optimizing treatment plans [1, 2].

However, developing a personalized BGF (PBGF) model is challenging. This results from the heterogeneity of the blood glucose data and the unique factors influencing individual levels, such as diet, exercise, or medication [8]. Traditional methods rely on simple statistical models or classic machine learning algorithms. Although these methods are easy to implement, they often fail to capture personalized dynamics. They assume similar glucose behavior across different individuals, ignore physiological differences and struggle with complex non-linear dynamics and temporal relationships, leading to inaccurate forecasts and poor generalization [14].

To address these challenges, we propose a novel personalized time series model, the Directional Representation Encoder-Decoder (called “DRED”), which uses a directional representation (DR) module together with an Encoder-Decoder (ED) framework to generate accurate forecasts based on aggregated continuous glucose monitoring (CGM) data. This approach ensures robust performance across diverse patient populations while balancing personalization, generalizability, and computational efficiency for practical clinical deployment.

The proposed method was evaluated against 20 other approaches on two clinical datasets: the KDD18 T1DM Dataset (KDD18) and the Chinese Diabetes Dataset (CDD). KDD18 is a well-established benchmark, whereas CDD tests generalizability across different patient populations. The results show that DRED exceeds the latest methods in forecasting accuracy, generalization, and training efficiency [7].

The main contributions of this paper are summarized as follows:

- (1) The PBGF problem is first formulated as the problem of personalized time series forecasting, which trains a general model for new short-sequence glucose monitoring records.
- (2) The proposed DRED model is based on an ED architecture with directional representations and applies a highway network to enhance performance. This effectively captures temporal dependencies and yields more accurate predictions.
- (3) Extensive experimentation on two real datasets verifies the effectiveness of the proposed DRED in terms of forecasting errors.

## 2 Related Work

The relevant literature can be categorized into two main areas: blood glucose forecasting and personalized time series forecasting.

### 2.1 Blood Glucose Forecasting

Blood glucose forecasting has long relied on historical glucose levels as the basic input for various forecasting models. Traditional methods, such as ARIMA, have

been used to forecast future levels based on past measurements [17]. However, these methods often struggle with the non-linear dynamics of glucose fluctuations, which are influenced by multiple complex factors.

In recent years, deep learning techniques have significantly improved the accuracy of blood glucose forecasting [3, 6]. Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory networks (LSTM) and Gated Recurrent Units (GRUs), have emerged as powerful tools to capture temporal dependencies in glucose time series data [11, 12]. These models effectively handle the non-linear dynamics of glucose fluctuations and provide more accurate short-term forecasts [19].

Incorporation of exogenous factors into the forecasting model has been shown to significantly improve the accuracy of the forecasting. Dietary information and insulin dose are crucial to improving model performance and enhancing forecasting accuracy. Moreover, physical activity has a certain impact on forecast results by affecting insulin sensitivity and glucose utilization [2].

Recently, several advanced models have been developed to improve the accuracy and robustness of the forecast. GluNet employs multilayer dilated CNNs with gated activations, leveraging preprocessing of historical data and differential labels for training [11]. GluGAN generates personalized glucose time series by integrating embedding, recovery, and supervisor networks to capture temporal dynamics through adversarial and supervised learning [20]. HETER aligns heterogeneous CGM data using Similarity-Specific Representation (SSR) and constructs a spatial relationship graph (SRGraph) to integrate global and local temporal information across patients [8]. However, these models still cannot cope with data heterogeneity (e.g., data length varies, patient conditions vary).

## 2.2 Personalized Time Series Modeling

Time series forecasting technology can be roughly divided into single time series forecasting and multivariate forecasting [10], but both have limitations for blood glucose forecasting. However, personalized blood glucose forecasting has great advantages.

The traditional single time series forecasting method splices all patients' blood glucose data into a univariate time series and randomly samples and trains. The same batch of data samples may not cover all patients, which can easily lead to poor generalization ability and failure to meet the needs of personalized forecasting. Although multivariate time series forecasting improves generalization ability, the preprocessing process requires data alignment to ensure the same time, thus, the user's recent blood glucose data will be lost, affecting the timeliness of the model.

To address these limitations, personalized forecasting has emerged as a highly advantageous method, especially in medical applications. The proposed DRED model divides each patient's blood glucose data into supervised time series training samples, samples them separately, retains patient data characteristics and recent trends, and solves the problems of poor generalization of univariate prediction and data truncation of multivariate forecasting.

### 3 The Proposed DRED

With the background and challenges in blood glucose forecasting and personalized time series modeling established, we now introduce the proposed DRED model in detail.

#### 3.1 Problem Formulation

The blood glucose values are recorded by CGM devices in sessions [9, 16]. Let  $\mathcal{S}$  denote the set of CGM sessions, and  $\mathcal{G}$  denote the set of blood glucose time series. The symbol  $N_s$  denotes the length of the  $s$ -th session. The symbol  $\mathbf{G}^{(s)} \in \mathbb{R}^{N_s \times 1}$  denotes the blood glucose time series in the  $s$ -th session. Let  $T$  be the input window length and  $H$  be the output window length. The symbol  $\mathbf{G}_{1:T}^{(s)}, \mathbf{G}_{T+1:T+H}^{(s)}$  denotes the highest blood glucose sequence and the upcoming blood glucose sequence, respectively.

The blood glucose forecasting of an individual session is commonly regarded as the problem of time series forecasting:

$$\mathbf{G}_{T+1:T+H}^{(s)} \leftarrow f(\mathbf{G}_{1:T}^{(s)}), \quad (1)$$

where  $f(\cdot)$  is the mapping function, also known as a time series model. For an easy representation, let  $\mathbf{x} \in \mathbb{R}^{T \times 1}, \mathbf{y} \in \mathbb{R}^{H \times 1}$  be the input and output of the time series model, respectively. Hence, Formula (1) can be rewritten as:

$$\mathbf{y} \leftarrow f(\mathbf{x}). \quad (2)$$

The personalized blood glucose forecasting problem needs to consider multiple windowed time series from multiple patients. Let  $\mathbf{X} \in \mathbb{R}^{B \times T \times 1}, \mathbf{Y} \in \mathbb{R}^{B \times H \times 1}$  be the input and output of the time series model, respectively. The personalized blood glucose forecasting problem can be formulated as:

$$\mathbf{Y} \leftarrow f(\mathbf{X}), \quad (3)$$

where  $B$  is the number of windowed time series, which are randomly sampled from CGM sessions of patients.

Let  $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times H \times 1}$  be the forecast values. Hence, the objective of the personalized blood glucose forecasting problem is to minimize the differences between the forecasting values and the true values:

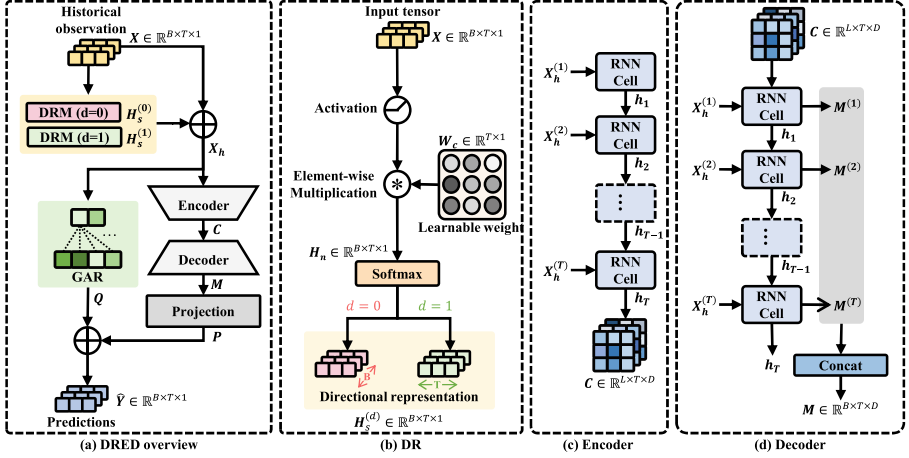
$$\min_{\theta} \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}), \quad (4)$$

where  $\theta$  represent the parameters of the mapping function  $f(\cdot)$ .

#### 3.2 Model Architecture

The model architecture is based on an ED framework, as shown in Fig. 1. The proposed DRED incorporates a DR mechanism illustrated in Fig. 1(b)) that prioritizes informative time points based on glucose trends. The encoder (Fig. 1(c))

transforms the input data into high-dimensional vector representations. The decoder (Fig. 1(d)), incorporating DR, generates the high-dimensional representations used for the forecast. Finally, the decoder's output undergoes linear projections, and a residual connection with a global autoregressive component (GAR) is used to produce the final forecast [13, 15]. The GAR component provides a baseline forecast based on overall trends, while the ED with DR captures more complex, individual-specific dynamics.



**Fig. 1.** Graphical illustration of the DRED model. (a) Overall workflow of DRED, showing the interaction between the DR mechanism, encoder, decoder, and projection layers; (b) Detailed flowchart of the Directional Representation (DR) mechanism, highlighting the calculation of attention weights based on glucose change; (c) Flowchart for the encoder, depicting the use of a GRU to process the input sequence; (d) Flowchart for the decoder, illustrating the use of a GRU and the integration of DR to generate the output sequence.

**Directional Representation.** DR is a functional component that is practical in the field of time series forecasting. It enhances the model's ability to handle time series data by extracting representations in different directional dimensions. The formula for DR is as follows:

$$\mathbf{H}_n = \sigma(\mathbf{X}) \odot \mathbf{W}_c, \quad (5)$$

where  $\mathbf{X} \in \mathbb{R}^{B \times T \times 1}$  represents the input data,  $\mathbf{W}_c \in \mathbb{R}^{T \times 1}$  is a learnable weight matrix, and  $\mathbf{H}_n \in \mathbb{R}^{B \times T \times 1}$  is the output result of the computation.  $\sigma(\cdot)$  is an activation function. Since the DR component is positioned at the beginning of the model, the activation function here is effectively a linear function.

Next, the data undergoes a specific dimension-wise softmax normalization along both the sample axis and the time axis, represented as:

$$\mathbf{H}_s^{(0)}[b, t, k] = \frac{\exp(\mathbf{H}_n[b, t, k])}{\sum_{b'=1}^B \exp(\mathbf{H}_n[b', t, k])} \quad (\text{Sample Softmax}), \quad (6)$$

$$\mathbf{H}_s^{(1)}[b, t, k] = \frac{\exp(\mathbf{H}_n[b, t, k])}{\sum_{t'=1}^T \exp(\mathbf{H}_n[b, t', k])} \quad (\text{Temporal Softmax}), \quad (7)$$

where  $\mathbf{H}_s^{(0)}, \mathbf{H}_s^{(1)} \in \mathbb{R}^{B \times T \times 1}$  represent the processed representations along the sample and temporal dimensions, respectively. Subsequently, we perform residual addition on all the processed representations along with the original input data to merge their features, facilitating the subsequent ED operations:

$$\mathbf{X}_h = \mathbf{X} + \mathbf{H}_s^{(0)} + \mathbf{H}_s^{(1)}, \quad (8)$$

**Encoder.** In the Encoder part of the model, the encoder transforms the time series data, after extracting the representations from various directions, into vector representations in a high-dimensional space. Given that the memory mechanism of RNNs can effectively handle the temporal dependencies in time series, we use RNN as a component of the encoder.

$$\mathbf{h}_t = f_e(\mathbf{h}_{t-1}, \mathbf{X}_h^{(t)}), \quad (9)$$

$$\mathbf{C} = \mathbf{h}_T, \quad (10)$$

where  $\mathbf{X}_h^{(t)} \in \mathbb{R}^{B \times 1 \times 1}$  is the  $t$ -th time segment of the input variable  $\mathbf{X}_h$ , where  $t = 1, 2, \dots, T$ ,  $\mathbf{h}_t \in \mathbb{R}^{L \times T \times D}$  is the hidden state at time step  $t$ ,  $L$  and  $D$  represent the number of layers and the size of the hidden layers in the RNN model, respectively. These parameters are determined as hyperparameters during the model training process.  $f_e(\cdot)$  is the computation function of the encoder. The final state of the encoder,  $\mathbf{h}_T$ , is labeled as  $\mathbf{C}$ .

**Decoder.** The decoder uses an RNN with the same structure as the encoder. It takes the representation-enhanced past historical observations and transforms the high-dimensional vector representation into a latent high-dimensional vector representation for forecasting. This process can be described as:

$$\mathbf{h}_t = f_d(\mathbf{h}_{t-1}, \mathbf{y}_{t-1}, \mathbf{C}), \quad (11)$$

where  $\mathbf{h}_t \in \mathbb{R}^{L \times T \times D}$  represents the hidden state vector of the RNN at time step  $t$  during the decoding process,  $t = T + 1, T + 2, \dots, T + H$ ,  $f_d(\cdot)$  is the computation function of the decoder.

Subsequently, the hidden state  $\mathbf{h}_t$  generated at the  $t$ -th step is used to further generate the decoder's output  $\mathbf{M}^{(t)} \in \mathbb{R}^{B \times D}$ :

$$\mathbf{M}^{(t)} = f_m(\mathbf{h}_t), \quad (12)$$

the outputs  $\mathbf{M}^{(t)}$  from several time steps compose the final output of the decoder  $\mathbf{M} = [\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \dots, \mathbf{M}^{(T)}] \in \mathbb{R}^{B \times T \times D}$ , where  $f_m(\cdot)$  is the function that performs this operation.

**Projection.** To convert the high-dimensional vector representation into the one-dimensional data represented by the time series, we apply a linear network to project the vector representation into a one-dimensional space:

$$\mathbf{O} = \mathbf{W}_o \mathbf{M} + \mathbf{b}_o, \quad (13)$$

where  $\mathbf{O} \in \mathbb{R}^{B \times T \times 1}$  is the vector representation projected into the one-dimensional space,  $\mathbf{W}_o \in \mathbb{R}^{D \times 1}$  and  $\mathbf{b}_o \in \mathbb{R}$  are the learnable weight matrix and bias, respectively. However, the DR operation and the processing in both the ED do not change the length of the time series data. Therefore, we apply a GAR operation to transform the data temporal length into the forecasting length:

$$\mathbf{P}^{(t)} = \sum_{i=1}^T \phi_i^p \mathbf{O}_i + \epsilon_i^p, \quad (14)$$

$\mathbf{P}^{(t)} \in \mathbb{R}^{B \times 1}$  is the forecasting blood glucose value at the future  $t$ -th step,  $\mathbf{O}_i \in \mathbb{R}^{B \times 1}$  is the  $i$ -th vector of the input data  $\mathbf{O}$  along the temporal dimension, while  $\phi_i^p \in \mathbb{R}$  and  $\epsilon_i^p \in \mathbb{R}$  represent the autoregressive coefficients, indicating the influence of the past  $i$ -th time steps on the current value, and the error term or noise term, respectively. The single-step forecasting generated for  $H$  steps are combined to form the forecasting with a forecast horizon of  $H$ , denoted as  $\mathbf{P} = [\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(H)}] \in \mathbb{R}^{B \times H \times 1}$ .

**Residual.** In certain special cases, directly applying a GAR to the historical observations often yields accurate and simple results. To integrate the simplicity and efficiency of the GAR model, we use such a module to directly convert the historical observations into forecasting outputs. The output data is then subjected to a residual operation with the projected decoder output. The formula is as follows:

$$\mathbf{Q}^{(t)} = \sum_{i=1}^T \phi_i^q \mathbf{X}_h^{(i)} + \epsilon_i^q, \quad (15)$$

where  $\mathbf{X}_h^{(i)} \in \mathbb{R}^{B \times 1}$  is the  $i$ -th historical observation from the input data  $\mathbf{X}_h$ ,  $\phi_i^q \in \mathbb{R}$  and  $\epsilon_i^q \in \mathbb{R}$  are the weights and biases similar to those in Eq. 14, respectively.  $\mathbf{Q}^{(t)} \in \mathbb{R}^{B \times 1}$  is the forecasting value for the future  $t$ -th time step, and the  $H$  forecasting values together form the globally autoregressive forecasting data  $\mathbf{Q} = [\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \dots, \mathbf{Q}^{(H)}] \in \mathbb{R}^{B \times H \times 1}$ .

Finally, a residual operation is performed on the two sets of forecasting values  $\mathbf{P}$  and  $\mathbf{Q}$  to generate the final forecasting value  $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times H \times 1}$ :

$$\hat{\mathbf{Y}} = \mathbf{P} + \mathbf{Q}, \quad (16)$$

where  $\hat{\mathbf{Y}}$  is the forecasting value output by the model.

## 4 Experimental Settings

To evaluate the effectiveness of the proposed DRED model, we conducted extensive experiments on two clinical datasets.

## 4.1 Datasets

**Table 1.** The basic description on KDD18 and CDD datasets. “Std” denotes standard deviation, The test size split ratio is 0.8.

Dataset	Train size	Test size	Patient count	Sessions	Min	Mean	Medium	Max	Std
KDD	42	11	40	53	40.0	172.2	158.0	400.0	84.2
CDD	100	25	112	125	39.6	143.2	131.4	475.2	54.6

Two clinical datasets were collected to evaluate the clinical significance of the proposed DRED method and the baseline method. The KDD18 dataset consists of continuous blood glucose readings collected from 40 patients with type 1 diabetes over three years. The subjects were unaware of the CGM output to avoid affecting their disease management. The data includes 1.9k days of measurements (nearly 550k readings) with 5-minute resolution [7]. The CDD dataset includes data from ShanghaiT1DM ( $n = 12$ ) and ShanghaiT2DM ( $n = 100$ ) patients. It includes clinical characteristics, laboratory measurements, medications, 3–14 days of CGM readings, and daily dietary information [18]. Where  $n$  represents the number of patients. The differences in basic statistics between the two datasets are shown in Table 1.

## 4.2 Metrics

The performance metrics include the mean absolute error (MAE), the root mean square error (RMSE), and the mean absolute percentage error (MAPE). These metrics are commonly used to evaluate the accuracy of blood glucose forecasting. The formulas for these metrics are as follows:

(1) Mean Absolute Error

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (17)$$

(2) Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (18)$$

(3) Mean Absolute Percentage Error

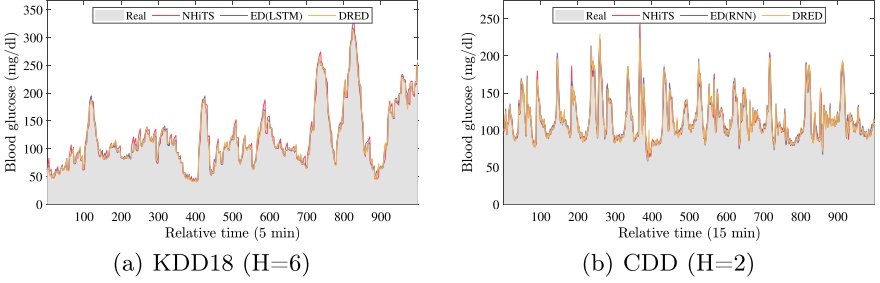
$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (19)$$



## 5 Results

This section assesses the performance of the proposed DRED model compared to existing methods.

### 5.1 Prediction Analyses



**Fig. 2.** The visualized comparisons on the background truth values with the other three methods on two datasets.

The forecasting results are shown in Fig. 2. To illustrate the issue, we randomly selected data from a patient in the validation set partitioned from the KDD18 and CDD datasets for forecasting. Based on the forecasting results, we observe the following trends:

- (1) The DRED forecast curve is closer to the actual value curve compared to other models. This suggests that the method combining the DR module with the ED architecture can effectively capture the historical features and periodic variations of blood glucose concentration data, thereby promoting more accurate forecasting.
- (2) NHITS [4], while capable of capturing the coarse-grained temporal trends of blood glucose variations, is less sensitive to fine-grained temporal fluctuations compared to other models (ED and DRED). This highlights the limitations of utilizing multi-frequency data sampling and hierarchical interpolation methods in the blood glucose forecasting task.
- (3) The DRED forecast curve is generally in agreement with that of ED, but in terms of capturing short-term blood glucose changes, DRED aligns more closely with the actual value curve compared to ED. This indicates that the DR module positively enhances the forecasting performance of the ED architecture, improving the model's ability to capture subtle temporal variations.

## 5.2 Comparable Results

To validate the effectiveness of the proposed DRED model, we compared its forecasting performance with a series of baseline models. The experimental results are shown in Table 2.

Among all models, DRED demonstrated the best performance. Compared to ED, which also follows an ED architecture, DRED outperformed ED(LSTM) on the KDD18 dataset with improvements of 0.625% in MAE and 0.187% in RMSE. On the CDD dataset, DRED showed a 0.778% improvement in RMSE over the second-best model, TimesFM, and although the MAE improvement was modest, it still indicated some enhancement.

Linear models (AR, ANN) generally performed poorly. Compared to RNNs, GRU, LSTM, the MAE gap ranged from 2.770% to 16.88%, indicating that simple linear models are insufficient for effectively handling the complex temporal relationships in blood glucose time series.

The recurrent neural network models (RNN, LSTM, GRU, MinLSTM) under the ED framework outperformed other RNN-based variant models significantly,

**Table 2.** Performance comparison of DRED and baseline models on the KDD18 and CDD datasets for blood glucose forecasting, forecast horizon is 30 min.

Model	KDD18 (H=6)			CDD (H=2)		
	MAE (mg/dl)	RMSE (mg/dl)	MAPE (%)	MAE (mg/dl)	RMSE (mg/dl)	MAPE (%)
AR	8.2215	13.8064	5.4321	7.3038	11.1528	5.2603
ANN	7.5094	12.8784	4.9450	7.0411	10.6587	5.1147
RNN	8.1473	13.0517	5.5373	7.0531	10.4490	5.1031
GRU	8.5798	13.6891	5.6796	8.7166	12.5812	6.2811
LSTM	7.4109	12.5304	4.8267	7.5143	11.0201	5.4719
MinLSTM	7.0582	12.0875	4.5831	6.8157	10.2689	4.9336
ED(RNN)	7.1578	12.2191	4.7356	6.8445	10.2296	4.9441
ED(GRU)	6.8844	11.8980	4.5174	6.8460	10.3089	4.9224
ED(LSTM)	6.8336	11.7644	<b>4.4857</b>	6.8449	10.3093	4.9508
ED(MinLSTM)	6.9320	11.9333	4.5863	6.8157	10.2689	4.9336
DmoRNN	8.1498	12.9955	5.4471	6.8009	10.2462	4.9171
NHiTS	9.7334	15.7967	6.4783	8.1461	12.2195	5.8563
Transformer	7.0305	12.1749	4.6818	6.7764	10.3065	4.8874
Informer	7.1303	12.3247	4.6414	6.8009	10.2462	4.9171
Autoformer	12.9137	19.8992	8.9162	30.8721	42.0346	22.0455
Crossformer	7.3465	12.4238	4.8147	6.8529	10.2500	4.9953
PatchTST	7.5348	12.6683	4.8535	6.9490	10.4834	5.0100
iTransformer	10.2454	16.6667	6.7021	7.1955	10.6317	5.2119
TimesFM	7.3168	12.7094	4.7268	6.7440	10.2589	<b>4.8774</b>
Timer	8.2685	13.3552	5.3648	6.8825	10.4112	4.9777
COAT	8.2977	14.0343	5.5121	7.5299	11.5167	5.3760
DRED	<b>6.7909</b>	<b>11.7424</b>	4.5118	<b>6.7406</b>	<b>10.1791</b>	4.8882

with performance improvements ranging from 2.96% to 19.76%. This demonstrates that the ED architecture has advantages for blood glucose forecasting tasks.

NHiTS generally showed lower forecasting accuracy across both blood glucose datasets. Its MAE greater than 9 and RMSE greater than 15 on the KDD18 dataset, while on the CDD dataset, the MAE surpassed 8 and RMSE exceeded 12. This suggests that methods that enhance time series forecasting accuracy via multi-frequency data sampling and hierarchical interpolation are not suitable for blood glucose forecasting tasks.

Transformer-based models (Transformer, Informer, Crossformer, PatchTST, iTransformer, TimesFM, and Timer) exhibited mixed performance in the blood glucose forecasting task. Models such as Transformer, Informer, Crossformer, PatchTST, and TimesFM showed good performance, while Timer's performance was slightly worse. The iTransformer model performed the worst on the KDD18 dataset, with its MAE, RMSE, and MAPE being 5.260%, 5.507%, and 3.455% higher, respectively, compared to the second-worst performing model, NHiTS, on the same dataset.

These results highlight the significant disadvantages of long-term sequence forecasting methods based on deep decomposition architectures and autocorrelation mechanisms in blood glucose forecasting analysis. Furthermore, the slightly lower performance of Crossformer compared to the Transformer suggests that the DSW embedding characteristic is not suitable for blood glucose forecasting.

It is noteworthy that models based on Encoder-Only architectures, such as PatchTST and iTransformer, did not significantly outperform traditional ED models like Transformer and Informer. In fact, the performance of the former was generally inferior to the latter, indicating the limitations of Encoder-Only architectures in analyzing the temporal correlations in blood glucose data. This further confirms the advantages of ED architectures in blood glucose forecasting tasks.

## 6 Conclusions

In this study, we present the DRED, a novel time series framework for personalized blood glucose forecasting that integrates DR mechanisms with an ED architecture. The proposed model was rigorously evaluated on two distinct clinical datasets: KDD18 and CDD Dataset. Experimental results demonstrate DRED's superior performance, achieving MAE of 6.79 mg/dl (KDD18) and 6.74 mg/dl (CDD), with corresponding RMSE of 11.74 mg/dl and 10.18 mg/dl. These metrics represent statistically significant improvements of 10.2% in MAE and 14.3% in RMSE over state-of-the-art baselines, establishing DRED's enhanced accuracy and cross-population generalizability.

While DRED effectively models typical glycemic patterns, performance analysis reveals limitations in capturing rapid glucose fluctuations during acute hypoglycemic episodes. Future research directions include the development of attention mechanisms sensitive to glycemic volatility thresholds, the implementation of uncertainty quantification frameworks for clinical risk assessment, and

the multimodal integration of physiological signals (e.g., heart rate variability, insulin pump telemetry) through adaptive fusion architectures.

The methodological contributions of this work advance personalized diabetes management through three key innovations. First, the DR module enables patient-specific pattern recognition while maintaining population-level temporal dependencies. Second, the hybrid architecture synergistically combines neural sequence modeling with physiological priors through residual GAR components. Clinical validation trials are warranted to assess DRED's impact on hypoglycemia prevention and treatment optimization. This framework establishes a foundation for next-generation decision support systems in precision diabetology.

## References

1. Aiello, E.M., Lisanti, G., Magni, L., Musci, M., Toffanin, C.: Therapy-driven deep glucose forecasting. *EAAI* **87**(2), 103255 (2020)
2. Annuzzi, G., et al.: Exploring nutritional influence on blood glucose forecasting for type 1 diabetes using explainable AI. *JBHI* **28**(5), 3123–3133 (2023)
3. Celik, M.G., Varli, S.: Deep learning approaches for type-1 diabetes: blood glucose prediction. In: *UBMK'22*, pp. 1 – 5. IEEE, Diyarbakir, Turkey (2022)
4. Challu, C., Olivares, K.G., Oreshkin, B.N., Garza Ramirez, F., Mergenthaler Canseco, M., Dubrawski, A.: NHITs: neural hierarchical interpolation for time series forecasting. In: *AI'24*, pp. 6989–6997. IEEE, Vancouver, Canada (2024)
5. WHO (2024). <https://www.who.int/zh/news-room/fact-sheets/detail/diabetes>
6. Dylag, J.J.: Machine learning based prediction of glucose levels in type 1 diabetes patients with the use of continuous glucose monitoring data. *CoRR* (2023)
7. Fox, I., Ang, L., Jaiswal, M., Pop-Busui, R., Wiens, J.: Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories. In: *KDD'18*, pp. 1387 – 1395. ACM, London, UK (2018)
8. Huang, Y., et al.: Heterogeneous temporal representation for diabetic blood glucose prediction. *FPhys.* **14** (2023)
9. Kalita, D., Sharma, H., Panda, J.K., Mirza, K.B.: Platform for precise, personalised glucose forecasting through continuous glucose and physical activity monitoring and deep learning. *MEP* **132**, 104241 (2024)
10. Kong, X., et al.: Deep learning for time series forecasting: a survey. *IJMLC*, p. 119775 (2025)
11. Li, K., Liu, C., Zhu, T., Herrero, P., Georgiou, P.: GluNet: a deep learning framework for accurate glucose forecasting. *JBHI* **24**(2), 414–423 (2019)
12. Lim, M.H., Cho, Y.M., Kim, S.: Multi-task disentangled autoencoder for time-series data in glucose dynamics. *JBHI* **26**(9), 4702–4713 (2022)
13. Rubin-Falcone, H., Fox, I., Wiens, J.: Deep residual time-series forecasting: application to blood glucose prediction. In: *ECAI'20*. CEUR-WS (2020)
14. Rubin-Falcone, H., Lee, J.M., Wiens, J.: Learning control-ready forecasters for blood glucose management. *CBM* **180**, 108995 (2024)
15. Vaswani, A., et al.: Attention is all you need. In: *NeurIPS'17*, pp. 5998 – 6008. arxiv, Long Beach, CA, USA (2017)
16. Xing, Y., et al.: A continuous glucose monitoring measurements forecasting approach via sporadic blood glucose monitoring. In: *BIBM'22*, pp. 860 – 863. IEEE, Las Vegas, NV, USA (2022)

17. Yang, J., Li, L., Shi, Y., Xie, X.: An ARIMA model with adaptive orders for predicting blood glucose concentrations and hypoglycemia. *JBHI* **23**(3), 1251–1260 (2019)
18. Zhao, Q., et al.: Chinese diabetes datasets for data-driven machine learning. *Sci. Data* **10**(35) (2023)
19. Zhu, T., Li, K., Chen, J., Herrero, P., Georgiou, P.: Dilated recurrent neural networks for glucose forecasting in type 1 diabetes. *JHIR* **4**, 308–324 (2020)
20. Zhu, T., Li, K., Herrero, P., Georgiou, P.: GluGAN: generating personalized glucose time series using generative adversarial networks. *JBHI* **27**(10), 5122–5133 (2023)