Comparing Machine Learning Techniques for Blood Glucose Forecasting Using Free-living and Patient Generated Data

Hadia Hameed HHAMEED@STEVENS.EDU *Electrical and Computer Engineering*
Stevens Institute of Technology
Hoboken, NJ, USA

Samantha Kleinberg SAMANTHA.KLEINBERG@STEVENS.EDU *Computer Science*
Stevens Institute of Technology
Hoboken, NJ, USA

Abstract

Managing a chronic disease like Type 1 diabetes (T1D) is both challenging and time consuming, but new technologies that allow continuous measurement of glucose and delivery of insulin have led to significant improvements. The development of an artificial pancreas (AP), which algorithmically determines insulin dosing and delivers insulin in a fully automated way, may transform T1D care but it is not yet widely available. Patient-led alternatives, like the Open Artificial Pancreas (OpenAPS), are being used by hundreds of individuals and have also led to a dramatic increase in the availability of patient generated health data (PGHD). All APs require an accurate forecast of blood glucose (BG). While there have been efforts to develop better forecasts and apply new ML techniques like deep learning to this problem, methods are often tested on small controlled datasets that do not indicate how they may perform in reality – and the most advanced methods have not always outperformed the simplest. We introduce a rigorous comparison of BG forecasting using both a small controlled research dataset and large heterogeneous PGHD. Our comparison advances the state of the art in BG forecasting by providing insight into how methods may fare when moving beyond small controlled studies to real-world use.

1. Introduction

Worldwide there are are over 400 million adults with diabetes, and this is expected to rise to over 600 million by 2045 (International Diabetes Federation, 2018). While the majority of cases are of Type 2 diabetes (T2D), the rate of Type 1 diabetes (T1D) is similarly increasing. T1D is a lifelong condition that usually develops in childhood, where the body does not produce insulin. As a result, people with T1D rely on exogenous insulin delivery to regulate their blood glucose (BG) levels (Liu et al., 2018). However this is a difficult and time consuming task that adds to the cognitive burden of T1D. For each meal and each activity an individual must calculate the right amount of insulin while accounting for all the other factors like stress that influence insulin needs. Yet getting the dose right is critical, as keeping BG within a healthy range can prevent many of the severe secondary complication of diabetes, such as chronic kidney disease (Levey et al., 2003) and heart disease and stroke (Mozaffarian et al., 2016).

A major advance in managing T1D is the introduction of continuous glucose monitors (CGMs). Unlike fingerstick BG measurements, which are regularly done before and after

meals, CGMs measure glucose every 5 minutes. An artificial pancreas (AP) system connects a CGM and pump with a controller that takes in CGM measurements, algorithmically determines the right insulin dosage and sends instructions to the pump. There are many AP projects, including some that also deliver glucagon, giving them the ability to raise BG as well as lowering it (El-Khatib et al., 2010). Any AP, though, must be able to accurately and reliably forecast where BG will be in the future and then determine what adjustments can safely bring this path into the target range. This is a challenging problem, as CGMs measure not BG but rather glucose from fluid between cells, which makes measurements delayed relative to BG. Further they have significant noise and error and can be affected by factors that don't influence BG (e.g. acetaminophen (Maahs et al., 2015)).

BG forecasting is a key place where ML can contribute to an AP. Prior work has shown that simple auto-regressive models can predict BG 30 minutes in advance with a root mean square error (RMSE) of 27.7 mg/dL (Botwey et al., 2014). Deep learning has recently been used for BG forecasting, including RNN (Li et al., 2019), CNN (Zhu et al., 2018) and LSTM (Mohebbi et al., 2020) (RMSE 21.07, 21.72, and 20.80 respectively). However these works primarily use data collected over a short duration of time from small sets of patients or simulation, so it is not yet known whether these methods can handle data collected in daily life, where error rates are higher and there are numerous unmeasured influences on glucose (e.g. running to catch a bus, having a stressful meeting). Further, as deep learning requires large training datasets it is not known what the upper bound on performance may be. Yet, there is now large-scale patient generated health data (PGHD) that can be used to more robustly evaluate ML for BG forecasting. The Open Artificial Pancreas (OpenAPS) is a project led by people with T1D who wanted a system they could use immediately and tailor to their needs (Lewis, 2019). The OpenAPS now has over 1500 users and contributors and a subset of participants have made their data available for research. The data has been used to study outcomes of the OpenAPS (Melmer et al., 2019), but despite being larger and more varied than any clinical dataset it has not yet been leveraged for testing ML algorithms.

In this work we focus on the task of forecasting BG in real-world settings. We provide a rigorous comparison of multiple ML methods and architectures and compare them on both the OpenAPS dataset and data collected for research use. We show how accuracy differs across the datasets from both statistical and clinical standpoints, and provide insights that can be used for selecting algorithms for BG forecasting. In the future, these improved forecasts can be incorporated into both research AP projects and potentially in OpenAPS systems to better close the loop. The code is publicly available at https://github.com/health-ai-lab/MLHC_BG_Forecasting.

Technical Significance Our primary technical contribution is rigorous comparison of ML methods for glucose forecasting. Prior works have applied existing methods to this task, often using small controlled datasets with few subjects (e.g. such as OhioT1DM (Zhu et al., 2018; Midroni et al., 2018; Xie and Wang, 2020) or simulated data (Samadi et al., 2017; Liu et al., 2018). However as PGHD becomes more prevalent, it is increasingly important to understand how methods fare on this considerably more challenging data, which has noise, error, and heterogeneity (e.g. individuals using different types of glucose monitors) not seen in controlled settings. Thus we present an in-depth and systematic comparison across 1) methods (e.g. baseline models like linear regression and random forest; traditional

forecasting models like ARMA; and deep learning), and 2) data types (small controlled datasets versus large observational data; multi-modality data versus CGM only). We further compare methods across task type (single-step and multi-output forecasting). This work provides new insight into the strengths and limitations of these ML approaches on real-world data and can provide insight into how to choose methods for real-world tasks.

Clinical Relevance T1D requires constant decision-making on how to manage glucose, primarily by adjusting insulin dosing for meals, activity, stress and all the other factors that affect BG. This is a significant cognitive burden, which is why an AP that fully automates insulin dosing is so transformative. Accurate management requires an accurate forecast of BG. Our clinical contribution is 1) the first rigorous comparison of BG forecasting using PGHD, and 2) comparison of methods across PGHD and benchmark research data. First, DIY systems are increasing in popularity, with over 1500 individuals using the OpenAPS (Lewis, 2019) alone. Further, research data often limits variation by ensuring all participants use the same devices (e.g. same CGM model), which may limit generalizability of results. Thus it is critical to understand how accurate methods are for forecasting BG on this much more challenging data compared to the highly controlled datasets used for research. More broadly, there is a need for more scholarship on such emerging techniques for diabetes management. Second, we provide a comparison of methods on PGHD (OpenAPS) and on a controlled research dataset (OhiotT1DM). OpenAPS is larger and covers many more days per individual than is possible in research datasets, so rigorously comparing methods on this diverse data to how they perform on controlled datasets can provide insight to guide future evaluation. In particular we examine whether methods that perform best on the research data are the best performers on the patient generated data, as this is critical for extrapolating from papers to real world settings in the future.

2. Related Work

In 2016, the FDA approved the first commercial AP (FDA, 2018), paving the way for intelligent decision support systems in diabetes management. An artificial pancreas has three key components: a CGM (to measure BG), an insulin pump (to deliver insulin to lower BG), and a control algorithm that takes in input from the CGM and sends instructions to the pump. There are numerous AP projects in various stages of development. While the algorithms and devices may differ, they all rely on accurate forecasts of where glucose is headed, so it can be kept within a target range. Thus we focus our efforts and our review of related work on approaches for BG forecasting. This is also a key focus for machine learning to improve BG control. The two primary categories of work for BG forecasting (i.e. predicting BG at a specific future time or window of times) are knowledge-based approaches that leverage models of BG dynamics, and data-driven approaches Table 1 summarizes key results from both groups..

Table 1: Related work in glucose forecasting and gylcemic event detection.

Task	Model	P_H (min)	Data	# Patients	# Days or # samples	Input			Performance
						Glucose	Insulin	Meals	
GF (Ståhl and Johansson, 2009)	ARMA(X)	60-120	Clinical	1	180	Fingerstick	✓	✓	RMSE
GF, Hypo, Hyper (Botwey et al., 2014)	cARX, RNN	15, 30, 45	Clinical	23	5.30 ±1.4	CGM	✗	✗	RMSE = 13.8, Time lag, correlation
GF, Hypo, Hyper (Liu et al., 2018)	Composite minimal model	30-120	Simulated	10	7	CGM	✓	✓	RMSE = 25.06, CEGA, F1, Sen, Spc, MCC
GF (Wang et al., 2013)	AR, ELM, SVR	30	Clinical	10	860 samples/subj	CGM	✗	✗	RMSE = 16.3*, CEGA, J-index
GF (Fox et al., 2018)	RNN (GRU)	30	FL	40	1.9k	CGM	✗	✗	APE = 4.59
GF (Li et al., 2019)	Convolutional RNN	30	Clinical, Simulated	10 each	180 (clin.), 360 (sim.)	CGM	✓	✓	RMSE = 21.07
GF (Mirshekarian et al., 2019)	LSTM	30, 60	OhioT1DM	6	56	CGM	✓	✓	RMSE = 18.74, CEGA

*average, GF: Glucose forecasting, FL: Free-living, P_H: Prediction horizon

2.1. Knowledge-based Approaches

Knowledge based approaches in BG forecasting aim to use physiological models of glucose-insulin dynamics to predict BG trajectory. Autoregressive (AR) models have been extensively used for BG forecasting in the past (Zarkogianni et al., 2013; Li et al., 2014; Eren-Oruklu et al., 2012; Novara et al., 2015; Cescon and Johansson, 2009; Gani et al., 2008), with different works exploring various configurations for autoregressive integerated moving average (ARIMA) model. Ståhl and Johansson (2009) used glucoregulatory subsystems based on compartmental models to develop several black-box and grey-box models such as ARMA, ARMA with exogenous inputs (ARMAX), and nonlinear ARMAX models. On fingerstick BG from a single individual they aimed to predict BG values 2 hours ahead with an RMSE of less than 18mg/dl (1 mmol/l) in 95% of the cases. The models did not achieve this goal but this work highlights some of the important limitations and challenges in BG forecasting using non-CGM data and system identification techniques, including the difficulty in accurately modeling glucoregulatory subsystems, handling missing data in sparsely sampled time-series, and modeling the inherent time-varying dynamics of the BG time-series data. Botwey et al. (2014) proposed a fusion strategy for combining an AR model with output correction (cARX) and a Recurrent Neural Network (RNN) for

glucose forecasting and subsequent prediction of hypo/hyperglycemia. They used data from 23 T1D patients in a clinical study with an average of 5.30 ±1.4 recorded days per subject. They achieved a median RMSE of 13.8 for $P_H = 30$ min using genetic algorithm fusion. Liu et al. (2018) propose a compartmental model of glucose-insulin dynamics with a deconvolution technique in estimating glucose levels, and showed that additional information such as on meals and physical activity improved performance over their uinvariate model. Using both simulated and clinical data from 10 subjects over one week, they achieved an RMSE of $35.96 \pm 4.65, p < 0.001$ for clinical test data. One of the limitations of classical autoregressive models is that because they estimate future values recursively, the estimations accumulate error over time. In order to mitigate this, these models usually require large amount of historical data to tune several parameters, which increases their already high computation time.

2.2. Data-driven Approaches

As CGMs improve in accuracy and more data is being collected, there has been increasing work toward fully data driven approaches. The key advantage of these methods is that they do not rely on prior knowledge about BG dynamics (which can be incomplete). Support Vector Regression (SVR) is one of the machine learning algorithms used in prior works including (Georga et al., 2013; Plis et al., 2014; Wang et al., 2013; De Bois et al., 2019; Plis et al., 2014) for both BG forecasting and glycemic event prediction. Wang et al. (2013) combined an AR model, extreme learning machine (ELM), and SVR, using a novel adaptive-weighting algorithm and achieved a higher performance than using each model separately. On data from 10 people with T1D the best and worst RMSE were 9.7±0.2 and 23.5±0.8, respectively. Since this is a time series forecasting problem, RNN and Long Short-term Memory (LSTM) are natural solutions, being widely used in sequence prediction (Lipton et al., 2015). Mirshekarian et al. (2019) used an LSTM for BG forecasting, evaluating it on both synthetic data and OhioT1DM data (6 subjects, 8 weeks), and report the best RMSE of 18.74±0.17 for the latter dataset. Fox et al. (2018) compared single-step and multi-output forecasting configurations, using an ensemble of polynomial and sequential functions and report a 50^{th} percentile absolute percentage error (APE) of 4.59, concluding that a multi-output approach outperforms single-step forecasting because of its ability to effectively capture the underlying glucoregulatory dynamics. Li et al. (2019) used a convolutional RNN and had high accuracy on simulated data (mean RMSE 9.38±0.71), but error increased significantly when applied to patient data (21.07±2.35).

3. Problem Setup

In this work we focus on the task of BG forecasting, which can be modeled in two main ways: recursively, where estimates are used for future forecasts, or directly, where prior estimates do not influence future estimates.

The basic recursive setup is as follows:

$$x'(t) = \phi[(x(t-1), x(t-2), ..., x(t-n))] \tag{1}$$

$$x'(t+1) = \phi[(x'(t), x(t-1), x(t-2), ..., x(t-n))] \tag{2}$$

$$x'(t+h) = \phi[(x'(t+(h-1)), ..., x(t-1), x(t-2), ..., x(t-n))]. \tag{3}$$

In this setup, x and x' are the actual and estimated value of the variable being forecast (BG in our task), t is the timepoint the prediction is being made for, and ϕ is the function being used to make predictions. P_H denotes the prediction horizon, meaning how many timepoints in advance are being forecast. Thus $h = 0, ..., P_H$. For a prediction horizon of 30 minutes with input values recorded every 5 minutes, $P_H = 6$. As shown, the prediction for the last timepoint $t + h$ is a function of all prior predictions.

In a direct method, on the other hand, the data is split into chunks of fixed sized sequences of historical and future points which then become features and target output for a classic regression task. With variables defined as before there is now no dependence between forecasts:

$$x'(t+h) = \phi_h[(x(t-1), x(t-2), ..., x(t-n))]. \tag{4}$$

We can further differentiate between methods based on their output. The basic setups shown here provide single step forecasts. That is, each time ϕ is called, it produces an estimate of a single BG value at a single time. Multi-output forecasts in contrast can estimate all values in the prediction horizon simultaneously, thereby predicting the progression or trajectory of the signal over the prediction horizon. Such approaches may better ensure smoothness and consistency in forecasts, and capture dependencies between the forecasts. A downside is that like for recursive forecasts, errors can propagate rather than being contained to a single output.

In this work, we aim to compare recursive and direct methods for single-step and multi-output BG forecasting. We also investigate the impact of using univariate and multivariate data on the accuracy of future predictions. For blood glucose prediction, let $X_{0:t-1} = \{x_0, x_1, ..., x_{t-1}\} \in \mathbb{R}^d$ denote a multi-variate time series for multi-modality data obtained from d different sources at t timepoints. Sources could be a CGM, insulin pump, amount of carbohydrate intake, and activity. We assume all data sources are continuous-valued, but they may be measured at different frequencies. The output $X'_{t:t+P_H} = \{ x'_t, x'_{t+1}, x'_{t+2}, ..., x'_{t+P_H}\} \in \mathbb{R}$ represents multiple future glucose values across a given prediction horizon P_H. In this context, single-step methods estimate $p(x_{t+P_H}|x_{0:t-1})$ with x_{t+P_H} being the glucose value P_H time instances in the future. This can be achieved in a recursive or direct way. Multi-output forecasting, on the other hand, aims to estimate the joint probability $p(x_{t:t+P_H}|x_{0:t-1})$ in one step so it is a direct method.

4. Methods

The methods used for blood glucose prediction are divided into three categories:

- **Baseline Models:** Popular machine learning models used for regression in healthcare applications, namely linear regression (LR), Random Forest Regression Trees (RF), and Support Vector Regression (SVR).
- **Recursive Model:** This includes the classic auto-regressive moving average (ARMA) model used in time-series forecasting problems.
- **Deep Learning Models:** A simple Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) model.

4.1. Baseline models

4.1.1. Linear regression

In an LR model, the core assumption is that future values depend linearly on past glucose levels, as shown in equation (5),

$$x'_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t}, ..., +\beta_n x_{n,t} + \varepsilon_t \tag{5}$$

where x'_t represents the future time series, β is the set of parameters for the model and $x_{k,t}$ represents the past values for the k^{th} feature. For the univariate case $k = 1$ and for the multivariate setting with insulin basal and bolus rates, meals, and glucose differences used as additional features, $k = 5$. ε_t denotes the random error present in the learned model with an underlying assumption that $\varepsilon_t, .., \varepsilon_T$ have zero mean and are independent of the predictor variables, where T is the total number of future values to be predicted (Hyndman and Athanasopoulos, 2018). We used a least square approach for learning the model parameters.

4.1.2. Random forest

RF regression is an ensemble of multiple decision trees trained using bootstrap samples and the final estimations are made by averaging the results obtained from each tree. We used 100 estimators in our RF model and use mean squared error (MSE) to measure the quality of each split.

4.1.3. Support vector regression

For SVR, there were three main parameters: cost C, insensitive parameter ε, and the type of kernel function along with the kernel parameters. We used the default parameter configuration of sklearn library in python with $C = 1$, $\varepsilon = 0.1$ and radial basis function (rbf) as a kernel with γ scaled according to the variance of the input samples used as support vectors.

4.2. Recursive modeling

4.2.1. ARMA

We use ARMA to perform recursive single-step forecasting. Equation (6) describes the time-series forecasting problem using ARMA model where x_{t-i} are previous glucose values with $(i = 0, ..., p + 1)$ and p being the lag order. e_{t+n} and e_{t-i} for $(i = 0, ..., q + 1)$ are the residual terms obtained by subtracting the estimated glucose value from the true value with q being the size of the moving average window.

$$x_{t+n} = (a_1 x_t + a_2 x_{t-1} + ... + a_p x_{t-p-1}) + (e_{t+n} + c_1 e_t + ... + c_q e_{t-q-1}). \tag{6}$$

An automatic grid search was performed using pyramid's auto.arima library in python (Pyramid, 2018) to select the best values of p and q. The function uses Akaike information criterion (AIC) (Konishi and Kitagawa, 2008) to determine the best set of parameters and instead of doing an exhaustive grid search, uses the step-wise algorithm proposed by (Hyndman and Khandakar, 2008). Despite this optimization, the model has high time and memory costs. The parameters $(p = 2, q = 1)$ gave the best performance.

4.3. Deep learning models

A typical feed-forward deep neural network takes a sequence of inputs and maps them to an output space through a combination of linear operations and non-linear activations. An RNN additionally has feedback from the preceding hidden states, thereby using not only the current input but also the state representations learned from the previous inputs in making the current estimation. Therefore, it takes into account the larger context and long-term dependencies by circulating past information within the network during the learning process. More advanced configurations of RNN include LSTM (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014), which better capture long-term dependencies in sequential data but have more complex architectures and require more data.

4.3.1. LSTM and RNN

We use an LSTM with a single hidden layer, $H(t)$ with 32 units, followed by a fully-connected output layer $O(t)$. We compare it with a fully-connected vanilla RNN which also has a single hidden layer with 32 units. The number of units for both LSTM and RNN were chosen after trying and testing [28, 32, 64, 128] and optimizing for the lowest RMSE. The output layer $O(t)$ predicts the glucose value(s) 30 minutes into the future. The input $I(t)$ at time instance t consists of the glucose values in the past hour $X(t)$ and the state representation $H(t-1)$ learned by the hidden layer in the previous time instance.

$$I(t) = X(t) + H(t-1) \tag{7}$$

At time instance t, the hidden layer H_j receives the input consisting of past glucose values and output of the hidden layer in the previous time step. It learns a set of weights θ_{ji} for every i^{th} input which then passes through an activation function giving the state vector representation for that time instance. In this work, we use a ReLU activation function given by equation (9).

$$H_j(t) = \sigma\left(\sum_i I_i(t)\theta_{ji}\right) \tag{8}$$

$$\sigma(z) = max(0, z) \tag{9}$$

For single-step forecasting, there is a single output unit $O(t)$ which learns a set of weights θ'_j to predict the future glucose value for a single time step. A multi-output RNN has $k = 6$ ($P_H = 30$ min) in the output layer because it predicts all the glucose values in the P_H simultaneously. We do not use any activation function in the output layer since we are using the model to estimate continuous values. Thus,

$$O_k(t) = \sum_j H_j(t)\theta'_{kj}. \tag{10}$$

5. Experiments

Our experiments aim to compare BG forecasting methods across a large PGHD dataset (1-4 years of data for over 50 people) and controlled research data (8 weeks of data for 12 people). We first describe the datasets and then the evaluations.

5.1. Data

5.1.1. OpenAPS

The OpenAPS Project (Melmer et al., 2019) is a participant-led artifical pancreas system. The system itself has been made open source, and participants in the project can also volunteer to donate their diabetes-related data for use by researchers. As a result, the data available for each individual varies, and includes CGM, insulin (basal and bolus rates), carbohydrate intake, and insulin doses calculated by the OpenAPS system. As individuals use their own devices, the exact models of insulin pump, CGM, and other devices will vary across participants. We use a subset of the available OpenAPS data in this work, only including participants with multiple calendar years of data even if data was recorded for only a portion of each year, in order to allow training on prior years and testing on the last one. This resulted in a final dataset of 55 participants with an average of 320±158.34 days of data (median 287 days). In total, this dataset has more than 17,000 days of data with more than 7 million raw glucose measurements (∼4.5M training, ∼3M testing).

5.1.2. OhioT1DM

All the experiments were repeated on OhioT1DM dataset which is a widely used benchmark dataset for BG prediction (Marling and Bunescu, 2020). The updated dataset released in 2020 has eight weeks of data for 12 people with T1D. In total this yielded 600 days of data, with 177,000 CGM values (∼ 144k training, ∼33k testing), making it more than 40 times smaller than OpenAPS dataset. The average and median number of recorded days were 54±3.02 and 56 days per person. Out of the twenty features originally recorded in the dataset, we used CGM values, insulin bolus and basal rates and carbs intake, for consistency with the OpenAPS dataset.

5.2. Data processing

For both datasets, we extracted four features for BG forecasting: raw CGM values (mg/dL), insulin basal rate, insulin bolus events, and carbs intake (mg). In addition to these four features, we also calculated the difference between consecutive glucose levels and used them for BG forecasting in the multivariate setting.

Because of the high error rates and noise in CGM data, we include a number of pre-processing steps to remove implausible values and fill in missing ones. First, we excluded any values below $15 mg/dL$, as glucose values cannot fall below this level for a person who is still conscious. Missing glucose values in the training set were imputed using linear interpolation for gaps less than 30 minutes in duration. Interpolation imputes a missing value x_t at time t by assuming a linear relationship between past data points ($x_{t_{-1}}$), and future data points ($x_{t_{+1}}$) and fits a straight line using equation (11) (Lepot et al., 2017).

$$x_t = \frac{x_{t_{-1}} - x_{t_{+1}}}{t_{-1} - t_{+1}}(t - t_{+1}) + x_{t_{+1}} \tag{11}$$

where $t1 < t < t2$. For test data, we used first-order extrapolation to avoid using data from the future.

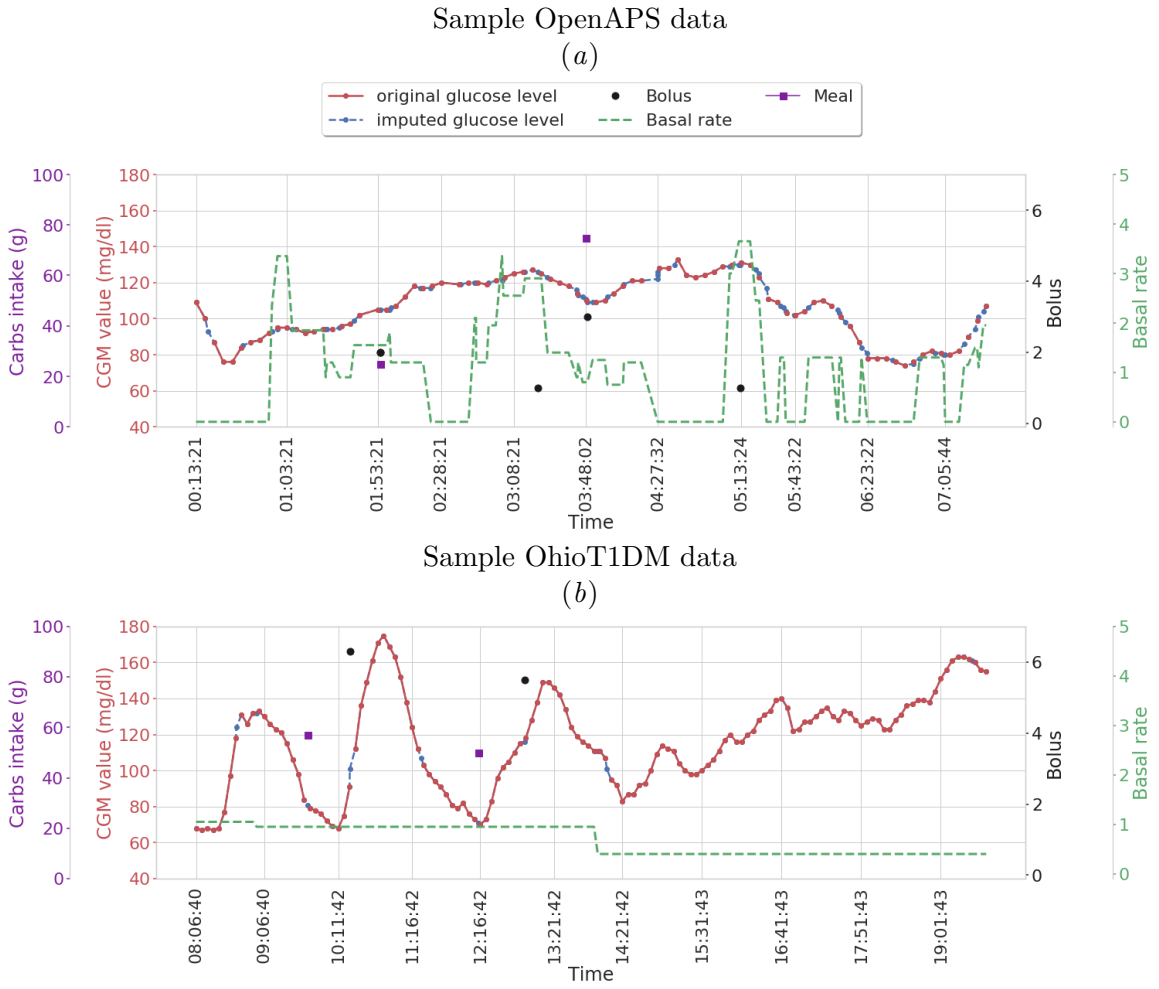


Figure 1: Multimodality data for (a) OpenAPS (b) OhioT1DM.

The gaps in basal insulin rate recordings were forward filled, meaning missing values were replaced with the last recorded basal rate. This is because a basal rate is recorded when it is changed, so we know that the value is constant between data points. If a temporary basal infusion rate was recorded for a given set of timestamps, it was used to supersede the recorded basal rate (Marling and Bunescu, 2020) by evenly distributing its value across the time duration which was divided into 5-minute intervals, as done in (Midroni et al., 2018; Xie and Wang, 2018). Bolus rates (discrete amounts of insulin given to cover events like meals) were handled in a similar manner by distributing each bolus event evenly across the specified duration. Outside of recorded bolus events, the value was set to zero as no insulin was bolused for those times. Similarly, the amount of carbohydrate intake was set to 0 when no value was recorded, as carbs are only recorded at meal times. Figure 1 shows an example of a short sequence of multivariate time series from each dataset. Different pre-processing steps are listed in the Appendix in table 4.

We used a median filter with a window size of 5 samples (Zhu et al., 2018) to smooth the CGM data and address discontinuities, which cause unnecessary variance in predictions. This was only done for training data and not for the test set to test robustness of the models for real-world use.

For direct forecasting, a sliding window was used to split the data into fixed sized sequences of past and future BG values. There were three parameters for the moving window i.e. size of the history window (size of historical data to use for forecasting), prediction offset and horizon (how far into the future and how many future values to predict) and stride (number of samples to skip while sliding the window). An hour (12 samples for 5-minute intervals) of past values were used with a prediction horizon of 30 minutes ($P_H = 6$ samples). We used a unit stride which means overlapping windows were used to partition the data. For the single-step setting, a single future glucose value was estimated whereas for multi-output prediction, six consecutive future glucose values were estimated for the next 30 minutes, simultaneously. Figure 2 shows the data used from each source before and after the processing steps which included synchronization, thresholding, and imputation. After processing, the average number of recorded days per subject were 320±158.34 and 54±3.02 for OpenAPS and OhioT1DM, respectively.

5.3. Training

We divided the experiments into four categories, 1) single-step univariate, 2) single-step multivariate, 3) multi-output univariate, and 4) multi-output multivariate. ARMA was used to perform only single-step univariate forecasting due to its high computation time. The baseline and deep learning models were used to test both category 1 and 2. For category 3 and 4, only deep learning models were compared since they have been used in prior works for multi-output forecasting (Fox et al., 2018).

For the univariate setting, the time-series consisting of only raw CGM values was given to the input layer, whereas in the multivariate setting, multimodality time series for CGM, insulin boluses, basal amounts and meal intake, along with the difference between adjacent glucose levels were included in the input. Early stopping was used to halt the training process if validation loss was not improving significantly, with the maximum number of epochs being 1000 with a batch size of 248 and 32 for OAPS and OhioT1DM, proportional to the size of each dataset. Glorot normal initialization (Glorot and Bengio, 2010) was used to initialize the weight matrix in deep learning models. Each experiment was repeated 3 times and average RMSE and standard deviation was calculated across all the trials for each subject. As we aim to train a single model using data from all the patients, we shuffled the ordered list of subjects before each trial so that the model's learning did not depend on the order in which data from each subject was passed to it.

5.4. Evaluation

We make our evaluations from three main perspectives, 1) difference between individual modeling methods, 2) impact of forecasting configurations, and 3) effect of data size and data quality on the accuracy of the predictions. We use both statistical and clinical metrics to quantify the performance.

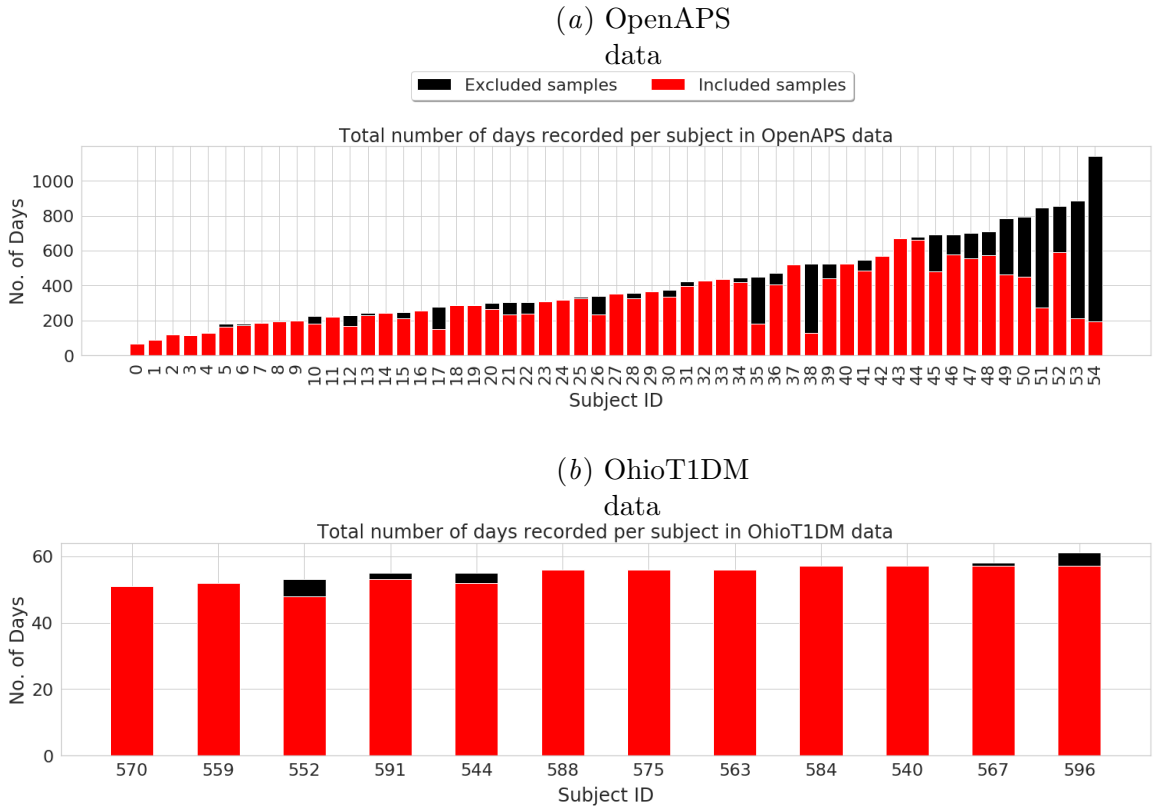


Figure 2: The total number of days for which multimodality data was recorded for each subject in (a) OpenAPS (b) OhioT1DM dataset. The black bars show the samples excluded post processing and the red bars represent the data actually used in the experiments.

5.4.1. Root Mean Square Error (RMSE)

The statistical measure used was RMSE given by equation (12), where x and $\hat{x}$ are the actual and estimated BG values, respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{n=1}^{n}(x_i - \hat{x}_i)^2} \tag{12}$$

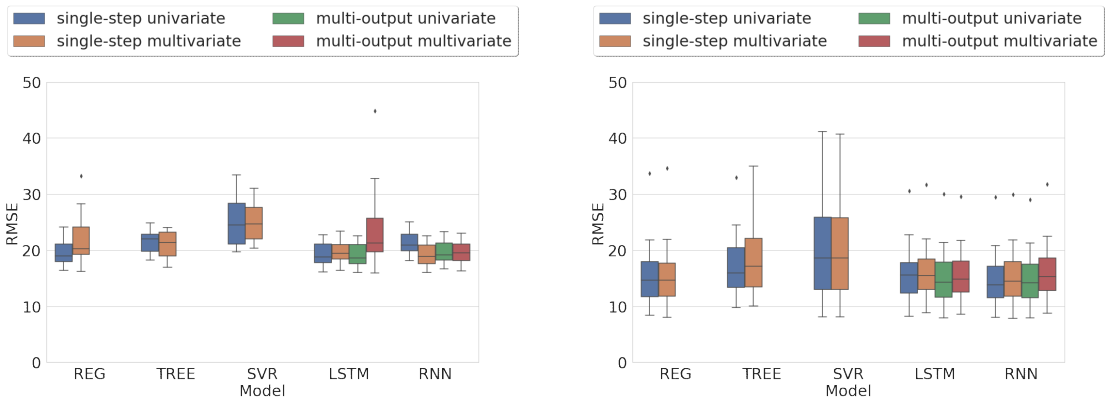
We use RMSE as our primary metric because it has been widely used in prior works to measure the deviation of estimated BG values from true BG levels. Since each experiment was repeated 3 times, we record the mean RMSE for each subject. We then take an average of all the RMSEs and final results are expressed as average RMSE $\pm$ standard deviation (SD) over all the subjects.

Table 2: Best(B), worst(W) and mean(M) RMSE for glucose prediction with $P_H = 30$ minutes across different subjects for each model. The best mean performance for each dataset is bolded.

		Output setting							
		Single-step						Multi-output	
	RMSE	LR	RF	SVR	ARMA	RNN	LSTM	RNN	LSTM
Univariate	Ohio: B	16.37	18.26	19.73	19.84	16.12	18.13	16.00	16.70
	Ohio: W	24.12	24.84	33.38	31.47	22.71	25.05	22.57	23.26
	Ohio: M	19.59	21.66	25.22	26.64	19.22	21.33	19.12	19.69
	OpenAPS: B	8.42	9.76	8.07	8.44	8.23	8.05	7.95	7.92
	OpenAPS: W	33.73	32.94	41.12	34.46	30.57	29.42	30.00	29.03
	OpenAPS: M	15.34	17.09	19.68	21.42	15.50	**14.53**	15.02	**14.73**
Multivariate	Ohio: B	16.22	16.95	20.37	—	16.36	15.99	15.91	16.27
	Ohio: W	33.26	24.02	31.06	—	23.37	22.53	44.79	22.98
	Ohio: M	22.01	20.97	25.06	—	19.68	19.25	23.95	19.57
	OpenAPS: B	8.00	10.03	8.08	—	8.83	7.84	8.59	8.73
	OpenAPS: W	34.58	35.02	40.69	—	31.64	29.95	29.57	31.72
	OpenAPS: M	15.33	18.12	19.82	—	15.95	15.07	15.42	15.95

5.4.2. Clarke's Error Grid Analysis (CEGA)

Clarke's Error Grid (Clarke, 2005) is another approach used for evaluating BG predictions. It is primarily used in clinical works, as it provides more detailed insight into the safety of predictions. Here we use it to assess the clinical significance of the best performing models for each of the dataset. For this analysis, actual CGM values (x-axis) are plotted against predicted values (y-axis). Points on the diagonal represent perfect accuracy, while those above and below are overestimations and underestimations, respectively. To capture the severity of these errors, there are 5 regions on the grid labelled A – E. Points within 20% of the actual values lie in Zone A, which is within sensor error ranges. Points located in Zone B represent estimations with an error of more than 20%, but which would not lead to incorrect treatment choices. Zones C-E can be potentially dangerous failures and are counted as clinically risky mistakes, as they may lead to overtreatment, undetected hypo/hyperglycemia or in the worst case conflation of hypo/hyperglycemia (Maran et al., 2002). We report the score by showing the percentage of total samples for all the subjects that lie in each of the five regions, for each of the two datasets.



(a) OhioT1DM (b) OpenAPS

Figure 3: Performance summary of deep learning models with four different configurations for each dataset.

6. Results

Table 2 shows the RMSE values for different model configurations and modality. We report the best (B), worst (W) across all 12 and 55 subjects in both OhioT1DM and OpenAPS datasets, respectively, and average RMSE across the iterations.

We now analyze the results from three main perspectives: methods, input/output (uni and multivariate, single and multi-step), and data. We also present an ablation study to better understand the impact of the pre-processing steps that were used for all tasks.

6.1. Modeling techniques

The recursive ARMA model had the highest average RMSE for the single-step univariate setting. Figure 3 gives a visual summary of the of results in Table 2 for baseline and deep learning models. In terms of variance, deep learning models tend to perform better than baseline models, shown by the height of the boxes and the number of outliers for the OpenAPS dataset. SVR has the the highest variability of all, but as shown in figure 3(b) it is the only model for which there are no outlier estimations, which shows its ability to generalize well in case of a bigger dataset at the cost of having the highest average RMSE. Among deep learning models, performance is comparable and there does not seem to be an overall best model. LSTM, however performs better with more data as compared to RNN across OhioT1DM and OpenAPS, for both single-step and multi-output settings. Direct comparison with prior work is difficult due to differences in the datasets used in terms of total number of subjects and recorded days. However, we note that RMSE of single-step, univariate LSTM on OhioT1DM is comparable to 19.51 achieved by Mirshekarian et al. (2019) using OhioT1DM with 6 subjects instead of 12 used in our work. This performance can be improved by using multiple techniques like adding drop-out layers, using memory augmentation and attention modules as done by Mirshekarian et al. (2019), reducing the RMSE to 18.75. We used simple, single-layer, deep learning models for fair comparison

between baseline models for different settings. Future work will be needed to compare more advanced architectures.

6.2. Forecasting configurations

6.2.1. Univariate vs Multivariate

For both baseline and deep learning models, the performance slightly decreased from univariate to multivariate setting for single-step forecasting, although it changed the least for LSTM, where the average error increased by a factor of only 0.04 for OpenAPS, compared to 0.11 for RNN. Adding more features is believed to improve forecasting accuracy, as shown by Georga et al. (2012), who used data from 15 subjects part of a clinical study over 5-22 days. That work achieved an average RMSE of 9.15 for single-step multivariate forecasts using insulin and meal information, versus 15.29 using CGM data only. They used an SVR model and tuned it iteratively using grid search. However, in our results, we observe that adding more features adds more perturbations in the BG predictions for some models than others. For example, among baseline models, linear regression was harmed the most by adding more features, more so for OpenAPS dataset than OhioT1D, with SD changing from 5.79 to 13.81 from univariate to multivariate setting for single-step forecasting. Overall, adding more features led to the same or slightly worse average RMSE for all methods. This suggests that simply adding more variables is not necessarily better and more work is needed to figure out how best to use this additional data.

6.2.2. Single-step vs Multi-output

There was not a significant change in performance for deep learning models between the single-step and multioutput setting. However, the best (B) RMSE was lowest using multi-output univariate LSTM (7.92 for OAPS), showing that learning the trajectory of BG values versus predicting a single future value, improves the accuracy of the predictions. This is in line with the observations made by Fox et al. (2018) who achieved an absolute percentage error of 5.01 using multi-output GRU versus 5.31 achieved using single-output recursive network on a large dataset collected from 40 patients over 3 years.

6.3. Data size and quality

For the univariate setting, we always see an improvement in best and average error from OhioT1DM to OpenAPS dataset, showing that more data not only helps in training more accurate deep learning models but also improves baseline models like LR. On the other hand, the worst case RMSE for OpenAPS dataset is in most cases significantly worse than for the OhioT1DM dataset, for all methods. This highlights the difficulty in training accurate models using large-scale, heterogeneous and noisy PGHD, and shows that variation in homogeneous research datasets may not predict performance for individuals using these methods in the real world. However, large data even when noisy can yield better average case performance than smaller controlled datasets. The worst case RMSE was lowest (best) for LSTM and highest (worse) for SVR, using OpenAPS. This shows the robustness of deep architectures to noisy data. This is also shown by the presence of not a single outlier

Table 3: Results for ablation study comparing the effect of median filtering and imputation technique on forecasting RMSE.

		Imputation Technique	
		No imputation	Interpolation
Unfiltered	Ohio: best	35.79	16.96
	Ohio: worst	50.793	23.61
	Ohio: mean	42.71	19.80
	OpenAPS: best	12.73	8.50
	OpenAPS: worst	37.02	33.86
	OpenAPS: mean	20.61	15.48
Filtered	Ohio: best	16.13	16.70
	Ohio: worst	23.29	23.26
	Ohio: mean	19.37	19.69
	OpenAPS: best	11.46	7.92
	OpenAPS: worst	31.46	29.03
	OpenAPS: mean	19.14	**14.73**

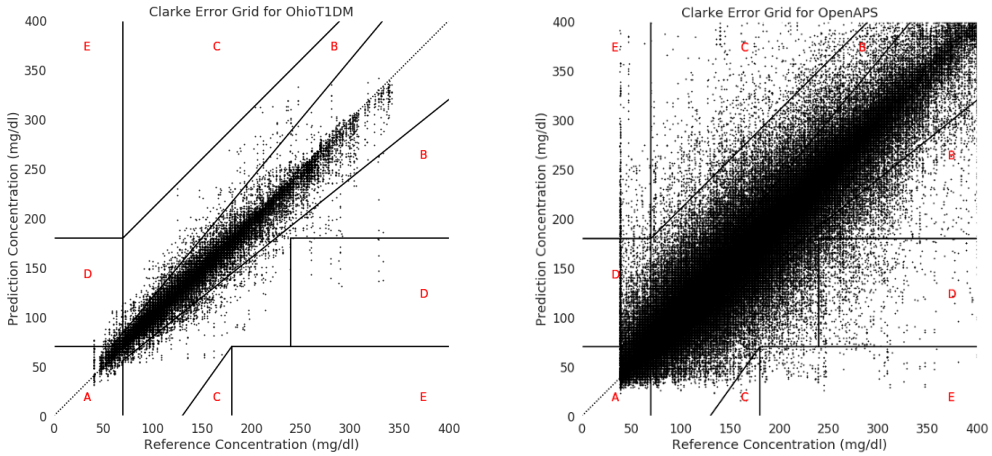
estimation in figure 3(a) for OhioT1DM dataset and almost every model except SVR having outliers in 3(b) for OpenAPS. The variance in performance increases from OhioT1DM to OpenAPS shown by the height of the boxes. The best and median values of error for OhioT1DM are much higher than those for OpenAPS, which also has outliers.

6.4. Ablation Study

We now examine the effect of our pre-processing steps that were conducted for all methods. In particular, we vary filtering (CGM data passed through a median filter, or not) and imputation (linear interpolation or deletion of missing instances) and test both combinations, leading to four different configurations. We repeat this for both the Ohio and OpenAPS datasets, using the multi-output, univariate LSTM algorithm for forecasting. Results are shown in table 3. First, we see that regardless of imputation approach or dataset, filtering the CGM data improves performance. This is due to the filter smoothing out some of the jumpiness inherent in CGM data. Next, we find that imputation can significantly improve results. On the unfiltered data, there is a dramatic improvement in accuracy on Ohio data (mean 42.71 RMSE no imputation, 19.80 with imputation). OpenAPS similarly improves from a mean of 20.61 RMSE to 15.48 with interpolation. Once data is filtered the improvement on OpenAPS is smaller, and results are similar with and without interpolation on Ohio. However, the best results out of the four configurations across two datasets was on OpenAPS with both filtering and imputation, suggesting these pre-processing steps and large data are keys to high accuracy.

6.5. Clinical Analysis

Finally, we used Clarke's error grid method to assess the clinical relevance of the results. As shown in Table 2, single-step LSTM using univariate input gave the least average RMSE for the bigger dataset. In Figure 4 we plot the BG values estimated by a multi-output, univariate LSTM model versus the actual BG values for the 12 and 55 subjects in OhioT1DM and OpenAPS datasets respectively. We find that for OhioT1DM, 99.36% ($n = 29$k points) of the values are in zone A and B, 0% in zone C, and 0.64%($n = 191$) in zone D. For OpenAPS, these values are as follows: Zone A and B: 99.43% ($n = 16$M), zone C: 0.029% ($n = 4$k), zone D: 0.52% ($n = 91$k). Thus the types of errors made and their impact on decision-making (i.e. treatment of BG highs and lows) are substantially similar across the two datasets. However, due to the size of the OpenAPS data, the number of predictions that may lead to an incorrect decision is substantial, numbering in the thousands.



(a) OhioT1DM (b) OpenAPS

Figure 4: Clarke's error grid analysis for all the subjects in each of the two datasets.

7. Conclusion

We have done a rigorous comparison between various configurations for time-series forecasting for BG prediction, in the context of PGHD like OpenAPS and benchmark datasets like OhioT1DM. We have shown that the state-of-the-art forecasting models known to perform well on small, controlled datasets in prior works, give statistically and clinically different performance on large-scale PGHD datasets like OpenAPS. Another observation we made was that different models are affected differently based on the dimension of feature space, and adding more information about meals and insulin does not always boost performance and can sometimes even degrade accuracy, such as in the case of linear regression. Conventional time-series forecasting models like ARMA failed to capture the underlying differences in the BG patterns of different subjects which becomes more prominent in larger datasets with more subjects with data recorded over a longer duration. In all our experiments, we used models with most basic configurations in order to do a fair comparison. Our aim was

not to propose an ultimate best model for BG forecasting, but to compare how different type of machine learning techniques in their most general form, fare across different forecasting settings and datasets. Overall we show that BG forecasting is a problem that should be studied in the context of different size and quality of data, output settings, and modality of input data.

Acknowledgments

This work was supported in part by the NSF under award number 1915182, NIH under award number R01LM011826, and a Fulbright Scholarship.

References

Ransford Henry Botwey, Elena Daskalaki, Peter Diem, and Stavroula G Mougiakakou. Multi-model data fusion to improve an early warning system for hypo-/hyperglycemic events. In *The 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4843–4846, 2014.

Marzia Cescon and Rolf Johansson. Glycemic trend prediction using empirical model identification. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3501–3506, 2009.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

William L. Clarke. The original Clarke error grid analysis (EGA). *Diabetes technology & therapeutics*, 7(5):776–779, 2005.

Maxime De Bois, Mounîm A El Yacoubi, and Mehdi Ammi. Study of short-term personalized glucose predictive models on type-1 diabetic children. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

Firas H El-Khatib, Steven J Russell, David M Nathan, Robert G Sutherlin, and Edward R Damiano. A bihormonal closed-loop artificial pancreas for type 1 diabetes. *Science translational medicine*, 2(27):27ra27–27ra27, 2010.

Meriyan Eren-Oruklu, Ali Cinar, Derrick K Rollins, and Lauretta Quinn. Adaptive system identification for estimating future glucose concentrations and hypoglycemia alarms. *Automatica*, 48(8):1892–1897, 2012.

FDA. Fda's efforts to advance artificial pancreas device systems. https://www.fda.gov/medical-devices/consumer-products/artificial-pancreas-device-system, 2018. [Online; accessed 03-Oct-2019].

Ian Fox, Lynn Ang, Mamta Jaiswal, Rodica Pop-Busui, and Jenna Wiens. Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1387–1395, 2018.

Adiwinata Gani, Andrei V Gribok, Srinivasan Rajaraman, W Kenneth Ward, and Jaques Reifman. Predicting subcutaneous glucose concentration in humans: data-driven glucose modeling. *IEEE Transactions on Biomedical Engineering*, 56(2):246–254, 2008.

Eleni I Georga, Vasilios C Protopappas, Diego Ardigo, Michela Marina, Ivana Zavaroni, Demosthenes Polyzos, and Dimitrios I Fotiadis. Multivariate prediction of subcutaneous glucose concentration in type 1 diabetes patients based on support vector regression. *IEEE Journal of Biomedical and Health Informatics*, 17(1):71–81, 2012.

Eleni I Georga, Vasilios C Protopappas, Diego Ardigò, Demosthenes Polyzos, and Dimitrios I Fotiadis. A glucose model based on support vector regression for the prediction of hypoglycemic events under free-living conditions. *Diabetes technology & therapeutics*, 15(8):634–643, 2013.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Robin John Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of Statistical Software*, 27:1–22, 2008.

International Diabetes Federation. IDF Diabetes Atlas Eighth edition. https://www.idf.org/e-library/epidemiology-research/diabetes-atlas/134-idf-diabetes-atlas-8th-edition.html, 2018.

Sadanori Konishi and Genshiro Kitagawa. *Information criteria and statistical modeling*. Springer Science & Business Media, 2008.

Mathieu Lepot, Jean-Baptiste Aubin, and François HLR Clemens. Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. *Water*, 9(10):796, 2017.

Andrew S Levey, Josef Coresh, Ethan Balk, Annamaria T Kausz, Adeera Levin, Michael W Steffes, Ronald J Hogg, Ronald D Perrone, Joseph Lau, and Garabed Eknoyan. National kidney foundation practice guidelines for chronic kidney disease: evaluation, classification, and stratification. *Annals of internal medicine*, 139(2):137–147, 2003.

Dana Lewis. History and perspective on DIY closed looping. *Journal of Diabetes Science and Technology*, 13(4):790–793, 2019.

Kezhi Li, John Daniels, Chengyuan Liu, Pau Herrero-Vinas, and Pantelis Georgiou. Convolutional recurrent neural networks for glucose prediction. *IEEE Journal of Biomedical and Health Informatics*, 2019.

Peng Li, Lei Yu, Jiping Wang, Liquan Guo, and Qiang Fang. Effect of meal intake on the quality of empirical dynamic models for type 1 diabetes. In *2014 IEEE International Symposium on Bioelectronics and Bioinformatics (IEEE ISBB 2014)*, pages 1–4, 2014.

Zachary C Lipton, David C Kale, and Randall C Wetzel. Phenotyping of clinical time series with LSTM recurrent neural networks. *arXiv preprint arXiv:1510.07641*, 2015.

Chengyuan Liu, Josep Vehi, Nick Oliver, Pantelis Georgiou, and Pau Herrero. Enhancing blood glucose prediction with meal absorption and physical exercise information. *arXiv preprint arXiv:1901.07467*, 2018.

David M Maahs, Daniel DeSalvo, Laura Pyle, Trang Ly, Laurel Messer, Paula Clinton, Emily Westfall, R Paul Wadwa, and Bruce Buckingham. Effect of acetaminophen on cgm glucose in an outpatient setting. *Diabetes Care*, 38(10):e158–e159, 2015.

Alberto Maran, Cristina Crepaldi, Antonio Tiengo, Giorgio Grassi, Emanuela Vitali, Gianfranco Pagano, Sergio Bistoni, Giuseppe Calabrese, Fausto Santeusanio, Frida Leonetti, et al. Continuous subcutaneous glucose monitoring in diabetic patients: a multicenter analysis. *Diabetes Care*, 25(2):347–352, 2002.

Cindy Marling and Razvan Bunescu. The OhioT1DM Dataset for Blood Glucose Level Prediction: Update 2020. 2020.

Andreas Melmer, Thomas Züger, Dana M Lewis, Scott Leibrand, Christoph Stettler, and Markus Laimer. Glycaemic control in individuals with type 1 diabetes using an open source artificial pancreas system (OpenAPS). *Diabetes, Obesity and Metabolism*, 21(10): 2333–2337, 2019.

Cooper Midroni, Peter J Leimbigler, Gaurav Baruah, Maheedhar Kolla, Alfred J Whitehead, and Yan Fossat. Predicting glycemia in type 1 diabetes patients: experiments with xgboost. *Heart*, 60(90):120, 2018.

Sadegh Mirshekarian, Hui Shen, Razvan Bunescu, and Cindy Marling. Lstms and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 706–712. IEEE, 2019.

Ali Mohebbi, Alexander R Johansen, Nicklas Hansen, Peter E Christensen, Jens M Tarp, Morten L Jensen, Henrik Bengtsson, and Morten Mørup. Short term blood glucose prediction based on continuous glucose monitoring data. *arXiv preprint arXiv:2002.02805*, 2020.

D Mozaffarian, EJ Benjamin, AS Go, DK Arnett, MJ Blaha, M Cushman, SR Das, S de Ferranti, JP Després, HJ Fullerton, et al. Heart disease and stroke statistics-2016 update: a report from the american heart association. *Circulation*, 133(4):e38, 2016.

Carlo Novara, N Mohammad Pour, Tyrone Vincent, and Giorgio Grassi. A nonlinear blind identification approach to modeling of diabetic patients. *IEEE Transactions on Control Systems Technology*, 24(3):1092–1100, 2015.

Kevin Plis, Razvan Bunescu, Cindy Marling, Jay Shubrook, and Frank Schwartz. A machine learning approach to predicting blood glucose levels for diabetes management. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

Python Pyramid. Pyramid auto.arima. http://alkaline-ml.com/pmdarima/0.9.0/modules/generated/pyramid.arima.auto_arima.html, 2018. [Online; accessed 10-Dec-2019].

Sediqeh Samadi, Kamuran Turksoy, Iman Hajizadeh, Jianyuan Feng, Mert Sevil, and Ali Cinar. Meal detection and carbohydrate estimation using continuous glucose sensor data. *IEEE Journal of Biomedical and Health Informatics*, 21(3):619–627, 2017.

Fredrik Ståhl and Rolf Johansson. Diabetes mellitus modeling and short-term prediction based on blood glucose measurements. *Mathematical biosciences*, 217(2):101–117, 2009.

Youqing Wang, Xiangwei Wu, and Xue Mo. A novel adaptive-weighted-average framework for blood glucose prediction. *Diabetes technology & therapeutics*, 15(10):792–801, 2013.

Jinyu Xie and Qian Wang. Benchmark machine learning approaches with classical time series approaches on the blood glucose level prediction challenge. In *KHD@ IJCAI*, pages 97–102, 2018.

Jinyu Xie and Qian Wang. Benchmarking machine learning algorithms on blood glucose prediction for type 1 diabetes in comparison with classical time-series models. *IEEE Transactions on Biomedical Engineering*, 2020.

Konstantia Zarkogianni, Eleni Litsa, Andriani Vazeou, and Konstantina S Nikita. Personalized glucose-insulin metabolism model based on self-organizing maps for patients with type 1 diabetes mellitus. In *13th IEEE International Conference on BioInformatics and BioEngineering*, pages 1–4. IEEE, 2013.

Taiyu Zhu, Kezhi Li, Pau Herrero, Jianwei Chen, and Pantelis Georgiou. A deep learning algorithm for personalized blood glucose prediction. In *KHD@ IJCAI*, pages 64–78, 2018.

Appendix A.

All the data wrangling and machine learning was done in Python. Figure 5 shows how the data was split into fixed-sized sequences of past and future BG levels, using overlapping moving windows. The three parameters for the moving window were history window (number of past samples), stride (how many samples to skip while sliding) and prediction horizon (number of future values to be estimated). For single-step forecasting, the last value in the prediction horizon (red window) was estimated, whereas in multi-output forecasting all 6 values were estimated by the deep learning models simultaneously but final RMSE was calculated by comparing only the last predicted value with the actual value to make a fair comparison between the output settings.

Table 4 summarizes the various pre-processing steps performed along with the parameters used for each technique. We aimed to do minimal pre-processing to evaluate the robustness of the models being compared.

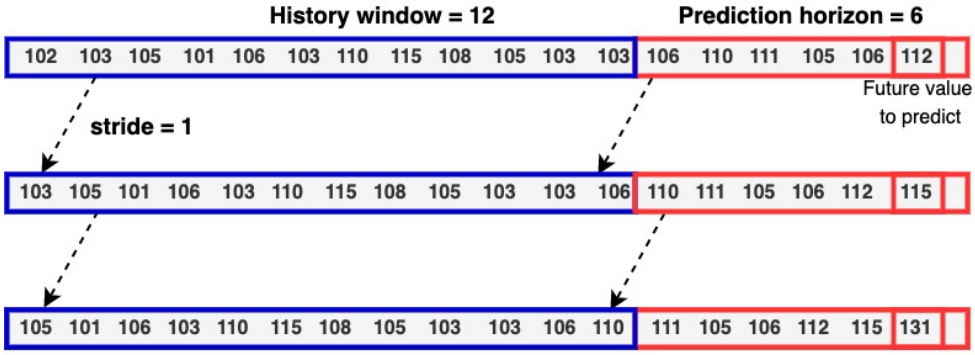


Figure 5: Overlapping windows to partition the data for time-series forecasting. The values represent raw CGM levels recorded over time.

Table 5 gives the parameter configurations used for each model. Baseline models were implemented using scikit-learn library in Python with default parameter values. For ARMA, we used the auto.arima function in the pyramid library. It automatically finds the most optimal parameters including the lag order and moving window size. However, the method was extremely time-intensive due to its automatic grid search using the Akaike information criterion (AIC). The deep learning models were implemented using Tensorflow and Keras and the state vector length and best activation function was found by doing manual grid search.

Table 4: Data pre-processing steps and techniques used.

Pre-processing step	**Technique**	**Parameters/constraints**
Windowing	Overlapping, sliding windows	history window = 12, stride = 1, $P_H = 6$
Thresholding CGM values	remove samples with CGM $\leq$ k	k = 15 mg/dL
Imputing CGM values	Interpolation, (training) Extrapolation (test)	linear and for gaps $\leq$ 30 min
Imputing basal rates	Forward filling	superseded by TBR
Imputing bolus and meals	0 when not recorded	–
Normalization	Min-max	x_{min} = minimum BG x_{max} = maximum BG
Filtering (training data only)	Median filter	window size = 5

P_H: Prediction horizon, BG: Blood glucose, TBR: Temporal basal rate

Table 5: Parameter configuration for different models.

Model	**Parameter selection**	**Best parameters**
	Baseline Models	
Linear Regression	Default	—
Random Forest	Default	no. of trees = 100
Support vector regression	Default	kernel = rbf, $\varepsilon = 0.1$ γ = 'scale', C = 1,
	Recursive model	
ARMA	Auto grid search	p = 2, q = 1
	Deep Learning Models	
RNN	Manual grid search	s = 32, activ = reLU
LSTM	Manual grid search	s = 32, activ = reLU

rbf: radial basis function s = state vector length