Prediction of Adverse Glycemic Events from Continuous Glucose Monitoring Signal

Matteo Gadaleta, Andrea Facchinetti, Enrico Grisan, Michele Rossi

***Abstract*—The most important objective of any diabetes therapy is to maintain the blood glucose concentration within the euglycemic range, avoiding or at least mitigating critical hypo/hyperglycemic episodes. Modern Continuous Glucose Monitoring (CGM) devices bear the promise of providing the patients with an increased and timely awareness of glycemic conditions as these get dangerously near to hypo/hyperglycemia. The challenge is to detect, with reasonable advance, the patterns leading to risky situations, allowing the patient to make therapeutic decisions on the basis of future (predicted) glucose concentration levels. We underline that a technically sound performance comparison of the approaches that have been proposed in recent years is still missing, and is thus unclear which one is to be preferred. The aim of this study is to fill this gap, by carrying out a comparative analysis among the most common methods for glucose event prediction. Both regression and classification algorithms have been implemented and analyzed, including static and dynamic training approaches. The dataset consists of 89 CGM time series measured in diabetic subjects for 7 subsequent days. Performance metrics, specifically defined to assess and compare the event prediction capabilities of the methods, have been introduced and analyzed. Our numerical results show that a static training approach exhibits better performance, in particular when regression methods are considered. However, classifiers show some improvement when trained for a specific event category, such as hyperglycemia, achieving performance comparable to the regressors, with the advantage of predicting the events sooner.**

***Index Terms*—Continuous glucose monitoring (CGM), event prediction, machine learning, signal processing, type 1 diabetes.**

I. Introduction

Diabetes is a metabolic disorder characterized by chronic hyperglycemia resulting from a deficient insulin production or utilization. This is due to the destruction of beta cells in the pancreas (type 1 diabetes), which requires daily administration of insulin, or an ineffective use of insulin (type 2 diabetes) [1]. Insulin-dependent diabetes requires a daily management, which commonly consists in diet, physical exercise and exogenous insulin administration [2], [3], which are tuned on the basis of self-monitoring blood glucose (SMBG) measurements usually performed 3-4 times per day.

In recent years, blood glucose (BG) monitoring has been revolutionized by the advent of Continuous Glucose Monitoring (CGM) sensors, consisting of wearable subcutaneous needle-based and minimally-invasive devices that allow measuring the BG concentration almost continuously (1-5 min sampling period) for several consecutive days/weeks. Thanks to this, and to the availability of acoustic/visual alerts for hypo/hyperglycemia, CGM sensors have become a key tool to effectively improve diabetes management and glucose control [4]. However, avoiding hypoglycemic and hyperglycemic events, or at least mitigating their frequency and duration, still remains an open challenge [5]. In particular, the real-time prediction of future levels of BG concentration from its past history could allow the patient to take therapeutic actions on the basis of predicted glycemic levels instead of the current glycemic status, possibly mitigating and/or avoiding imminent critical events [6], [7]. Most of the modern sensors include hardware capable of performing heavy computations. The relatively low sensors' sampling frequency allows for near-realtime operation. Nevertheless, most of the wearable sensors usually delegate this task to an additional, more powerful device, to enhance their battery time and to limit the memory requirements [8].

Several algorithms for the real-time prediction of hypoglycemic and hyperglycemic events from CGM data have been proposed in the literature. Glucose prediction by using as input only the past history of the CGM signal has been explored with auto-regressive models [9], [10], [11], artificial neural networks [12], and kernel-based methods [13]. Since glucose dynamics are influenced by the quantity of ingested carbohydrates (CHO), injected insulin, physical activity, etc., glucose prediction algorithms that consider some (or even all) of these signals have been proposed, e.g., autoregressive-moving average with exogenous inputs models [14], [15], [16], random forests [17], support vector based algorithms [18], Gaussian processes [19], linear multi-step predictors [20], neural networks [21], [22], [23], [24], [25] and multi-model systems [26]. An increasing spread of data-driven approaches can be observed, and the availability of high quality measurements is of primary importance. In [27], for example, the authors make use of artificially generated data to obtain a large dataset. Nevertheless, real data is always preferable and leads to a more accurate prediction analysis. Despite these attempts, commercial CGM sensors still do not embed any prediction algorithm for the early detection of hypoglycemia and hyperglycemia [4], [5], at variance with the "artificial pancreas" systems that couple a sensor with an insulin/glucose pump connected in close loop. However, basic methods are typically used for these devices. For example, a simple linear projection algorithm is embedded in Medtronic systems, both sensor-augmented pumps [8] and hybrid closed loop systems [28], to predict hypoglycemia and suspend the basal insulin delivery trying to mitigate hypoglycemic states (*"suspend before low"*). One of the reasons for this, is that

M. Gadaleta, A. Facchinetti, E. Grisan, and M. Rossi are with the Department of Information Engineering (DEI), University of Padova, Italy.

E. Grisan is also with the Division of Imaging Sciences and Biomedical Engineering, King's College London, UK.

E-mail: {gadaleta, facchine, enrigri, rossi}@dei.unipd.it

Manuscript submitted April 3, 2018.

Corresponding author: Matteo Gadaleta (E-mail: gadaleta@dei.unipd.it).

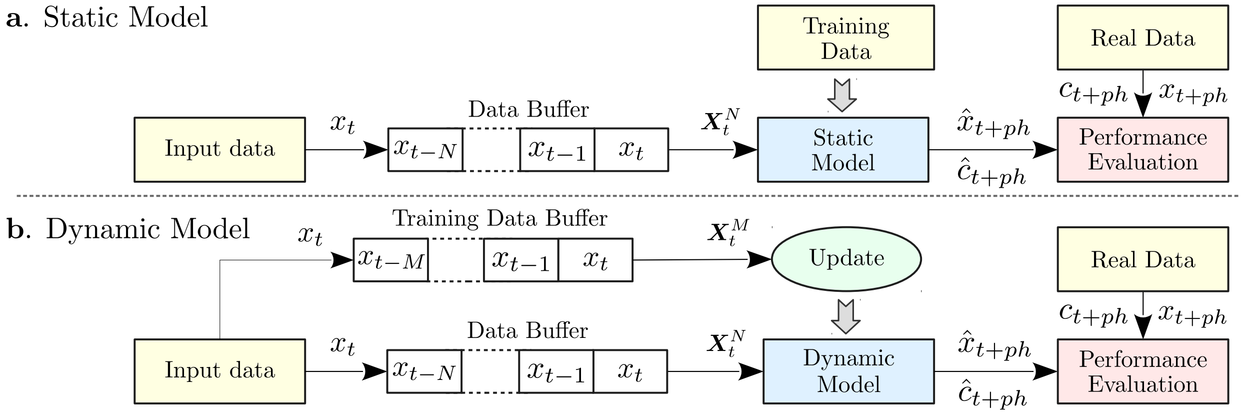

Fig. 1: System diagram. Static (a) and Dynamic (b) models.

prediction algorithms have been developed and tested using different datasets (either real or simulated), and that an in depth comparison of their performance on the *same* dataset is still missing [29].

The main contribution of this paper is to fill the aforementioned gap, offering a performance comparison among regression and classification algorithms for hypoglycemic and hyperglycemic event prediction based on CGM data. Two different approaches have been implemented and analyzed: *static* and *dynamic* methods, which are detailed in Section II. Numerical results are presented in Section III. Most of the previous works focus on the prediction performance of the algorithms. Here, we go one step further, looking at the event detection problem, introducing a specific analysis for this purpose, and showing that common prediction metrics are not always in accordance with event detection capabilities. Finally, in Section IV we provide some concluding remarks.

II. Methods

The present work is meant to be a comparative analysis among the most common machine learning algorithms for glucose events prediction. Only the Continuous Glucose Monitoring (CGM) readings are considered for this study, whereas exogenous inputs or any additional information are not taken into account as they are not available in our dataset.

In this study, two different approaches have been analyzed and tested (see Fig. 1), defined as follows:
Static - A single model is trained once, in an offline fashion, using a subset of the available data. The data may belong to subjects different from the one considered in the test phase. The model is never updated during the test and a fixed amount N of previous samples are used as input.
Dynamic - The model is updated with each new sample. M previous samples are used to update the model, and N previous samples are used to estimate future glucose levels or events. Note that $M > N$ in order to have enough training samples for updating the model. In this case, the data used for updating the model and the data used for the test belong to the same subject.

In this paper, several machine learning techniques have been analyzed. These models can be divided in two main categories: *regression* and *classification* algorithms, detailed in the following sections.

A. Regression algorithms

The aim of this type of algorithms is to predict the future value of the BG concentration, that is considered to be a real number, from measurements acquired in the past. In general, they implement a function r such that:

$$r(\boldsymbol{X}_t^N, \boldsymbol{\beta}_t) = \hat{x}_{t+ph} \qquad \hat{x}_{t+ph} \in \mathbb{R}, \tag{1}$$

where $\boldsymbol{X}_t^N = [x_{t-N+1}, \dots, x_t] \in \mathbb{R}^N$ is a vector containing the last N CGM readings, ph is the prediction horizon, and $\boldsymbol{\beta}_t$ represents a vector containing algorithm-specific parameters. This parameter vector is unknown and is to be found during the training process. The sub-index t represents the discrete time progression. In general, $\boldsymbol{\beta}_t$ is not constant. In particular, if a dynamic approach is used, the model parameters continuously change at each iteration; when considering a static model, instead, the parameters are optimized in the training phase, and then remain fixed ($\boldsymbol{\beta}_t = \boldsymbol{\beta}$). As performance metric, we define the regression error, corresponding to a prediction horizon ph, as:

$$e_t^{ph} = x_{t+ph} - \hat{x}_{t+ph} \, . \tag{2}$$

Furthermore, a Root Mean Square Error (RMSE) can be associated with an entire signal of T samples as follows:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(e_t^{ph}\right)^2} \, . \tag{3}$$

Sum of Squares of the Glucose Prediction Error (SSGPE), defined in Eq. (4), is also considered as a relative error estimation metric.

$$SSGPE = \sqrt{\frac{\sum_{t=1}^{T}\left(e_t^{ph}\right)^2}{\sum_{t=1}^{T}\left(x_{t+ph}\right)^2}} \, . \tag{4}$$

The methods considered in this study that fall into this category are: linear, quadratic and spline fitting, Linear Regression, Bayesian Regression [30], Support Vector Regression (SVR) [31] and Autoregressive Integrated Moving Average (ARIMA) models.

B. Classification algorithms

In order to perform a classification of the events corresponding to specific levels of BG concentration, the BG has been quantized into a finite number of classes. Following a well-accepted and commonly used definition [32], the following states have been considered: severe hyperglycemic (SHyper), hyperglycemic (Hyper), normal (Norm), hypoglycemic (Hypo) and severe hypoglycemic (SHypo). The function q that maps the CGM readings x_t onto the class set $\mathbb{C} = \{\text{SHyper, Hyper, Norm, Hypo, SHypo}\}$ is defined as follows:

$$c_t = q(x_t) = \begin{cases} \text{SHypo} & \text{if } x_t \leqslant 50 \\ \text{Hypo} & \text{if } 50 < x_t \leqslant 70 \\ \text{Norm} & \text{if } 70 < x_t < 180 \\ \text{Hyper} & \text{if } 180 \leqslant x_t < 250 \\ \text{SHyper} & \text{if } x_t \geqslant 250\,. \end{cases} \tag{5}$$

All the measurement units are mg/dL and have been omitted for the sake of conciseness. The aim of the classification algorithms is to predict the future state given the past CGM readings. In particular, all the classification models presented in this study implement a function f such as:

$$f(\boldsymbol{X}_t^N, \boldsymbol{\theta}_t) = \hat{c}_{t+ph} \qquad\qquad \hat{c}_{t+ph} \in \mathbb{C}, \tag{6}$$

where $\boldsymbol{X}_t^N$ is a vector containing the previous N glucose readings, $\boldsymbol{\theta}_t$ is the unknown parameters vector and ph is the prediction horizon. As for Eq. (1), upon completion of the training phase, $\boldsymbol{\theta}_t$ is a constant vector or it changes at each iteration for the static and dynamic approach, respectively.

An *accuracy* measure is used as the performance metric for the classification algorithms, which is here computed as the percentage of classes that are correctly predicted. Note that the accuracy metric is evaluated including the samples belonging to all the considered classes (hypo, hyper and normal states). Therefore, the result may be affected by an imbalanced dataset.

In this study, the following classification algorithms were considered: Support Vector Machine (SVM) [33], k-Nearest Neighbor (kNN) [34], Naive Bayes (NB) [35], Classification Tree (CT) [36], Random Forest (RF) [37], AdaBoost [38], Linear Discriminant Analysis (LDA).

C. Events detection

The ultimate goal of this study is to present a performance evaluation framework involving a large number of prediction and classification models, trying to figure out the most appropriate approach to solve glucose event prediction problems. We consider as *events* all the time instants $\bar{t}$ whose corresponding samples verify one of the following conditions:

$$\begin{array}{llll} x_{\bar{t}-1} > 50 & \text{and} & x_{\bar{t}} \leqslant 50 & \text{Severe Hypoglycemia Event} \\ x_{\bar{t}-1} > 70 & \text{and} & x_{\bar{t}} \leqslant 70 & \text{Hypoglycemia Event} \\ x_{\bar{t}-1} < 180 & \text{and} & x_{\bar{t}} \geqslant 180 & \text{Hyperglycemia Event} \\ x_{\bar{t}-1} < 250 & \text{and} & x_{\bar{t}} \geqslant 250 & \text{Severe Hyperglycemia Event} \end{array}$$

Four different sets can be created according to this criterion, with each set containing all the events that meet one of the above defined conditions. Due to the presence of measurement noise and signal fluctuations, multiple consecutive events of the same type could be observed within a short time frame. To avoid this behavior, which is quite unrealistic, a settling time of 30 minutes is also considered [39], [40]. That is, if a specific event occurs, additional events of same type are not considered for the following 30 minutes. The same procedure can also be applied to the predicted samples $\hat{x}_t$, and a similar approach is used for the predicted classes $\hat{c}_t$, where the event is mapped onto a class change.

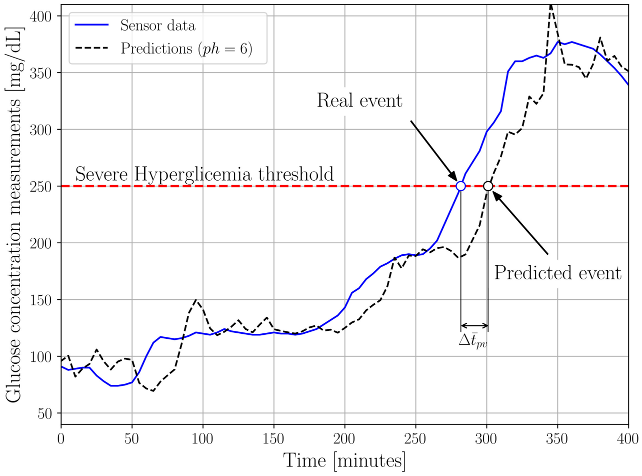

Fig. 2: Example of severe hyperglycemic event onset. A Linear Regression model has been considered.

In the following, we refer to a specific set, containing a generic type of event. The considerations that follow apply to all events.

Let $\mathbb{V} = \{\bar{t}_1, \ldots, \bar{t}_V\}$ be the generic set containing the V time instants associated with the real events, obtained analyzing the CGM measurements x_t, and let $\mathbb{P} = \{\bar{t}_1, \ldots, \bar{t}_P\}$ be the set containing the P predicted events, found using the predicted BG concentration values $\hat{x}_t$ (or $\hat{c}_t$ in case of classification methods). Each element $\bar{t}_p \in \mathbb{P}$ is individually analyzed and marked as either a True Positive (TP) or a False Positive (FP), according to the following criterion.

Let define $\Delta\bar{t}_{pv} = \bar{t}_p - \bar{t}_v$ as the distance between a generic predicted event time $\bar{t}_p$ and a real event $\bar{t}_v$ (see Fig. 2). With ph we mean the temporal prediction step into the future. For each $\bar{t}_p \in \mathbb{P}$, if there exists $\bar{t}_v \in \mathbb{V}$ such as $-k < \Delta\bar{t}_{pv} < ph$, $\bar{t}_p$ is considered to be a TP and $\bar{t}_v$ is removed from the set $\mathbb{V}$ and no longer considered in the following steps, to prevent that the same event be considered multiple times. A TP condition is indeed assigned to a predicted event that, in general, actually occurs in the future, within a tolerance range. If the condition above is not met, $\bar{t}_p$ is considered to be a FP, i.e., the predicted event will not occur or has already occurred. $k \in \mathbb{N}$ is a positive constant that prevents the association of events that are too distant apart. When there are multiple events meeting the above condition, only the one with the minimum $|\Delta\bar{t}_{pv}|$ distance is considered. After having checked all the elements in set P, all the remaining events in the set $\mathbb{V}$, if any, are tagged as False Negatives (FN). In fact, at this point, the set $\mathbb{V}$ contains the observed events that have not

TABLE I: List of possible conditions associated with a predicted event.

$\Delta\bar{t}_{pv} < -k$	The event is out of the analysis range.
$-k \leqslant \Delta\bar{t}_{pv} < 0$	The real event occurs later than the predicted time $\bar{t}_p$, but still within the tolerance range.
$\Delta\bar{t}_{pv} = 0$	The real event occurs exactly when predicted.
$0 < \Delta\bar{t}_{pv} < ph$	The real event $\bar{t}_v$ occurs earlier than the predicted time $\bar{t}_p$, but still in the future.
$\Delta\bar{t}_{pv} \geqslant ph$	The real event has already happened. Its prediction has failed.

been correctly predicted. In Tab. I, a list of possible conditions associated with a predicted event is shown for clarity. In Fig. 2 an example of the onset of a severe hyperglycemic event is shown, along with the predictions of a linear regressor with a prediction horizon $ph = 6$.

Once all the TP, FP and FN events have been found, Precision, Recall and F-measure can be evaluated as follows [41]:

$$\texttt{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{7}$$

$$\texttt{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{8}$$

$$\texttt{F-measure} = 2 \cdot \frac{\texttt{Precision} \cdot \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}} \tag{9}$$

where N_{TP}, N_{FP} and N_{FN} are respectively the total number of TP, FP and FN events. With these three metrics, an overall assessment of the algorithm performance is possible, along with a probabilistic interpretation. The Precision estimates the fraction of correctly predicted events (i.e., the true positives) among all the predicted events (including the false positive outcomes). It can be interpreted as the probability that a predicted event will actually happen in the near future. The Recall, instead, estimates the fraction of real events that is correctly predicted by the algorithm. It can be seen as the probability that a real event is detected in advance. Both these metrics are extremely important and should be jointly considered to provide a complete performance analysis. The F-measure combines these two metrics with an harmonic mean, providing a good way to compare different algorithms' outcomes.

Note that the metrics defined in Eqs. (7), (8) and (9) refer to a specific set $\mathbb{V}$, that is related to a particular event type. When another event is considered, these results may vary considerably. To provide an overall assessment, all the different conditions can be aggregated and all the events can be considered at the same time. In this study, we focus both on the overall performance and on the specific hyperglycemic event, which is considered to be of great importance for diabetic patients.

A compact and clear overview of the event prediction capabilities of a method can be assessed with these metrics. Nevertheless, there is still a limitation, due to the loss of information about the time difference $\Delta\bar{t}_{pv}$ between the predicted and the real events, referred to as *prediction distance* in the following. Therefore, an additional statistical analysis is devoted to the comparison of the prediction distances achieved by each of the selected methods.

To provide more clarity, we remark the clear difference between *signal prediction* and *event prediction*, and all the corresponding metrics. The former is related to the capability of the algorithm to predict the future value of the signal, or the corresponding class in case of classifiers, given some historical data. The event prediction analysis, instead, is specifically intended to evaluate the ability to predict the occurrence of a specific condition.

D. Training process

A different training process is used for the static and dynamic approaches. We recall that the main difference between them is that the static model is trained only once, the dynamic model, instead, is updated at each iteration. The input data for each of these models, however, is of the same type, i.e., they both take the past N CGM readings as input and output the predicted BG concentration (for regressors) or the corresponding class (for classifiers) at a preset prediction horizon ph.

Let consider a generic vector $\boldsymbol{X}^{\boldsymbol{M}} = [x_1, \ldots, x_M]$ containing M measurements, and let ph be the prediction horizon, with $M > N + ph$. A series of input samples $\boldsymbol{i}_{\boldsymbol{n}}^{\boldsymbol{N}} \in \mathbb{R}^N$, and the corresponding output label o_n, can be extracted from this vector, according to the following criterion:

$$\begin{aligned} &\text{Regression} && \boldsymbol{i}_{\boldsymbol{n}}^{\boldsymbol{N}} = [x_{n-N+1}, \ldots, x_n] && o_n = x_{n+ph} \\ &\text{Classification} && \boldsymbol{i}_{\boldsymbol{n}}^{\boldsymbol{N}} = [x_{n-N+1}, \ldots, x_n] && o_n = q(x_{n+ph}) \\ & && \forall n \in \{N, \ldots, M - ph\} && \end{aligned} \tag{10}$$

All the pairs $(\boldsymbol{i}_{\boldsymbol{n}}^{\boldsymbol{N}}, o_n)$ generated according to Eq. (10) are included in a set used for training or testing the analyzed models. In the following, the specific training procedure for static and dynamic approaches is detailed.

1) Static model: Data acquired from K different subjects is considered in this phase. Here, a Leave One Patient Out validation procedure is used to evaluate the performance metrics. In particular, at each round of the validation a subject is considered as a test subject and referred to as the *target user*. All the data belonging to the other $K - 1$ subjects is used to generate a single training set according to Eq. (10). Then, this set is employed to find the parameters $\boldsymbol{\beta}$ or $\boldsymbol{\theta}$ in Eqs. (1) and (6). Once the model is trained, its parameters remain fixed. A test set, created with the target user data, is used to evaluate the performance metrics, both for prediction capabilities (residuals for regressors and accuracy for classifiers) and event detection (Precision, Recall, F-measure). This procedure is repeated for each subject in the dataset.

2) Dynamic model: For the dynamic model approach, only the data belonging to the target user is considered, both for training and testing procedures. Let us consider a generic time instant t, and a vector containing the current (time t) and the previous $M - 1$ measurements $\boldsymbol{X}_{\boldsymbol{t}}^{\boldsymbol{M}} = [x_{t-M+1}, \ldots, x_t] \in \mathbb{R}^M$. A training set is generated from this vector using Eq. (10), and it is used to find the model parameters $\boldsymbol{\beta}_t$ or $\boldsymbol{\theta}_t$ associated with the current time instant. Then, the last N samples $\boldsymbol{X}_{\boldsymbol{t}}^{\boldsymbol{N}}$ are employed to make a prediction $\hat{x}_{t+ph}$, or $\hat{c}_{t+ph}$ in case of classifiers, where $N < M$ represents the number of inputs for

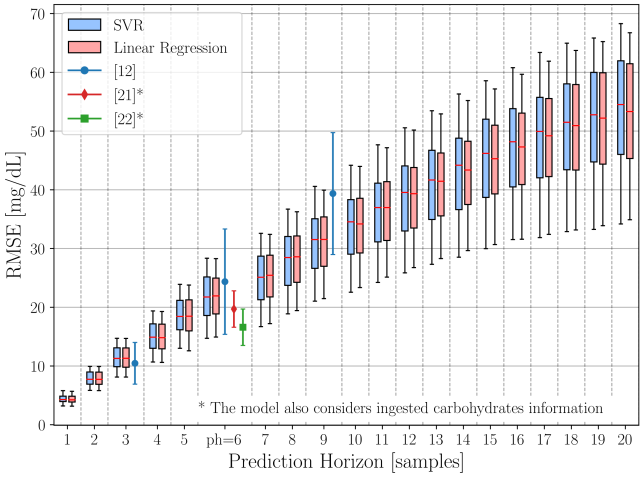

Fig. 3: RMSE for increasing prediction horizon. As comparison, the results reported in [12], [21], [22] have been included.

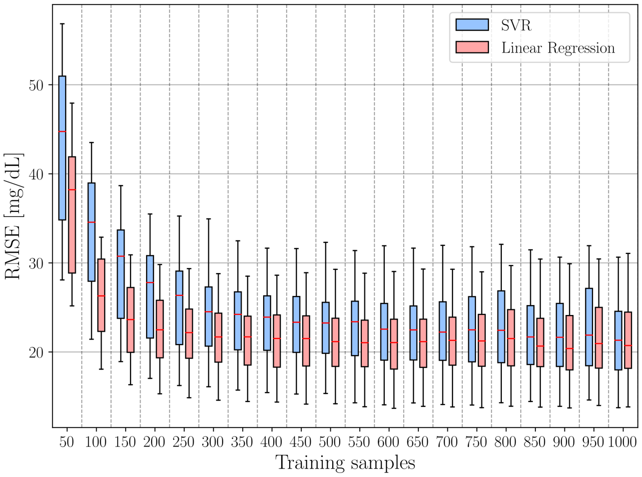

Fig. 4: RMSE for increasing training samples of the dynamic model.

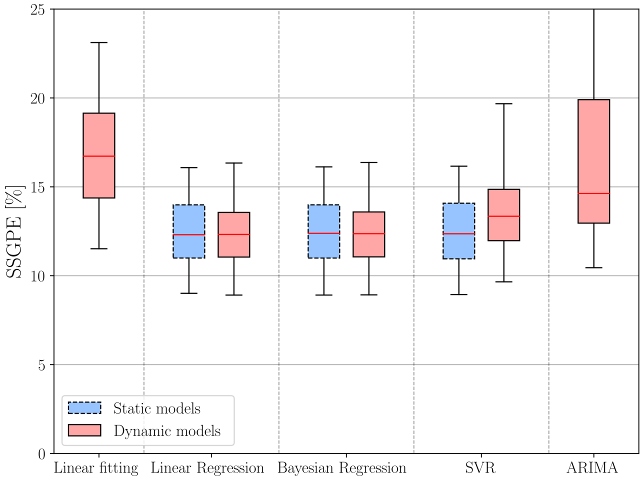

Fig. 5: SSGPE for each regression method.

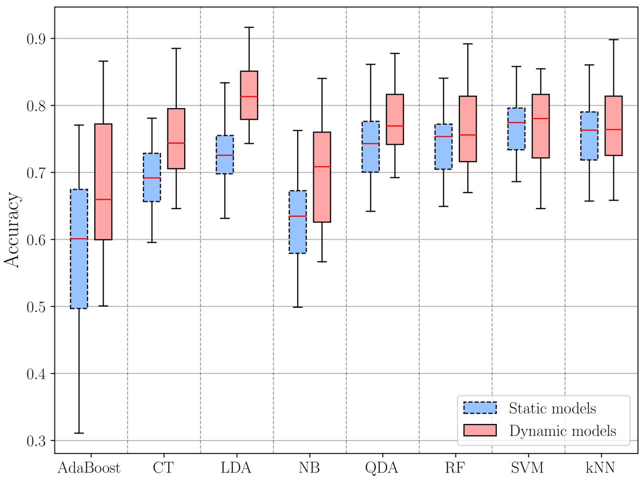

Fig. 6: Accuracy for each classification method.

the selected model. Finally, the prediction is compared to the real value x_{t+ph} or c_{t+ph}, to evaluate the performance metrics.

This procedure is repeated for each time instant t for which both the future value x_{t+ph} and $\boldsymbol{X}_t^M$ are available. At each iteration, the new measurement is added to the vector $\boldsymbol{X}_t^M$, and the oldest one is dropped, in order to always have a training vector of fixed length, containing the last M readings.

Note that: **1)** the model is trained using only the past data, which is always available in a real implementation of the algorithm, and **2)** each model is used to predict a single value.

III. Results

Data consist of CGM monitoring for 7 consecutive days of 89 Type-1 Diabetes (T1D) patients, which are extracted from a larger datasets. CGM traces were measured by the Dexcom G4 Platinum CGM sensor (Dexcom Inc., San Diego, CA), which has a sampling period of 5 minutes. The CGM sensor has been inserted at the beginning of day 1 of the study and calibrated accordingly to manufacturer instructions (i.e., twice at the beginning of the monitoring, and then every 12 hours). In this period, patients were admitted to the clinical research center (CRC) at days 1, 4, and 7, each time for a time period of 12 hours. During CRC admission, subjects underwent a delayed-and-increased insulin bolus at meal-time in order to simulate rapid glucose fluctuations, aimed at testing CGM sensors in more challenging conditions. During the rest of the monitoring, patients were asked to behave as usual (free-living conditions). No additional signals, e.g., CHO content of the meals or injected insulin, are available. Additional clinical information, e.g., the cohort, are specified in [42].

Since during acquisition on real subjects, failures and irregular sampling may happen, the CGM data was first interpolated using a third-order spline for obtaining a regular sampling period, and then the value of any missing data reading was imputed. In case of too many consecutive missing samples, data is discarded and not considered in the corresponding period of time. All the presented results are shown using boxplots, which provide a statistical and compact view of the results. The boxes represent the Interquartile Range (IQR), the line is the median value, and the whiskers have a coverage factor of 90%. A time window of $N = 5$ samples has been considered for all the models. A further increase in the window size N did not show significant improvements.

In Fig. 3, the RMSE values for an increasing prediction horizon ph are shown. As expected, the error considerably increases for long-term predictions. SVR and Linear Regression algorithms have been used here, but other methods exhibit a similar behavior. For the sake of comparison, the results reported in [12], [21], [22] are also shown in the plot, bearing in mind that the dataset used is different. Note that the algorithms implemented in [21] and [22] make use of additional information on ingested carbohydrates, which allows them to achieve better performance. If available, these additional sources of information are expected to lead to improved results. A prediction horizon of $ph = 6$ samples (corresponding to 30 minutes) is considered for the following analyses, as it provides a good trade-off between error and prediction time.

The dimension M of the training data buffer (see Fig. 1) for the dynamic approach is an important parameter to take into account. The RMSE has been measured varying the number of historical samples in the training set. The result is shown in Fig. 4, where the RMSE performance of SVR and Linear Regression dynamic models are shown as examples. A saturation effect is observed for a buffer size larger than $M = 400$ samples, beyond which no additional improvements are observed. This value, which approximately corresponds to 30 hours of training data, is then chosen to train the final models.

The signal prediction performance of all the methods considered in this study is shown in Fig. 5 (regressors) and Fig. 6 (classifiers). SSGPE has been chosen as representative metric for regression methods, and accuracy for classifiers, as defined in Section II. Note that only a dynamic implementation has been considered for linear fitting and ARIMA models. This is why the results for their static versions do not appear in Fig. 5. Quadratic and Spline fitting provide poor results, and for this reason have also been omitted from the plot.

Some of the algorithms require to set a number of hyper-parameters, which can heavily affect the performance of the learning algorithms. Prior to the proper training of the algorithms requiring such hyper-parameters, an optimization phase is always required. This pre-train explores different set of hyper-parameters, looking at the performance of the algorithm on a validation set (separated from the training and testing data). Once the pre-training phase is finished, the hyper-parameters value is kept fixed during the subsequent training and testing phases. Several ARIMA models of different orders have been validated, considering all the possible combinations up to a maximum of the second order. The model that provided the best results has an autoregressive term $ar = 2$, a differencing term $i = 0$ and a moving average term $ma = 0$. All the presented results refer to these parameters. Radial Basis Function (RBF) kernel is considered for kernel based methods, i.e., SVM and SVR: a grid search procedure has been applied during pre-training to optimize the penalty parameter of the error term and the RBF kernel coefficient. A number of 3, 5 and 10 neighbors have been considered for the kNN algorithm, the results that we show use a value of 10. Finally, a maximum of 50 weak classifiers have been used for the AdaBoost method. Please refer to Table II for further details on the setting of hyper-parameters.

TABLE II: Hyper-parameters settings.

Method	Parameters settings
Bay. Reg.	All the shape and the inverse scale parameters for the Gamma distribution priors have been set to 10^{-6}.
SVR	A radial basis function kernel has been considered. The penalty parameter of the error term has been set to 5^3 and the kernel coefficient to 5^{-6}.
ARIMA	A second order auto-regressive model has been considered.
AdaBoost	A maximum of 50 estimators and a learning rate of 1 have been considered [43].
CT	Gini impurity has been used to measure the quality of a split [44].
LDA	Singular value decomposition has been used as the solver method.
NB	The likelihood of the features is assumed to be Gaussian.
RF	10 estimators have been considered. Gini impurity has been used to measure the quality of a split.
SVM	A radial basis function kernel has been considered. The penalty parameter of the error term has been set to 10^5 and the kernel coefficient to 1^{-7}.
kNN	10 neighbors have been considered.

Regression methods (Fig. 5) exhibit very similar performance both for static and dynamic approaches, except for the Linear Fitting method, which provides higher errors. As for the classification methods, all the dynamic models perform better, in term of accuracy, than their static counterparts. LDA is the model with the highest average accuracy across all classes. As a general consideration, AdaBoost and NB perform worse than other classification methods, on average.

Analyzing signal prediction performance is certainly a good comparison strategy. Nevertheless, this approach could be misleading when event detection capabilities need to be assessed. An event, as defined in Sec. II-C, only occurs when a predefined condition is met, such as the overcoming of a threshold. However, the metrics that are commonly used to evaluate prediction algorithms usually assume that all samples are equally important, but this could lead to inappropriate results, especially when the dataset is highly imbalanced, due, for example, by the prevalence of normal states. In this case, a small prediction error will be driven by the ability of a method to follow the normal range, considering the hypo- and hyper-glycemic events as outliers. For this reason, and also to allow a direct comparison between regression and classification algorithms, a more insightful analysis has been devoted to event detection performance, as detailed in Section II-C. A simple prediction method that considers the last measurement as the predicted value is typically used as a lower bound on the performance evaluation. Since this method is not able to predict any event, as defined in this study, it has no prediction capabilities and it has not been considered in the results. The event-based performance metrics are included in Fig. 7, where an exhaustive comparison among all the methods is shown. In this figure, we show the Recall (blue boxes), the Precision (red boxes) and the F-measure (green boxes), dividing the results into two groups: regressors and classifiers (separated through a vertical dashed line). The

TABLE III: Events detection metrics for all the analyzed methods, static and dynamic implementations, and for each event type. The reported uncertainty refers to the standard deviation.

		Severe Hypoglycemia		Hypoglycemia		Hyperglycemia		Severe Hyperglycemia	
		Static	Dynamic	Static	Dynamic	Static	Dynamic	Static	Dynamic
Lin. Fit.	Recall	n/a	0.96 ± 0.11	n/a	0.93 ± 0.14	n/a	0.92 ± 0.08	n/a	0.92 ± 0.10
	Precision	n/a	0.24 ± 0.11	n/a	0.38 ± 0.16	n/a	0.51 ± 0.11	n/a	0.43 ± 0.13
	F-measure	n/a	0.37 ± 0.14	n/a	0.52 ± 0.17	n/a	0.65 ± 0.10	n/a	0.58 ± 0.13
Lin. Reg.	Recall	0.60 ± 0.33	0.62 ± 0.36	0.62 ± 0.22	0.51 ± 0.29	0.89 ± 0.08	0.78 ± 0.20	0.80 ± 0.17	0.61 ± 0.30
	Precision	0.49 ± 0.28	0.55 ± 0.34	0.47 ± 0.20	0.41 ± 0.24	0.66 ± 0.11	0.63 ± 0.17	0.61 ± 0.17	0.50 ± 0.27
	F-measure	0.51 ± 0.27	0.57 ± 0.34	0.53 ± 0.20	0.44 ± 0.25	0.75 ± 0.09	0.69 ± 0.17	0.68 ± 0.16	0.54 ± 0.27
Bay. Reg.	Recall	0.61 ± 0.32	0.59 ± 0.38	0.62 ± 0.22	0.50 ± 0.30	0.89 ± 0.09	0.77 ± 0.21	0.80 ± 0.17	0.60 ± 0.29
	Precision	0.50 ± 0.28	0.54 ± 0.35	0.47 ± 0.20	0.41 ± 0.25	0.66 ± 0.11	0.62 ± 0.18	0.61 ± 0.17	0.49 ± 0.26
	F-measure	0.52 ± 0.27	0.55 ± 0.35	0.53 ± 0.20	0.44 ± 0.26	0.75 ± 0.09	0.68 ± 0.18	0.68 ± 0.16	0.53 ± 0.27
SVR	Recall	0.49 ± 0.33	0.44 ± 0.39	0.75 ± 0.22	0.52 ± 0.29	0.87 ± 0.09	0.73 ± 0.19	0.83 ± 0.16	0.66 ± 0.27
	Precision	0.43 ± 0.30	0.39 ± 0.34	0.51 ± 0.19	0.45 ± 0.26	0.64 ± 0.11	0.60 ± 0.17	0.62 ± 0.17	0.53 ± 0.24
	F-measure	0.43 ± 0.27	0.39 ± 0.33	0.59 ± 0.19	0.46 ± 0.25	0.73 ± 0.09	0.65 ± 0.17	0.70 ± 0.15	0.58 ± 0.24
ARIMA	Recall	n/a	0.53 ± 0.38	n/a	0.42 ± 0.29	n/a	0.68 ± 0.25	n/a	0.53 ± 0.27
	Precision	n/a	0.49 ± 0.36	n/a	0.37 ± 0.25	n/a	0.59 ± 0.22	n/a	0.46 ± 0.25
	F-measure	n/a	0.50 ± 0.36	n/a	0.39 ± 0.26	n/a	0.63 ± 0.23	n/a	0.49 ± 0.25
AdaBoost	Recall	0.28 ± 0.34	0.05 ± 0.12	0.54 ± 0.30	0.23 ± 0.23	0.82 ± 0.14	0.31 ± 0.19	0.29 ± 0.29	0.21 ± 0.18
	Precision	0.11 ± 0.17	0.08 ± 0.17	0.28 ± 0.21	0.24 ± 0.25	0.53 ± 0.17	0.22 ± 0.14	0.32 ± 0.30	0.21 ± 0.21
	F-measure	0.12 ± 0.15	0.06 ± 0.13	0.34 ± 0.21	0.22 ± 0.23	0.63 ± 0.14	0.25 ± 0.16	0.24 ± 0.20	0.20 ± 0.17
CT	Recall	0.95 ± 0.12	0.21 ± 0.31	0.89 ± 0.16	0.53 ± 0.30	0.94 ± 0.06	0.70 ± 0.18	0.92 ± 0.10	0.48 ± 0.32
	Precision	0.13 ± 0.07	0.12 ± 0.17	0.23 ± 0.11	0.29 ± 0.17	0.32 ± 0.09	0.32 ± 0.11	0.28 ± 0.09	0.22 ± 0.15
	F-measure	0.23 ± 0.11	0.14 ± 0.19	0.35 ± 0.14	0.36 ± 0.20	0.47 ± 0.10	0.44 ± 0.13	0.43 ± 0.10	0.30 ± 0.20
LDA	Recall	0.96 ± 0.12	0.10 ± 0.20	0.77 ± 0.21	0.25 ± 0.22	0.93 ± 0.06	0.71 ± 0.20	0.88 ± 0.10	0.41 ± 0.30
	Precision	0.27 ± 0.13	0.18 ± 0.33	0.31 ± 0.16	0.29 ± 0.26	0.61 ± 0.13	0.51 ± 0.17	0.65 ± 0.15	0.30 ± 0.21
	F-measure	0.41 ± 0.16	0.12 ± 0.22	0.42 ± 0.18	0.25 ± 0.21	0.73 ± 0.11	0.59 ± 0.17	0.74 ± 0.13	0.34 ± 0.24
NB	Recall	0.72 ± 0.26	0.20 ± 0.25	0.66 ± 0.20	0.44 ± 0.23	0.34 ± 0.14	0.23 ± 0.19	0.05 ± 0.09	0.15 ± 0.20
	Precision	0.34 ± 0.21	0.12 ± 0.15	0.38 ± 0.18	0.32 ± 0.22	0.34 ± 0.13	0.21 ± 0.16	0.05 ± 0.09	0.13 ± 0.16
	F-measure	0.44 ± 0.21	0.14 ± 0.17	0.45 ± 0.16	0.36 ± 0.21	0.33 ± 0.12	0.22 ± 0.17	0.05 ± 0.09	0.13 ± 0.17
QDA	Recall	0.88 ± 0.20	0.10 ± 0.19	0.79 ± 0.18	0.35 ± 0.26	0.93 ± 0.08	0.67 ± 0.19	0.81 ± 0.15	0.44 ± 0.27
	Precision	0.30 ± 0.14	0.10 ± 0.19	0.33 ± 0.16	0.29 ± 0.22	0.52 ± 0.13	0.41 ± 0.16	0.39 ± 0.14	0.25 ± 0.16
	F-measure	0.43 ± 0.17	0.10 ± 0.19	0.45 ± 0.17	0.31 ± 0.23	0.66 ± 0.12	0.50 ± 0.16	0.51 ± 0.14	0.31 ± 0.18
RF	Recall	0.93 ± 0.20	0.19 ± 0.27	0.86 ± 0.16	0.46 ± 0.28	0.92 ± 0.09	0.60 ± 0.20	0.91 ± 0.10	0.38 ± 0.28
	Precision	0.18 ± 0.09	0.12 ± 0.17	0.30 ± 0.14	0.30 ± 0.19	0.43 ± 0.11	0.34 ± 0.14	0.44 ± 0.11	0.21 ± 0.15
	F-measure	0.29 ± 0.13	0.14 ± 0.19	0.43 ± 0.17	0.36 ± 0.21	0.58 ± 0.11	0.43 ± 0.15	0.59 ± 0.11	0.26 ± 0.19
SVM	Recall	0.98 ± 0.08	0.31 ± 0.34	0.86 ± 0.19	0.62 ± 0.29	0.95 ± 0.06	0.81 ± 0.18	0.93 ± 0.08	0.58 ± 0.34
	Precision	0.32 ± 0.13	0.13 ± 0.15	0.36 ± 0.17	0.32 ± 0.17	0.58 ± 0.13	0.47 ± 0.17	0.60 ± 0.14	0.34 ± 0.20
	F-measure	0.47 ± 0.15	0.18 ± 0.19	0.49 ± 0.18	0.41 ± 0.21	0.71 ± 0.11	0.59 ± 0.17	0.72 ± 0.12	0.43 ± 0.25
kNN	Recall	0.96 ± 0.11	0.11 ± 0.20	0.86 ± 0.18	0.28 ± 0.24	0.92 ± 0.07	0.38 ± 0.21	0.85 ± 0.13	0.24 ± 0.21
	Precision	0.26 ± 0.12	0.15 ± 0.29	0.37 ± 0.17	0.25 ± 0.22	0.56 ± 0.12	0.28 ± 0.17	0.55 ± 0.15	0.17 ± 0.16
	F-measure	0.40 ± 0.15	0.12 ± 0.21	0.50 ± 0.17	0.25 ± 0.21	0.69 ± 0.10	0.32 ± 0.18	0.66 ± 0.13	0.20 ± 0.17

results in the same group refer to the static implementation (dashed boxes) and to the dynamic model (solid line boxes). As a general behavior, a high recall and a low precision can be observed. This is typically due to a tendency to predict events that will not really happen. This effect is particularly emphasized for most of the static classifiers and for linear fitting. Considering the F-measure as the most representative score, static models always perform better, or at least are comparable with the corresponding dynamic implementations. It is worth noting that the signal prediction performance results show a different behavior, especially for classifiers, where the dynamic implementations show even better performance. This result empirically demonstrates that the signal prediction performance metrics, evaluated considering the entire signals, could be very imbalanced and they are not always a good indicator of event prediction capabilities, which require a more thorough investigation.

The classifier showing the best performance is SVM, followed by LDA and kNN. Regression methods, even if with a lower recall, provide a significantly higher precision, which leads to an improved F-measure. In particular, the static implementation of SVR, Linear and Bayesian Regression provide the best results across all methods.

We remark that these results refer to all the possible events, including hypoglycemic and hyperglycemic conditions. It is also interesting to evaluate the performance when only a specific event is considered. For a more detailed analysis, Table III reports all the event detection metrics, along with their standard deviations, separating the results over the four considered event types. Under this assumption, the classification problem needs to be revised. The class space set $\mathbb{C}$, introduced in Section II-B, is now reduced to a binary set $\mathbb{C}_H$ = {Hyper, Norm}, and also the mapping function defined in Eq. (5) is modified accordingly. All the classifiers are retrained solving this reduced binary classification problem. The regressors, instead, remain the same, but their performance is now evaluated on the specific event only, to obtain a fair comparison.

In this study, we focus our attention on hyperglycemic events. The frequent occurrence of these events in our dataset makes them suitable for a more in-depth analysis. Furthermore, in a preventive scenario, where a patient takes

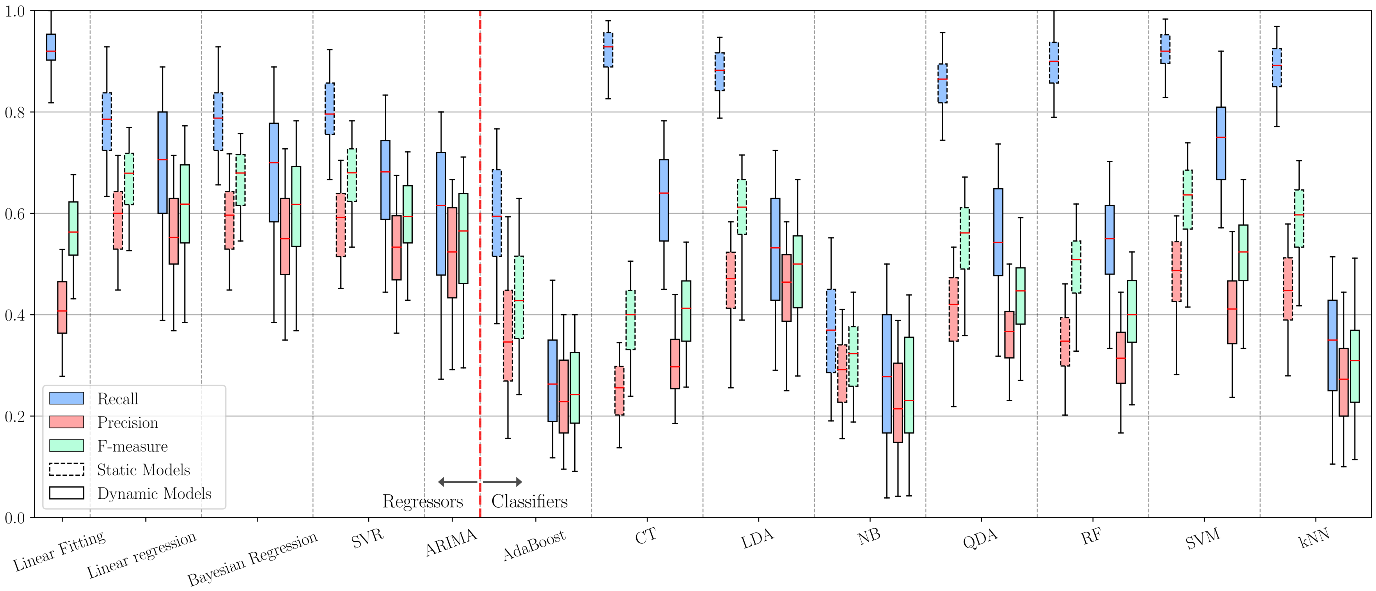

Fig. 7: Events detection metric for all the analyzed algorithms (prediction horizon ph = 6).

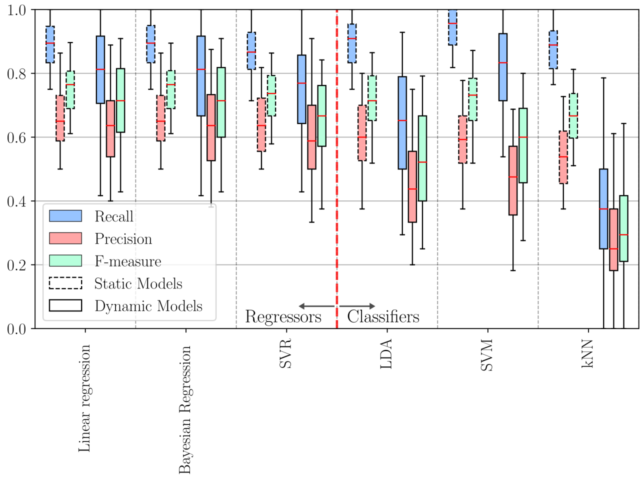

Fig. 8: Events detection metric for all the analyzed algorithms only considering hyperglycemic events (prediction horizon ph = 6).

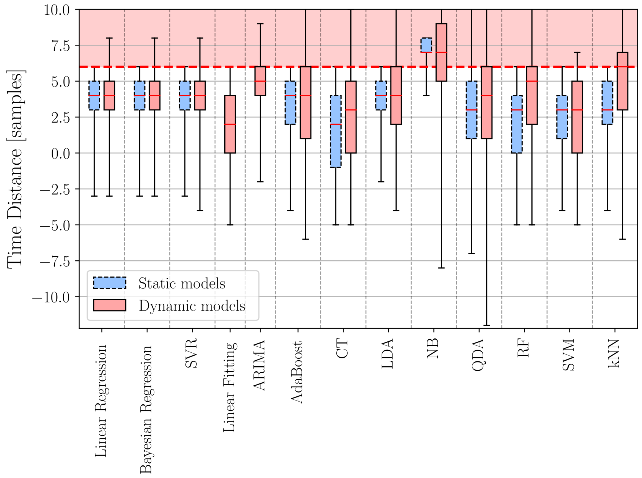

Fig. 9: Temporal distance between real and predicted events (prediction horizon ph = 6).

pre-emptive actions based on the predicted glucose level, the hyperglycemic condition is of primary importance. In Fig. 8, the event detection performance of the three best regressors and classifiers is shown, when only the hyperglycemic events are considered. A significant overall improvement can be observed. In particular, the classifiers exhibit improved performance when specialized to detect a specific event (hyperglycemia in this case). Also in this case the static models provide better results than their dynamic counterparts.

The best methods remain the Linear and Bayesian Regression, which perform almost the same and have a median F-measure value of 0.76. The best performing classifier is the SVM, which has a median F-measure value of 0.73, and the highest median recall value of about 0.95.

The last analysis concerns the prediction distance, i.e., the period between the real event and the predicted one. Statistical results are shown in Fig. 9. Since the prediction horizon in our implementation is ph = 6 samples, the dashed red line in the figure corresponds to the actual time of the event. Even if the prediction horizon is 30 minutes, events are not always detected that early. Most of the times, even when an event is correctly predicted, it actually happens sooner than expected.

Let consider the best performing regressor and classifier, i.e., the Linear or Bayesian Regression method and the SVM. The former has a median prediction time of only 2 samples, which means that half of the correctly predicted events actually happen at least 10 minutes later. Furthermore, the IQR goes from 5 to 15 minutes, i.e., half of the times the event will happen after a time period within this range. The SVM, instead, has much better performance, with an IQR ranging from 10 to 25 minutes. Therefore, even if the SVM F-measure is slightly lower, this classifier is able to predict the events earlier than the regression methods. This makes a specialized classifier the best choice for glucose event prediction algorithms, and a good performance benchmark for future developments.

IV. Conclusion

A comparison between a number of selected regression and classification algorithms for the prediction of hyper and hypoglycemic events based on CGM signals has been carried out in this study. The value of the glucose concentration in the blood depends on many external factors that affect glucose-insulin dynamics, such as the amount of CHO of a meal or of a snack, the insulin injected, physical activity, etc.. In addition, CGM devices are often affected by a high level of noise and error, mainly due to the interaction with the human body, that is dynamic by nature. The lack of additional information from the user, which plays a fundamental role, makes the prediction a difficult task. Two different approaches have been investigated: static and a dynamic implementations. Both signal prediction and event detection metrics are considered in this study. The former, typically used in the literature, quantifies the capability of the model to make good predictions; the latter, instead, refers to the capacity of the model to correctly predict future hyper/hypoglycemic events. Static methods exhibit better performance for most of the analyses considered in this work, with particular focus on F-measure values, as shown in Tab. III. However, prediction metrics (RMSE and SSGPE) are not always in agreement with the event detection capabilities of the algorithms, which require a specific analysis. The best results, in terms of event prediction capabilities, have been obtained with Linear and Bayesian Regression methods. All the classifiers show some improvement when trained for a specific and single event, such as hyperglycemia. In particular, Support Vector Machine (SVM) performs nearly the same as the best regressor, under this working assumption. An additional analysis based on the prediction time shows that classification methods also tend to predict the events sooner with respect to regressors. Based on these considerations, a specialized SVM yields the best overall performance.

There are several ways in which this work could be extended. A further optimization of the proposed algorithms or the inclusion of new prediction techniques could provide improved performance. Furthermore, a promising research avenue is the implementation of a combined approach, towards subject-adaptive and personalized algorithms. Having data covering longer time periods, we may define and train a static method, and then progressively tune it in a dynamic way, adapting it to a specific subject as new measurements become available.

Acknowledgment

The authors would like to thank the company Dexcom Inc. (San Diego, CA) for having provided the G4 Platinum sensor data used in this paper [45].

References

[1] R. Taylor, “Type 2 diabetes: etiology and reversibility,” *Diabetes care*, vol. 36, no. 4, pp. 1047–1055, Apr 2013.

[2] C. Hayes and A. Kriska, “Role of physical activity in diabetes management and prevention,” *Journal of the American Dietetic Association*, vol. 108, no. 4, pp. S19–S23, Apr 2008.

[3] S. H. Ley, O. Hamdy, V. Mohan, and F. B. Hu, “Prevention and management of type 2 diabetes: dietary components and nutritional strategies,” *The Lancet*, vol. 383, no. 9933, pp. 1999–2007, Jun 2014.

[4] D. Rodbard, “Continuous glucose monitoring: a review of recent studies demonstrating improved glycemic outcomes,” *Diabetes Technology & Therapeutics*, vol. 19, no. 3, pp. 25–37, Jun 2017.

[5] A. Facchinetti, “Continuous glucose monitoring sensors: past, present and future algorithmic challenges,” *Sensors*, vol. 16, no. 12, p. 2093, Dec 2016.

[6] C. Zecchin, A. Facchinetti, G. Sparacino, and C. Cobelli, “Reduction of number and duration of hypoglycemic events by glucose prediction methods: a proof-of-concept in silico study,” *Diabetes technology & therapeutics*, vol. 15, no. 1, pp. 66–77, Jan 2013.

[7] B. Buckingham, H. P. Chase, E. Dassau, E. Cobry, P. Clinton, V. Gage, K. Caswell, J. Wilkinson, F. Cameron, H. Lee, B. W. Bequette, and F. J. Doyle, “Prevention of nocturnal hypoglycemia using predictive alarm algorithms and insulin pump suspension,” *Diabetes Care*, vol. 33, no. 5, pp. 1013–1017, May 2010.

[8] A. Zhong, P. Choudhary, C. McMahon, P. Agrawal, J. B. Welsh, T. L. Cordero, and F. R. Kaufman, “Effectiveness of automated insulin management features of the MiniMed® 640G sensor-augmented insulin pump,” *Diabetes technology & therapeutics*, vol. 18, no. 10, pp. 657–663, Oct 2016.

[9] G. Sparacino, F. Zanderigo, S. Corazza, A. Maran, A. Facchinetti, and C. Cobelli, “Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series,” *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 5, pp. 931–937, May 2007.

[10] A. Gani, A. V. Gribok, S. Rajaraman, W. K. Ward, and J. Reifman, “Predicting subcutaneous glucose concentration in humans: data-driven glucose modeling,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 2, pp. 246–254, Feb 2009.

[11] M. Eren-Oruklu, A. Cinar, L. Quinn, and D. Smith, “Estimation of future glucose concentrations with subject-specific recursive linear models,” *Diabetes Technology and Therapeutics*, vol. 11, no. 4, pp. 243–253, Apr 2009.

[12] C. Perez-Gandia, A. Facchinetti, G. Sparacino, C. Cobelli, E. J. Gomez, M. Rigla, A. de Leiva, and M. E. Hernando, “Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring,” *Diabetes Technology and Therapeutics*, vol. 12, no. 1, pp. 81–88, Jan 2010.

[13] V. Naumova, S. V. Pereverzyev, and S. Sivananthan, “A meta-learning approach to the regularized learning-case study: blood glucose prediction,” *Neural Networks*, vol. 33, pp. 181–193, Sep 2012.

[14] D. A. Finan, F. J. Doyle, C. C. Palerm, W. C. Bevier, H. C. Zisser, L. Jovanovic, and D. E. Seborg, “Experimental evaluation of a recursive model identification technique for type 1 diabetes,” *Journal of Diabetes Science and Technology*, vol. 3, no. 5, pp. 1192–1202, Sep 2009.

[15] M. Eren-Oruklu, A. Cinar, D. K. Rollins, and L. Quinn, “Adaptive system identification for estimating future glucose concentrations and hypoglycemia alarms,” *Automatica*, vol. 48, no. 8, pp. 1892–1897, Aug 2012.

[16] K. Turksoy, E. S. Bayrak, L. Quinn, E. Littlejohn, D. Rollins, and A. Cinar, “Hypoglycemia early alarm systems based on multivariable models,” *Industrial & engineering chemistry research*, vol. 52, no. 35, pp. 12 329–12 336, Sep 2013.

[17] E. I. Georga, V. C. Protopappas, D. Polyzos, and D. I. Fotiadis, “A predictive model of subcutaneous glucose concentration in type 1 diabetes based on random forests,” in *34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, San Diego, CA, USA, Aug 2012.

[18] E. I. Georga, V. C. Protopappas, D. Ardigo, M. Marina, I. Zavaroni, D. Polyzos, and D. 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*, vol. 17, no. 1, pp. 71–81, Jan 2013.

[19] E. I. Georga, V. C. Protopappas, D. Polyzos, and D. I. Fotiadis, “Evaluation of short-term predictors of glucose concentration in type 1 diabetes combining feature ranking with regression models,” *Medical and Biological Engineering and Computing*, vol. 53, no. 12, pp. 1305–1318, Dec 2015.

[20] M. Cescon, R. Johansson, and E. Renard, “Subspace-based linear multi-step predictors in type 1 diabetes mellitus,” *Biomedical Signal Processing and Control*, vol. 22, pp. 99–110, Sep 2015.

[21] C. Zecchin, A. Facchinetti, G. Sparacino, G. De Nicolao, and C. Cobelli, “Neural network incorporating meal information improves accuracy of short-time prediction of glucose concentration,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1550–1560, Jun 2012.

[22] C. Zecchin, A. Facchinetti, G. Sparacino, and C. Cobelli, “Jump neural network for online short-time prediction of blood glucose from continuous monitoring sensors and meal information,” *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 144–152, Jan 2014.

[23] C. Zecchin, A. Facchinetti, G. Sparacino, and C. Cobelli, “How much is short-term glucose prediction in type 1 diabetes improved by adding insulin delivery and meal content information to CGM data? A proof-of-concept study,” *Journal of diabetes science and technology*, vol. 10, no. 5, pp. 1149–1160, Sep 2016.

[24] K. Zarkogianni, K. Mitsis, E. Litsa, M.-T. Arredondo, G. Fico, A. Fioravanti, and K. S. Nikita, “Comparative assessment of glucose prediction models for patients with type 1 diabetes mellitus applying sensors for glucose and physical activity monitoring,” *Medical & biological engineering & computing*, vol. 53, no. 12, pp. 1333–1343, Dec 2015.

[25] E. Daskalaki, A. Prountzou, P. Diem, and S. G. Mougiakakou, “Real-time adaptive models for the personalized prediction of glycemic profile in type 1 diabetes patients,” *Diabetes technology & therapeutics*, vol. 14, no. 2, pp. 168–174, Feb 2012.

[26] P. Tkachenko, G. Kriukova, M. Aleksandrova, O. Chertov, E. Renard, and S. V. Pereverzyev, “Prediction of nocturnal hypoglycemia by an aggregation of previously known prediction approaches: proof of concept for clinical application,” *Computer methods and programs in biomedicine*, vol. 134, pp. 179–186, Oct 2016.

[27] I. Contreras, S. Oviedo, M. Vettoretti, R. Visentin, and J. Veh, “Personalized blood glucose prediction: A hybrid approach using grammatical evolution and physiological models,” *PloS one*, vol. 12, no. 11, pp. 1–16, Nov 2017.

[28] R. M. Bergenstal, S. Garg, S. A. Weinzimer, B. A. Buckingham, B. W. Bode, W. V. Tamborlane, and F. R. Kaufman, “Safety of a hybrid closed-loop insulin delivery system in patients with type 1 diabetes,” *Jama*, vol. 316, no. 13, pp. 1407–1408, 2016.

[29] T. Koutny, “Modelling of glucose dynamics for diabetes,” in *International Conference on Bioinformatics and Biomedical Engineering*, Seoul, South Korea, Nov 2017.

[30] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011, vol. 40.

[31] D. Basak, S. Pal, and D. C. Patranabis, “Support vector regression,” *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, Oct 2007.

[32] M. Vettoretti, A. Facchinetti, G. Sparacino, and C. Cobelli, “Type 1 diabetes patient decision simulator for in silico testing safety and effectiveness of insulin treatments,” *IEEE Transactions on Biomedical Engineering*, vol. PP, no. 99, pp. 1–1, Aug 2017.

[33] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, Mar 1995.

[34] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan 1967.

[35] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine Learning*, vol. 29, no. 2, pp. 131–163, Feb 1997.

[36] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[37] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, Jan 2001.

[38] H. Drucker, “Improving regressors using boosting techniques,” in *14th International Conference on Machine Learning (ICML)*, San Francisco, CA, USA, Jul 1997.

[39] A. Facchinetti, S. Del Favero, G. Sparacino, J. R. Castle, W. K. Ward, and C. Cobelli, “Modeling the glucose sensor error,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 620–629, Mar 2014.

[40] A. Facchinetti, S. Del Favero, G. Sparacino, and C. Cobelli, “Model of glucose sensor error components: identification and assessment for new dexcom g4 generation devices,” *Medical & biological engineering & computing*, vol. 53, no. 12, pp. 1259–1269, Dec 2015.

[41] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Butterworth-Heinemann, 1979.

[42] M. Christiansen, T. Bailey, E. Watkins, D. Liljenquist, D. Price, K. Nakamura, R. Boock, and T. Peyser, “A new-generation continuous glucose monitoring system: improved accuracy and reliability compared with a previous-generation system,” *Diabetes Technology and Therapeutics*, vol. 15, no. 10, pp. 881–888, Oct 2013.

[43] T. Hastie, S. Rosset, J. Zhu, and H. Zou, “Multi-class adaboost,” *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, Feb 2009.

[44] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone, “Classification and regression trees,” *Wadsworth*, 1984.

[45] Dexcom Inc. (San Diego, CA). [Online]. Available: https://www.dexcom.com