Continuous Glucose, Insulin and Lifestyle Data Augmentation in Artificial Pancreas Using *Adaptive* Generative and Discriminative Models

Deepjyoti Kalita, *Student Member, IEEE*, Hrishita Sharma, *Student Member, IEEE*, and Khalid B. Mirza, *Senior Member, IEEE*

***Abstract*—Artificial pancreas requires data from multiple sources for accurate insulin dose estimation. These include data from continuous glucose sensors, past insulin dosage information, meal quantity and time and physical activity data. The effectiveness of closed-loop diabetes management systems might be hampered by the absence of these data caused by device error or lack of compliance by patients. In this study, we demonstrate the effect of output sequence length-driven generative and discriminative model selection in high quality data generation and augmentation. This novel generative adversarial network (GAN) based architecture automatically selects the generator and discriminator architecture based on the desired output sequence length. The proposed model is able to generate glucose, physical activity, meal information data for individual patients. The discriminative scores for Ohio T1DM (2018) dataset were 0.17±0.03 (Inputs: CGM, CHO, Insulin) and 0.15±0.02 (Inputs: CGM, CHO, Insulin, Heart Rate, Steps) and for Ohio T1D (2020) dataset was 0.16±0.02 (Inputs: CGM, CHO, Insulin) and 0.15±0.02 (Inputs: CGM, CHO, Insulin, acceleration). A mixture of generated and real data was used to test predictive scores for glucose forecasting models. The best RMSE and MARD achieved for OhioT1DM patients were 17.19±3.22 and 7.14±1.76 for PH=30 min with CGM, CHO, Insulin, heartrate and steps as inputs. Similarly, the RMSE and MARD for real+synthetic data were 15.63±2.57 and 5.86±1.69 respectively. Compared to existing generative models, we demonstrate that sequence length based architecture selection leads to better synthetic data generation for multiple output sequences (CGM, CHO, Insulin) and forecasting accuracy.**

***Index Terms*—Continuous glucose monitoring, data augmentation, discriminative score, generative adversarial network, predictive scores, TimeGAN.**

I. Introduction

RESEARCH has consistently demonstrated that continuous monitoring of glucose and other physiological parameters leads to better clinical outcomes in patients suffering from diabetes mellitus [1], [2]. Closed-loop diabetes management systems, also referred to as *artificial pancreas*, are recent therapies used in management of diabetes that integrate automated administration of insulin with continuous glucose monitoring (CGM) [1], [3]. Real-time glucose data from CGM sensors enables the closed-loop system's algorithm to determine and give accurate insulin doses in accordance with a person's glucose trends [4]. By improving glucose regulation and lowering the risk of hypo- and hyperglycemia, this dynamic and responsive method seeks to mimic the operation of a healthy pancreas [4].

Current artificial pancreas are a *hybrid closed-loop* system that require manual intervention to calibrate CGM sensors, collect and combine data on meals and physical activity [5]. While closed-loop systems effectively employ data derived from continuous glucose monitoring (CGM) to regulate insulin delivery, the automated operation of the system is interfered with, when people have to manually input information about their meals and physical activity [5]. This manual entry is difficult and cumbersome, and it could result in user compliance concerns, delayed responses, and human errors [6]. It also leads to missing datapoints. Automated meal and activity data integration into closed-loop systems is essential to achieving seamless and efficient glycemic control. The majority of currently available hybrid CGM devices measure glucose levels in the interstitial fluid (ISF). Compared to blood glucose levels, ISF glucose levels move more slowly due to physiological reasons [7]. Hence, it is crucial to *forecast* glucose levels in order to avoid hyperglycemia resulting from inaccurate or delayed insulin administration. Yet, calculating the amount of insulin on glucose forecasts increases the danger of insulin overdosing due to inaccurate insulin dose estimation, leading to hypoglycemia. Therefore accuracy is an important criterion, data-driven techniques like deep learning have lately been used to obtain high accuracy [8], [9].

The prevalence of missing data points in CGM sensor, meal and physical activity datasets is a crucial problem to be addressed in artificial pancreas.Missing data can occur for a number of causes such as sensor insertion or replacement transitions, sensor malfunctions, signal interference, low battery/power loss, user behaviour, data transmission errors, sensor warm-up periods, skin irritation or dislodgment, and sensor end-of-life [10].

In addition to CGM data, other sensors used to record physical activity can also be used to improve glucose forecasts [11].

Manuscript received 10 November 2023; revised 10 March 2024; accepted 30 April 2024. Date of publication 6 May 2024; date of current version 7 August 2024. This work was supported by the Department of Science and Technology (DST), Government of India, under Grant TDP/BDTD/01/2021. *(Corresponding author: Khalid. B Mirza.)*

The authors are with the Department of Bioetchnology and Medical Engineering, National Institute of Technology, Rourkela 769008, India (e-mail: deepjyoti_kalita@nitrkl.ac.in; baigm@nitrkl.ac.in).

Digital Object Identifier 10.1109/JBHI.2024.3396880

Another aspect is the incorporation of meal and past insulin dosage information [11]. Due to lack of patient compliance, data points can be missed. In order to replace data which is similar to the daily habits of individuals, it is crucial to use the correct *interpolation* model to develop a *personalised* approach. Generative Adversarial Networks provide immense flexibility in modelling different types of datasets [12].

Generative Adversarial Networks for generation have been developed and tested in many applications [13]. In this study, a novel approach is put forth that, to the best of our knowledge, is the first hybrid framework capable of producing accurate glucose time series and replacing missing data points with the identical timestamps. Additionally, it addresses missing values for insulin levels, values for physical activity like steps, heart rate, acceleration and carbs from meal intake within the same time series data. The problem statement involving data augmentation to enhance the effectiveness of deep learning-based short and midterm glucose forecasting algorithms for glycemic control are two use cases for the suggested architecture in this work. A recent GAN-based model, TimeGAN, was developed for generation of time series [14]. In our work, a modified TimeGAN architecture is used to generate continuous glucose data, meal information data, insulin dosage and physical activity data. In the proposed model, a GRU (Gated Recurrent Unit) or LSTM (Long Short-Term Memory) is selectively used as a generator and discriminator depending on sequence length, with a supervisor network for enhancing latent representations. In this model, adversarial training is performed not only in the data space but also in the latent space.Previous works have demonstrated the increasing utility of GANs in time series data augmentation [15], [16], [17]. Some of the earliest works in GAN-based time series augmentation is use of GAN/WGAN with TSF-MSE based loss function for generation of EEG signals [15]. Private, synthetic medical data was also generated using Rényi Differential Privacy and Convolutional Generative Adversarial Networks (RDP-CGAN) [16]. In the domain of diabetes management, blood glucose prediction was attempted, where synthetic data generation of blood glucose levels was performed using transfer learning and TimeGAN [14]. The reported performance metrics were further improved using another TimeGAN based model where RNNs are used for implementing generators and discriminators [17].

Furthermore, by extending LSTM-based GANs to the RC-GAN (recurrent conditional GAN), Esteban et al. [18] made it possible to generate medical time series data with absolute values, conditional inputs with differential personalised training. Bidirectional RNNs with the LSTM cells were first used in [19] to create music using a GAN model. WaveGAN, a method for producing synthetic audio, was developed in [20].

In Sections II and III the proposed architecture and its step by step implementation process has been described. The performance metrics for the proposed approach have been presented and with detailed comparison using clinical dataset in Section IV. This is followed by Discussion in Section V followed by Conclusion in Section VI.

II. Methodology

The network for generation of time series glucose and lifestyle data consists of GANs in which the generator and the discriminator are designed to take data from multiple sensors. The sensor data consists of CGM data, data from physical activity monitoring devices, insulin dosage data and meal information data. In this Section, the detailed architecture of the network is presented. In order to train and test the network, OhioT1DM dataset was used. [21].

A. Dataset

The OhioT1DM dataset consists of glucose, insulin and lifestyle datasets of patients suffering from Type 1 Diabetes. Data was collected in two phases, 2018 and 2020. The 2018 dataset consisted of 8 weeks data of 6 individuals, which was increased to 12 individuals in 2020. All these individuals used closed loop diabetes management system consisting of a glucose sensor (Enlite sensor, Medtronic) and an insulin pump (530 G, Medtronic). Their meal information was manually recorded and their physical activity data was recorded using Empatica or Basis band.

As shown in Fig. 1, the time series dataset consists of missing datapoints in (CGM) sensor values sometimes for any of the reasons like sensor insertion or replacement transitions, sensor malfunctions, user behavior, low battery/power loss events, data transmission errors, and sensor warm-up times [22].

Since insulin, meal and the physical activity data (like: heart rate, steps, acceleration etc.) are closely correlated with glucose dynamics, they can be used to generate accurate glucose time series during the missing values utilizing generative models. Time series data can typically be broken down into two types of features: static features and temporal features. Static features are variables that do not change during the course of the time series or for a long time. Due to significant volatility in real-world circumstances and high temporal connections between glucose levels and external factors, it is challenging to map time series data in applications such as artificial pancreas with static features, first suggested in [14].

In the paper, as we formulate the time series generation problem, S represents the static component, and X for the temporal component. As a result, inputs such as meal's carbohydrate content, insulin bolus, and physical activity data were used in this study as a multivariate time series with a sequence length T. In this case, the model's inputs can be viewed as a tuple of $(S, X_{1:t})$ with a joint distribution of p. A generative model's goal is to learn using training data, which is an approximate representation of the initial distribution $p(S, X)$ or $\hat{p}(S, X)$. Here p represents distribution's density function over real data and $\hat{p}$ describes estimated output distribution of the generator. Given the length, dimensionality, and distribution of the data, it may be challenging to optimise this high-level goal using the conventional GAN framework. So another goal is to concurrently learn this joint distribution and the autoregressive decomposition of $P(S, X_{1:T}) = p(S) \prod_t p(X_t|S, X_{1:t-1})$. The

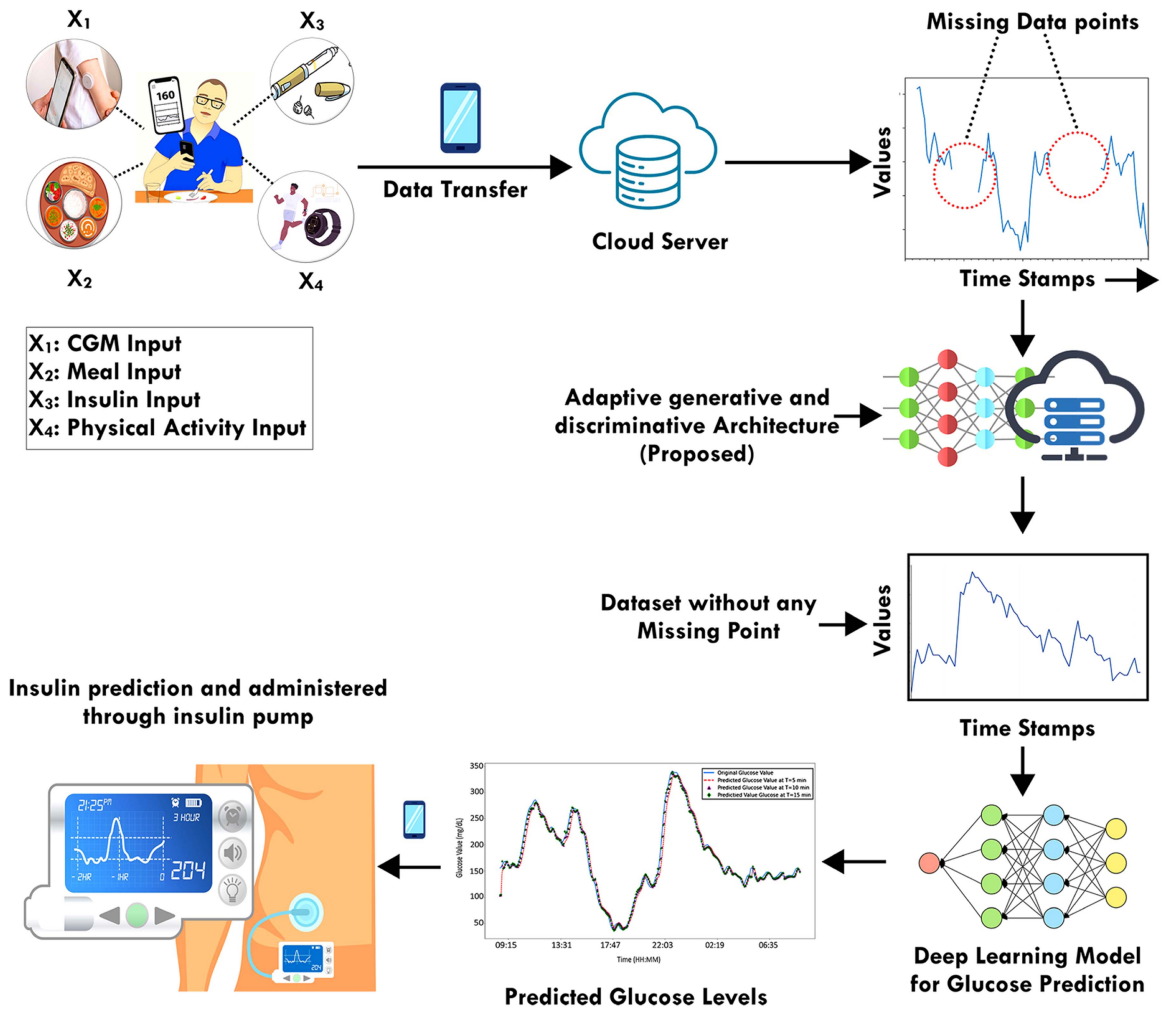

Fig. 1. Overview of Data augmentation process in closed-loop diabetes management using adaptive generative and discriminative models.

two objective functions that result are as follows [14]:

$$min_{\hat{p}} D(p(S, X_{1:T}) \,||\, \hat{p}(S, X_{1:T})) \tag{1}$$

$$min_{\hat{p}} D(p(X_t|S, X_{1:t-1}) \,||\, \hat{p}(X_t|S, X_{1:t-1})) \tag{2}$$

where D is a suitable way to compare distributions' distances from one another in (1) and 2. In the GAN framework, the (1) assumes the form of the Jensen-Shannon divergence [23] under an ideal discriminator. The Kullback-Leibler divergence [13] is the outcome of maximum-likelihood (ML) training employing the original data as supervision. Equation (1) showed similarity between two probability distributions and (2) showed conditional distributions.

B. Proposed Generative Model

The embedding function, recovery function, sequence generator, and sequence discriminator are the four network elements which make together proposed GAN as shown in Fig. 2. The significant discovery is that the adversarial components (last two) and autoencoding components (first two) are trained simultaneously so that our proposed GAN all together learns to encode all the features, produce representations, and iterate over time. The embedding network provides a latent space for the adversarial network to operate in, and the supervised loss synchronises the latent dynamics for both real and artificial data. Using our proposed network, data production is optimised for high quality and adaptability across varied applications by dynamically choosing between GRU and LSTM architectures based on required output sequence lengths in the generator and discriminator block. It improves the model's adaptability by enabling it to be used for a variety of time series data generating jobs while automatically modifying its internal structure for optimum performance.

From the model perspective, the novelty lies in selective architecture selection for generators and discriminators based on length of the sequence to be generated. Furthermore, another aspect of novelty in the model is that, in the proposed model, adversarial training is done in the latent space and not in the data space only. This ensures that the generated synthetic data not only matches the distribution of the data but also has similar latent representations. Furthermore, a supervisor network which enhances the latent representations generated by the embedding network. By supervising the latent space, the model can generate synthetic sequences with improved quality and diversity.

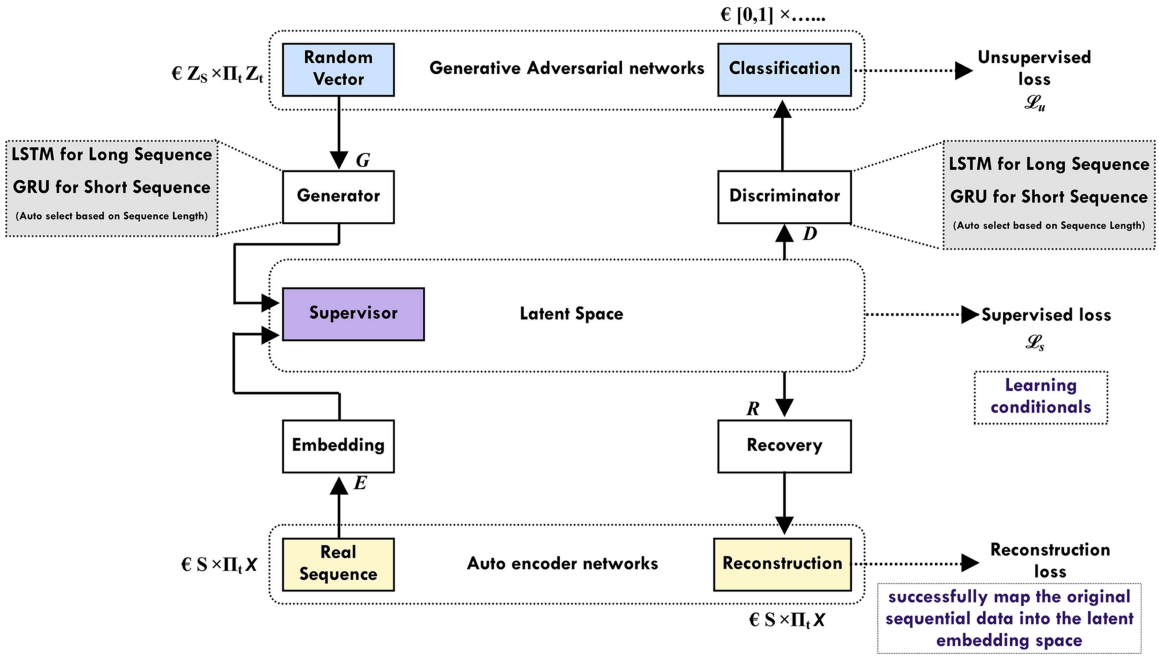

Fig. 2. The architecture of the proposed generative model is shown here with automatic model selection depending on sequence lengths. Here $\mathcal{L}_u$ and $\mathcal{L}_s$ indicate unsupervised loss and supervised loss respectively.

1) Functions for Embedding and Recovering: The autoencoder is a paradigm for unsupervised learning that combines an encoder and a decoder. In proposed architecture, the encoder (Embedder) maps the sequential data $X_1 : T$ into the latent space S, while the decoder (Recovery) reconstructs the original sequential data $X_1 : T$ from the latent space S. The embedding function converts the input time series data into a representation of a lower dimension in latent space. TimeGAN can auto-encode the data to capture the key temporal dependencies and characteristics within a small latent space [24]. To enhance the performance of data generation, this latent space enables more efficient adversarial learning and supervised learning [25]. The recovery function restores the original data space with the latent space representations. This ensures that the generated sequences are accurate and consistent with the distribution of the input data. The latent vector spaces that correspond to the feature spaces S and X are denoted as $H_S; H_X$. The latent codes $h_s, h_{1:T} = e(s, x_{1:T})$ of static and temporal features are then obtained via the embedding function $e : S \times \prod_t X \to H_S \times \prod_t H_X$. So, we can write,

$$h_s = eS(s) \tag{3}$$

$$h_t = eX(h_S, h_{t-1}, x_t) \tag{4}$$

Here, $e_x : H_S \times H_X \times X \to H_X$ is a recurrent embedding structure for temporal features and $e_s : S \to H_S$ represents an embedding network for static features. The recovery function $r : H_s \times \prod_t H_x \to S \times \prod_t X$ returns static and temporal codes to their representations of features $\widetilde{s}, \widetilde{x}_{1:T} = r(h_S, h_{1:T})$ in the opposite direction. So, we can write,

$$\widetilde{S} = rs(h_s) \tag{5}$$

$$\widetilde{x}_t = rx(h_t) \tag{6}$$

where the recovery networks for static and temporal embeddings are denoted by $rx : H_X \longrightarrow X$ and $rs : H_s \longrightarrow S$, respectively. The *embedding function* enables the model to understand the underlying patterns and dependencies in the multi-dimensional input data from the CGM sensor data creation with meal, insulin, and physical activity data. The *recovery function* then makes sure that the generated sequences of meal, insulin, physical activity, and CGM glucose values are consistent with the distribution of the actual data. The model may produce customised and context-aware CGM sensor data that accurately depicts the dynamics of the person's glucose levels in response to numerous stimuli by integrating these functions with the GAN architecture and the conditioning on meal, insulin, and physical activity data.

2) Discriminator and Sequence Generator: The Sequence Generator strives to produce data that deceives the Discriminator while the Discriminator aims to correctly identify real and synthetic data. The Discriminator and Sequence Generator are trained adversarially [26]. Iteratively, this competitive training process produces high-quality synthetic time series data that closely mimic actual data [27]. With the ability to learn complex temporal patterns and dependencies present in the data, the Discriminator and Sequence Generator combined in a GAN architecture provide a powerful framework for time series data generation. This makes it suitable for tasks like producing realistic CGM sensor data with meal, insulin, and physical activity information.The generator initially outputs into the embedding space instead of directly providing synthetic output in feature space. Let Z_S and Z_X represent vector spaces where existing distributions are defined and where random vectors are generated

as input to generate H_S and H_X. Then, the generating function $g : Z_s \times \prod_t Z_X \to H_S \times \prod_t H_X$ uses a tuple of static and dynamic vectors that are random to create artificial latent codes $\widehat{h}_S, \widehat{h}_{1:T} = g(z_S, z_{1:T})$. Now we can write,

$$\widehat{h}_S = gs(zs) \tag{7}$$

$$\widehat{h}_t = gx(\widehat{h}_S, \widehat{h}_{t-1}, z_t) \tag{8}$$

Here, the generator network with static features can be expressed as $g_s : Z_s \to H_S$ and for temporal features g_x can be expressed as $gx : H_s \times H_x \times Z_x \to H_x$. The static and temporal codes are passed to the discrimination function $d : H_s \times H_x \to [0, 1] \times \prod_t [0, 1]$, which produces classifications as $\widetilde{y}_s, \widetilde{y}_{1:T} = d(\widetilde{h}_s, \widetilde{y}_{1:T})$. The discriminator's goal is to effectively learn how to tell actual data apart from generated data. By successfully classifying created data as synthetic $(y = 0)$ and real data as real $(y = 1)$ for both the static along with temporal codes, it is taught to minimise its classification error. Bidirectional recurrent networks allow the discriminator to efficiently capture temporal dependencies in the input time series data and offer helpful feedback to the generator, assisting in creating more realistic synthetic time series data. This can be expressed as

$$\widetilde{y}_s = ds(\widetilde{h}_s) \tag{9}$$

$$\widetilde{y}_t = dx\left(\underset{u_t}{\leftarrow}, \underset{u_t}{\rightarrow}\right) \tag{10}$$

As a result, the Generator network uses a four-layer RNN with gated recurrent units to produce realistic and customised sequences of CGM data and other parameters. The Discriminator network uses a bidirectional recurrent network with a feedforward output layer to distinguish between actual and artificial sequences. The quality and authenticity of the generated sequences are increased by using the temporal and static latent codes that the Supervisor and Embedder networks taught the Generator network.

3) Training and Generation Simultaneously: The embedding and recovery functions should first allow precise reconstructions $(\widetilde{S}, \widetilde{X}_{1:T})$ of the initial data $S, X_{1:T}$ from their latent representations $h_s, h_{1:T}$, strictly speaking as a reversible translation among feature and latent spaces. Therefore, the reconstruction loss can be best described as:

$$\mathcal{L}_R = \mathbb{E}_{S,X_1:T\sim P}[\|s - \widetilde{s}\|_2 + \sum \|x_t - \widetilde{x_t}\|_2] \tag{11}$$

Two different sorts of inputs are presented to the GAN architecture generator during training. In order to create the subsequent synthetic vector $\widehat{h}_t$, the generator, which is autoregressive, must first receive synthetic embeddings of its own prior outputs $\widehat{h}_S, \widehat{h}_{1:t-1}$ in pure openloop mode. Gradients are then calculated on the unsupervised loss. The likelihood of producing the right classifications $\widehat{y}_S, \widehat{y}_{1:T}$ in both the training data $h_S, h_{1:T}$ and the synthetic output $\widehat{h}_S, \widehat{h}_{1:T}$ through the generator can be maximised (for the discriminator) or minimised (for the generator), as is expected, and it can be represented as:

$$\mathcal{L}_U = \mathbb{E}_{S,X_1:T\sim P}[\log y_s + \sum \log y_t] + \mathbb{E}_{S,X_1:T\sim P}$$

$$[\log(1 - \widehat{y}_s) + \sum_t \log(1 - \widehat{y}_t)] \tag{12}$$

The generator will not be sufficiently motivated to detect the progressive conditional distribution in the data set if it depends solely on the discriminator's binary adversarial feedback. We introduce a different loss to enhance constrained learning in order to accomplish this more effectively. Alternately, we train in a closed-loop setting, where the generator is fed streams of embeddings of real data $h_{1:t-1}$ in order to produce the subsequent latent vector. Now, it is possible to compute gradients on a loss that accurately depicts the difference among probabilities $p(H_t|H_s, H_{1:t-1})$ and $\widehat{p}(H_t|H_s, H_{1:t-1})$. Consequently, the supervised loss is created by the maximum likelihood of this and is denoted by:

$$\mathcal{L}_S = \mathbb{E}_{S,X_1:T\sim P}[\sum_t \|h_t - g_x(h_S, h_t - 1, z_t)\|_2] \tag{13}$$

Here, $\mathcal{L}_U$ and $\mathcal{L}_S$ denoted as the unsupervised loss and supervised loss of the generator and discriminator accordingly.

C. Optimization

Let θ_r, θ_e, θ_d and θ_g, respectively, stand for the recovery, embedding, discriminator and generator's networks parameters. The first two parts are trained on supervised losses (L_S) as well as reconstruction losses (L_R) and expressed as follows:

$$\min_{\theta_c,\theta_r}(\lambda\mathcal{L}_S + \mathcal{L}_R) \tag{14}$$

where the two losses are balanced by the hyperparameter $\lambda \geqslant 0$. Following an adversarial training of the generator and discriminator networks, the following can be stated:

$$\min_{\theta_g}(\eta\mathcal{L}_S + \underset{\theta_d}{max}\mathcal{L}_U) \tag{15}$$

where the two losses are balanced by the hyperparameter $\eta \geqslant 0$.Thus, the generator reduces the supervised loss in addition to the unsupervised minimax game that is played over classification accuracy. As a result, our proposed architecture has been trained to encode (using feature vectors), produce (using latent representations), and iterate (over time).

III. Implementation

The model described in previous section was tested using the OhioT1DM dataset [21]. The dataset contains data from participants whose glucose levels were monitored using CGM sensors and physical activity monitoring was performed using wearables. Meal and dosage information was also collected and Incorporated in the dataset. The dataset and any details regarding the dataset, such as ethical approvals, may be requested from the original owners of the dataset [21]. The different loss functions described in this Section are used to optimise the model.

A. Data Preprocessing, Splitting and Analysis

To produce high-quality multivariate time series from clinical datasets, data preparation is a necessary step. By using a set of maximum and minimum thresholds according to physiological

features, outliers were eliminated for each input feature. Insulin doses greater than 50 units were rejected as well as negative readings for each feature and blood glucose levels greater than 500 mg/dL [17]. Notably, there are several gaps in the CGM data which can be attributed to a variety of issues, including sensor replacement, and signal loss, as were discussed in the previous section. Using min-max normalisation, all input features are scaled to a range of values. The proposed GAN model consists of a Generator that produces synthetic data, a Supervisor that guides the generator by transforming synthetic data into meaningful latent space representations, a Discriminator that distinguishes between real and synthetic data, and a Recovery and Embedder pair that forms an autoencoder, responsible for mapping real data into the latent space and then recovering it back to the original space. These components work together in a carefully designed training process to generate synthetic time series data that captures the essential characteristics of the real data. To create the all modules, we have utilised the build method inside every module design. It was built using the Keras Sequential application programming interface. Depending on the particular requirements of the formulated problem, the length of the sequences, and the features of the data being processed, the build method is chosen automatically based on the sequence length to be LSTM or GRU based.

Details of the generative adversarial model used for data augmentation for the CGM sensor data generation in diabetes control are shown in Fig. 2. The implementation part includes a description of the workflow and the specific training procedures. All of the network modules in the proposed model must be optimised using the loss functions specified in (11), (13), and (14). The training iterations for embedding learning, supervised learning, and joint learning are τ_R, τ_S, and τ_J, respectively. Using ground truth data and $\mathcal{L}_R$, we optimise the embedding and recovery networks before training the supervisor module by itself with $\mathcal{L}_S$. The final step is to jointly train all five modules using a combination of supervised and unsupervised losses. Specifically, when paired with reconstruction and unsupervised losses, two hyperparameters, η and λ, are used to modify the ratios of $\mathcal{L}_S$. Thus, the generator minimises the supervised loss in addition to the unsupervised minimax game that is played over classification accuracy. Our proposed GAN architecture is simultaneously trained to encode (feature vectors), produce (latent representations), and iterate (through time) by merging the objectives in this way. As a result, our suggested GAN model can produce synthetic glucose data with batch inputs for all other parameters, including CHO, Insulin, and physical activity.

B. Model Training Process

The training process iterates through these steps, optimizing the trainable variables of the various model components (Generator, Discriminator, Embedder, Supervisor) using the specified optimization methods. The overall objective is to train the Generator to produce synthetic time series data that closely resembles the distribution of the real data. The training continues until the model converges or achieves the desired performance. Model parameters like model_parameters, hidden_dim, seq_len (T), n_seq (features: like if we have used CGM, CHO, Insulin and Physical activity then n_seq will be 4), and gamma (γ, Discriminator loss) are used to initialise the GAN class. The architecture and hyperparameters of the GAN model are defined by these parameters.

The dataset for a single patient was split into 80% for training and 20% for testing. The training dataset was further split into 70% for training and 30% as validation dataset for optimising hyperparameters. The proposed GAN model was trained for 8000 epochs for generation of short sequences (L=24 datapoints maximum) and 15000 for generation of long sequences (L= 288 datapoints maximum), for Supervised and embedded learning. For Joint learning, the number of epochs used for training was 60000. $Early\_stopping\_patience$ was set at 50 for all types of training mentioned above. We have observed that the discriminative score does not improve significantly when the model is trained beyond the number of epochs mentioned above.

The proposed model was trained and validated on a computing platform based on the NVIDIA GPU Tesla T4 16 GB GDDR6 with CUDA Cores 2560. The software packages used are TensorFlow version: 2.15.0, Python version: 3.10.12, scikit-learn version: 1.2.2, NumPy version: 1.23.5, Pandas version: 1.5.3, Matplotlib version: 3.7.1.

IV. Results

The effect of generator or discriminator model selection based on output sequence length was tested with the help of PCA, tSNE plots and using discriminative scores. Furthermore, the performance of the generative models was also evaluated by mixing the generated data (CGM, CHO, Insulin) with clinical data and tested for improvement in glucose forecasting accuracy using deep learning for benchmarking.

A. Comparison Using t-SNE and PCA

The temporal dimension of the generated dataset was flattened following which t-distributed Stochastic Neighbour Distribution (t-SNE) [28] and Principal Component Analysis (PCA) [29] analyses were performed on the original and synthetic datasets. This displays in 2-dimensional space how closely the distribution of generated samples reflects that of the original, providing a qualitative evaluation. Fig. 3 shows the distribution with respect to the different dimensions, where each dot represents a glucose sequence, from temporal L-dimension sequential data in two dimensions.

It was observed in Fig. 3 that the spread in the generated dataset was less when GRU is used for the generation of short sequences and LSTM is used for generation of long sequences.

B. Discriminative Score

In order to differentiate between sequences from the original and produced datasets, we trained a post-hoc time series classification model by optimising a 2-layer LSTM for long sequences and a 2-layer GRU layer for short sequences. Each original sequence was first marked as real, whereas every generated sequence is marked as not real. We then provide

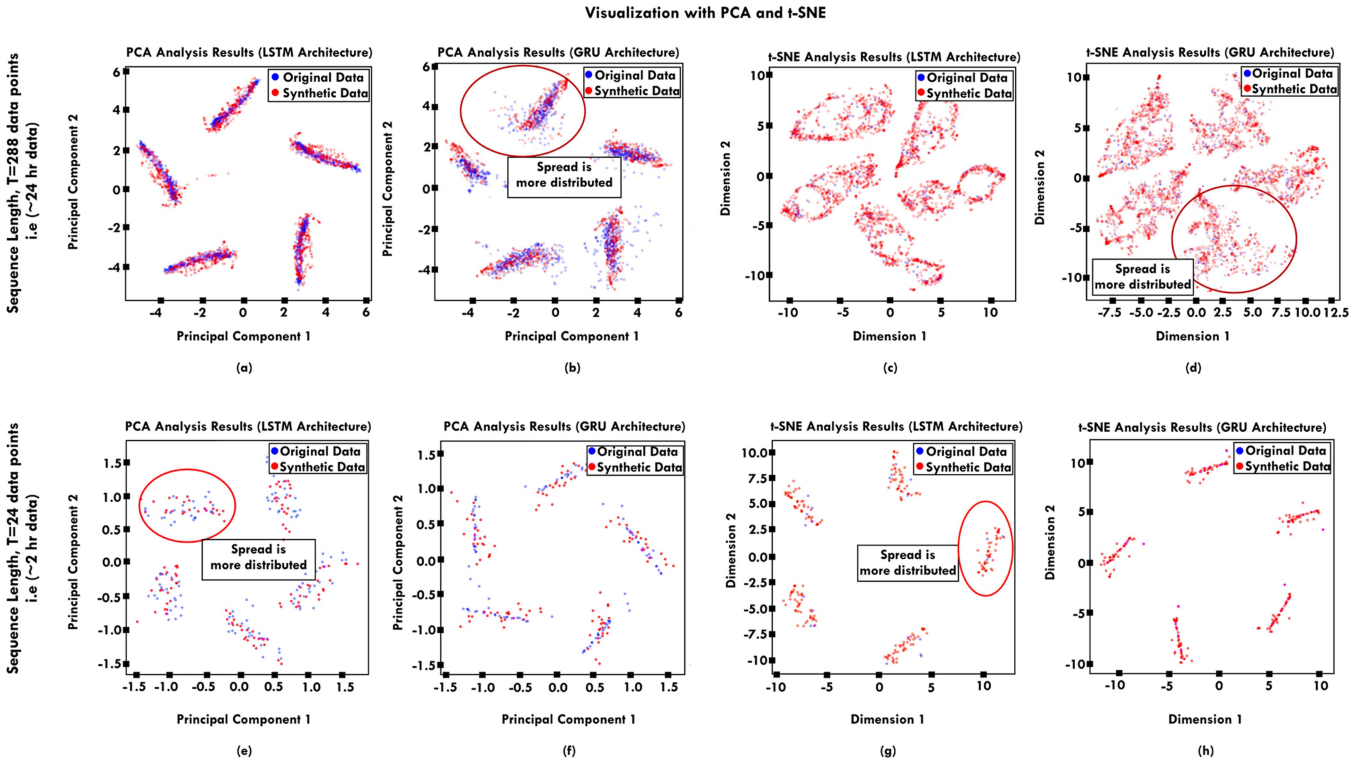

Fig. 3. The distributions of actual and synthetic glucose sequences for the OhioT1DM Dataset are displayed in red and blue dots, respectively, after PCA (left) and t-SNE (right) analysis.

TABLE I
RESULTS OF DISCRIMINATIVE SCORE EVALUATED ON THE ALL 12 OHIOT1DM SUBJECTS (2018 AND 2020 CLINICAL DATASET) VALIDATED WITH PROPOSED ARCHITECTURE

Ohio T1D Dataset (2018)			Ohio T1D Dataset (2020)		
Patient ID	**Discriminative Score (Inputs: CGM, CHO, Insulin)**	**Discriminative Score (Inputs: CGM, CHO, Insulin, Heart Rate, Steps)**	**Patient ID**	**Discriminative Score (Inputs: CGM, CHO, Insulin)**	**Discriminative Score (Inputs: CGM, CHO, Insulin, Acceleration)**
559	0.19	0.18	**540**	0.14	0.13
563	0.15	0.14	**544**	0.16	0.15
570	0.11	0.11	**552**	0.17	0.16
575	0.21	0.16	**567**	0.18	0.18
588	0.20	0.16	**584**	0.19	0.17
591	0.17	0.15	**596**	0.14	0.13
Avg. Discriminative Score	**0.17±0.03**	**0.15±0.02**	**Avg. Discriminative Score**	**0.16±0.02**	**0.15±0.02**

a quantitative evaluation by reporting the categorization error on the held-out test set. We generated a discriminative score expressed as $|accuracy - 0.5|$. The discriminative score drops when the post-hoc architecture's accuracy is close to 0.5 (i.e., a random guess), which shows that incorrect synthetic data cannot be distinguished from real data. The discriminative scores are shown in Table I.

C. Predictive Score

In order to assess the possibility of replacing missing data in the clinical dataset, a predictive model is designed with the goal of assessing the improvement in glucose forecasting accuracy with and without the synthetic data.

The model consisted of a wide-deep combination of GRU and LSTM layers, known as WDNet was used for glucose forecasting [30]. The Mean Absolute Relative Difference (MARD) and Root Mean Squared Error (RMSE) quantify the difference between predicted glucose levels and also actual glucose levels both when real data was mixed with synthetic datapoints and used as training data, as shown in Fig. 5 for CGM, Fig. 6 for CHO, Fig. 7 for insulin dosage and when not mixed with synthetic datapoints. The predictive scores for different *glucose forecasting* models are shown in Tables II and III.

For the T1D subjects from OhioT1DM, the results of PCA and t-SNE analysis are shown in Fig. 3, respectively. It should be noticed that there is considerable overlap between the

TABLE II
PERFORMANCE ANALYSIS USING RMSE (MG/DL), MAE (MG/DL) AND MARD (%) FOR GLUCOSE FORECASTING USING WDNET, LSTM, GRU MODELS ON VARIOUS AUGMENTED TRAINING DATASETS

Method Used	WDNet			LSTM			GRU		
	RMSE (mg/dL)	MAE (mg/dL)	MARD (%)	RMSE (mg/dL)	MAE (mg/dL)	MARD (%)	RMSE (mg/dL)	MAE (mg/dL)	MARD (%)
					PH= 30Min				
T1D (IN_3) 2018	17.26±3.15	12.71±2.23	7.16±1.77	23.15±2.48	14.94±2.04	9.14±2.41	23.04±2.40	14.57±2.34	8.94±2.17
AugT1D (IN_3) 2018	16.09±2.63	12.05±2.29	5.91± 1.67	22.30±2.37	13.75±2.27	8.13±2.17	22.02±2.66	13.05±2.63	7.52±1.99
T1D (IN_5) 2018	17.19±3.22	12.58±2.34	7.14±1.76	22.84±2.42	14.42±2.16	8.33±2.39	22.65±2.43	14.04±2.29	8.35±2.10
AugT1D (IN_5) 2018	**15.63±2.57**	**11.69±2.17**	**5.86±1.69**	21.71±2.81	13.15±2.37	7.42±2.11	21.25±2.73	12.92±2.43	7.12±1.81
T1D (IN_3) 2020	18.64±2.67	13.46±1.80	7.97±1.35	24.31±2.75	14.88±2.15	10.03±1.29	23.74±2.60	14.12±2.57	8.91±1.41
AugT1D (IN_3) 2020	17.12±2.62	11.97±2.14	6.75±1.26	23.27±2.55	13.98±2.21	9.14±1.26	22.53±2.53	13.07±2.64	8.05±1.56
T1D (IN_4) 2020	18.36±2.66	13.14±1.79	7.65±1.31	23.48±2.59	14.28±2.17	9.36±1.45	23.20±2.64	13.58±2.62	8.32±1.27
AugT1D (IN_4) 2020	17.11±2.35	11.90±1.73	6.78±1.15	22.17±±2.28	12.81±1.81	8.06±1.27	22.03±2.73	12.65±2.62	7.10±1.27
					PH= 60Min				
T1D (IN_3) 2018	29.63±6.16	22.25±4.83	12.58±3.36	35.59±5.10	26.60±4.46	15.25±3.80	34.98±4.27	26.04±4.65	14.70±3.82
AugT1D (IN_3) 2018	27.98±5.59	20.12±4.30	10.37±2.21	33.89±4.90	24.47±4.93	13.81±3.81	32.82±4.41	24.50±5.05	12.69±3.23
T1D (IN_5) 2018	29.41±5.92	21.96±4.67	12.33±3.15	34.97±4.85	25.90±4.49	14.43±3.82	34.52±4.17	25.57±4.77	14.29±3.70
AugT1D (IN_5) 2018	**27.41±5.44**	**19.77±4.31**	**9.99±2.16**	32.95±4.41	23.58±4.89	12.75±3.67	32.44±4.35	23.27±4.82	12.43±3.58
T1D (IN_3) 2020	30.13±3.31	22.20±2.94	13.12±2.39	36.14±3.21	26.35±3.24	15.64±2.00	35.74±4.29	27.18±4.54	14.68±1.88
AugT1D (IN_3) 2020	27.61±3.19	20.37±3.09	11.24±2.09	35.11±3.56	25.05±3.17	14.43±1.99	33.67±4.27	25.46±4.26	12.76±1.97
T1D (IN_4) 2020	29.82±3.32	22.02±3.06	12.95±2.40	35.52±3.26	25.93±3.35	15.09±2.01	35.35±4.33	26.76±4.57	14.28±1.81
AugT1D (IN_4) 2020	28.02±3.27	19.91±2.67	10.88±2.51	34.05±3.23	23.60±2.64	13.06±2.09	33.39±4.13	24.97±4.33	12.47±1.94

TABLE III
COMPARISON OF DISCRIMINATIVE AND PREDICTIVE SCORES BETWEEN THIS WORK AND RECENT WORK

	Related Works	Dataset Used	Model Used	RMSE (mg/dL)	MAE (mg/dL)	MARD (Percent)
			PH =30 Min			
Predictive Scores (Lower is the better)	[17] [2023]	OhioT1DM (Original Dataset)	LSTM	21.30±2.43	15.55±1.57	-
			DRNN	20.39±2.48	14.64±1.71	-
			SVR	23.89±4.51	17.28±2.19	-
		OhioT1DM (Augmented Dataset)	LSTM	20.28±2.42	14.42±1.42	-
			DRNN	19.80±2.35	14.10±1.55	-
			SVR	22.00±5.26	15.01±2.17	-
	This work	OhioT1DM (Original Dataset)	WDNet	17.19±3.22	12.58±2.34	7.14±1.76
			LSTM	22.84±2.42	14.42±2.16	8.33±2.39
			GRU	22.65±2.43	14.04±2.29	8.35±2.10
		OhioT1DM (Augmented Dataset)	WDNet	**15.63±2.57**	**11.69±2.17**	**5.86±1.69**
			LSTM	21.71±2.81	13.15±2.37	7.42±2.11
			GRU	21.25±2.73	12.92±2.43	7.12±1.81
			PH =60 Min			
Predictive Scores	[17] [2023]	OhioT1DM (Original Dataset)	LSTM	35.61±4.50	26.64±3.61	-
			DRNN	34.38±3.73	25.37 ±2.97	-
			SVR	36.46 ±5.87	27.06 ± 3.66	-
		OhioT1DM (Augmented Dataset)	LSTM	33.80±3.64	25.05± 2.96	-
			DRNN	33.43 ±3.56	24.80 ±2.72	-
			SVR	35.47±5.99	25:59±3.70	-
	This work	OhioT1DM (Original Dataset)	WDNet	29.41±5.92	21.96±4.67	12.33±3.15
			LSTM	34.97±4.85	25.90±4.49	14.43±3.82
			GRU	34.52±4.17	25.57±4.77	14.29±3.70
		OhioT1DM (Augmented Dataset)	WDNet	**27.41±5.44**	**19.77±4.31**	**9.99±2.16**
			LSTM	32.95±4.41	23.58±4.89	12.75±3.67
			GRU	32.44±4.35	23.27±4.82	12.43±3.58
Discriminative Scores	**Related Works**		**Dataset Used**			**Discriminative Scores**
	[17] [2023]		OhioT1DM (Inputs: CGM, CHO, insulin, SMBG)			0.17± 0.09
	This work		OhioT1DM (Inputs: CGM, CHO, insulin, heart rate and steps)			**0.15±0.02**

distributions of the synthetic and actual glucose time series indicating robustness of the sythetic data. It has been noted that the synthetic curve crosses the CGM values, fills in the missing points, shows trends, and has a strong correlation with the real CGM measurements. Figs. 6 and 7 show the representation of an entire day of synthetic meal and insulin time series for a T1D subject in the OhioT1DM dataset. Similarly other generated lifestyle data which could have been dropped or missed due to technical issue can be viewed in Fig. 8, in Appendix A.

V. DISCUSSION

Previous work on generating time series data for diabetes management applications have demonstrated the suitability and

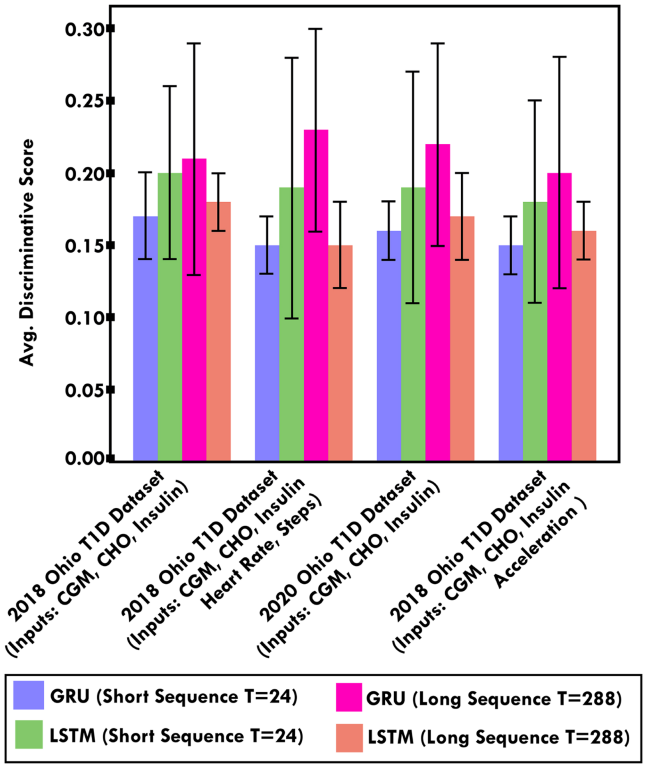

Fig. 4. Different discriminative Score evaluated on the OhioT1DM Dataset in different input setting validated with proposed architecture for long sequences (T=288 datapoints) and Short sequences (T=24 datapoints).

feasibility of data generation using GANs for T1D datasets [17], [31]. In this study, we propose a new model which can be used to produce data of better quality to replace missing datapoints using selective layer selection in generator and discriminator architectures.

This aspect was firstly demonstrated in Fig. 3 where in long sequences, data sequence generated using GRU exhibits lower spread and resembles the clinical data better (Fig. 3(f) and (h)) whereas the spread is more if GRU is used for longer sequences (Fig. 3(b) and (d)). This was also observed in the average discriminative score across the entire dataset where data was generated for missing datapoints of each patient. We have determined that for shorter sequences (length=24, equivalent to 2 hr), a GRU based generator and discriminator led to better generated/synthetic data. For longer sequences (length=288, equivalent to 24 hr), an LSTM based generator and discriminator led to better quality of generated/synthetic data.

The standard deviation of the discriminative score was less when GRU was used for generation of shorter sequences (24 timepoints which is equal to data of about 2 hours) as shown in Fig. 4. The same effect was observed i.e spread in PCA, tSNE plots and standard deviation of discriminative scores being less if LSTM was used for longer output sequences (288 timepoints which are equal to data to upto 24 hours). This effect could be due to the ability of LSTMs to capture both long-term and short-term states, whereas GRUs are limited in their abilities to capture very long-term states.

The discriminative scores were determined without the utilization of embedding networks, the exclusion of the supervised loss, and the absence of joint training of the embedding and adversarial networks on the supervised loss to assess the relative value of each contribution, as detailed in Table VI, in Appendix A. All three components significantly improve the quality of the time series data that are created. When the data exhibited strong temporal correlations, as in the glucose dataset, the supervised loss was more crucial. Additionally, we discovered that embedding networks and collaborative training with adversarial network (thereby aligning the two networks' goals) significantly and consistently boost generative performance in all domains.

The discriminative scores are described in the Table I. In GAN architecture, discriminative scores are essential for monitoring convergence, guiding hyperparameter adjustments, avoiding mode collapse, and preserving the equilibrium between the generator and discriminator throughout training. The discriminative scores in proposed architecture were lower compared to previous work, indicating that the generator is producing more realistic and high-quality generated time series data. The discriminative scores for the 2018 Ohio dataset were 0.17 ± 0.03 when CGM, CHO and insulin were taken as inputs from a clinical dataset with all missing points. Similarly, for the 2020 Ohio T1D dataset the score was 0.16 ± 0.02 for the same inputs. When more inputs were considered with the existing inputs such as heart rate, steps, acceleration then the score improved to 0.15 ± 0.02 for 2018 and 2020 dataset.

In Table V provided in Appendix A, we have shown and compared the discriminative scores for different methods. Here we have also shown the results for the proposed method of selective layers based on output sequence lengths. This approach has the lowest discriminative scores among all the above and previous methods.

A. Predictive Scores Analysis

The performance of glucose forecasting models using various data-driven prediction methods over a 30 and 60-minute prediction horizon is shown in Table II. Every prediction is put to the test using various training set compositions as described below. The different training set compositions used for evaluating glucose forecasting are as:

- T1D (IN_3) 2018: Trained on for 6 T1D adult from 2018 dataset *with Missing Points* (Inputs: CGM, CHO. Insulin)
- AugT1D (IN_3) 2018: Trained on for 6 T1D adult from 2018 augmented dataset *without any missing points* (Inputs: CGM, CHO. Insulin)
- T1D (IN_5) 2018: Trained on for 6 T1D adult from 2018 dataset *with Missing Points* (Inputs: CGM, CHO. Insulin, Heart Rate, Steps)
- AugT1D (IN_5) 2018: Trained on for 6 T1D adult from 2018 augmented dataset *without any missing points* (Inputs: CGM, CHO. Insulin, Heart Rate, Steps)

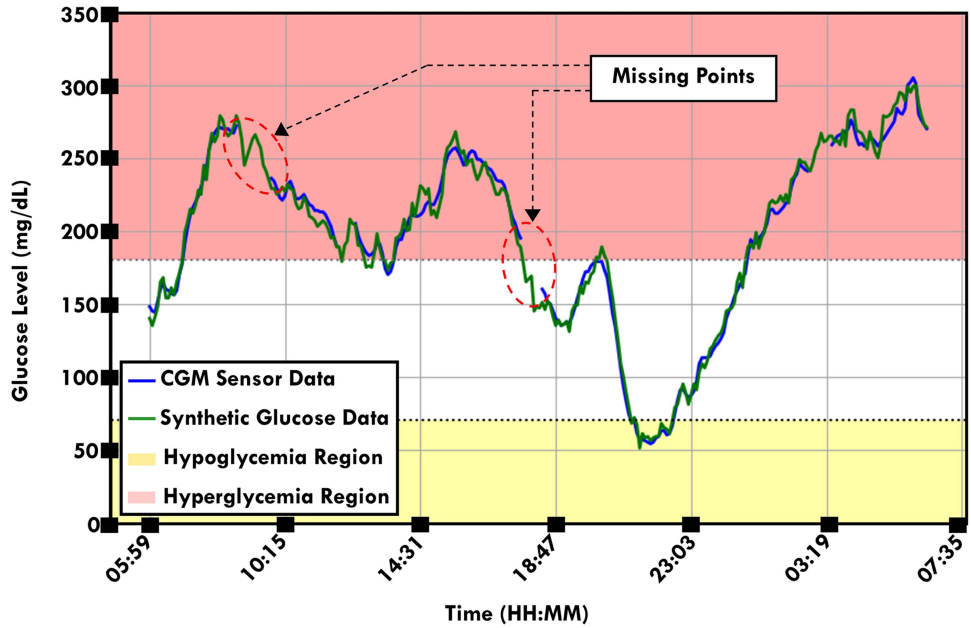

Fig. 5. Representation of an entire day of synthetic glucose time series for a T1D subject in the OhioT1DM dataset. The blue and green solid lines, respectively, represent the actual CGM measurements and the synthetic glucose values produced by the proposed GAN architecture. Light yellow and pink shaded patches, respectively, indicate the hypoglycemic and hyperglycemic zones.

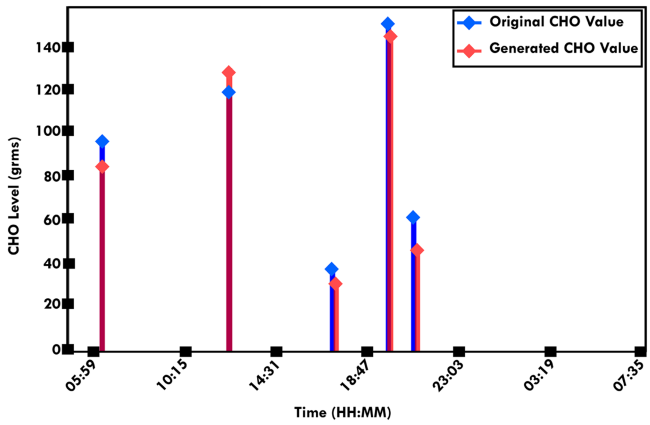

Fig. 6. Representation of an entire day of synthetic meal time series for a T1D subject in the OhioT1DM dataset. The blue and pink solid lines, respectively, represent the actual Meal measurements and the synthetic Meal values produced by the proposed architecture.

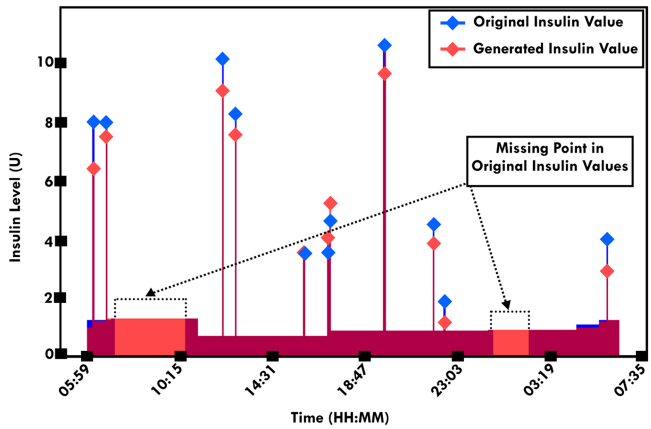

Fig. 7. Representation of an entire day of synthetic insulin time series for a T1D subject in the OhioT1DM dataset. The blue and pink solid lines, respectively, represent the actual insulin measurements and the insulin values produced by the proposed architecture.

- T1D (IN_3) 2020: Trained on for 6 T1D adult from 2020 dataset *with Missing Points* (Inputs: CGM, CHO. Insulin)
- AugT1D (IN_3) 2020: Trained on for 6 T1D adult from 2020 dataset augmented dataset *without any missing points* (Inputs: CGM, CHO. Insulin)
- T1D (IN_4) 2020: Trained on for 6 T1D adult from 2020 dataset *with Missing Points* (Inputs: CGM, CHO. Insulin, Acceleration)
- AugT1D (IN_4) 2020: Trained on for 6 T1D adult from 2020 dataset augmented dataset *without any missing points* (Inputs: CGM, CHO. Insulin, Acceleration)

It is important to note that each dataset's RMSE, MAE, and MARD scores considerably decreased with the inclusion of supplemented training sets and this combination of real and synthetic datapoints used in the augmented dataset as per clinical recommendations [32].

Table II shows the predictive scores in terms of RMSE (mg/dL), MAE (mg/dL) and MARD (%) for WDNet [30], LSTM [33] and GRU [30] architecture, which indicates WDNet leads to highest predictive scores of RMSE: 15.63±2.57, MAE: 11.69±2.17, MARD: 5.86±1.69 for 30 min PH and for 60 min PH RMSE was 27.41±5.44, MAE: 19.77±4.31 and MARD: 9.99±2.16) in the augmented dataset.

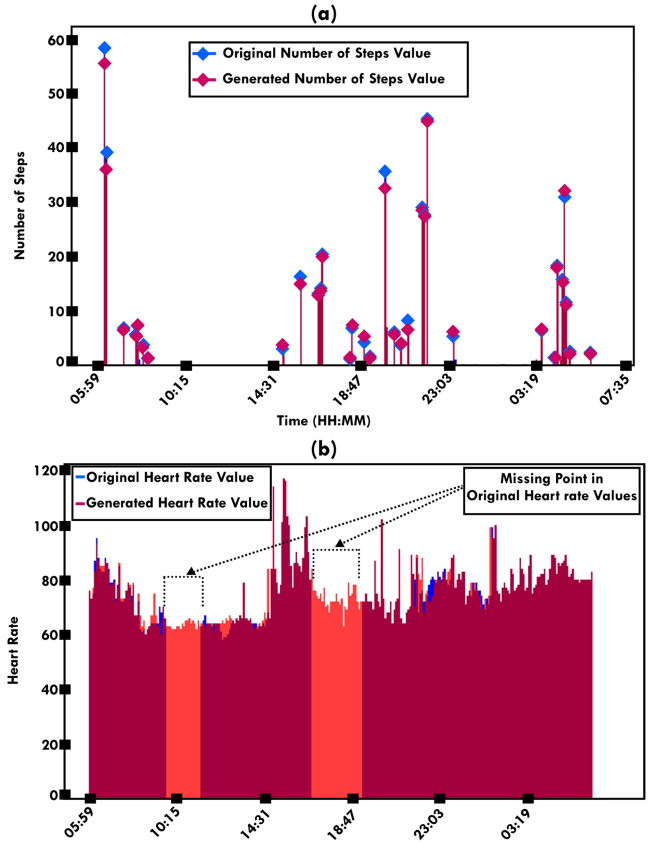

Fig. 8. Representation of an entire day of synthetic (a) number of steps and (b) heart rate values time series for a T1D subject in the OhioT1DM dataset. The blue and pink solid lines, respectively, represent the actual Meal measurements and the synthetic values produced by the proposed architecture.

VI. Conclusion

In this paper, a novel method to improve data augmentation using GANs has been proposed and studied in detail to resolve the issue of missing data in datasets involving sensor augmented pump platforms and possibly artificial pancreas platforms in future. Selective use of different architectures for discriminitators and generators leads to better data generation for different data types such as CGM levels, CHO values or insulin doses. The performance improvement is also demonstrated not only through PCA, t-SNE plots, and discriminatives but also on a regression task in artificial pancreas, such as glucose forecasting. This approach of generating high-fidelity data, which closely mirrors the probability distributions of the original data, will lead to better clinical outcomes even if data points are missed due to device or human error. Future work will be focussed on testing this approach in real time.

Appendix A
Hyperparameters

The hyperparameters utilised in this work are listed in Table IV.

TABLE IV
List of Hyperparameters

Parameter	Values (For short Sequence)	Values (For Long Sequence)
Length of Glucose Time Series (L)	24	288
Supervised learning iteration	8000	15000
Embedding learning iteration	8000	15000
Joint Learning iteration	32000	60000
Ratio of unsupervised loss	10	10
Ratio of reconstruction loss	1	1
Dimensionality of the random noise	32	64
Number of inner loop steps	2	2
Threshold of discriminator loss	0.15	0.15
Batch Size	128	128
Hidden Units of RNN Layers	256	512
Number of Layers	4	4
Learning rate of Adam optimizer	0.0005	0.0005
Early stopping patience	50	50

Appendix B
Discriminative Scores Analysis With Different Training Strategy

TABLE V
Results of Discriminative Score Evaluated on the All 12 OhioT1DM Subjects Validated With Proposed Architecture for Different Length of Sequence

Datasets	Method	Avg Discriminative Score (Lower the Better)
2018 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin)	GRU (Short Sequence T=24)	**0.17±0.03**
	LSTM (Short Sequence T=24)	0.20±0.06
	GRU (Long Sequence T=288)	0.21±0.08
	LSTM (Long Sequence T=288)	**0.18±0.02**
2018 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin, Heart Rate, Steps)	GRU (Short Sequence T=24)	**0.15±0.02**
	LSTM (Short Sequence T=24)	0.19±0.09
	GRU (Long Sequence T=288)	0.23±0.07
	LSTM (Long Sequence T=288)	**0.15±0.03**
2020 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin)	GRU (Short Sequence T=24)	**0.16±0.02**
	LSTM (Short Sequence T=24)	0.19±0.08
	GRU (Long Sequence T=288)	0.22±0.07
	LSTM (Long Sequence T=288)	**0.17±0.03**
2020 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin, Accelaration)	GRU (Short Sequence T=24)	**0.15±0.02**
	LSTM (Short Sequence T=24)	0.18±0.07
	GRU (Long Sequence T=288)	0.20±0.08
	LSTM (Long Sequence T=288)	**0.16±0.02**

TABLE VI
Results of Discriminative Score Evaluated on OhioT1DM Subjects With Proposed Architecture for Different Training Methods

Datasets	Method	Avg Discriminative Score (Lower the Better)
2018 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin)	Proposed architecture	**0.17±0.03**
	Without embedding model	1.89±0.81
	Without supervised loss	1.72±0.97
	Without joint training method	0.25±0.09
2018 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin, Heart Rate, Steps)	Proposed architecture	**0.15±0.02**
	Without embedding model	1.55±0.77
	Without supervised loss	1.4±0.91
	Without joint training method	0.21±0.07
2020 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin)	Proposed architecture	**0.16±0.02**
	Without embedding model	1.72±0.92
	Without supervised loss	1.56±0.65
	Without joint training method	0.27±0.11
2020 Ohio T1D Dataset (Inputs: CGM, CHO, Insulin, Accelaration)	Proposed architecture	**0.15±0.002**
	Without embedding model	1.59±0.32
	Without supervised loss	1.35±0.77
	Without joint training method	0.23±0.08

REFERENCES

[1] S. A. Brown et al., "Six-month randomized, multicenter trial of closed-loop control in type 1 diabetes," *New England J. Med.*, vol. 381, pp. 1707–1717, 2019.

[2] E. W. Gregg, N. Sattar, and M. K. Ali, "The changing face of diabetes complications," *Lancet Diabetes Endocrinol.*, vol. 4, no. 6, pp. 537–547, 2016.

[3] W. Tamborlane et al., "Juvenile diabetes research foundation continuous glucose monitoring study group continuous glucose monitoring and intensive treatment of type 1 diabetes," *New England J. Med*, vol. 359, no. 14, pp. 1464–1476, 2008.

[4] Y. Jiao et al., "A systematic review: Cost-effectiveness of continuous glucose monitoring compared to self-monitoring of blood glucose in type 1 diabetes," *Endocrinol., Diabetes Metab.*, vol. 5, no. 6, 2022, Art. no. e369.

[5] M. Tejedor, A. Z. Woldaregay, and F. Godtliebsen, "Reinforcement learning application in diabetes blood glucose control: A systematic review," *Artif. Intell. Med.*, vol. 104, 2020, Art. no. 101836.

[6] M. Vettoretti, G. Cappon, A. Facchinetti, and G. Sparacino, "Advanced diabetes management using artificial intelligence and continuous glucose monitoring sensors," *Sensors*, vol. 20, no. 14, 2020, Art. no. 3870.

[7] D. B. Keenan, J. J. Mastrototaro, G. Voskanyan, and G. M. Steil, "Delays in minimally invasive continuous glucose monitoring devices: A review of current technology," *J. Diabetes Sci. Technol.*, vol. 3, no. 5, pp. 1207–1214, 2009.

[8] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation," *IEEE J. Biomed. Health Inform.*, vol. 25, no. 4, pp. 1223–1232, Apr. 2021.

[9] Y. Zou, Z. Chu, J. Guo, S. Liu, X. Ma, and J. Guo, "Minimally invasive electrochemical continuous glucose monitoring sensors: Recent progress and perspective," *Biosensors Bioelectron.*, vol. 225, 2023, Art. no. 115103.

[10] G. Cappon, M. Vettoretti, G. Sparacino, and A. Facchinetti, "Continuous glucose monitoring sensors for diabetes management: A review of technologies and applications," *Diabetes Metab. J.*, vol. 43, no. 4, pp. 383–397, 2019.

[11] M. R. Askari et al., "Detection of meals and physical activity events from free-living data of people with diabetes," *J. Diabetes Sci. Technol.*, vol. 17, no. 6, pp. 1482–1492, 2023.

[12] I. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[13] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[14] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5508–5518.

[15] T.-J. Luo, Y. Fan, L. Chen, G. Guo, and C. Zhou, "EEG signal reconstruction using a generative adversarial network with wasserstein distance and temporal-spatial-frequency loss," *Front. Neuroinform.*, vol. 14, 2020, Art. no. 15.

[16] A. Torfi, E. A. Fox, and C. K. Reddy, "Differentially private synthetic medical data generation using convolutional GANs," *Inf. Sci.*, vol. 586, pp. 485–500, 2022.

[17] T. Zhu, K. Li, P. Herrero, and P. Georgiou, "GluGAN: Generating personalized glucose time series using generative adversarial networks," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 10, pp. 5122–5133, Oct. 2023.

[18] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," 2017, *arXiv:1706.02633*.

[19] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016, *arXiv:1611.09904*.

[20] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," 2018, *arXiv:1802.04208*.

[21] C. Marling and R. Bunescu, "The OhioT1DM dataset for blood glucose level prediction: Update 2020," in *Proc. CEUR Workshop*, 2020, vol. 2675, pp. 71–74.

[22] L. Morrow et al., "Evaluation of a novel continuous glucose measurement device in patients with diabetes mellitus across the glycemic range," *J. Diabetes Sci. Technol.*, vol. 5, no. 4, pp. 853–859, 2011.

[23] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung, "On data augmentation for GAN training," *IEEE Trans. Image Process.*, vol. 30, pp. 1882–1897, 2021.

[24] Y. Zhang, Z. Zhou, J. Liu, and J. Yuan, "Data augmentation for improving heating load prediction of heating substation based on timegan," *Energy*, vol. 260, 2022, Art. no. 124919.

[25] M. Fochesato, F. Khayatian, D. F. Lima, and Z. Nagy, "On the use of conditional timeGAN to enhance the robustness of a reinforcement learning agent in the building domain," in *Proc. 9th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, 2022, pp. 208–217.

[26] T. Zheng, L. Song, J. Wang, W. Teng, X. Xu, and C. Ma, "Data synthesis using dual discriminator conditional generative adversarial networks for imbalanced fault diagnosis of rolling bearings," *Measurement*, vol. 158, 2020, Art. no. 107741.

[27] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: A systematic literature review," *ACM Comput. Surv.*, vol. 55, no. 10, pp. 1–31, 2023.

[28] Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik, "Understanding how dimension reduction tools work: An empirical approach to deciphering T-SNE, UMAP, TRIMAP, and PACMAP for data visualization," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 9129–9201, 2021.

[29] F. B. Bryant and P. R. Yarnold, "Principal-components analysis and exploratory and confirmatory factor analysis," Amer. Psychol. Assoc., 1995.

[30] D. Kalita and K. B. Mirza, "Glucose prediction using wide-deep LSTM network for accurate insulin dosing in artificial pancreas," in *Proc. 44th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2022, pp. 4426–4429.

[31] Y. Deng et al., "Deep transfer learning and data augmentation improve glucose levels prediction in type 2 diabetes patients," *NPJ Digit. Med.*, vol. 4, no. 1, 2021, Art. no. 109.

[32] T. Battelino et al., "Continuous glucose monitoring and metrics for clinical trials: An international consensus statement," *Lancet Diabetes Endocrinol.*, vol. 11, no. 1, pp. 42–57, 2023.

[33] D. Kalita et al., "LS-GRUNet: Glucose forecasting using deep learning for closed-loop diabetes management," in *Proc. IEEE 7th Int. Conf. Convergence Technol.*, 2022, pp. 1–6.