Original Article

Journal of Diabetes Science and Technology
1–9
© 2023 Diabetes Technology Society
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/19322968231181138
journals.sagepub.com/home/dst
Sage

Simulating Realistic Continuous Glucose Monitor Time Series By Data Augmentation

Louis A. Gomez, Msc[1], Adedolapo Aishat Toye, Msc[1], R. Stanley Hum, MD, MA[2], and Samantha Kleinberg, PhD[1]

Abstract

Background: Simulated data are a powerful tool for research, enabling benchmarking of blood glucose (BG) forecasting and control algorithms. However, expert created models provide an unrealistic view of real-world performance, as they lack the features that make real data challenging, while black-box approaches such as generative adversarial networks do not enable systematic tests to diagnose model performance.

Methods: To address this, we propose a method that learns missingness and error properties of continuous glucose monitor (CGM) data collected from people with type 1 diabetes (OpenAPS, OhioT1DM, RCT, and Racial-Disparity), and then augments simulated BG data with these properties. On the task of BG forecasting, we test how well our method brings performance closer to that of real CGM data compared with current simulation practices for missing data (random dropout) and error (Gaussian noise, CGM error model).

Results: Our methods had the smallest performance difference versus real data compared with random dropout and Gaussian noise when individually testing the effects of missing data and error on simulated BG in most cases. When combined, our approach was significantly better than Gaussian noise and random dropout for all data sets except OhioT1DM. Our error model significantly improved results on diverse data sets.

Conclusions: We find a significant gap between BG forecasting performance on simulated and real data, and our method can be used to close this gap. This will enable researchers to rigorously test algorithms and provide realistic estimates of real-world performance without overfitting to real data or at the expense of data collection.

Keywords

blood glucose forecasting, continuous glucose monitoring, data augmentation, error, missing data, simulation

Introduction

Continuous glucose monitors (CGMs) have revolutionized care for individuals with type 1 diabetes (T1D),[1] as CGMs allow continuous insight into blood glucose (BG) and are essential components of artificial pancreas (AP) systems that automate BG management. Blood glucose simulators have enabled testing new BG control algorithms[2] and benchmarking BG forecasting performance. For simulations to be predictive of real-world performance, though, the data must contain the properties that make BG forecasting and control challenging.

Most BG simulations use domain knowledge to create generative models that capture the dynamics between BG, insulin, meals (eg, UVA/PADOVA model),[3] and physical activity.[4] However, there is a significant BG forecasting performance gap between simulations and real data (root mean square error [RMSE] simulated: 9.38; real: 21.07)[5] with similar gaps reported elsewhere.[6-8] Part of this is due to the incompleteness of the simulation system, as it does not include factors affecting BG such as stress[9] and the influence of fats and protein on glycemic profile of a meal.[10] Adding intra-subject variation via model parameters can improve simulations but does not close the performance gap even with ±30% variation (simulated: 10.90; real: 19.03).[8] More

[1]Stevens Institute of Technology, Hoboken, NJ, USA
[2]The Montreal Children's Hospital, McGill University Health Centre, Montreal, QC, Canada

Corresponding Author:
Samantha Kleinberg, PhD, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, NJ 07030, USA.
Email: samantha.kleinberg@stevens.edu

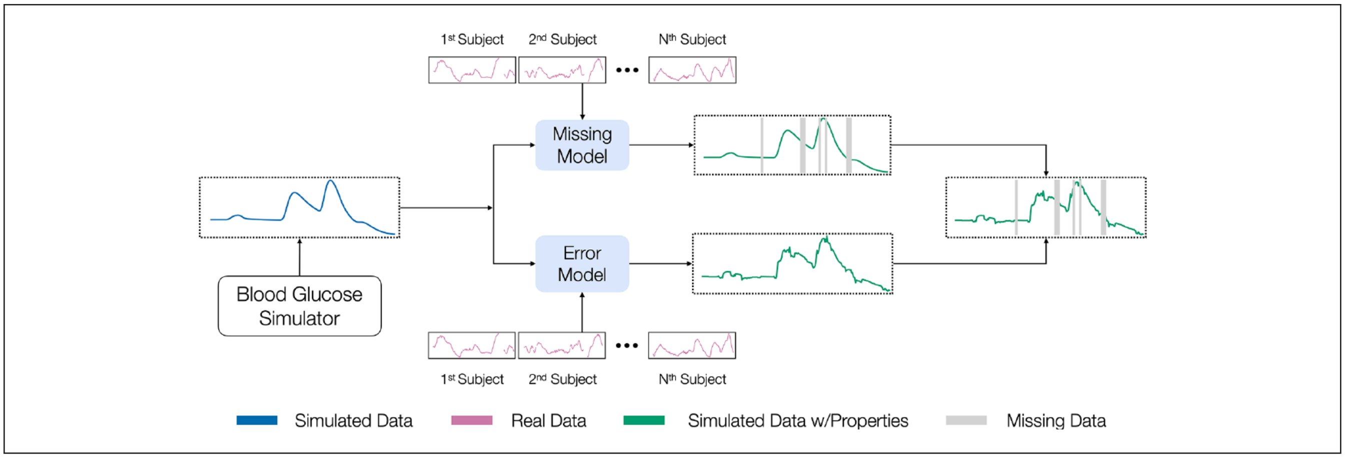


Figure 1. An overview of our methods for making simulated BG data more realistic. We begin with a simulation model that generates simulated BG, learns missing data and error patterns from real diabetes data sets, and finally augments simulated BG data with these learned properties.
Abbreviation: BG, blood glucose.

knowledge may help overcome these obstacles, but we hypothesize that the complications of real data pose the main challenge for machine learning (ML). Hence, we focus on data challenges of missing data and error as (1) they are inherent to CGM data, (2) smoothing and imputation are common preprocessing steps for CGM data, and (3) performing both leads to smaller RMSE compared with when neither (or just one) is performed.[11]

In most BG simulations, a CGM error model[12,13] is used to capture sensor error. However, these models: (1) focus only on sensor-related properties and do not capture variations in data sets like demographics that affect BG values differently[14,15]; (2) require subjects to wear multiple sensors and have frequent finger-stick BG readings, which are impractical for routine use[13,16]; and (3) do not capture factors like motion artifacts[17] and other forms of error like pressure-induced sensor attenuations.[18,19] Missing data are often simulated using random dropout[20] where a percentage of observations is deleted.[21,22] However, this assumes data are missing completely at random, which is not true for CGM. Drecogna et al[23] modeled gaps in CGM data using a two-state Markov model, but only replicated statistics (eg, number of gaps) and did not account for gaps due to other observed variables (eg, a person's vigorous exercise causing a CGM sensor to be disconnected).

One way to overcome these challenges is to learn a simulation model directly from data. Recent work using this approach has simulated CGM using generative adversarial networks (GANs) conditioned on HbA1C[24] or the presence of hypo/hyperglycemia.[25,26] However, GANs have privacy issues as individuals can be re-identified,[27]and in addition, they are black-box models, which means ablation studies (eg, BG forecasting with or without missing data) cannot be performed.

To address the challenges of creating realistic simulated BG, we propose data-augmented simulation (DAS), a hybrid approach that is modular (like knowledge-based methods) and realistic (like data-driven methods) by augmenting simulated data with real data properties. We focus on BG simulation for individuals with T1D and incorporate data set–specific errors and missingness patterns. We apply DAS to real-world data sets collected from different populations (varied age, race) with different protocols (data duration and sampling frequency).

Methods

We first discuss how we learn data properties from CGM data, and then how we augment simulated BG with these properties (see Figure 1 for an overview).

Learning Data Properties From Real Data

We define a multivariate time series as $X = \{x_1, x_2, \ldots, x_T\} \in R^{D\times T}$, where x_t is a vector of observations. We learn the patterns of missing data and error for CGM readings $(v_{1:T})$, which is a noisy estimate of finger-stick BG readings $(g_{1:T})$. While finger-stick BG readings are error prone,[28,29] we use them as ground truth as they are more accurate than CGM readings.

Missing data model. To learn the patterns of missingness, we frame it as a two-part problem where we predict the start and then duration of a missing interval. Predicting the start of a missing interval at time t is a multivariate classification task where we use a window of observations $X_{t-w:t-1}$ to determine whether v_t is the start of a missing interval. The window of observations consists of CGM data, binary meal indicator

Table 1. Time Series Features used for Error Modeling and Predicting the Length of Missing Intervals.

Statistical features	Interquartile range, kurtosis, maximum, minimum, mean, mean absolute deviation, median, median absolute deviation, root mean square, skew, standard deviation, variance
Temporal features	Area under the curve, autocorrelation, centroid, entropy, mean absolute difference, mean difference, median absolute difference, median difference, negative turning points, neighborhood peaks, peak to peak distance, positive turning points, signal distance, sum absolute difference

(1 during a meal, 0 otherwise), time since last observed CGM value, hour of the day, and day of the week. We select these variables to account for when CGM data are both missing at random and missing not at random. We use w in the range $[w_{min} : f : w_{max}]$, where w_{min} and f are the sampling frequency of v, to learn patterns of different lengths as the time between missing intervals varies.

To predict the duration of a missing interval z, we learn a second function. As there is no established feature set for learning the duration of missing intervals, we examined many features (Table 1) extracted from a window w_l of CGM data from the start of the missing interval. While the duration z can vary in length, we restrict the maximum value because: (1) larger missing intervals are often not representative of actual missing intervals and (2) there are typically fewer samples to learn from. The maximum duration, $z_{\max}$, set based on domain knowledge, is two hours here.

Error model. We first discuss how to learn an error model when we have both finger-stick and CGM data, before discussing the case when only CGM is available. When BG is available, we align CGM and BG values using the Poincaré plot approach[30] on the entire data set. This is necessary as CGM values are delayed relative to finger-stick BG and this delay may vary across individuals. When BG is absent, we approximate it using smoothed CGM data as our reference. While we cannot identify errors such as all CGM readings being shifted higher or lower than finger-stick BG due to errors in sensor calibration, smoothing allows us to identify outlying values and erroneous spikes. Given an observation v_t and its corresponding ground truth g_t, we extract time series features in Table 1 to predict the error $e_t := v_t - g_t$ using a regression model (see Supplemental Appendix A for feature importance for missing data and error).

Augmenting Simulated Blood Glucose Data With Real Data Properties

To add the learned properties to simulated BG, we use a post-processing step. The input is simulated BG and the outputs are a missingness vector $m_{1:T} \in \{0,1\}$ (0 indicates missing) and a vector of predicted error values $e_{1:T} \in R$. Simulated BG is generated using the UVA/PADOVA simulator (see Supplemental Appendix B for details on data simulation).

Augmenting simulated blood glucose with missing data. To predict when missing data occur, we begin by extracting windows of length w_{min} to w_{max} and iteratively performing prediction until a window is classified as missing. To determine whether to accept these predictions, we use the precision P from the learning phase as the probability of correctly predicting the start of a missing interval using a binomial trial $\mathcal{B}(1, P)$. Hence, for each prediction, we have:

$$y^* = \begin{cases} B(1,P)\, if\ output = 1, \\ 0\, if\ output = 0, \end{cases}$$

where y^* indicates whether a timepoint is the start of a missing interval. Once the start of an interval is identified, we predict the length of the missing data using a history window w_l from that time point. We move forward to the time point after the last predicted missing entry and repeat the process until the end of the time series. For time points t that are missing, we set $m_t = 0$ and mask them as NaNs (see Figure 1 in Supplemental Appendix C for an overview of this process).

Augmenting simulated blood glucose with error. To add error to simulated data, we extract history windows of size w_{error} to create feature vectors, and then use this to predict the errors, leading to a vector of error values, $e_{1:T}$. We then sum each t and its predicted error e_t to get the final simulated data set.

Experiments

Data Sets

We first describe the real data sets (see Table 2) and then simulated data sets generated to match the features of each. To avoid leakage, we divide real data sets into a 70/30% subset for learning data properties and forecasting except for OhioT1DM where it is six weeks/two weeks due to its size (see Supplemental Appendix D for details on data subsets).

OhioT1DM[31]: It has been used for benchmarking BG forecasting methods. As it is relatively small and controlled, results may not be representative of performance on other data sets.[11]

OpenAPS[32]: It is from people using an open-source AP who elect to donate their data. It is patient generated; so,

Table 2. Characteristics of Real T1D Data Sets.

Data sets	OhioT1DM	OpenAPS	RCT	Racial-Disparity
Number of subjects	12	86	226	227
Average days of data	54 ± 3.02	307.7 ± 246.48	234.3 ± 36	74.4 ± 20.5
Median days of data[a]	54.9 (2.52)	233.1 (289.8)	257.4 (45.56)	83.9 (8.32)
Variables collected	CGM, finger-stick BG, basal rates, bolus doses, physiological sensor readings, and meal size and times	CGM, basal rates, bolus doses, and meal size and times	CGM, finger-stick BG, basal rates, bolus doses, and self-reported information	CGM, finger-stick BG, basal rates, bolus doses, and mealtimes
Number of BG values	4762	NA	506,394	49,130
Number of CGM values	166,533	7,221,371	13,841,924	1,259,072
Percentage of CGM values missing for less than two hours[b]	1.84%	7.09%	4.01%	0.24%

Abbreviations: T1D, type 1 diabetes; T1DM, type 1 diabetes mellitus; CGM, continuous glucose monitor; BG, blood glucose; NA, not applicable.

[a] Median is reported with the interquartile range in parenthesis.

[b] We selected 2 hours because it is the maximum length of a missing window we consider for CGM.

it provides a useful comparison with highly controlled data sets.

RCT[33]: It was collected during a randomized trial testing the use of CGM without BG confirmations. It is larger than OhioT1DM and thus provides a better comparison for data collected in a controlled setting.

Racial-Disparity[34]: It was collected in a study testing if there is a difference between mean glucose and HbA1c in non-Hispanic black and white people with T1D. It has a higher proportion of black participants (54%) and a larger age range (5-72 years) compared with other data sets, which allows us to test with a different population. Note that the CGM sampling interval is 15 minutes compared with 5 minutes in other data sets.

Simulated Data: It is generated to match the characteristics (average days of data, meal information, age group) of real data sets. For example, *Sim-OpenAPS* is generated for 10 adults with 308 days of data per subject (see Supplemental Appendix B for details on simulation).

Experiments for learning data properties. To predict the start of missing intervals, we train a recurrent neural network (RNN) with a hidden layer of 32 units, batch size of 128, maximum epochs of 100, and early stopping of 15 epochs. We use a fivefold cross-validation split across patients (a further split of the 70%) and report performance using the area under the receiver operating curve (AUROC) and the area under the precision-recall curve (AUPRC). For predicting the length of missing intervals, we train a random forest (RF) regressor. For predicting errors, we train an Xgboost regressor except for Racial-Disparity, where RF was used as this provided better performance. We evaluate the performance of predicting the length of missing intervals and errors using the RMSE (RMSE is defined as: $\sqrt{\sum_{i=1}^{n}(h_i^* - h_i)^2 / n}$. When computing the RMSE for duration of missing intervals, h_i^* is the predicted duration and h_i is the actual duration. For computing the RMSE when predicting error values, h_i is the actual error of CGM data relative to BG at timepoint t (ie, $CGM_t - BG_t$) while h_i^* is the predicted error value). We use a leave-one-out cross-validation for OhioT1DM due to its small size (see Supplemental Appendix E for full details).

Experiments for blood glucose forecasting on simulated data. We aim to test whether our method for augmenting simulated BG (*Dropout-predicted* and *Error-predicted*) brings performance closer to real CGM compared with current practice. We compare against current BG simulation practices:

- Dropout-random: We delete a percentage of observations based on the average percent of missing data (up to the maximum gap size of two hours from missing data experiments) in each real data set. These are 1.71% for OhioT1DM, 7.40% for OpenAPS, 4.12% for RCT, and 0.27% for Racial-Disparity.
- Error-Gaussian: We add Gaussian noise of N(0.4mg/dL) within the 15% required error range for CGM sensors.
- Error-CGM: We add noise using a CGM error model[12] that is not specific to any data set.

For forecasting, we use the same set of models as Hameed & Kleinberg,[11] including linear regression (REG), RF, RNN, and long short-term memory (LSTM). We report the difference in the mean RMSE (over ten runs) between real and simulated data as $mRMSE_{diff} = mRMSE_{real} - mRMSE_{sim}$, where $mRMSE_{real}$ and $mRMSE_{sim}$ are the mean RMSE on real and simulated data sets. A smaller $mRMSE_{diff}$ indicates a closer performance of simulated data to real data. For REG

Table 3. Results for Predicting the Start of Missing Intervals, Duration of Missing Intervals, and the CGM Error for Real Data Sets.

	Missing data			
	Start of interval		Duration	Error
Data sets	Mean AUROC	Mean AUPRC	RMSE (minutes)	RMSE (mg/dL)
OhioT1DM	0.66	0.20 (0.12)	23.27	13.87
OpenAPS	0.70	0.55 (0.30)	14.50	5.07[a]
RCT	0.68	0.22 (0.07)	18.54	21.94
Racial-Disparity	0.62	0.38 (0.30)	22.19	31.53

The value in parenthesis is the mean AUPRC baseline, which is the percentage of positive samples in test data.
Abbreviations: CGM, continuous glucose monitor; AUROC, area under the receiver operating curve; AUPRC, area under the precision-recall curve; RMSE, root mean square error; T1DM, type 1 diabetes mellitus.
[a]We generated ground truth for OpenAPS, unlike the other data sets where ground truth was present.

and RF, we use default parameters from scikit-learn.[35] For RNN and LSTM, we use a hidden layer with 32 units, batch size of 248, maximum epoch of 50, and early stopping of 15 epochs (see Supplemental Appendix F for additional experimental details).

Results

We now discuss results on learning data properties and BG forecasting to test improved simulated data performance.

Results on Learning Data Properties of Missing Data and Error

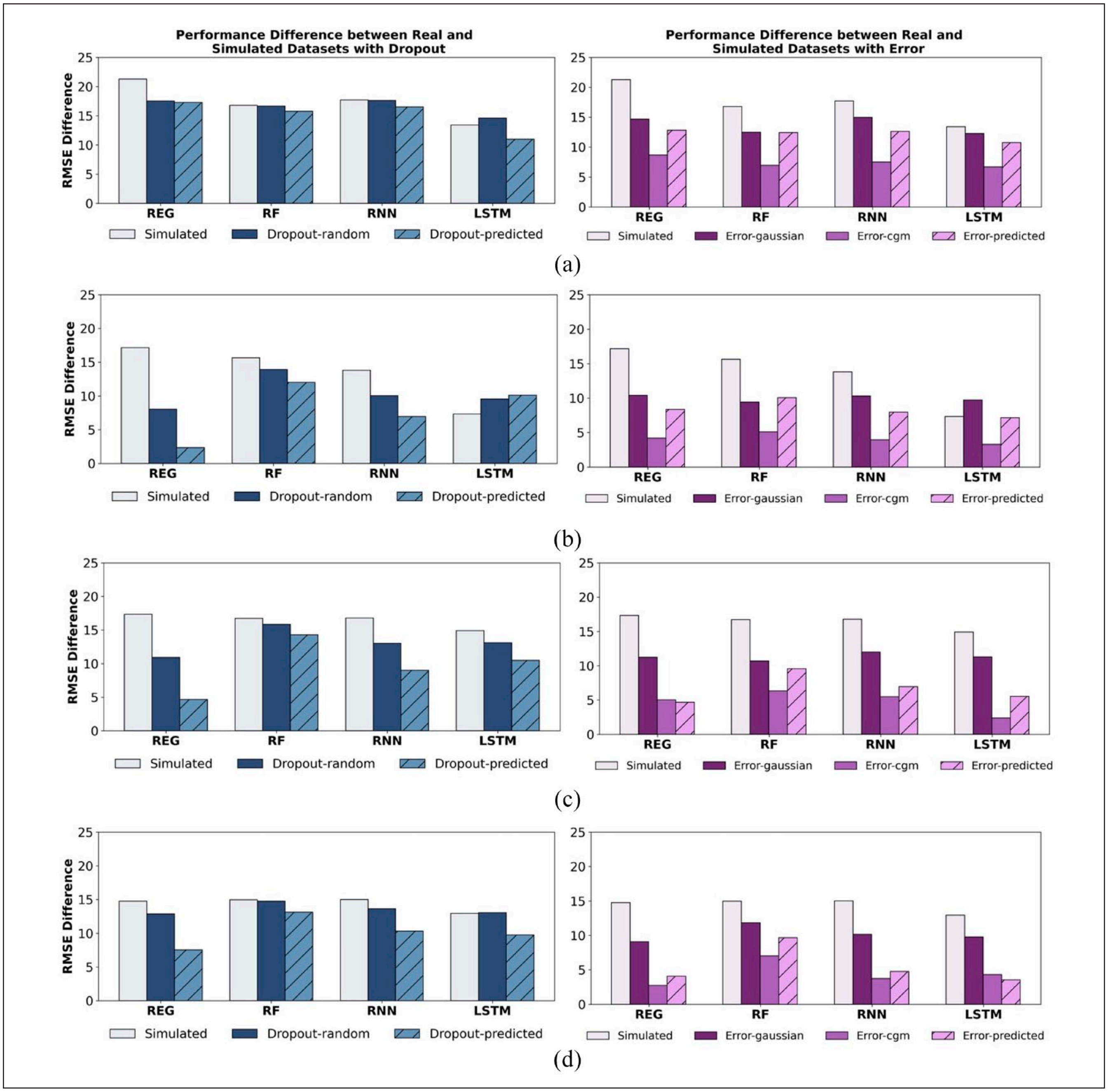
Table 3 shows results for missing data and error prediction on all real data sets. For predicting the start of missing intervals, a prediction is only correct if the first timepoints in a sequence of missing datapoints are predicted. While this is a strict evaluation (as the time identified may be slightly early or slightly late), our method performs better than the AUROC baseline of 0.5. Our method has the best improvements over the mean baseline for the AUPRC on OpenAPS and RCT compared with OhioT1DM and Racial-Disparity. For OhioT1DM, the performance is likely due to its smaller size, while for Racial-Disparity, its heterogeneous population coupled with a lower sampling rate (which may not be the right timescale for observing patterns) makes learning challenging. These challenges also likely apply when predicting the duration of missing intervals as our method performs better for OpenAPS and RCT compared with OhioT1DM and Racial-Disparity, which have a higher RMSE.

Similarly, when predicting error, we had the worst performance on Racial-Disparity which suggests that its heterogeneity makes learning challenging. We performed best on OpenAPS due to the use of synthetic ground truth which creates error values closer to CGM data compared with other data sets where actual ground truth is present.

Results for Blood Glucose Forecasting on Simulated Data

We now turn to the task of BG forecasting, comparing our approach to baseline methods for adding error and dropout. To understand how dropout and error individually affect performance, we test them separately. For each property (eg, error), we compare our approach (*Error-predicted*) to baselines and raw simulated data. As shown in Figure 2, raw simulated data have the largest $mRMSE_{diff}$, which in practice would lead to erroneous forecasts of future BG and subsequently incorrect insulin dosing. For dropout, *Dropout-predicted* had the smallest $mRMSE_{diff}$, outperforming dropout-random in all cases except for LSTM on OpenAPS. For error, *Error-predicted* had a smaller $mRMSE_{diff}$ compared with Error-Gaussian except for RF on OpenAPS but ranks second compared with Error-CGM. However, Error-CGM requires a significant amount of domain knowledge coupled with the use of multiple sensors, while our approach is purely data-driven.

Next, we test BG forecasting performance when our methods for dropout and error are combined, showing that both are needed to further bring simulated data performance closer to real data. For each error module, we vary the dropouts applied to it and report the $mRMSE_{diff}$ in Tables 4 and 5. We test for significant differences between *Error-Predicted* and each error model across dropouts using a *t*-test (eg, Error-CGM + *Dropout-Predicted* against *Error-Predicted* + *Dropout-Predicted*). When testing between Error-Gaussian and *Error-Predicted*, *Error-Predicted* was significantly better than Error-Gaussian on OpenAPS (all $P \leq .004$), RCT (all $P \leq .04$) and Racial-Disparity (all $P \leq .002$). Next, when comparing *Error-Predicted* with Error-CGM, Error-CGM performed better in two of four ML models for (1) *Dropout-Predicted* on OpenAPS ($P \leq .05$) and (2) Dropout-Random for RCT ($P \leq .005$). On Racial-Disparity though, *Error-Predicted* was significantly better in three of four ML models for *Dropout-Predicted* ($P \leq .04$). This suggests that *Error-Predicted* performs better when the data set properties are



(a)

(b)

(c)

(d)

Figure 2. Mean RMSE difference between real and simulated data sets ($mRMSE_{diff}$) with missingness (left) and error (right). A smaller difference indicates results closer to real data. (a) Performance on OhioT1DM, (b) performance on OpenAPS, (c) performance on RCT, and (d) performance on Racial-Disparity.

Abbreviations: RMSE, root mean square error; T1DM, type 1 diabetes mellitus; REG, linear regression; RF, random forest; RNN, recurrent neural network; LSTM, long short-term memory.

significantly different from the one Error-CGM was modeled on. This is supported by evidence showing a difference in glucose variability, control, and diabetes management in the population Error-CGM was modeled with compared with the population of the Racial-Disparity data set (heterogeneous by age[15] and majority black population[14]). For OhioT1DM, *Error-Predicted* was better than Error-Gaussian in two of four ML Models for Dropout-random ($P \leq .03$) but performed worse compared with Error-CGM (all $P \leq .02$). Overall, in most comparisons for OpenAPS, RCT, and Racial-Disparity, there were no significant differences in $mRMSE_{diff}$ between Error-CGM and *Error-Predicted*, which means that we achieved the same performance, while not requiring the same amount of background knowledge.

Table 4. Mean RMSE Difference Between Real and Simulated Data Sets ($mRMSE_{diff}$) When Combining Dropout and Error.

		OhioT1DM				OpenAPS			
		REG	RF	RNN	LSTM	REG	RF	RNN	LSTM
Raw simulated BG		21.29	16.78	17.71	13.42	17.15	15.66	13.81	7.34
Error	Dropout								
Gaussian	Random	**14.60**[a]	**12.38**	15.11[a]	12.16	9.95[a]	9.18[a]	10.31[a]	9.97[a]
	Predicted	14.83	13.15	**14.87**	**10.11**	**8.91**[a]	**8.74**[a]	**8.75**[a]	**6.98**[a]
CGM	Random	**8.48**[b]	**6.88**[b]	**7.76**[b]	**5.46**[b]	3.37	4.11[b]	2.69	**2.07**
	Predicted	9.76[b]	8.28[b]	8.22[b]	7.30[b]	**1.87**	**3.24**[b]	**0.96**[b]	2.88
Predicted	Random	**12.75**	12.38	12.77	11.10	2.97	6.88	4.13	3.55
	Predicted	13.19	**11.70**	**12.47**	**9.60**	**2.16**	**5.50**	**2.91**	**3.00**

Lower values indicate performance closer to real data. The best performance within each error group is shown in bold.
Abbreviations: RMSE, root mean square error; REG, linear regression; RF, random forest; RNN, recurrent neural network; LSTM, long short-term memory; BG, blood glucose; CGM, continuous glucose monitor.
[a] *Error-Predicted* is significantly better ($P < .05$) when compared with either Error-CGM or Error-Gaussian across all dropout modules.
[b] *Error-Predicted* is significantly worse ($P < .05$) when compared with either Error-CGM or Error-Gaussian across all dropout modules.

Table 5. Mean RMSE Difference Between Real and Simulated Data Sets ($mRMSE_{diff}$) When Combining Dropout and Error.

		RCT				Racial-Disparity			
		REG	RF	RNN	LSTM	REG	RF	RNN	LSTM
Raw simulated BG		17.34	16.74	16.79	14.91	14.76	14.96	15.01	12.94
Error	Dropout								
Gaussian	Random	10.99[a]	10.55[a]	11.64[a]	11.61[a]	8.95[a]	**11.68**[a]	9.90[a]	**9.94**[a]
	Predicted	**9.88**[a]	**9.98**[a]	**9.64**[a]	**9.79**[a]	**8.39**[a]	12.39[a]	**9.31**[a]	10.08[a]
CGM	Random	4.60	5.84[b]	4.72[b]	4.45	**2.67**	**6.92**	**3.82**	**3.93**
	Predicted	**3.53**	**4.90**[b]	**3.56**	**2.89**	5.37[a]	9.22	6.31[a]	6.93[a]
Predicted	Random	4.60	9.26	7.00	5.58	3.97	9.49	4.65	5.19
	Predicted	**4.30**	**8.30**	**5.80**	**5.20**	**1.66**	**5.75**	**2.29**	**0.78**

Lower values indicate performance closer to real data. The best performance within each error group is shown in bold.
Abbreviations: RMSE, root mean square error; REG, linear regression; RF, random forest; RNN, recurrent neural network; LSTM, long short-term memory; BG, blood glucose; CGM, continuous glucose monitor.
[a] *Error-Predicted* is significantly better ($P < .05$) when compared with either Error-CGM or Error-Gaussian across all dropout modules.
[b] *Error-Predicted* is significantly worse ($P < .05$) when compared with either Error-CGM or Error-Gaussian across all dropout modules.

Our combination of methods reduced $mRMSE_{diff}$ from: (1) 21.29 to 13.19 for OhioT1DM, (2) 17.15 to 2.16 for OpenAPS, (3) 17.34 to 4.30 for RCT, and (4) 15.01 to 2.29 for Racial-Disparity on the ML model with the largest $mRMSE_{diff}$ on raw simulated BG. This improvement in performance provides a more realistic estimation of performance when using simulated data for testing new algorithms for diabetes-related applications.

Discussion

Simulated BG data are vital for evaluating the performance of BG control and forecasting algorithms as it enables testing under varied conditions (eg, different meal schedules and exercise). However, current simulation methods require prior knowledge and do not guarantee the same performance as real data or replicate data without understanding how data properties affect performance. To address this gap, we introduce DAS, a modular data-driven approach to simulation that allows us to learn specific properties of CGM data and encode them into simulated BG data sets to bring performance closer to real data. All code will be made available on publication. A limitation of our work is the need for ground truth for learning error models. While we use finger-stick BG as ground truth, these values are also subject to error.[28,29] This in turn affects our error estimates as our model learns how CGM differs from finger-tick measurements rather than actual BG. In future work, we plan to explore other ways of learning error models without requiring ground truth. Second, as in many ML applications, our approach requires sufficient data to learn reliable models for missingness and error. OhioT1DM was significantly smaller than the others tested, and performance gains on it were thus correspondingly smaller.

Conclusions

We demonstrate that DAS learns the patterns of missing data and errors in several real diabetes data sets. On BG forecasting, adding these properties brings performance on simulated data closer to real data compared with other baselines in most of our real data sets. This has real-world implications as it allows researchers to test algorithms with simulated BG that provides realistic estimates of performance and better understand how different features of the data contribute to performance on ML tasks. Our results motivate a hybrid approach for simulating time series data sets to enable greater control over the types of properties encoded in them. Future work will involve adding more properties like non-stationarity to further aid in simulating more realistic BG values and helping researchers debug their algorithms. Code is available at: https://github.com/health-ai-lab/Data-Augmented-Simulation

Abbreviations

AUPRC, area under the precision-recall curve; AUROC, area under the receiver operating curve; BG, blood glucose; CGM, continuous glucose monitor; DAS, data-augmented simulation; GAN, generative adversarial network; LSTM, long short-term memory; REG, linear regression; RF, random forest; RMSE, root mean square error; RNN, recurrent neural network; T1D, type 1 diabetes.

Declaration of Conflicting Interests

The author(s) declared the following potential conflicts of interest with respect to the research, authorship, and/or publication of this article: RSH works as a health care data science consultant for US Retina and has equity in CapsicoHealth, Inc, which are both health care data science firms, but they do not work in the glucose simulation field. SK, LG, and AAT do not have any conflicts of interest.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the NLM of the NIH under award umber R01LM011826 with SK as the grant recipient. The funding agency had no involvement in the study or paper preparation.

ORCID iDs

Louis A. Gomez https://orcid.org/0000-0002-7712-340X
Samantha Kleinberg https://orcid.org/0000-0001-6964-3272

Supplemental Material

Supplemental material for this article is available online.

References

1. Battelino T, Conget I, Olsen B, et al. The use and efficacy of continuous glucose monitoring in type 1 diabetes treated with insulin pump therapy: a randomised controlled trial. *Diabetologia*. 2012;55(12):3155-3162.
2. Fox I, Lee J, Pop-Busui R, Wiens J. Deep reinforcement learning for closed-loop blood glucose control. *Mach Learn Healthc Conf*. 2020;126:508-536.
3. Kovatchev BP, Breton M, Man CD, Cobelli C. In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes. *J Diabetes Sci Technol*. 2009;3(1):44-55.
4. Man CD, Breton MD, Cobelli C. Physical activity into the meal glucose—insulin model of type 1 diabetes: in silico studies. *J Diabetes Sci Technol*. 2009;3(1):56-67.
5. Li K, Daniels J, Liu C, Herrero P, Georgiou P. Convolutional recurrent neural networks for glucose prediction. *IEEE J Biomed Health Inform*. 2020;24(2):603-613.
6. Li K, Liu C, Zhu T, Herrero P, Georgiou P. GluNet: a deep learning framework for accurate glucose forecasting. *IEEE J Biomed Health Inform*. 2020;24(2):414-423.
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(3):308-324.
8. Liu C, Vehí J, Avari P, et al. Long-term glucose forecasting using a physiological model and deconvolution of the continuous glucose monitoring signal. *Sensors*. 2019;19(19):4338.
9. Riazi A, Pickup J, Bradley C. Daily stress and glycaemic control in type 1 diabetes: individual differences in magnitude, direction, and timing of stress-reactivity. *Diabetes Res Clin Pract*. 2004;66(3):237-244.
10. Bell KJ, Smart CE, Steil GM, Brand-Miller JC, King B, Wolpert HA. Impact of fat, protein, and glycemic index on postprandial glucose control in type 1 diabetes: implications for intensive diabetes management in the continuous glucose monitoring era. *Diabetes Care*. 2015;38(6):1008–1015.
11. Hameed H, Kleinberg S. Comparing machine learning techniques for blood glucose forecasting using free-living and patient generated data. *Proc Mach Learn Res*. 2020;126:871-894.
12. Breton M, Kovatchev B. Analysis, modeling, and simulation of the accuracy of continuous glucose sensors. *J Diabetes Sci Technol*. 2008;2(5):853-862.
13. Vettoretti M, Favero SD, Sparacino G, Facchinetti A. Modeling the error of factory-calibrated continuous glucose monitoring sensors: application to Dexcom G6 sensor data. *Annu Int Conf IEEE Eng Med Biol Soc*. 2019;2019:750-753.
14. Spanakis EK, Golden SH. Race/ethnic difference in diabetes and diabetic complications. *Curr Diab Rep*. 2013;13(6):814-823.
15. Hirsch IB, Balo AK, Sayer K, Garcia A, Buckingham BA, Peyser TA. A simple composite metric for the assessment of glycemic status from continuous glucose monitoring data: implications for clinical practice and the artificial pancreas. *Diabetes Technol Ther*. 2017;19(suppl 3):S38-S48.
16. Facchinetti A, Del Favero S, Sparacino G, Castle JR, Ward WK, Cobelli C. Modeling the glucose sensor error. *IEEE Trans Biomed Eng*. 2014;61(3):620-629.

17. Mensh BD, Wisniewski NA, Neil BM, Burnett DR. Susceptibility of interstitial continuous glucose monitor performance to sleeping position. *J Diabetes Sci Technol*. 2013;7(4):863-870.
18. Baysal N, Cameron F, Buckingham BA, et al. A novel method to Detect Pressure-Induced Sensor Attenuations (PISA) in an artificial pancreas. *J Diabetes Sci Technol*. 2014;8(6):1091-1096.
19. Navarathna P, Cameron F, Sontakke M, Yang S, Diamond T, Bequette BW. Machine-learning-based detection of pressure-induced faults in continuous glucose monitors. *Ind Eng Chem Res*. 2023;62(5):2255-2262.
20. Lin W-C, Tsai C-F. Missing value imputation: a review and analysis of the literature (2006–2017). *Artif Intell Rev*. 2020;53(2):1487-1509.
21. Yoon J, Jordon J, van der Schaar M. Gain: missing data imputation using generative adversarial nets. *Int Conf Mach Learn*. 2018;80:5689-5698.
22. Cao W, Wang D, Li J, Zhou H, Li L, Li Y. Brits: bidirectional recurrent imputation for time series. *Neur Inf Process Syst*. 2018;31:6776-6786.
23. Drecogna M, Vettoretti M, Favero SD, Facchinetti A, Sparacino G. Data gap modeling in continuous glucose monitoring sensor data. *Annu Int Conf IEEE Eng Med Biol Soc*. 2021;2021:4379-4382.
24. Cichosz SL, Xylander AAP. A conditional generative adversarial network for synthesis of continuous glucose monitoring signals. *J Diabetes Sci Technol*. 2022;16(5):1220-1223.
25. Deng Y, Lu L, Aponte L, et al. Deep transfer learning and data augmentation improve glucose levels prediction in type 2 diabetes patients. *NPJ Digit Med*. 2021;4(1):109.
26. Noguer J, Contreras I, Mujahid O, Beneyto A, Vehi J. Generation of individualized synthetic data for augmentation of the type 1 diabetes data sets using deep learning models. *Sensors*. 2022;22(13):4944.
27. Hitaj B, Ateniese G, Perez-Cruz F. Deep models under the GAN: information leakage from collaborative deep learning. Paper presented at Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security; October 2017; Dallas, TX.
28. Ekhlaspour L, Mondesir D, Lautsch N, et al. Comparative accuracy of 17 point-of-care glucose meters. *J Diabetes Sci Technol*. 2017;11(3):558-566.
29. Klonoff DC, Parkes JL, Kovatchev BP, et al. Investigation of the accuracy of 18 marketed blood glucose monitors. *Diabetes Care*. 2018;41(8):1681-1688.
30. Kovatchev BP, Shields D, Breton M. Graphical and numerical evaluation of continuous glucose sensing time lag. *Diabetes Technol Ther*. 2009;11(3):139-143.
31. Marling C, Bunescu R. The OhioT1DM dataset for blood glucose level prediction: update 2020. *CEUR Workshop Proc*. 2020;2675:71-74.
32. Melmer A, Züger T, Lewis DM, Leibrand S, Stettler C, Laimer M. Glycaemic control in individuals with type 1 diabetes using an open source artificial pancreas system (OpenAPS). *Diabetes Obes Metab*. 2019;21(10):2333-2337.
33. Aleppo G, Ruedy KJ, Riddlesworth TD, et al. REPLACE-BG: a randomized trial comparing continuous glucose monitoring with and without routine blood glucose monitoring in adults with well-controlled type 1 diabetes. *Diabetes Care*. 2017;40(4):538-545.
34. Bergenstal RM, Gal RL, Connor CG, et al. Racial differences in the relationship of glucose concentrations and hemoglobin A1c levels. *Ann Intern Med*. 2017;167(2):95.
35. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python. *JMLR*. 2011;12:2825-2830.