frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 30 August 2021
doi: 10.3389/fams.2021.707884

A Function Approximation Approach to the Prediction of Blood Glucose Levels

H. N. Mhaskar[1], S. V. Pereverzyev[2] and M. D. van der Walt[3]*

[1]Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA, , United States, [2]The Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Linz, Austria, [3]Department of Mathematics and Computer Science, Westmont College, Santa Barbara, CA, United States

The problem of real time prediction of blood glucose (BG) levels based on the readings from a continuous glucose monitoring (CGM) device is a problem of great importance in diabetes care, and therefore, has attracted a lot of research in recent years, especially based on machine learning. An accurate prediction with a 30, 60, or 90 min prediction horizon has the potential of saving millions of dollars in emergency care costs. In this paper, we treat the problem as one of function approximation, where the value of the BG level at time $t + h$ (where h the prediction horizon) is considered to be an unknown function of d readings prior to the time t. This unknown function may be supported in particular on some unknown submanifold of the d-dimensional Euclidean space. While manifold learning is classically done in a semi-supervised setting, where the entire data has to be known in advance, we use recent ideas to achieve an accurate function approximation in a supervised setting; i.e., construct a model for the target function. We use the state-of-the-art clinically relevant PRED-EGA grid to evaluate our results, and demonstrate that for a real life dataset, our method performs better than a standard deep network, especially in hypoglycemic and hyperglycemic regimes. One noteworthy aspect of this work is that the training data and test data may come from different distributions.

Keywords: hermite polynomial, continuous glucose monitoring, blood glucose prediction, supervised learning, prediction error-grid analysis

OPEN ACCESS

Edited by:
Hau-Tieng Wu,
Duke University, United States

Reviewed by:
Valeriya Naumova,
Simula Research Laboratory, Norway
Xin Guo,
Hong Kong Polytechnic University, China

****Correspondence:***
H. N, Mhaskar
hrushikesh.mhaskar@cgu.edu

Specialty section:
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *10 May 2021*
Accepted: *09 July 2021*
Published: *30 August 2021*

Citation:
Mhaskar HN, Pereverzyev SV and van der Walt MD (2021) A Function Approximation Approach to the Prediction of Blood Glucose Levels. Front. Appl. Math. Stat. 7:707884. doi: 10.3389/fams.2021.707884

1 INTRODUCTION

Diabetes is a major disease affecting humanity. According to the 2020 National Diabetes Statistics report [3], more than 34,000,000 people had diabetes in the United States alone in 2018, contributing to nearly 270,000 deaths and costing nearly 327 billion dollars in 2017. There is so far no cure for diabetes, and one of the important ingredients in keeping it in control is to monitor blood glucose levels regularly. Continuous glucose monitoring (CGM) devices are used increasingly for this purpose. These devices typically record blood sugar readings at 5 min intervals, resulting in a time series. This paper aims at predicting the level and rate of change of the sugar levels at a medium range prediction horizon; i.e., looking ahead for 30–60 min. A clinically reliable prediction of this nature is extremely useful in conjunction with other communication devices in avoiding unnecessary costs and dangers. For example, hypoglycemia may lead to loss of consciousness, confusion or seizures [9]. However, since the onset of hypoglycemic periods are often silent (without indicating symptoms), it can be very hard for a human to predict in advance. If these predictions are getting communicated to a hospital by some wearable communication device linked with the CGM device, then a nurse could

alert the patient if the sugar levels are expected to reach dangerous levels, so that the patient could drive to the hospital if necessary, avoiding a dangerous situation, not to mention an expensive call for an ambulance.

Naturally, there are many papers dealing with prediction of blood glucose (BG) based not just on CGM recordings but also other factors such as meals, exercise, etc. In particular, machine learning has been used extensively for this purpose. We refer to [9] for a recent review of BG prediction methods based on machine learning, with specific focus on predicting and detecting hypoglycemia.

The fundamental problem of machine learning is the following. We have (training) data of the form $\{(\mathbf{x}_j, f(\mathbf{x}_j) + \epsilon_j)\}$, where, for some integer $d \geq 1$, $\mathbf{x}_j \in \mathbb{R}^d$ are realizations of a random variable, f is an unknown real valued function, and theϵ_j's are realizations of a mean zero random variable. The distributions of both the random variables are not known. The problem is to approximate f with a suitable model, especially for the points which are not in the training data. For example, in the prediction of a time series $x_0, x_1, \ldots$ with a prediction horizon of h based on d past readings can be (and is in this paper) viewed as a problem of approximating an unknown function f such that $x_{j+h} = f(x_{j-d+1}, \ldots, x_j)$ for each j. Thus, $\mathbf{x}_j = (x_{j-d+1}, \ldots, x_j)$ and $y_j = x_{j+h}$.

Such problems have been studied in approximation theory for more than a century. Unfortunately, classical approximation theory techniques do not apply directly in this context, mainly because the data does not necessarily become dense on any known domain such as a cube, sphere, etc., and we cannot control the locations of the points $\mathbf{x}_j$. This is already clear from the example given above–it is impossible to require that the BG readings be at a pre-chosen set of points.

Manifold learning tries to ameliorate this problem by assuming that the data lies on some unknown manifold, and developing techniques to learn various quantities related to the manifold, in particular, the eigen-decomposition of the Laplace-Beltrami operator. Since these objects must be learnt from the data, these techniques are applicable in the semi-supervised setting, in which we have all the data points $\mathbf{x}_j$ available in advance. In [8], we have used deep manifold learning to predict BG levels using CGM data. Being in the semi-supervised setting, this approach cannot be used directly for prediction based on data that was not included in the original data set we worked with.

Recently, we have discovered [7] a more direct method to approximate functions on unknown manifolds without trying to learn anything about the manifold itself, and giving an approximation with theoretically guaranteed errors on the entire manifold; not just the points in the original data set. The purpose of this paper is to use this new tool for BG prediction based on CGM readings which can be used on patients not in the same data sets, and for readings for the patients in the data set which are not included among the training data.

In this context, the numerical accuracy of the prediction alone is not clinically relevant. The Clarke Error-Grid Analysis (C-EGA) [2] is devised to predict whether the prediction is reasonably accurate, or erroneous without serious consequences (clinically uncritical), or results in unnecessary treatment (overcorrection), or dangerous (either because it fails to identify a rise or drop in BG or because it confuses a rise in BG for a drop or vice versa). The Prediction Error-Grid Analysis (PRED-EGA), on the other hand, categorizes a prediction as accurate, erroneous with benign (not serious) consequences or erroneous with potentially dangerous consequences in the hypoglycemic (low BG), euglycemic (normal BG), and hyperglycemic (high BG) ranges separately. This stratification is of great importance because consequences caused by a prediction error in the hypoglycemic range are very different from ones in the euglycemic or the hyperglycemic range. In addition, the categorization is based not only on reference BG estimates paired with the BG estimates predicted for the same moments (as C-EGA does), but also on reference BG directions and rates of change paired with the corresponding estimates predicted for the same moments. It is argued in [14] that this is a much better clinically meaningful assessment tool than C-EGA, and is therefore the assessment tool we choose to use in this paper.

Since different papers on the subject use different criterion for assessment it is impossible for us to compare our results with those in most of these papers. Besides, a major objective of this paper is to illustrate the use and utility of the theory in [7] from the point of view of machine learning. So, we will demonstrate the superiority of our results in the hypoglycemic and hyperglycemic regimes over those obtained by a blind training of a deep network using the MATLAB Deep Learning Toolbox. We will also compare our results with those obtained by the techniques used in [6, 8, 10]. However, we note that these methods are not directly comparable to ours. In [10], the authors use the data for only one patient as training data, which leads to a meta-learning model that can be used serendipitously on other patients, although a little bit of further training is still required for each patient to apply this model. In contrast, we require a more extensive "training data," but there is no training in the classical sense, and the application to new patients consists of a simple matrix vector multiplication. The method in [6] is a linear extrapolation method, where the coefficients are learnt separately on each patient. It does not result in a model that can be trained once for all and applied to other patients in a simple manner. As pointed out earlier, the method in [8] requires that the entire data set be available even before the method can start to work. So, unlike our method, it cannot be trained on a part of the data set and applied to a completely different data set, not available at the beginning.

We will explain the problem and the evaluation criterion in **Section 3**. Prior work on this problem is reviewed briefly in **Section 2**. The methodology and algorithm used in this paper are described in **Section 4**. The results are discussed in **Section 5**. The mathematical background behind the methods described in **Section 4** is summarized in **Appendix A**.

2 PRIOR WORK

Given the importance of the problem, many researchers have worked on it in several directions. Below we highlight the main trends relevant to our work.

The majority of machine learning models (30% of those studied in the survey paper [9]) are based on artificial neural networks such as convolutional neural networks, recurrent neural networks and deep learning techniques. For example, in [17], the authors develop a deep learning model based on a dilated recurrent neural network to achieve 30-min blood glucose prediction. The authors assess the accuracy of their method by calculating the root mean squared error (RMSE) and mean absolute relative difference (MARD). While they achieve good results, a limitation of the method is the reliance on various input parameters such as meal intake and insulin dose which are often recorded manually and might therefore be inaccurate. In [11], on the other hand, a feed-forward neural network is designed with eleven neurons in the input layer (corresponding to variables such as CGM data, the rate of change of glucose levels, meal intake and insulin dosage), and nine neurons with hyperbolic tangent transfer function in the hidden layer. The network was trained with the use of data from 17 patients and tested on data from 10 other patients for a 75-min prediction, and evaluated using C-EGA. Although good results are achieved in the C-EGA grid in this paper, a limitation of the method is again the large amount of additional input information necessary to design the model, as described above.

A second BG prediction class employs decision trees. The authors of [13], for example, use random forests to perform a 30-min prediction to specifically detect postprandial hypoglycemia, that is, hypoglycemia occurring after a meal. They use statistical analyses like sensitivity and specificity to evaluate their method. In [16], prediction is again achieved by random forests and incorporating a wide variety of additional input features such as physiological and physical activity parameters. Mean Absolute Percentage Error (MAPE) is used as assessment methodology. The authors of [5] use a classification and regression tree (CART) to perform a 15-min prediction using BG data only, again using sensitivity and specificity to test the performance of their method.

A third class of methods employ kernel-based regularization techniques to achieve prediction (for example [10], and references therein), where Tikhonov regularization is used to find the best least square fit to the data, assuming the minimizer belongs to a reproducing kernel Hilbert space (RKHS). Of course, these methods are quite sensitive to the choice of kernel and regularization parameters. Therefore, the authors in [10] develop methods to choose both the kernel and regularization parameter adaptively, or through meta-learning ("learning to learn") approaches. C-EGA and PRED-EGA are used as performance metrics.

Time series techniques are also employed in the literature. In [12], a tenth-order auto-regression (AR) model is developed, where the AR coefficients are determined through a regularized least square method. The model is trained patient-by-patient, typically using the first 30% of the patient's BG measurements, for a 30-min or 60-min prediction. The method is tested on a time series containing glucose values measured every minute, and evaluation is again done through the C-EGA grid. The authors in [15] develop a first-order AR model, patient-by-patient, with time-varying AR coefficients determined through weighted least squares. Their method is tested on a time series containing glucose values measured every 3 min, and quantified using statistical metrics such as RMSE. As noted in [10], these methods seem to be sensitive to gaps in the input data.

A summary of the methods described above is given in **Table 1**.

It should be noted that the majority of BG prediction models (63.64% of those analyzed in [9]), including many of the references described above, are dependent on additional input features (such as meal intake, insulin dosage and physical activity) in addition to historical BG data. Since such data could be inaccurate and hard to obtain, we have intentionally made the design decision in our work to base our method on historical BG data only.

3 THE SET-UP

We use two different clinical data sets provided by the DirectNet Central Laboratory [1], which lists BG levels of different patients taken at 5-min intervals with the CGM device; i.e., for each patient p in the patient set P, we are given a time series $\{s_p(t_j)\}$, where $s_p(t_j)$ denotes the BG level at time t_j. The relevant details of the two data sets are described in **Table 2**.

Our goal is to predict for each j, the level $s_p(t_{j+h})$, given readings $s_p(t_j), \ldots, s_p(t_{j-d+1})$ for appropriate values of h and d. We took $d = 7$ (a sampling horizon $t_j - t_{j-d+1}$ of 30 min has been suggested as the optimal one for BG prediction in [4, 6]). We tested prediction horizons of 30 min ($h = 6$) (the most common prediction horizon according to the analysis in [9]), 60 min ($h = 12$) and 90 min ($h = 18$).

4 METHODOLOGY IN THE CURRENT PAPER

4.1 Data Reformulation

For each patient p in the set P of patients in the data sets, the data is in the form of a time series $\{s_p(t_j)\}_{j=1}^{N_p}$ of BG levels at time t_j, where $t_j - t_{j-1} = 5$ minutes. We re-organize the data in the form

$$\mathcal{P}^* = \left\{\left(\left(s_p\left(t_{j-d+1}\right), \cdots, s_p\left(t_j\right)\right), s_p\left(t_{j+h}\right)\right): j = d, \cdots, N_p - h,\ p \in P\right\}.$$

For any patient $p \in P$, we will abbreviate

$$\mathbf{x}_{p,j} = \left(s_p\left(t_{j-d+1}\right), \cdots, s_p\left(t_j\right)\right), \qquad y_{p,j} = s_p\left(t_{j+h}\right),$$

and write

$$\mathcal{P} := \left\{\mathbf{x}_{p,j} = \left(s_p\left(t_{j-d+1}\right), \cdots, s_p\left(t_j\right)\right): j = d, \cdots, N_p - h,\ p \in P\right\}.$$

The problem of BG prediction is then seen as the classical problem of machine learning; i.e., to find a functional relationship f such that $f(\mathbf{x}_{p,j}) \approx y_{p,j}$, for all p and $j = d, \cdots, N_p - h$.

4.2 Training Data Selection

To obtain training data, we form the training set C by randomly selecting (according to a uniform probability distribution) c% of

TABLE 1 | A summary of related work on BG prediction.

Ref	Method	PH (min)	Data	Assessment
[17]	DL	30	BG, meal intake, insulin dose	RMSE, MARD
[11]	NN	75	BG, BG rate of change, meal intake, insulin dose	C-EGA
[13]	RF	30	BG	Sensitivity/specificity
[16]	RF	60	BG, physical activity, physiological measurements, meal intake	MAPE
[5]	CART	15	BG	Sensitivity/specificity
[10]	KBR	30, 60, 75	BG	C-EGA
		10, 20	BG	PRED-EGA
[12]	AR	30, 60	BG	C-EGA
[15]	AR	30	BG	RMSE

PH, prediction horizon; DL, deep learning; NN, neural network; RF, random forest; CART, classification and regresssion tree; KBR, kernel-based regularization; AR, autoregression. RMSE, root mean squared error; MARD, mean absolute relative difference; C-EGA, Clarke error-grid analysis; MAPE, mean absolute percentage error; PRED-EGA, prediction error-grid analysis.

TABLE 2 | Summary statistics for datasets D and J.

	Dataset D	Dataset J
No of patients	25	25
Total no of observations	5,542	5,837
Avg no of observations per patient	221.68	233.48
Min BG	40	48.07
Max BG	356	390.93
Mean BG	128.15	163.92
SD in BG	60.61	67.59
Avg % hypoglycemic readings per patient	13.50	4.33
Avg % euglycemic readings per patient	69.83	57.98
Avg % hyperglycemic readings per patient	16.67	37.69

the patients in P. The training data are now defined to be all the data of each patient in C, that is,

$$\mathcal{C}^\star := \Big\{\big(\mathbf{x}_{p,j}, y_{p,j}\big) = \Big(\big(s_p\big(t_{j-d+1}\big), \cdots, s_p\big(t_j\big)\big), s_p\big(t_{j+h}\big)\Big): \\ j = d, \cdots, N_p - h,\ p \in C\Big\}.$$

Since we define the training and test data in terms of patients, choosing the entire data for a patient in the training (respectively, test) data, we may now simplify the notation by writing $(\mathbf{x}_j, y_j)$ rather than $(\mathbf{x}_{p,j}, y_{p,j})$, with the understanding that if $\mathbf{x}_j$ represents a data point for a patient p, y_j is the corresponding value for the same patient p. For future reference, we also define

$$\mathcal{C} := \Big\{\mathbf{x}_j = \Big(s_p\big(t_{j-d+1}\big), \cdots, s_p\big(t_j\big)\Big): p \in C\Big\}$$

and the testing patient set $Q = P\backslash C$, with

$$\mathcal{Q} := \mathcal{P}\backslash\mathcal{C} = \Big\{\mathbf{x}_j = \Big(s_p\big(t_{j-d+1}\big), \cdots, s_p\big(t_j\big)\Big): p \in Q\Big\}.$$

4.3 BG Range Classification

To get accurate results in each BG range, we want to be able to adapt the training data and parameters used for predictions in each range–as noted previously, consequences of prediction errors in the separate BG ranges are very different. To this end, we divide the measurements in $\mathcal{C}$ into three clusters $\mathcal{C}_o, \mathcal{C}_e$ and $\mathcal{C}_r$, based on y_j, the BG value at time t_{j+h}, for each $\mathbf{x}_j$:

$$\mathcal{C}_o = \Big\{\mathbf{x}_j \in \mathcal{C}\colon\ 0 \le y_j \le 70\Big\}\ (\text{hyp}\mathbf{o}\text{glycemia});$$
$$\mathcal{C}_e = \Big\{\mathbf{x}_j \in \mathcal{C}\colon\ 70 < y_j \le 180\Big\}\ (\mathbf{e}\text{uglycemia});$$
$$\mathcal{C}_r = \Big\{\mathbf{x}_j \in \mathcal{C}\colon\ 180 < y_j \le 450\Big\}\ (\text{hyp}\mathbf{r}\text{glycemia}),$$

with

$$\mathcal{C}_\ell^\star = \Big\{\big(\mathbf{x}_j, y_j\big)\colon \mathbf{x}_j \in \mathcal{C}_\ell\Big\}, \quad \ell \in \{o, e, r\}.$$

In addition, we also classify each $\mathbf{x}_j \in \mathcal{Q}$ as follows:

$$\mathcal{Q}_o = \Big\{\mathbf{x}_j \in \mathcal{Q}\colon\ 0 \le x_{j,d} \le 70\Big\}\ (\text{hyp}\mathbf{o}\text{glycemia});$$
$$\mathcal{Q}_e = \Big\{\mathbf{x}_j \in \mathcal{Q}\colon\ 70 < x_{j,d} \le 180\Big\}\ (\mathbf{e}\text{uglycemia})\,;$$
$$\mathcal{Q}_r = \Big\{\mathbf{x}_j \in \mathcal{Q}\colon\ 180 < x_{j,d} \le 450\Big\}\ (\text{hyp}\mathbf{r}\text{glycemia}).$$

Ideally, the classification of each $\mathbf{x}_j \in \mathcal{Q}$ should also be done based on the value of y_j (as was done for the classification of the training data $\mathcal{C}$). However, since the values y_j are only available for the training data and not for the test data, we use the next best available measurement for each $\mathbf{x}_j$, namely $x_{j,d}$, the BG value at time t_j.

4.4 Prediction

Before making the BG predictions, it is necessary to scale the input readings so that the components of each $\mathbf{x}_j \in \mathcal{P}$ are in $[-1/2, 1/2]$. This is done through the transformation

$$\mathbf{x}_j \mapsto \frac{2\mathbf{x}_j - (M+m)}{2(M-m)},$$

where $M\colon = \max\{x_{j,k}\colon \mathbf{x}_j \in \mathcal{C}\}$ and $m\colon = \min\{x_{j,k}\colon \mathbf{x}_j \in \mathcal{C}\}$.

With $\widehat{F}_{n,\alpha} = \widehat{F}_{n,\alpha}(Y, \mathbf{x}_j)$ defined as in **(Appendix A)** used with training data Y and evaluated at a point $\mathbf{x}_j$, we are finally ready to compute the BG prediction for each $\mathbf{x}_j \in \mathcal{Q}$ by

$$f\big(\mathbf{x}_j\big) := \begin{cases} \widehat{F}_{n_o,\alpha_o}\big(\mathcal{C}_o^\star, \mathbf{x}_j\big), & \mathbf{x}_j \in \mathcal{Q}_o, \\ \widehat{F}_{n_e,\alpha_e}\big(\mathcal{C}_e^\star, \mathbf{x}_j\big), & \mathbf{x}_j \in \mathcal{Q}_e, \\ \widehat{F}_{n_r,\alpha_r}\big(\mathcal{C}_r^\star, \mathbf{x}_j\big), & \mathbf{x}_j \in \mathcal{Q}_r. \end{cases}$$

The construction of the estimator $\widehat{F}_{n,\alpha}$ involves several technical details regarding the classical Hermite polynomials.

TABLE 3 | Average PRED-EGA scores (in percent) for different prediction horizons on **dataset D**.

	Hypoglycemia: BG ≤ 70 (mg/dL)			Euglycemia: BG 70 – 180 (mg/dL)			Hyperglycemia: BG > 180 (mg/dL)		
	Acc.	**Benign**	**Error**	**Acc.**	**Benign**	**Error**	**Acc.**	**Benign**	**Error**
30 min prediction horizon									
Our method	85.05	13.34	1.61	84.96	11.93	3.11	64.18	22.89	12.93
Matlab DL Toolbox	0.24	0.26	99.50	83.42	13.94	2.65	47.00	16.73	36.27
Diffusion geometry (2017)	88.72	4.49	6.79	80.32	17.36	2.32	64.88	21.90	13.22
Legendre polynomials (2013)	49.49	25.85	24.66	53.88	37.28	8.84	41.40	34.90	23.70
FARL (2012)	72.42	8.80	18.78	74.54	18.79	6.67	55.45	26.42	18.13
60 min prediction horizon									
Our method	80.44	12.33	7.23	82.78	11.59	5.63	63.83	20.92	15.25
Matlab DL Toolbox	0.39	0.62	98.99	82.30	14.70	3.00	47.48	18.70	33.82
Diffusion geometry (2017)	54.30	4.72	40.98	79.99	17.29	2.72	59.78	21.13	19.09
Legendre polynomials (2013)	25.66	34.54	39.80	36.20	49.44	14.36	33.27	32.21	34.52
FARL (2012)	47.11	6.80	46.10	71.62	21.29	7.09	48.57	23.91	27.51
90 min prediction horizon									
Our method	80.60	10.05	9.35	80.19	9.96	9.85	56.07	15.44	28.49
Matlab DL Toolbox	0.32	0.38	99.30	83.01	14.26	2.73	53.05	13.77	33.18
Diffusion geometry (2017)	33.07	4.42	62.51	80.12	15.87	4.01	56.73	20.84	22.43
Legendre polynomials (2013)	20.67	39.28	40.05	27.98	51.64	20.38	20.00	30.94	49.06
FARL (2012)	35.98	2.53	61.49	74.41	16.97	8.62	46.17	26.69	27.14

TABLE 4 | Average PRED-EGA scores (in percent) for different prediction horizons on **dataset J**.

	Hypoglycemia: BG ≤ 70 (mg/dl)			Euglycemia: BG 70 – 180 (mg/dl)			Hyperglycemia: BG > 180 (mg/dl)		
	Acc.	**Benign**	**Error**	**Acc.**	**Benign**	**Error**	**Acc.**	**Benign**	**Error**
30 min prediction horizon:									
Our method	82.80	6.60	10.60	84.30	10.90	4.80	75.33	11.48	13.19
Matlab DL Toolbox	0.06	0.06	99.88	81.40	14.50	4.10	59.87	8.85	31.28
Diffusion geometry (2017)	72.09	1.52	26.39	83.12	14.74	2.14	73.91	12.03	14.06
Legendre polynomials (2013)	70.68	11.75	17.57	74.48	19.67	5.85	62.82	26.08	11.10
FARL (2012)	69.61	7.98	22.41	83.87	12.31	3.82	73.95	16.43	9.62
60 min prediction horizon:									
Our method	84.27	5.66	10.07	75.43	11.11	13.46	63.13	8.49	28.38
Matlab DL Toolbox	0.06	0.13	99.81	81.12	14.64	4.24	59.83	8.21	31.96
Diffusion geometry (2017)	35.89	0.87	63.24	79.50	14.96	5.54	71.37	12.53	16.10
Legendre polynomials (2013)	50.16	22.65	27.19	52.44	34.36	13.20	42.89	36.08	21.03
FARL (2012)	42.28	1.61	56.11	76.74	14.48	8.78	67.56	16.75	15.69
90 min prediction horizon:									
Our method	71.07	5.94	22.99	73.64	9.00	17.36	60.28	6.30	33.42
Matlab DL Toolbox	0.10	0.50	99.85	81.09	14.75	4.16	59.47	9.49	31.04
Diffusion geometry (2017)	6.84	1.03	92.13	77.85	13.93	8.22	68.29	13.04	18.67
Legendre polynomials (2013)	52.58	15.71	31.71	38.46	40.44	21.10	27.92	37.13	34.95
FARL (2012)	31.58	1.23	67.18	74.96	13.10	11.95	61.96	14.06	23.98

For ease of reading, the details of this construction are deferred to the **Appendix A**.

4.5 Evaluation

To evaluate the performance of the final output $f(\mathbf{x}_j)$, $\mathbf{x}_j \in \mathcal{Q}$, we use the PRED-EGA mentioned in **Section 3**. Specifically, a PRED-EGA grid is constructed by using comparisons of $f(\mathbf{x}_j)$ with the reference value y_j. This involves comparing

$$f(\mathbf{x}_j) \quad \text{with} \quad y_j$$

as well as the rates of change

$$\frac{f(\mathbf{x}_{j+1}) - f(\mathbf{x}_{j-1})}{2(t_{j+1} - t_{j-1})} \quad \text{with} \quad \frac{y_{j+1} - y_{j-1}}{2(t_{j+1} - t_{j-1})},$$

for all $\mathbf{x}_j \in \mathcal{Q}$. Based on these comparisons, PRED-EGA classifies $f(\mathbf{x}_j)$ as being Accurate, Benign or Erroneous.

We repeat the entire process described in **Subsections 4.2–4.5** for a fixed number of trials, after which we report the average of the PRED-EGA grid placements, over all $\mathbf{x}_j \in \mathcal{Q}$ and over all trials, as the final evaluation.

A Summary of the method is given in **Algorithm 1**.

Algorithm 1 Deep Network for BG prediction

input : Time series $\{s_p(t_j)\}$, $p \in P$, formatted as $\mathcal{P} = \{\mathbf{x}_j\}$ with
$\mathbf{x}_j = (s_p(t_{j-d+1}), \cdots, s_p(t_j))$ and $y_j = s_p(t_{j+h})$
$d \in \mathbb{N}$ (specifies sampling horizon), $h \in \mathbb{N}$ (specifies prediction horizon)
$c \in (0, 100)$ (percentage of data used for training)
$n_o, n_e, n_r \in \mathbb{N}$, $\alpha_o, \alpha_e, \alpha_r \in (0, 1]$ (parameters for function approximation)
Let C contain $c\%$ of patients from P (drawn according to uniform prob. distr.)
Set $\mathcal{C} = \{\mathbf{x}_j\}$ and $\mathcal{C}^\star = \{(\mathbf{x}_j, y_j)\}$ for all patients $p \in C$
Set $\mathcal{Q} = \mathcal{P} \setminus \mathcal{C}$

output: Prediction $f(\mathbf{x}_j) \approx s_p(t_{j+h})$ for $\mathbf{x}_j \in \mathcal{Q}$.

Set $\mathcal{C}_o = \{\mathbf{x}_j \in \mathcal{C} : 0 \le y_j \le 70\}$
Set $\mathcal{C}_e = \{\mathbf{x}_j \in \mathcal{C} : 70 < y_j \le 180\}$
Set $\mathcal{C}_r = \{\mathbf{x}_j \in \mathcal{C} : 180 < y_j \le 450\}$
Set $\mathcal{C}^\star_\ell = \{(\mathbf{x}_j, y_j) : \mathbf{x}_j \in \mathcal{C}_\ell\}$, $\ell \in \{o, e, r\}$

Set $\mathcal{Q}_o = \{\mathbf{x}_j \in \mathcal{Q} : 0 \le x_{j,d} \le 70\}$
Set $\mathcal{Q}_e = \{\mathbf{x}_j \in \mathcal{Q} : 70 < x_{j,d} \le 180\}$
Set $\mathcal{Q}_r = \{\mathbf{x}_j \in \mathcal{Q} : 180 < x_{j,d} \le 450\}$

Set $M = \max\{x_{j,k} : \mathbf{x}_j \in \mathcal{C}\}$ and $m = \min\{x_{j,k} : \mathbf{x}_j \in \mathcal{C}\}$
for $\mathbf{x}_j \in \mathcal{P}$ **do**
  Rewrite $\mathbf{x}_j = \frac{2\mathbf{x}_j - (M+m)}{2(M-m)}$
end
for $\ell \in \{o, e, r\}$ **do**
  for $\mathbf{x}_j \in \mathcal{Q}_\ell$ **do**
    Compute $f(\mathbf{x}_j) = \hat{F}_{n_\ell, \alpha_\ell}(\mathcal{C}^\star_\ell, \mathbf{x}_j)$
  end
end

5 RESULTS AND DISCUSSION

As mentioned in **Section 3**, we apply our method to data provided by the DirecNet Central Laboratory. Time series for the 25 patients in data set D and the 25 patients in data set J that contain the largest number of BG measurements are considered. These specific data sets were obtained to study the performance of CGM devices in children with Type I diabetes; as such, all of the patients are less than 18 years old. Summary statistics of the two data sets are provided in **Table 2**. Our method is a general purpose algorithm, where these details do not play any significant role, except in affecting the outcome of the experiments.

We provide results obtained by implementing our method in MATLAB, as described in **Algorithm 1**. For our implementation, we employ a sampling horizon $t_j - t_{j-d+1}$ of 30 min ($d = 7$), 50% training data ($c = 50$) (which is comparable to approaches followed in for example [12]) and a total of 100 trials ($T = 100$). For all the experiments, the function approximation parameters n_o, n_e and n_r referenced in **Algorithm 1** were chosen in the range $\{3, \ldots, 7\}$ with $\alpha_o = \alpha_e = \alpha_r = 1$. We provide results for prediction horizons $t_{j+m} - t_j$ of 30 min ($h = 6$), 60 min ($h = 12$) and 90 min ($h = 18$) (again, comparable to prediction horizons in for example [12, 13, 17]). After testing our method on all 25 patients in each data set separately, the average PRED-EGA scores (in percent) are displayed in **Tables 3**, **4**.

For comparison, **Tables 3**, **4** also display predictions on the same data sets using four other methods:

i) MATLAB's built-in Deep Learning Toolbox. We used a network architecture consisting of three fully connected convolutional and output layers and appropriate input, normalization, activation, and averaging layers. As before, we used 50% training data and averaged the experiment over 100 trials, for prediction windows of 30 min, 60 and 90 min and data sets D and J.

ii) The diffusion geometry approach, based on the eigen-decomposition of the Laplace-Beltrami operator, we followed in our 2017 paper [8]. As mentioned in **Section 1**, this approach can be classified as semi-supervised learning, where we need to have all the data points $\mathbf{x}_j$ available in advance. Because of this requirement, this approach cannot be used for prediction based on data that was not included in the original data set we worked with. Indeed, this is one of the main motivations for our current method, which is not dependent on this requirement. Nevertheless, we include a comparison of the diffusion geometry approach with our current method for greater justification of the accuracy of our current method. Again, we used 50% training data and averaged the experiment over 100 trials, for prediction windows of 30, 60 and 90 min and data sets D and J.

iii) The Legendre polynomial prediction method followed in our 2013 paper [6]. In this context, the main mathematical problem can be summarized as that of estimating the derivative of a function at the end point of an interval, based on measurements of the function in the past. An important difference between our current method and the Legendre polynoimal approach is that with the latter, learning is based on the data for each patient separately (each patient forms a data set in itself, as explained in **Section 1**). Therefore, the method is philosophically different from ours. We nevertheless include a comparison for prediction windows of 30 min, 60 and 90 min and data sets D and J.

iv) The Fully Adaptive Regularized Learning (FARL) approach in the 2012 paper [10], which uses meta-learning to choose the kernel and regularization parameters adaptively (briefly described in **Section 2**). As explained in [10], a strong suit of FARL is that it is expected to be portable from patient to patient without any readjustment. Therefore, the results reported below were obtained by implementing the code and parameters used in [10] directly on data sets D and J for prediction windows of 30 min, 60 and 90 min, with no adjustments for the kernel and regularization parameters. As explained earlier, this method is also not directly compared to our theoretically well founded method.

The percentage accurate predictions and predictions with benign consequences in all three BG ranges, as obtained using all five methods, for a 30 min, 60 and 90 min prediction window, are displayed visually in **Figures 1**, **2**.

By comparing the percentages in Tables 3, 4 and the bar heights in **Figures 1**, **2**, it is clear that our method far outperforms the competitors in the hypoglycemic range, except for the 30 min prediction on data set D, where the diffusion geometry approach is slightly more accurate–but it is important to remember that the diffusion geometry approach has all the data points available from the outset.

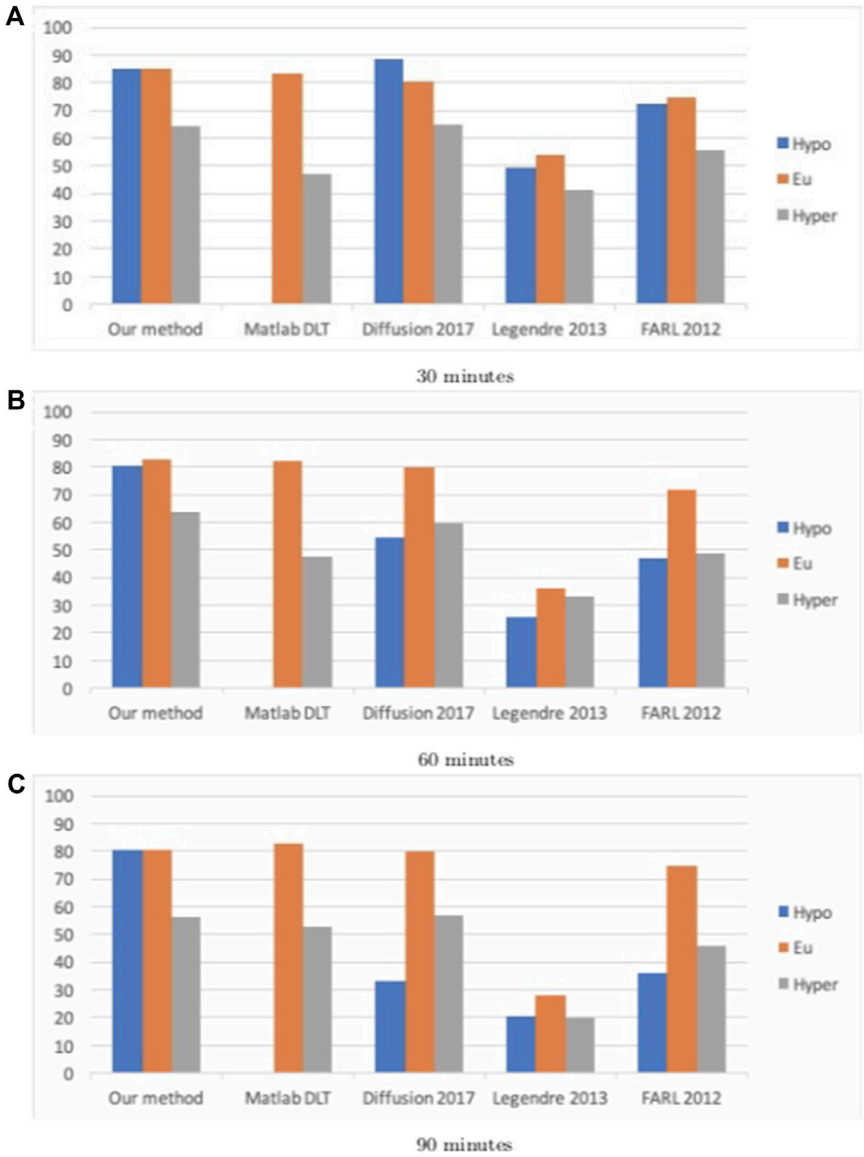

FIGURE 1 | Percentage accurate predictions and predictions with benign consequences in all three BG ranges (blue: hypoglycemia; orange: euglycemia; grey: hyperglycemia) for dataset D, for all five methods for different prediction windows (30, 60, and 90 min).

The starkest difference is between our method and Matlab's Deep Learning Toolbox–the latter achieves less than 1.5% accurate and benign consequence predictions regardless of the size of the prediction window or data set. There is also a marked difference between our method and the Legendre polynomial approach, especially for longer prediction horizons. Our method achieves comparable or higher accuracy in the hyperglycemic range, except for the 60 and 90 min predictions on data set J, where the diffusion geometry and FARL approaches achieve more accurate predictions. The methods display comparable accuracy in the euglycemic range, except for the 60 and 90 min predictions on dataset J, where Matlab's toolbox, the diffusion geometry approach and the FARL method achieve higher accuracy.

Tables 5, **6** display the results when testing those methods that are dependent on training data selection (that is, our current method, the Deep Learning Toolbox and the diffusion geometry approach in our 2017 paper) on different training set sizes (namely, 30, 50, 70 and 90%).

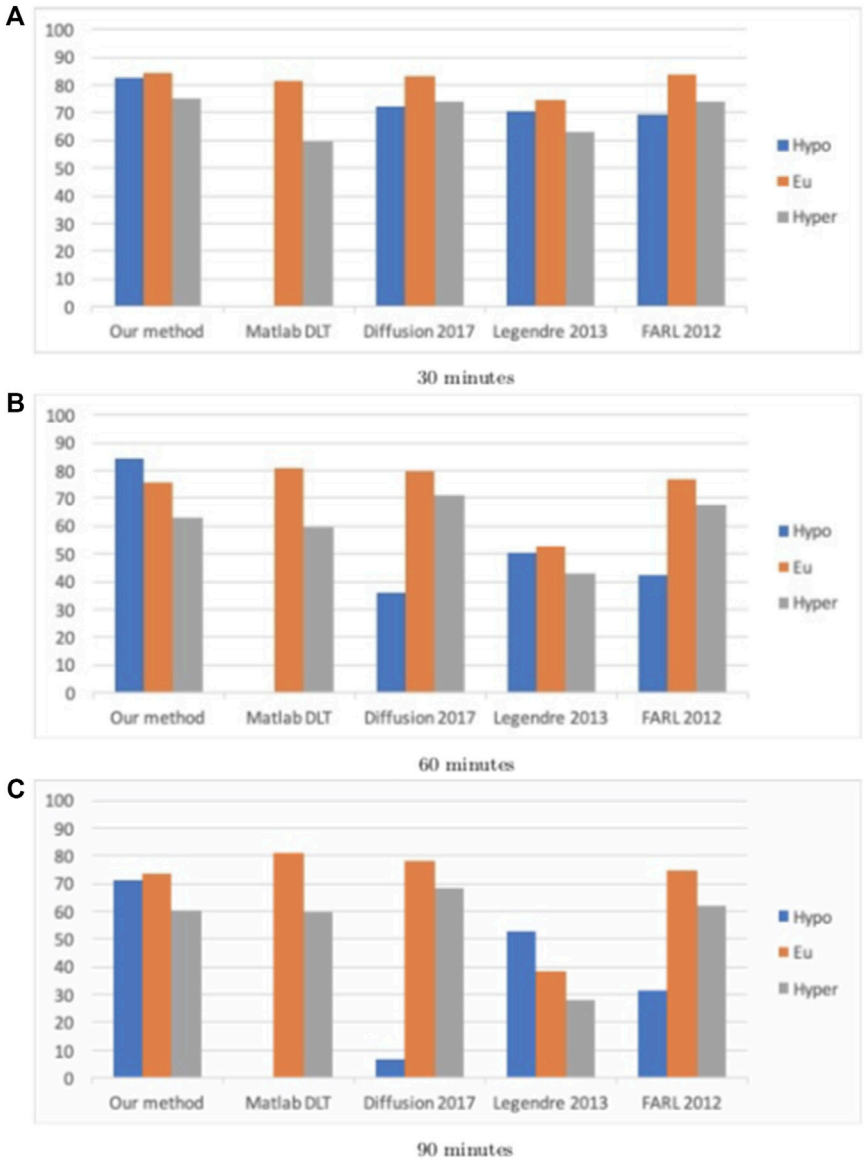

FIGURE 2 | Percentage accurate predictions and predictions with benign consequences in all three BG ranges (blue: hypoglycemia; orange: euglycemia; grey: hyperglycemia) for dataset J, for all five methods for different prediction windows (30, 60, and 90 min).

For these experiments, we used a fixed prediction window of 30 min, and training data was drawn randomly according to a uniform probability distribution in each case. While the Deep Learning Toolbox and diffusion geometry approach sometimes perform slightly better than our method in the euglycemic and hypoglycemic ranges, respectively, our method consistently outperforms both of these in the other two blood glucose ranges. The results are displayed visually as well in **Figures 3**, **4**.

As mentioned in **Section 1**, a feature of our method is that it does not require any information about the data manifold other than its dimension. So, in principle, one could "train" on one data set and apply the resulting model to another data set taken from a different manifold. We illustrate this by constructing our model based on one of the data sets D or J and testing it on the other data set. Building on this idea, we also demonstrate the accuracy of our method as compared to MATLAB's Deep Learning Toolbox when trained on data set D and applied to perform prediction on

TABLE 5 | Percentage accurate predictions for different training set sizes and a 30 min prediction horizon on **dataset D**.

	Hypoglycemia: **BG ≤ 70 (mg/dl)**	**Euglycemia:** **BG 70 – 180 (mg/dl)**	**Hyperglycemia:** **BG > 180 (mg/dl)**
30% training data:			
Our method	83.02	86.15	64.14
Matlab DL Toolbox	0.88	77.89	44.50
Diffusion geometry (2017)	82.56	80.83	63.89
50% training data:			
Our method	85.05	84.96	64.18
Matlab DL Toolbox	0.24	83.42	47.00
Diffusion geometry (2017)	88.72	80.32	64.88
70% training data:			
Our method	86.62	85.79	64.47
Matlab DL Toolbox	0.06	88.14	48.94
Diffusion geometry (2017)	90.72	80.34	64.99
90% training data:			
Our method	86.77	86.34	65.20
Matlab DL Toolbox	0.09	89.71	46.97
Diffusion geometry (2017)	93.22	80.87	65.84

TABLE 6 | Percentage accurate predictions for different training set sizes and a 30 min prediction horizon on **dataset J**.

	Hypoglycemia: **BG ≤ 70 (mg/dl)**	**Euglycemia:** **BG 70 – 180 (mg/dl)**	**Hyperglycemia:** **BG > 180 (mg/dl)**
30% training data:			
Our method	81.79	83.34	76.12
Matlab DL Toolbox	0.07	74.75	60.81
Diffusion geometry (2017)	67.02	83.99	75.33
50% training data:			
Our method	82.80	84.30	75.33
Matlab DL Toolbox	0.06	81.40	59.87
Diffusion geometry (2017)	72.09	83.12	73.91
70% training data:			
Our method	86.40	81.89	77.21
Matlab DL Toolbox	0.00	86.26	60.93
Diffusion geometry (2017)	74.45	81.28	73.47
90% training data:			
Our method	87.67	81.43	76.40
Matlab DL Toolbox	0	88.30	62.45
Diffusion geometry (2017)	73.90	81.33	70.98

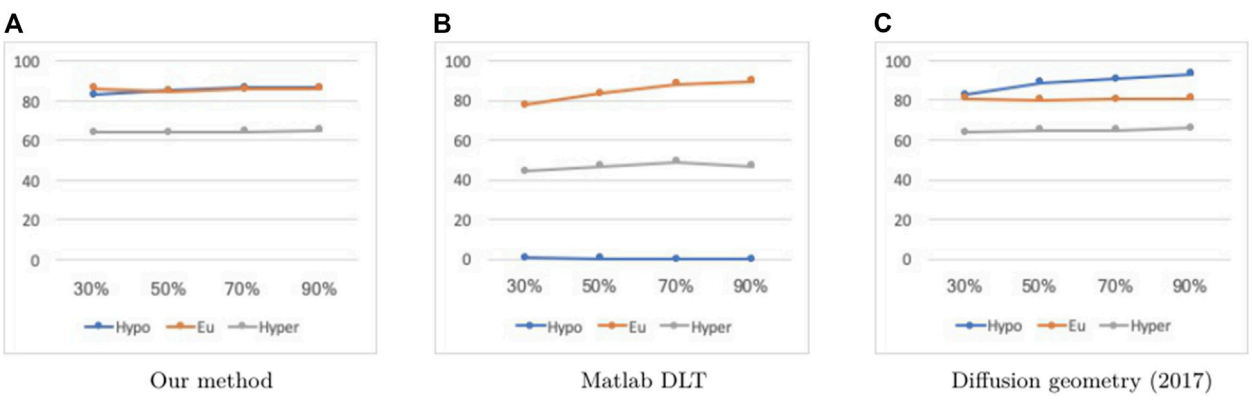

FIGURE 3 | Percentage accurate predictions and predictions with benign consequences in all three BG ranges (blue: hypoglycemia; orange: euglycemia; grey: hyperglycemia) with a 30 min prediction window on dataset D, for different training set sizes (30, 50, 70 and 90%).

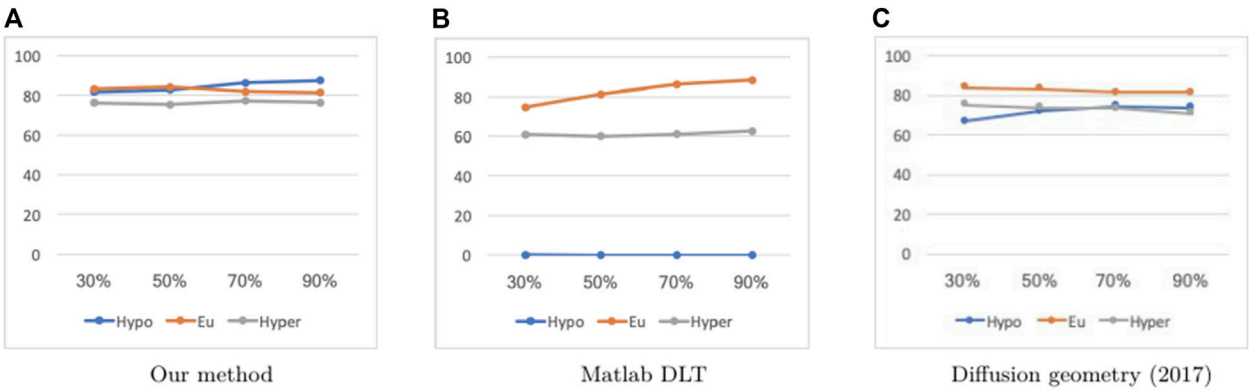

FIGURE 4 | Percentage accurate predictions and predictions with benign consequences in all three BG ranges (blue: hypoglycemia; orange: euglycemia; grey: hyperglycemia) with a 30 min prediction window on dataset J, for different training set sizes (30, 50, 70 and 90%).

TABLE 7 | Average PRED-EGA scores (in percent) for different prediction horizons on **dataset D**, with **training data selected from dataset J**.

	Hypoglycemia: BG ≤ 70 (mg/dl)			Euglycemia: BG 70 – 180 (mg/dl)			Hyperglycemia: BG > 180 (mg/dl)		
	Acc.	**Benign**	**Error**	**Acc.**	**Benign**	**Error**	**Acc.**	**Benign**	**Error**
30 min prediction horizon:									
Our method	85.85	12.23	1.92	85.51	11.48	3.01	54.95	24.21	20.84
Matlab DL Toolbox	0.01	0.00	99.99	90.91	8.09	1.01	50.43	14.99	34.58
60 min prediction horizon:									
Our method	85.05	12.30	2.65	79.86	10.59	9.55	45.73	18.52	35.75
Matlab DL Toolbox	0.00	0.01	99.99	90.57	8.39	1.03	52.37	16.08	31.55
90 min prediction horizon:									
Our method	78.01	9.41	12.58	79.65	10.16	10.19	43.18	12.36	44.46
Matlab DL Toolbox	0.00	0.01	99.99	89.83	8.74	1.43	56.35	12.58	31.07

TABLE 8 | Average PRED-EGA scores (in percent) for different prediction horizons on **dataset J**, with **training data selected from dataset D**.

	Hypoglycemia: BG ≤ 70 (mg/dl)			Euglycemia: BG 70 – 180 (mg/dl)			Hyperglycemia: BG > 180 (mg/dl)		
	Acc.	**Benign**	**Error**	**Acc.**	**Benign**	**Error**	**Acc.**	**Benign**	**Error**
30 min prediction horizon:									
Our method	82.18	17.59	0.23	79.72	14.04	6.24	74.19	16.30	9.51
Matlab DL Toolbox	0.01	0.00	99.99	89.16	9.44	1.40	60.47	4.62	34.91
60 min prediction horizon:									
Our method	88.08	10.07	1.85	73.50	14.41	12.09	63.90	11.64	24.46
Matlab DL Toolbox	0.05	0.01	94.94	89.79	9.05	1.16	60.94	4.20	34.86
90 min prediction horizon:									
Our method	74.79	10.36	14.85	67.83	13.20	18.97	57.70	10.55	31.75
Matlab DL Toolbox	0.09	0.14	99.77	89.38	9.38	1.24	58.85	4.98	36.17

data set J, and vice versa, which is the only other method that is directly comparable with the philosophy behind our method. These results are reported in **Tables 7**, **8**. It is clear that we are successful in our goal of training on one data set and predicting on a different data set. The percentage accurate predictions and predictions with benign consequences when training and testing on different datasets are comparable to the cases when we are training and testing on the same dataset; in fact, we obtain even better results in

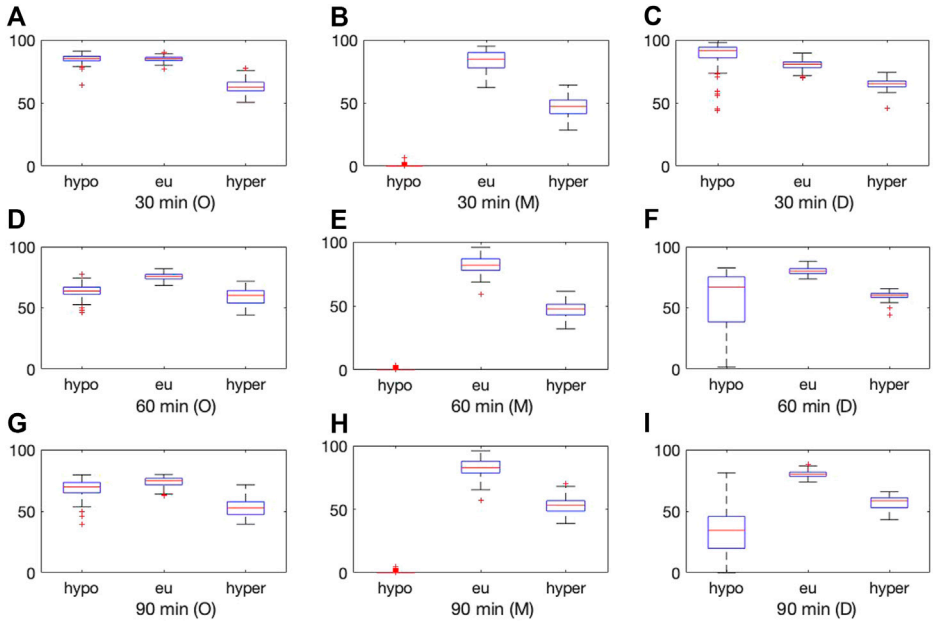

FIGURE 5 | Boxplot for the 100 experiments conducted with 50% training data for each prediction method (O = our method, M = MATLAB Deep Learning Toolbox, D = diffusion geometry approach) with a 30 min (**top**), 60 min (**middle**) and 90 min (**bottom**) prediction horizon and dataset D. Each of the graphs show the percentage accurate predictions in the hypoglycemic range (**left**), euglycemic range (**middle**) and hyperglycemic range (**right**).

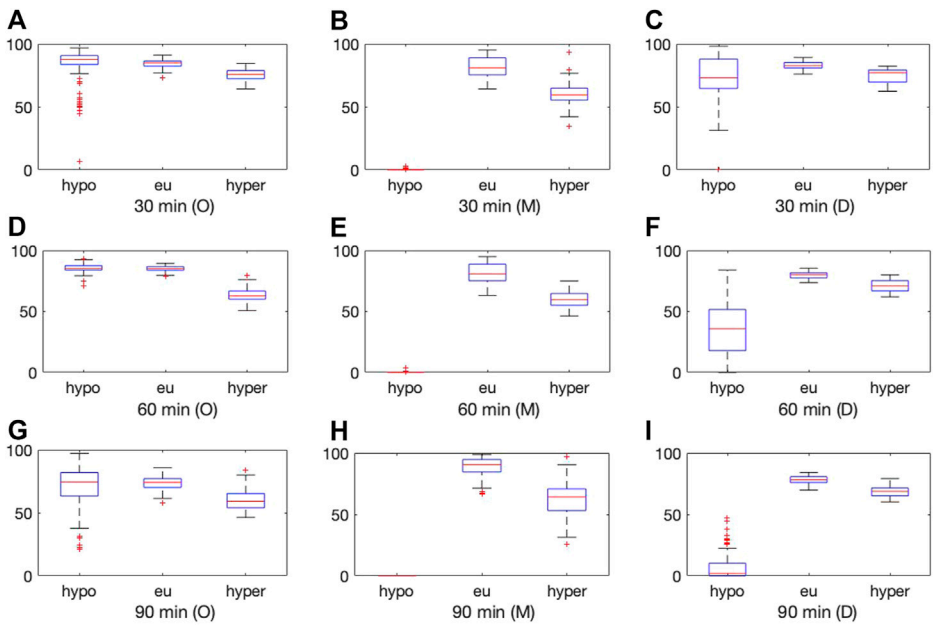

FIGURE 6 | Boxplot for the 100 experiments conducted with 50% training data for each prediction method (O = our method, M = MATLAB Deep Learning Toolbox, D = diffusion geometry approach) with a 30 min (**top**), 60 min (**middle**) and 90 min (**bottom**) prediction horizon and dataset J. Each of the graphs show the percentage accurate predictions in the hypoglycemic range (**left**), euglycemic range (**middle**) and hyperglycemic range (**right**).

the hypoglycemic BG range when training and testing across different datasets.

Figures 5, **6** display box plots for the 100 trials for the methods dependent on training data selection (that is, our current method, the Deep Learning Toolbox and the 2017 diffusion geometry approach) and prediction windows using data sets D and J, respectively. Our method performs fairly consistently over different trials.

6 CONCLUSION

In this paper, we demonstrate a direct method to approximate functions on unknown manifolds [7] in the context of BG prediction for diabetes management. The results are evaluated using the state-of-the-art PRED-EGA grid [14]. Unlike classical manifold learning, our method yields a model for the BG prediction on data not seen during training, even for test data which is drawn from a different distribution from the training data. Our method outperforms other methods in the literature in 30, 60, and 90 min predictions in the hypoglycemic and hyperglycemic BG ranges, and is especially useful for BG prediction for patients that are not included in the training set.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://direcnet.jaeb.org/Studies.aspx.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

The research of HM is supported in part by NSF grant DMS 2012355 and ARO grant W911NF2110218. The research by SP has been partly supported by BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG in the COMET Module S3AI.

ACKNOWLEDGMENTS

The authors would like to thank Professor Sampath Sivananthan at the Indian Institute of Technology, Delhi, India for providing us with the code used in [10] for the FARL approach.

REFERENCES

1. DirecNet Central Laboratory (2005). Available at: http://direcnet.jaeb.org/Studies.aspx (Accessed January 1, 2021).
2. Clarke WL, Cox D, Gonder-Frederick L. A, Carter W, and Pohl S. L. Evaluating Clinical Accuracy of Systems for Self-Monitoring of Blood Glucose. *Diabetes care* (1987) 10(5):622–8. doi:10.2337/diacare.10.5.622
3. Centers for Disease Control, Prevention. *National Diabetes Statistics Report*. Atlanta, GA: Centers for Disease Control and Prevention, US Department of Health and Human Services (2020). p. 12 15.
4. Hayes AC, Mastrototaro J. J, Moberg SB, Mueller JC, Jr, Clark HB, Tolle MCV, et al. Algorithm Sensor Augmented Bolus Estimator for Semi-closed Loop Infusion System. *US Patent* (2009) 7(547):281, 2009 . June 16. Available at: https://patents.google.com/patent/US20060173406A1/en
5. Jung M, Lee Y-B, Jin S-M, and Park S-M. Prediction of Daytime Hypoglycemic Events Using Continuous Glucose Monitoring Data and Classification Technique. arXiv preprint arXiv:1704.08769, 2017.
6. Mhaskar HN, Naumova V, and Pereverzyev SV. Filtered Legendre Expansion Method for Numerical Differentiation at the Boundary point with Application to Blood Glucose Predictions. *Appl Maths Comput* (2013) 224:835–47. doi:10.1016/j.amc.2013.09.015
7. Mhaskar H. *Deep Gaussian Networks for Function Approximation on Data Defined Manifolds* (2019). arXiv preprint arXiv:1908.00156.
8. Mhaskar HN, Sergei V, and Mariavan der Walt D. A Deep Learning Approach to Diabetic Blood Glucose Prediction. *Front Appl Maths Stat* (2017) 3(14). doi:10.3389/fams.2017.00014 Pereverzyev.
9. Mujahid O, Contreras I, and Vehi J. Machine Learning Techniques for Hypoglycemia Prediction: Trends and Challenges. *Sensors* (2021) 21(2):546. doi:10.3390/s21020546
10. Naumova V, Pereverzyev SV, and Sivananthan S. A Meta-Learning Approach to the Regularized Learning-Case Study: Blood Glucose Prediction. *Neural Networks* (2012) 33:181–93. doi:10.1016/j.neunet.2012.05.004
11. Pappada SM, Cameron BD, Rosman PM, Bourey RE, Papadimos TJ, Olorunto W, et al. Neural Network-Based Real-Time Prediction of Glucose in Patients with Insulin-dependent Diabetes. *Diabetes Tech Ther* (2011) 13(2):135–41. doi:10.1089/dia.2010.0104
12. Reifman J, Rajaraman S, Gribok A, and Ward WK. Predictive Monitoring for Improved Management of Glucose Levels. *J Diabetes Sci Technol* (2007) 1(4): 478–86. doi:10.1177/193229680700100405
13. Seo W, Lee YB, Lee S, Jin SM, and Park SM. A Machine-Learning Approach to Predict Postprandial Hypoglycemia. *BMC Med Inform Decis Mak* (2019) 19(1):210–3. doi:10.1186/s12911-019-0943-4
14. Sivananthan S, Naumova V, Man CD, Facchinetti A, Renard E, Cobelli C, et al. Assessment of Blood Glucose Predictors: the Prediction-Error Grid Analysis. *Diabetes Tech Ther* (2011) 13(8):787–96. doi:10.1089/dia.2011.0033
15. Sparacino G, Zanderigo F, Corazza S, Maran A, Facchinetti A, and Cobelli C. Glucose Concentration Can Be Predicted Ahead in Time from Continuous Glucose Monitoring Sensor Time-Series. *IEEE Trans Biomed Eng* (2007) 54(5): 931–7. doi:10.1109/tbme.2006.889774
16. Reza Vahedi M, MacBride KB, Woo W, Kim Y, Fong C, Padilla AJ, et al. "Predicting Glucose Levels in Patients with Type1 Diabetes Based on Physiological and Activity Data," In *Proceedings of the 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop* (2018). p. 1–5. doi:10.1145/3220127.3220133
17. Zhu T, Li K, Chen J, Herrero P, and Georgiou P. Dilated Recurrent Neural Networks for Glucose Forecasting in Type 1 Diabetes. *J Healthc Inform Res* (2020) 4(3):308–24. doi:10.1007/s41666-020-00068-2

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

APPENDIX

A Theoretical aspects

In this section, we describe the theoretical background behind the estimator $\widehat{F}_{n,\alpha}(\Gamma)$ used in this paper to predict the BG level. We focus only on the essential definitions, referring to [7] for details.

Let $d \geq q \geq 1$ be integers, $\mathbb{X}$ be a q dimensional, compact, connected, sub-manifold of $\mathbb{R}^d$ (without boundary), with geodesic distance ρ and volume measure μ^*, normalized so that $\mu^*(\mathbb{X}) = 1$. The operator $\widehat{F}_{n,\alpha}$ does not require any knowledge of the manifold other than q. Our construction is based on the classical Hermite polynomials which are best introduced by the recurrence relations (**A.1**). The orthonormalized Hermite polynomial h_k of degree k is defined recursively by

$$h_k(x) := \sqrt{\frac{2}{k}}xh_{k-1}(x) - \sqrt{\frac{k-1}{k}}h_{k-2}(x), \qquad k = 2, 3, \cdots,$$
$$h_0(x) := \pi^{-1/4},\ h_1(x) := \sqrt{2}\pi^{-1/4}x. \tag{A.1}$$

We write

$$\psi_k(x) := h_k(x)\exp\left(-x^2/2\right), \qquad x \in \mathbb{R},\ k \in \mathbb{Z}_+.$$

The functions $\{\psi_k\}_{k=0}^{\infty}$ are an orthonormal set with respect to the Lebesgue measure:

$$\int_{\mathbb{R}} \psi_k(x)\psi_j(x)dx = \begin{cases} 1, & \text{if } k = j, \\ 0, & \text{otherwise.} \end{cases}$$

In the sequel, we fix an infinitely differentiable function $H\colon [0,\infty)$, such that $H(t) = 1$ if $0 \leq t \leq 1/2$, and $H(t) = 0$ if $t \geq 1$. We define for $x \in \mathbb{R}$, $m \in \mathbb{Z}_+$:

$$\mathcal{P}_{m,q}(x):$$
$$= \begin{cases} \pi^{-1/4}(-1)^m\frac{\sqrt{(2m)!}}{2^m m!}\psi_{2m}(x), & \text{if } q = 1, \\ \frac{1}{\pi^{(2q-1)/4}\Gamma((q-1)/2)}\sum_{\ell=0}^{m}(-1)^{\ell}\frac{\Gamma((q-1)/2+m-\ell)}{(m-\ell)!}\frac{\sqrt{(2\ell)!}}{2^{\ell}\ell!}\psi_{2\ell}(x), & \text{if } q \geq 2, \end{cases}$$

and the kernel $\widetilde{\Phi}_{n,q}$ for $x \in \mathbb{R}$, $n \in \mathbb{Z}_+$ by

$$\widetilde{\Phi}_{n,q}(x) := \sum_{m=0}^{\lfloor n^2/2\rfloor} H\left(\frac{\sqrt{2m}}{n}\right)\mathcal{P}_{m,q}(x).$$

We assume here a noisy data of the form $(\mathbf{y}, \epsilon)$, with a joint probability distribution τ and assume further that the marginal distribution of $\mathbf{y}$ with respect to τ has the form $d\nu^* = f_0 d\mu^*$ for some $f_0 \in C(\mathbb{X})$. In place of $f(\mathbf{y})$, we consider a noisy variant $\mathcal{F}(\mathbf{y}, \epsilon)$, and denote

$$f(\mathbf{y}) := \mathbb{E}_{\tau}(\mathcal{F}(\mathbf{y}, \epsilon)|\mathbf{y}). \tag{A.2}$$

Remark A.1

In practice, the data may not lie on a manifold, but it is reasonable to assume that it lies on a tubular neighborhood of the manifold. Our notation accommodates this - if $\mathbf{z}$ is a point in a neighborhood of $\mathbb{X}$, we may view it as a perturbation of a point $\mathbf{y} \in \mathbb{X}$, so that the noisy value of the target function is $\mathcal{F}(\mathbf{y}, \epsilon)$, where ϵ encapsulate the noise in both the $\mathbf{y}$ variable and the value of the target function. ■

Our approximation process is simple: given by

$$\widehat{F}_{n,\alpha}(Y; \mathbf{x}) := \frac{n^{q(1-\alpha)}}{M}\sum_{j=1}^{M}\mathcal{F}\left(\mathbf{y}_j, \epsilon_j\right)\widetilde{\Phi}_{n,q}\left(n^{1-\alpha}\left|\mathbf{x} - \mathbf{y}_j\right|_{2,d}\right), \quad \mathbf{x} \in \mathbb{R}^d, \tag{A.3}$$

where $0 < \alpha \leq 1$.

The theoretical underpinning of our method is described by Theorem A.1, and in particular, by Corollary A.1, describing the convergence properties of the estimator. In order to state these results, we need a smoothness class $W_{\gamma}(\mathbb{X})$, representing the assumptions necessary to guarantee the rate of convergence of our estimator to the target function. It is beyond the scope of this paper to describe the details of this smoothness class; an interested reader will find them in [7]. We note that the definition of the estimator in **(A.3)** is *universal*, and does not require any assumptions. The assumptions are needed only to gurantee the right rates of convergence.

Theorem A.1

Let $\gamma > 0$, τ be a probability distribution on $\mathbb{X} \times \Omega$ for some sample space Ω such the marginal distribution of τ restricted to $\mathbb{X}$ is absolutely continuous with respect to μ^* with density $f_0 \in W_{\gamma}(\mathbb{X})$. We assume that

$$\sup_{\mathbf{x}\in\mathbb{X}, r>0}\frac{\mu^*(\mathbb{B}(\mathbf{x}, r))}{r^q} \leq c.$$

*Let $\mathcal{F}\colon \mathbb{X} \times \Omega \to \mathbb{R}$ be a bounded function, f defined by **(A.2)** be in $W_{\gamma}(\mathbb{X})$, the probability density $f_0 \in W_{\gamma}(\mathbb{X})$. Let $M \geq 1$, $Y = \{(\mathbf{y}_1, \epsilon_1), \cdots, (y_M, \epsilon_M)\}$ be a set of random samples chosen i.i.d. from τ. If*

$$0 < \alpha < \frac{4}{2+\gamma}, \qquad \alpha \leq 1,$$

then for every $n \geq 1$, $0 < \delta < 1$ and $M \geq n^{q(2-\alpha)+2\alpha\gamma}\sqrt{\log(n/\delta)}$, we have with τ-probability $\geq 1 - \delta$:

$$\left\|\widehat{F}_{n,\alpha}(Y) - ff_0\right\|_{\mathbb{X}} \leq c_1\frac{\sqrt{f_0\mathbb{X}}\mathcal{F}_{\mathbb{X}\times\Omega} + \|ff_0\|_{W_{\gamma}(\mathbb{X})}}{n^{\alpha\gamma}}.$$

Corollary A.1

With the set-up as in Theorem A.1, let $f_0 \equiv 1$ (i.e., the marginal distribution of $\mathbf{y}$ with respect to τ is μ^*). Then we have with τ-probability $\geq 1 - \delta$:

$$\left\|\widehat{F}_{n,\alpha}(Y) - f\right\|_{\mathbb{X}} \leq c_1\frac{\|\mathcal{F}\|_{\mathbb{X}\times\Omega} + \|f\|_{W_{\gamma}(\mathbb{X})}}{n^{\alpha\gamma}}.$$