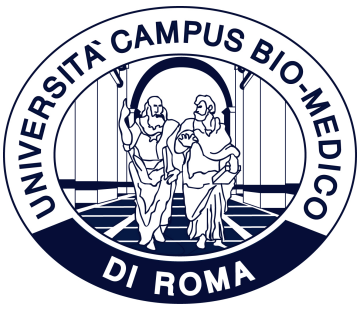


Università Campus Bio-Medico di Roma

Corso di Dottorato di Ricerca in
Scienze e Ingegneria per l'Uomo e l'Ambiente
XXXV ciclo a.a. 2019-2020

Artificial Intelligence Models for the Management of Type 1 Diabetes

Federico D'Antoni

Coordinatore
Prof. Giulio Iannello

Tutori
Dott. Ing. Mario Merone
Prof. Giulio Iannello

13 Marzo 2023

Contents

List of Figures

List of Tables

List of Acronyms

T1D Type 1 Diabetes Mellitus

T2D Type 2 Diabetes Mellitus

CHO carbohydrates

SMBG Self-Monitoring of Blood Glucose

CGM Continuous Glucose Monitoring

RMSE Root Mean Square Error

SSGPE Sum of Squares of Glucose Prediction Error

ARTiDe Auto-Regressive Time Delayed

UTS Univariate Time Series

MTS Multivariate Time Series

AI Artificial Intelligence

IOB Insulin-On-Board

LSTM Long Short-Term Memory

CNN Convolutional Neural Network

RL Reinforcement Learning

PH Prediction Horizon

CEGA Clarke Error Grid Analysis

Abstract

Type 1 Diabetes mellitus (T1D) is a chronic metabolic disease due to which the pancreas is not able to produce an adequate amount of insulin, resulting in an increased blood glucose concentration. If not treated properly, it can lead to short- and long-term complications requiring emergency care and life-threatening conditions. The advent of Continuous Glucose Monitoring (CGM) sensors has considerably improved the management of T1D, as it allows people suffering from this disease to monitor their glycemic levels for 24 hours a day. These sensors are usually coupled with an insulin pump, a device able to continuously provide small amounts of subcutaneous insulin and larger amounts at the patient's request. Since the final decision on glycemic control is taken by the patient, who is a part of the control loop, such a device is defined as a hybrid closed-loop artificial pancreas.

In the last decade, CGM data have been utilized together with Artificial Intelligence (AI) and time-series techniques with the aim of improving T1D management and increasing the quality of life of people with T1D. In this frame, regression is by far the most widely investigated task. In practice, CGM and other features such as injected insulin are given as input to a predictive model in order to forecast future glycemic levels; in this way, the patients are warned in advance of what their blood glucose level is going to be in the next future and are thus able to take the appropriate countermeasures if the glycemia is predicted to exit the target range.

A different approach resorts to classification, in which the AI model is trained to predict whether or not the patient is going to experience an adverse event, without predicting the exact value of the future glycemic level. These studies are relevant because they usually achieve better accuracy with regard to the prediction of hypoglycemic events compared to the regression approach.

While regression and classification limit to provide the patient with a decision support based on the prediction of future glycemic levels or events, leaving to the patient the management of the disease, a third approach focuses on the control of glycemia, i.e., decides what is the optimal amount of medication that must be provided in order to

maintain the blood glucose level within the target range.

This manuscript aims to provide significant and several contributions in the field of the application of AI methodologies to T1D management. With regard to the regression task, a novel neural network is presented for the forecasting on adult patients during daily-life activity, and the comparison of different learning techniques is performed on data of patients during sports; the optimal amount of data for training an AI algorithm for the application on an edge-computing device is investigated; an edge-computing application is developed for the forecasting of glycemic levels of pediatric patients. With regard to the classification task, a layered meta-learning approach is presented for the prediction of hypoglycemic and hyperglycemic events of adult patients during daily-life activity and during sports, and the system is implemented on an edge-computing device. With regard to the control task, a new glycemic closed-loop control based on Dyna-Q is presented that does not necessitate information on carbohydrates, thus not requiring any human intervention and providing a fully closed-loop control.

1. Introduction

1.1. Medical Background

Diabetes mellitus (DM) is a common metabolic disorder characterized by a chronic state of hyperglycemia (increased blood glucose concentration above 160-180 mg/dL), which can be due to inadequate pancreatic insulin production and/or a state of insulin resistance in peripheral cells [1]. Glucose homeostasis is maintained by an intricate balance between different hormonal signals in the body, the most important of which are represented by insulin and glucagon. Insulin (produced by pancreatic β cells) promotes glucose utilization and storage thus lowering glycemia, whereas glucagon (excreted by pancreatic α cells) opposes these actions and promotes glucose production, increasing glycemia [2]. This complex regulatory system maintains blood glycemia within a narrow physiological range, despite pre- and postprandial fluctuations.

There are two broad categories of DM, designated as either type 1 or type 2 DM, which differ in their pathogenesis and clinical management.

- Type 1 Diabetes Mellitus (T1D) is also known as insulin-dependent DM, and is usually diagnosed at a young age. It results from an autoimmune-mediated destruction of the insulin-producing β cells of the pancreas. This leads to a complete or near-total insulin deficiency, with consequent persistent hyperglycemia. The mechanisms that trigger the autoimmune process underlying this disease are still uncertain, however an important role is played by an individual's genetic predisposition as well as infectious or environmental stimuli. The management of T1D relies on patients performing several daily injections of insulin, in order to maintain their blood glucose level within the normal range.

- Type 2 Diabetes Mellitus (T2D) is typical of adult age and represents the most common type of DM (accounting for about 90% of all cases). T2D is characterized by variable degrees of insulin resistance and impaired insulin secretion. It is often related to an elevated Body Mass Index, but also genetic and environmental factors

as well as incorrect lifestyle and stress may play a role in its development. In the case of T2D, insulin injections are not initially necessary: a healthy diet and orally administered drugs are often sufficient to manage glycemia in the first phases of this disease.

Since the long-term complications of DM are related to poor glycemic control, the goal of these patients is to maintain a condition of normoglycemia for as long as possible. While T2D subjects maintain low glucose variability thanks to their residual pancreatic function, patients with T1D often have more difficulty controlling their blood sugar levels, despite optimal medical management [3]. In both cases, blood glucose levels often exceed the euglycemic range, becoming either too high or too low. Hypoglycemia is defined as a blood glucose level lower than 70 mg/dL, and can be caused by excess aerobic activity without the ingestion of an adequate amount of glucose, excessive insulin administration or other diabetes medications [4]. The consequences of severe hypoglycemia can be life-threatening due to its effects on the brain, ranging from mild cognitive impairment to a state of hypoglycemic coma. Hyperglicemia, the underlying metabolic alteration in DM, is equally dangerous. Acute hyperglycemia can be responsible of diabetic ketoacidosis, a condition requiring immediate emergency care as it can also lead to a state of coma. More importantly however, the impact of chronic hyperglycemia can have devastating consequences in the long term, leading to diabetic retinopathy, nephropathy and neuropathy, which cause blindness, renal failure and nerve damage respectively. Macro-vascular complications are related to hyperglycemia as well, i.e. coronary heart disease and stroke. Overall, the glycemic fluctuations that can be seen in patients with DM are responsible for the complications and recurrent hospitalizations that lower patient's quality of life and overall life expectancy [5].

According to the World Health Organization, the number of people suffering from diabetes rose from 108 million in 1980 to 422 million in 2014, whereas in 2019 diabetes was the direct cause of 1.5 million deaths and 48% of all deaths due to diabetes occurred before the age of 70 years [6]. It is considered to be the 8^{th} cause of death worldwide, as it caused more than 1.5 million deaths in 2019.

1.2. Biomedical Background

The most established and used technique to monitor blood glucose concentration is Self-Monitoring of Blood Glucose (SMBG). The most common test for measuring blood glucose involves pricking a finger with a needle to obtain a small drop of blood to be

applied onto a reagent test strip, and determining the glucose concentration by inserting the strip into a measurement device. Different manufacturers use different technologies, but most systems measure an electrical characteristic proportional to the amount of glucose in the blood sample. The measurements are painful and uncomfortable for the patient because of the prick. Reading a hyperglycemic or hypoglycemic value, patients can decide to adjust their glucose levels by injecting an insulin dose, e.g. by an insulin pen, or by ingesting some carbohydrates (CHO) manually, respectively. This adjustment would require, anyway, a further measurement of the blood glucose level after 30 to 60 minutes in order to assess the effectiveness of the adjustment, but this procedure cannot be performed several dozens of times a day, due to its drawbacks. Some of the latest produced devices are able to connect with the user's smartphone in order to register all the information that a patient needs, and can interact with a software that runs on the smartphone. Due to the small amount of samples taken per day, SMBG cannot give complete information on glycemic dynamics, thus the glucose may happen to subtly exceed the safe euglycemic range without the patient's awareness.

In 1999 the first Continuous Glucose Monitoring (CGM) device was approved by the Food and Drug Administration FDA. Such devices overcome the limitations of SMBG and have become very popular and widely adopted by people with diabetes in the last years. A CGM device mainly consists of a sensor and a receiver. The sensor consists in a miniature device placed in the subcutaneous adipose tissue having a miniature needle capable of measuring the blood glucose level in a wide range after being inserted in the subcutaneous tissue through an *ad hoc* clamping device. The receiver is usually a hand-held device which allows the patient to know their glucose level at any moment, by communicating wireless with the sensor.

CGM devices are much more user friendly than SMBG: the sensor is indeed placed in the subcutaneous tissue, thus it is not necessary to perform dozens of needle pricks a day. In other words, this monitoring system is much less invasive than SMBG, because it measures the glucose level in the interstitial fluid rather than in the blood compartment, and only a few finger pricks a day are necessary to calibrate the sensor. Furthermore, most sensors have a lifetime spanning from 7 to 14 days, and are thus able to considerably reduce the number of interventions requested to the patient; plus, in February 2022 the FDA approved a CGM sensor that lasts up to 6 months [7]. Another fundamental feature of CGM systems is that the glucose level is measured in real time with a 1-5 minute sampling period, which makes the glucose monitoring almost continuous and gives a wide outlook on the patient's glycemic excursions during the day. This also allows the

patient to promptly detect a hypoglycemic or hyperglycemic event, in order to quickly take the appropriate countermeasures.

In most cases, CGM devices are supplied with an insulin pump: this combination represents the most recent model of semiautomatic integrated system, also known as sensor-augmented pump (SAP) therapy. Since such a system consists of a monitoring system (the CGM sensor) and a device that provides insulin (the pump), this system is also defined "artificial pancreas". Since these devices close the loop between control and management, they are usually referred to as closed-loop control systems. The insulin pump is a continuously-connected device and is capable of giving insulin doses in a continuous way, plus larger amounts (a bolus) in the event of a meal, imitating physiological pancreatic activity. Insulin is injected through the medium of a catheter and a needle inserted in the patient's subcutaneous tissue. The physician can assess the value of the basal insulin and the carbohydrates-to-insulin ratio of the individual patient. The latter is exploited by devices that are able to automatically calculate the bolus magnitude, according to the amount of CHO that the patient reports to have ingested. This may lead to issues, such as the patient forgetting to insert the bolus with a consequent hyperglycemic event; furthermore, this approach is susceptible to human error, due to user's inability to perfectly calculate the amount of CHO he or she is going to ingest.

Cutting edge insulin pumps, such as the Medtronic MiniMed 670G, are able to automate the basal insulin delivery according to the values read by CGM sensors [8]. Differently, although systems like this are able to suggest the optimal bolus of insulin after knowing the amount of ingested CHO, the final decision on the bolus of insulin to be injected relies on the patient. For this reason, the currently available systems are usually referred to as hybrid closed-loop artificial pancreas, because the patient is still an active part of the control loop. However, in the light of an ever more automatic paradigm of health care, recent research efforts are moving toward the development of devices that are capable of excluding the patient from the control loop; this would provide the first fully closed-loop artificial pancreas, in which the insulin delivery is fully automated [9].

It is important to stress that, despite the fact that both the glycemic measurements and the insulin injections are subcutaneous and thus minimally invasive, a considerable drawback in terms of delay of action rises: both the insulin injection and the glucose measurement happen in the interstitial fluid and, thus, a delay is present compared to the blood dynamics both on the glucose read value and on the action of the injected insulin.

1.3. State of the art of data driven models

Although CGM sensors are widely adopted by people with T1D, hypo- and hyperglycemic events are still frequently reported [10, 11, 12]. This is in contrast with of T1D management itself, the main goal of which is the avoidance, or at least limitation, of adverse events such as hypoglycemia and hyperglycemia. For this reason, in the past two decades many studies have been presented, aiming at the improvement of the management of this disease. In particular, the growing availability of CGM data paved the way to the development of several data-driven models, aimed at the prediction of future glycemic excursions. Indeed, CGM can be regarded as a time series, as it presents temporal sequences of evenly-spaced data points, and thus typical Artificial Intelligence (AI) methodologies related to the field of time-series can be utilized for the analysis. In particular, studies present in the literature concerning the application of AI methodologies to T1D management can be resumed into 3 main categories:

1. Regression: it is the most widely adopted approach [13], and consists in forecasting the exact future glucose level given a prediction horizon. Predicting the future glycemic level in the next 15-30 minutes would allow the patient to take countermeasures in the case the glucose level is forecasted to exceed the target range;

2. Classification: rather than forecasting the exact future glycemic level, this approach limits to predict whether or not the patient is going to experience an adverse glycemic event;

3. Control: this approach aims at providing control strategies to manage insulin infusion, including the development of fully-automated insulin delivery systems.

Regardless of the specific category in which each study falls, they can be further divided according to the amount of features they exploit into:

- Univariate approach: the CGM track is the only feature used;

- Multivariate approach: further features such as injected insulin and CHO are considered in addition to CGM.

A further difference regards the validation procedure adopted by these studies. In particular, 2 major approaches are present:

- Precision medicine aims to develop a predictive model suited to patient-specific data, and, as a consequence, the available data from each patient are split into training and test sets. The training set is used to fit the model on data from one subject, and the test set is used to evaluate performance on other data from the same subject;

- k-fold Cross Validation is a statistical technique consisting in splitting the whole dataset, composed of data from several patients, into k subsets to evaluate model performance on the entire dataset by using, in turns, one fold as the test set and the remaining $k-1$ folds as discovery (training and validation) set. In many cases, the Leave-1-Patient-Out Cross Validation is used, a special kind of k-fold Cross-validation in which each fold consists of all the data of a single subject.

All the models that perform any kind of prediction must first define a Prediction Horizon (PH), i.e. how far forward in time a prediction is performed. In most studies, a PH of 30 minutes is selected, as it would be a sufficient time advance to avoid most adverse events. Each of the 3 main approaches will be described in detail in the next sections.

1.3.1. Regression

In the frame of T1D, the regression task consists in the forecasting of future blood glucose levels. In general, time series forecasting resorts to different types of approaches, including kernel machines, forests of trees, symbolic representation, generative models, and artificial neural networks. Kernel machines and forests of trees can be used for regression tasks, using well-established methods in the literature [14, 15, 16, 17, 18]. Symbolic representation aims at transforming real-time data into symbolic values, and includes methods such as Symbolic Aggregate approXimation (SAX) [19] and Bag-of-Words (BoW) [20]. A generative model is a dynamic model which can be used to generate random outcomes of an observation y given a target value $\tilde{y}$. Such models include Auto-Regressive (AR) models for UTS and Vector Auto-Regressive (VAR) models for MTS forecasting, just to mention a few [21, 22]. Finally, artificial neural networks [23, 24, 25, 26, 27, 28] are mathematical models inspired by the functioning of the human brain and have been used in many applications for both classification and regression purposes. The latter case mainly includes feed-forward neural networks, Nonlinear Auto-Regressive with eXogenous input (NARX) neural networks, and recurrent neural networks.

With the specific application to T1D, over the last decade, many algorithms belonging to all the aforementioned approaches have been developed to help patients keep their blood glucose levels as constant as possible. Most models only focus on T1D patients, whereas the literature concerning predictions on T2D patients is scarce.

Different metrics are used to evaluate forecasting performance. Most studies present their performance in terms of Root Mean Square Error (RMSE). The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}[y(t_i + PH) - \tilde{y}(t_i + PH|t_i)]^2}{N}} \tag{1.1}$$

where N stands for the number of timestamps in the validation set, t_i denotes the i-th timestamp, and $y(t_i + PH)$ represents the true value given a prediction horizon PH. Considering that such a metric refers to an error, the smaller the value, the better the model performance. RMSE returns an error that strongly depends on the order of magnitude of the analyzed quantity, thus, some studies investigated Sum of Squares of Glucose Prediction Error (SSGPE) reported in [29] as:

$$SSGPE = \sqrt{\frac{SSE}{\sum_{i=1}^{N-PH}[y(t_i + PH)]^2}} \tag{1.2}$$

which returns a percentage error score, and thus gives quantitative information about the prediction error which is independent of the order of magnitude of the analyzed quantity. It is based on the Sum of Squared Errors (SSE) defined in equation 4.2. Similarly, another used metric is the Mean Absolute Relative Difference (MARD) defined as:

$$MARD = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \tilde{y}_i}{y_i}\right| \tag{1.3}$$

where $\tilde{y}_i$ and y_i are the value predicted by the network and the real value at the i-th timestamp for a sequence of N samples, respectively. In addition, we considered the Clarke Error Grid Analysis (CEGA) as a measure of the clinical accuracy of the predictions [30]. The CEGA is based on a grid that is divided into 5 zones, from A to E, and provides a plot of the actual and the predicted CGM values on the horizontal and the vertical plot axis, respectively. Values in zones A and B represent good or acceptable glucose predictions; values in zone C represent mistaken predictions that may lead to unnecessary treatment; values in zone D represent a dangerous failure to predict; finally, values in zone E represent a completely wrong prediction that would lead to erroneous treatment. In brief, a good performance corresponds to small values of RMSE, SSGPE,

and MARD, and to a small percentage of values in the C-D-E zones of the Clarke Error Grid.

In the remainder of this section are listed some works which achieved remarkable results resorting to a regression approach, and their main features are reported in Table 1.1. In the frame of kernel machines, Bunescu et al. [14] used a three-compartmental physiological model of blood glucose dynamics to generate features for a Support Vector Regressor (SVR) that is trained on patient-specific data. The physiological model exploits data concerning CGM, insulin, and meal intake to simulate the glucose dynamics. The model is validated on data from 5 T1D patients from a private dataset. The experiments carried out to forecast blood glucose levels with a 30- and 60-minute prediction horizon attain RMSE values equal to 22.6 mg/dL and 35.8 mg/dL, respectively. Hamdi et al. [15] fed an SVR with CGM data after identifying the optimal model parameters through the medium of a differential evolution algorithm. They take into account data of 12 T1D patients from a private dataset, attaining RMSE values equal to 9.4 ± 3.7 mg/dL and 10.8 ± 3.9 mg/dL for prediction horizons of 15 and 30 minutes.

With regard to the forests of trees, Georga et al. [16] presented a personalized predictive model of the glucose concentration in T1D patients which employs the Random Forests regression technique. This multivariate model takes as input CGM data, physiological features, and lifestyle information. Experiments were carried out on data from 27 T1D patients from a private dataset. High-accuracy predictions are derived in case all the available features are used with a multivariate approach (RMSE $= 6.6 \pm 1.3\, mg/dL$ for 15-minute prediction horizon), whereas the performance considerably deteriorates when using a univariate approach (RMSE $= 11.3 \pm 2.2\, mg/dL$). Midroni et al. [17] combined an XGBoost expanded Random Forest Regression model with feature-engineering methods to predict blood glucose levels. After observing time-dependent patterns in the data, they include features concerning hour-of-day and day-of-week. The model was validated using features including CGM data, physical activity, insulin, and CHO intake of patients from the Ohio T1DM dataset [11], i.e. a publicly available dataset consisting of 6 T1D patients, which is described in section 4.1.2. They achieve an RMSE value of 20.4 ± 2.4 mg/dL considering a 30-minute prediction horizon.

In the frame of symbolic representation, Contreras et al. [20] used a search-based algorithm to generate models capable of capturing the dynamics of blood glucose at a personalized patient level. The grammar-based feature generation enables the construction of empirical models using data gathered from a CGM sensor, the glucose dynamics, and the daily energy expenditure. Raw data are pre-processed using three different

physiological models to simulate the effects of insulin, glucose absorption, and physical activity. The system requires the definition of a problem-specific objective function (RMSE is selected), which evaluates the solutions, and a customized grammar, which defines the structure of the generated prediction. It iteratively combines solutions to incrementally improve the prediction and to reach a final solution that minimizes the objective function. The model is tested on the Ohio T1DM dataset [11] and provides blood glucose levels estimations using prediction horizons of 30, 60, and 90 minutes, attaining RMSE values equal to 21.2, 31.3, and 36.3 mg/dL, respectively.

Generative models are used to cope with the intra-subject variability characterizing glucose dynamics. Authors adopt static (i.e. the model is trained once to perform tests on the incoming data) or dynamic (i.e. the model is identified recursively every time new data income) auto-regressive models, assigning a relative weight to a limited number of past data, which depends on the selected forgetting factor. In an early study, Reifman et al. [21] proposed a time-invariant AR model of order 10. Parameters are optimized using regularized least squares, considering prediction horizons of 30 minutes. CGM data of 9 T1D subjects from a private dataset are used to fit the model. Both subject-specific and subject-invariant models are evaluated: subject-specific models (RMSE = $22.3 \pm 3.9 mg/dL$) achieve better results than inter-subject models (RMSE = $24.8 \pm 3.2 mg/dL$), confirming the effectiveness of precision medicine in diabetes management. Sparacino et al. [22] propose a first-order AR model with time-varying parameters, which are estimated at each timestamp using recursive least squares. They test several values of the forgetting factor with prediction horizons of 30 and 45 minutes. The model is identified on CGM data of 28 T1D patients from a private dataset. Results are accurate enough to potentially avoid or mitigate critical hypo- and hyperglycemic events (RMSE = 18.3 ± 11.8 and $34.9 \pm 21.3\, mg/dL$).

Artificial neural networks have recently been proposed as an alternative approach for time series forecasting. The main advantage of artificial neural networks is their flexible nonlinear modeling ability [31]. Indeed, although traditional generative models generally achieve good results in time series forecasting, they are linear in the prediction of future values, and thus they find it difficult to capture nonlinear behaviors that can be observed in many real-world applications. Within this context, Zecchin et al. [23] proposed a jump neural network exploiting as input the CGM data, its recent trend, the glucose rate of appearance in blood, and its derivative. This multivariate model consists in a feed-forward neural network with the addition of direct connections from each input to the output neuron, combining linear and nonlinear connections between

input and output in a single structure. This allows the simulation of the linear and nonlinear dependencies of the output from the input using a rather simple model. The jump neural network is tuned on data of 10 T1D patients and then assessed on 10 different subjects from a private dataset. Experiments predict future blood glucose levels up to 30 minutes ahead of time (RMSE = $16.6 \pm 3.1\ mg/dL$). Martinsson et al. [24] proposed a Long Short-Term Memory (LSTM) model, which pursues a UTS approach exploiting only the CGM and its recent trend. An LSTM is a recurrent neural network where the cell at each timestamp contains an internal memory vector and three gates controlling what parts of the internal memory will be kept (the forget gate), what parts of the input will be stored in the internal memory (the input gate), and what will be included in the output (the output gate). The model is developed using data from patients in the Ohio T1DM dataset [11] to predict blood glucose levels up to 30 minutes ahead of time (RMSE = $20.1 \pm 2.5\, mg/dL$). Zhu et al. [25] used a Convolutional Neural Network (CNN) to forecast blood glucose levels. The authors put the change of glucose values in 256 categories as a target, with a difference of $1\ mg/dL$ between each class, through the medium of quantization. The input features of the neural network are the CGM values, the CHO intake, the insulin events, and the time index. They predict blood glucose levels of patients from the Ohio T1DM dataset [11] with a 30-minute prediction horizon (RMSE = $22.1 \pm 2.5\ mg/dL$). Chen et al. [26] proposed a dilated recurrent neural network comprising multi-resolution, recurrent skip connections, allowing the network to learn different temporal dependencies at different layers. This model exploits the previous knowledge of CGM data, insulin doses, and CHO intake from the past 60 minutes. Tests were performed on the Ohio T1DM dataset [11] considering a 30-minute prediction horizon (RMSE = $19.0 \pm 2.6\, mg/dL$). Bertachi et al. [27] proposed a feed-forward neural network with 8 neurons in its hidden layer, which takes as input information on CGM data, ingested CHO, injected insulin, and physical activity. Predictions 30 and 60 minutes ahead of time are performed on patients from the Ohio T1DM dataset [11], attaining RMSE values equal to $19.3\, mg/dL$ and $31.7\, mg/dL$, respectively. Li et al. [28] present GluNet, a dilated convolutional neural network capable of processing multidimensional long signals concerning CGM data, insulin, meal information, and lifestyle factors. The model was tuned on data from 20 virtual patients generated from the UVA/Padova T1D simulator [32] and validated on both a private dataset and the Ohio T1DM dataset [11]. In the latter case, prediction horizons of 30 and 60 minutes produced RMSE values of $19.3 \pm 2.8\ mg/dL$ and $31.8 \pm 3.5\ mg/dL$.

In recent years, models that combine multiple types of neural networks have been

Table 1.1.: Previous works in the literature exploiting different machine learning approaches for blood glucose levels forecasting with a regression approach. KM: Kernel Machine, FT: Forest of Trees, SR: Symbolic Representation, GM: Generative Model, NN: Neural Network, Multi-NN: multiple types of neural networks. Univariate (UTS) and multivariate (MTS) time series approaches are highlighted, together with the number of patients involved and the type of dataset.

Main author	Model	Approach	UTS/MTS	T1D patients/Dataset
Bunescu [14]	SVR	KM	MTS	5/private
Hamdi [15]	SVR + DE	KM	UTS	12/ private
Georga [16]	Random Forest Regression	FT	MTS	27/private
Midroni [17]	XGBoost Random Forest	FT	MTS	6/Ohio T1DM [11]
Contreras [20]	Search-based algorithm	SR	MTS	6/Ohio T1DM [11]
Reifman [21]	Auto-regressive model	GM	UTS	9/private
Sparacino [22]	Auto-regressive model	GM	UTS	28/private
Zecchin [23]	Jump NN	NN	MTS	20/private
Martinsson [24]	Long-Short Term Memory	NN	UTS	6/Ohio T1DM [11]
Zhu [25]	CNN	NN	MTS	6/Ohio T1DM [11]
Chen [26]	Dilated Recurrent NN	NN	MTS	6/Ohio T1DM [11]
Bertachi [27]	Feed-Forward NN	NN	MTS	6/Ohio T1DM [11]
Li [28]	Dilated Convolutional NN	NN	MTS	16/ private, Ohio T1DM [11]
Kalita [33]	LSTM-GRU	Multi-NN	MTS	UVA/Padova simulator [32]
Jaloli [34]	CNN-LSTM	Multi-NN	MTS	Replace-BG [35], DIAdvisor
Lu [36]	Stacked MLP, Bi-GRU, RNN +AM	Multi-NN	UTS	RT_CGM [37]

presented to catch deeper relations between the input and the desired output. Kalita et al. [33] fed an LSTM with Gated Recurrent Units (GRU) with CGM data and meal information of in silico patients from the UVA/Padova simulator [32] for predictions 15 and 30 minutes ahead of time, achieving an average RMSE of 5.27 and 14.85 mg/dL, and an average MARD of 2.9 and 8.4%, respectively. Jaloli et al. [34] presented a stacked CNN-LSTM neural network with CGM, meal information, and insulin intakes of patients from two datasets, namely Replace-BG [35] (free-living conditions) and DIAdvisor (intensive care), to forecast blood glucose levels with a PH of 30, 60, and 90 minutes, achieving an average RMSE for the longest PH of 23.45 ± 3.18 and $25.12 \pm 4.65\, mg/dL$, respectively. Finally, [36] proposed a method to forecast future blood glucose levels with a 30-minute PH with a hybrid deep learning model, which integrates multi-layer perceptron (MLP), stacked bidirectional gated recurrent unit (Bi-GRU) based recurrent neural network (RNN), and the attention mechanism (AM), fed with CGM data of real patients in the RT_CGM dataset [37] with a precision-medicine approach, and achieved an average RMSE of 11.76 mg/dL.

1.3.2. Classification

A different approach resorts to the classification task, in the sense that the model aims at predicting whether or not the patient will experience an adverse event within the time window defined by the PH, regardless of the exact future glycemic level. The need for this approach arises from the limitations observed for the regression approach, as it has been proven by recent studies that predicting adverse glycemic events using classification rather than regression leads to improved performance [29, 38]. As in other classification tasks, the performance is evaluated in terms of total accuracy and of metrics derived from the confusion matrix, namely Recall and Precision per class, defined as:

$$Recall = TP/(TP + FN) \qquad Precision = TP/(TP + FP) \tag{1.4}$$

where TP, FP and FN are the total numbers of true positives, false positives, and false negatives per class. Usually, these studies consider two main classes to be predicted, according to the read CGM value, namely hypoglycemia and hyperglycemia, although some studies take into account further thresholds (e.g., severe hypoglycemia if $CGM \leq 50\,mg/dL$) or only focus on predicting a specific event (e.g., postprandial hypoglycemia). For example, the vast majority of studies focus only on the prediction of hypoglycemia [39, 40, 41, 42, 43, 44]. It is a sensible choice because this condition can arrive unannounced also in severe cases, leading to serious short-term complications. In this regard, in a recent review on machine learning techniques for hypoglycemia prediction, Mujahid et al. [42] stated that *is important to understand that hypoglycemia prediction is blood glucose level prediction in essence.* Nonetheless, most of such works mainly aim at maximizing the true positive rate at the expense of a considerably low precision score, which is often not reported [39, 40] or impossible to compute [45, 46, 38, 47, 41, 48]. Indeed, it is acknowledged that any prediction algorithm has to "decide" between raising a lot of alerts to detect all events (good recall, bad precision, a lot of false positives) or trying to minimize the nuisance of the patient (good precision, limited false positives, at the expense of a lower recall). Works focusing on hypoglycemia prediction usually choose the former approach [49], with few exceptions [47]. It reduces patient engagement with the technology. Two main approaches fall into this area:

- sample-based prediction [39, 40, 41, 47] in which, at each timestamp, a prediction is performed according to the PH; in this way, each sample is classified, and the model performance is evaluated based on the predictions performed for all the timestamps;

Table 1.2.: State of the art of the glycemic events prediction task with a classification approach. For each work, we report the main author together with the number of patients in the dataset and the validation strategy, the adopted model, the specific sample-based or event-based approach, and, where available, the average classification Recall (R) and Precision (P) of predictions up to 30 minutes ahead of time for the classes hypoglycemia (Hypo), normoglycemia (Norm) and hyperglycemia (Hyper). We mark as not available (n/a) the performance values that were not reported and are not possible to compute.

First author	# Patients	Validation	Model	Approach	Results [%]			
						Hypo	Norm	Hyper
Gadaleta [29]	89	Leave-1-Patient-Out	SVM	Event-Based	R	86	n/a	95
					P	36	n/a	56
Daskalaki [45]	23	Precision Medicine	cARX + RNN	Event-Based	R	100	n/a	100
					P	n/a	n/a	n/a
Cappon [46]	100 in silico	Precision Medicine	XGBoosted Tree	Sample-Based	R	92	76	86
					P	n/a	n/a	n/a
Seo [39]	104	5-fold Cross Validation	Random Forest	Sample-Based	R	89.6	91.3	n/a
					P	38.9	n/a	n/a
Dave [40]	112	10-fold Cross Validation	Random Forest day + Random Forest night	Sample-Based	R	93.7	94.4	n/a
					P	15.1	99.8	n/a
Marcus [47]	11	Precision Medicine	Kernel Ridge Regression	Sample-Based	R	64	96	61
					P	n/a	n/a	n/a
Cichosz [41]	10	Precision Medicine	Linear Logistic Regression	Sample-Based	R	79	99	n/a
					P	n/a	n/a	n/a
Yang [38]	124	Precision Medicine	Long Short-Term Memory (LSTM) classifier	Event-Based	R	92.6	92.5	n/a
					P	n/a	n/a	n/a
Prendin [48]	112	Precision Medicine	Autoregressive Integrated Moving Average (ARIMA)	Event-Based	R	82	n/a	n/a
					P	64	n/a	n/a
Jensen [43]	463	5-fold Cross Validation	Linear Discriminant Analysis (LDA) classifier	Sample-Based	R	73	75	n/a
					P	22	97	n/a
Zhu [50]	49	Holdout Validation	Bidirectional RNN with meta-learning	Event-Based	R	84.1	n/a	n/a
					P	65.6	n/a	n/a
D'Antoni [51]	33	Precision Medicine	ARTiDe Jump NN	Event-Based	R	59.8	n/a	47.2
					P	86.4	n/a	58.0

- event-based prediction [46, 29, 48], in which consecutive timestamps of hypo- or hyperglycemia are considered as a single event; a prediction of an event is considered a true positive if an actual event occurs in the next minutes.

A summary of the state of the art of glycemic events prediction is reported in Table 1.2.

1.3.3. Control

Although regression and classification AI models can provide a reliable prediction for patients with a good balance between performance and applicability [49], their functioning is different from what a controller is expected to do. In practice, they advise patients to intervene directly to prevent adverse events by ingesting CHO or providing a bolus of insulin, sometimes providing the optimal amounts to reach a target glucose

value [52]. This is a limitation when aiming at the development of a fully-automated artificial pancreas system. Consequently, another branch of the research has focused on blood glucose control strategies. Typical T1D control algorithms are based on mathematical models [53] that depend on output error and its proportional-integral-derivative (PID) behavior [54], or on lookup-table and rule-based control [55]. However, in recent years Reinforcement Learning (RL) techniques have become more and more popular for T1D control. Such algorithms include an agent which, after training, can decide the optimal action to perform in order to generate a modification in the environment that will provide a reward as large as possible. In the case of T1D, the environment consists of the patient or a simulator, whereas the action typically consists of an insulin bolus or a glucagon injection; however, the latter is limited in practice by the fact that most insulin pumps limit to injecting insulin without the possibility of injecting glucagon. A control system able to inject both insulin and glucagon takes the name of "dual-hormone control". By performing an action on the environment and by computing the correspondent reward, the agent can learn a policy that can minimize a given loss function [56]. Nonetheless, RL agents typically start with a poor understanding of the environment. This limits the applicability of such models because an exploration of the environment based on poor initial decisions (i.e., insulin boluses) could result in catastrophic consequences for the patients.

RL algorithms can be roughly split into model-free and model-based algorithms, depending on whether the agent provides its action directly on the environment - without a model of it - or on a model of it, respectively. To date, almost all the studies that apply RL techniques to T1D use model-free algorithms [57, 58, 59], also known as Direct RL [60], in which the agent is trained through trial-and-error processes directly on the environment (i.e., the patient), without knowing in advance how it will change as a result of the action performed. On the one hand, such algorithms do not include any kind of model that simulates the behavior of the environment, and have thus the advantage of being simple to implement and optimize; on the other hand, two main issues limit the application of these algorithms on real devices: the large amount of data required for efficient training, and the difficulty in replicating a trial-and-error learning process on a real patient.

Since the main goal of the control algorithms is to maintain the blood glucose levels in the euglycemic range - or a custom target range - as long as possible, the time in range (TIR) is often considered as a performance metric to evaluate the goodness of a

control system, and it is usually defined as:

$$TIR = \frac{time \text{ s.t. } 70 \leq CGM \leq 180}{time_{tot}} \tag{1.5}$$

nonetheless, since any control algorithm mustn't generate severe hypoglycemic events, other custom metrics such as the number of hypoglycemic events generated are often used to evaluate performance too.

As mentioned, almost all studies that apply RL techniques to diabetes use model-free algorithms, in which the agent is trained through trial-and-error processes directly on the environment, without knowing in advance how it will change as a result of the action performed [61]. Indeed, such algorithms do not include any kind of model that simulates the behavior of the environment and, consequently, they have the advantage of being simple to implement and optimize. In particular, Zhu et al. [62] presented a deep RL approach for both single-hormone (insulin) and dual-hormone (insulin and glucagon) basal delivery. They used a double deep Q-learning with deep recurrent neural networks, able to capture the complexity of glucose-insulin-glucagon dynamics due to its enlarged receptive field. CGM values, ingested CHO, insulin, and glucagon data are the state variables utilized to describe the environment. The reward function is a piecewise linear function dependent on blood glucose levels. The same authors also proposed a personalized bolus calculator intending to provide more patient-specific recommendations by exploiting the Deep Deterministic Policy Gradient (DDPG) algorithm [58]. This algorithm allows working with continuous action space and state variables. On the contrary, the previous Double Q-Learning requires a discrete set of actions, still admitting a continuous environment state variables description. Again, the proposed reward function, reported in equation 9.4, is a discrete function, with a higher penalization for actions leading to hypoglycemia (glucose $< 70\, mg/dL$), and the state variables adopted are CGM, insulin data, and CHO. Although these studies show a quite good performance, with a TIR of 80.9%, it is worth noting that both use the amount of CHO ingested by the patient as a state variable to describe the environment. Such models are also referred to as hybrid closed-loop models because human intervention is necessary. However, with a view to a fully-automated closed-loop artificial pancreas, information on CHO should not be provided: in a real-life application, the patient would be asked to manually enter such a value, which would defeat the purpose of closed-loop control of the device. In this context, a step forward is achieved by Fox et al. [57], who exploited only CGM and insulin data as state variables of the environment, thus requiring no meal announcement.

They used a model-free Soft Actor-Critic (SAC) RL algorithm, which allows modeling both actions and state variables as continuous, and optimizes a stochastic policy in an off-policy way, forming a bridge between simple Actor-Critic methods (stochastic and on-policy) and DDPG (deterministic and off-policy). Another novelty introduced in this study is the reward function used to train the agent, which is the Magni risk function. Specifically, the authors develop several variants of the RL method: RL from scratch, where the patient-specific algorithm is trained from scratch for each individual; RL transfer, which fine-tunes an RL-Scratch model previously trained on data from an arbitrary child/adolescent/adult; RL-MA, which uses RL-Scratch trained using the automated meal boluses from the bolus calculator or PID controller; Oracle architecture, which replaces observed states with ground-truth states returned by the UVA/Padova simulator. Among these, the best performance (about 78.8% TIR) is achieved by the Oracle model; nonetheless, such a model could not be used in reality, because the ground truth would not be available. The second best model is RL-MA, which requires the meal announcement for the bolus calculator to generate the optimal boluses and, therefore, would not be suitable for closed-loop control. In addition, the two models Scratch and Trans RL achieve TIR of about 72% and 71%, respectively, only after a very long training phase, which includes more than 2 years of data for RL-Trans and up to 16.5 years for RL-Scratch, which would make their real application practically impossible.

However, since in the matter of glycemic control the environment is exactly the patient, it would be highly difficult and extremely unsafe to replicate a trial-and-error learning process on a real person. This is why in our opinion model-based RL algorithms, i.e., based on an environment modeling and planning strategy to take an action [61], would be more appropriate. To the best of our knowledge, in the literature, a unique attempt has been made to combine model-based RL techniques with glycemic control. In particular, the model from Yamagata et al. [63] relies on an ensemble system of Echo State Networks (ESNs) to predict blood glucose levels, coupled with a Model Predictive Controller (MPC) for planning. In other words, ESNs predict blood glucose levels, and MPC generates the insulin dose suggestion as a consequence of blood glucose levels predicted by the ESNs. The results achieved by the study suggest that model-based RL can perform equally or better than the model-free approach while considerably reducing the risks for the patient. However, the main limitation of such an architecture is that it uses information on CHO intake as input for the ESNs, as well as the insulin data, to predict blood glucose levels, which makes the algorithm impossible to be embedded in an actual closed-loop artificial pancreas.

1.4. Edge computing application

The increasing development of new, more powerful, dedicated hardware enables the emergence of a branch of artificial intelligence known as inference at the edge [64, 65]. It involves the machine learning models being run directly from a proximity device using data collected from associated sensors. With the growing interest in the telemedicine approach [66, 67], the inference at the edge can enable predictive models that work in real-time with patient data to improve both medical quality and efficiency. For this reason, to date, several works exploit the potential of edge computing not only from a more methodological and general point of view (e.g., [68]) but also in the field of glycemic level prediction. Zhu et al. [69], for example, proposed an Embedded Edge Evidential Neural Network to predict future glycemic levels of adult T1D patients in real-time by exploiting CGM sensor readings and an edge-computing device.

Although all the predictive algorithms described in the previous sections could run on the Cloud, this would introduce some practical limitations, as the patients would be requested to have a device that is constantly connected to the internet. In this respect, an eventual disconnection or lack of service would leave the patient without a predictive algorithm available. An interruption of the service must be avoided when applied to medical devices. In addition, a time delay is introduced for web communication. Differently, performing a prediction at the edge, i.e. as close as possible to where the data are gathered, considerably reduces the delays in communication, and eliminates the risk related to a no-service area. On the other hand, this approach introduces other challenges, such as the necessity to identify the target hardware able to run complex deep learning algorithms in a reasonable time while limiting the costs, and the need to find a way to work around the hardware limitations and the memory limits of edge-computing devices.

1.5. Motivations

Despite the noteworthy step forwards in the biomedical devices for T1D and the huge advances made by AI to aid subjects with T1D manage their condition in daily life, some major limitations still exist that prevent the widespread application of such systems among patients, which concern both the hardware and the software.

Concerning the hardware limitations, although CGM devices have considerably improved the quality of life of people with T1D, they present some major drawbacks that

need to be addressed, including:

- Accuracy and reliability: the measurement of blood glucose in the interstitial fluid is not as accurate as traditional blood glucose monitoring methods including finger pricking, and includes a measurement delay in case of rapid variations; moreover, like many battery-powered devices, they may occur to run out of battery charge, or may also interfere with other wireless devices;
- Cost: CGM devices can be expensive, and not everyone can afford them. In the United States, the cost of a CGM sensor without insurance coverage can range from around 50 to 150$ per sensor, depending on the brand and type, and usually last between 7 and 14 days. This can be a significant barrier for people with T1D who need these devices to manage their condition, especially considering that even patients with years of experience happen to puncture a blood vessel when applying a new CGM sensor, making it unusable and thus wasting that money.
- Privacy and Security: they collect a lot of personal sensitive data that could be hacked;
- User experience: CGM devices can be uncomfortable to wear and require frequent calibration and maintenance. This can make them a burden for some users, who may be less likely to use them consistently as a result.

Concerning the software, although AI systems are promising for the management of T1D, they present the following drawbacks:

- Data availability: AI systems rely on large amounts of data for training. However, data can be difficult to collect, as people with T1D typically need to manually measure their blood glucose levels and intake. This can make it challenging to collect the necessary data to train and improve AI systems;
- Variability in disease progression: T1D is a highly variable disease, and its progression can differ significantly between individuals. This can make it difficult for AI systems to make accurate predictions and recommendations, as they may not be able to account for all of the individual differences between people with T1D;
- Complexity of insulin dosing: insulin dosing for T1D can be complex, as it requires careful consideration of a variety of factors, including diet, exercise, and stress levels. AI systems may not be able to take all of these factors into account when making recommendations;

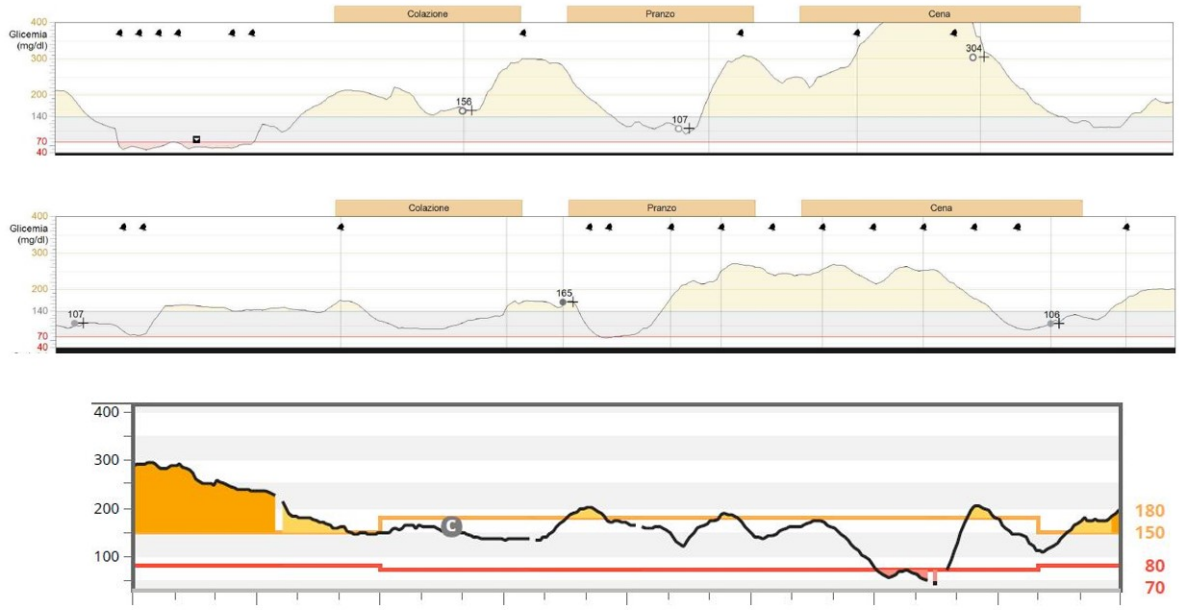


Figure 1.1.: Examples of a 1-day monitoring (0-24) of 3 real patients from the UCBM dataset (see Chapter 4.1.1).

- Lack of regulatory approval: while many AI systems have shown promise for T1D management, they have rarely been approved by regulatory agencies;
- Technical limitations: AI systems can be limited by the technology used to implement them. For example, if the sensors used to measure blood glucose levels are not accurate or reliable, this can limit the effectiveness of the AI system that relies on these measurements. Also, efficient and effective wearable technology is necessary to implement such algorithms for edge computing.

The current limitations of T1D management may result in sub-optimal glycemic management like the one that is observed in Figure 1.1. It reports as an example one day of CGM measurements of 3 subjects from the UCBM dataset (see Chapter 4.1.1). The first 2 patients exploit the Medtronic 640G system, whereas the third exploits Dexcom G5. As can be observed, the first patient experienced, within the same day, prolonged nocturnal hypoglycemia and severe daytime hyperglycemia; the second patient experienced a quick glycemia decrease from hyperglycemia to the hypoglycemic threshold after lunch, which resulted in successive prolonged hyperglycemia during the afternoon; the third patient experienced nocturnal hyperglycemia, hypoglycemia at dinner time, and several sensor disconnections. Many of these undesired events could have been

avoided or mitigated by the utilization of effective AI systems joined with the CGM sensor.

2. Contributions

This thesis presents contributions in each field of AI application in T1D management described in the previous sections. It aims to address different challenges that are relevant to T1D management and specifically 4 of the major software limitations described in the previous chapter, which are data availability, variability in disease progression, the complexity of insulin dosing, and technical limitations. Each contribution will be thoroughly described in the next chapters.

Concerning the regression task, Chapter 3 describes an early development of a novel artificial neural network for forecasting future glycemic levels using regression and a precision-medicine approach. The proposed neural network only necessitates 24 hours of CGM data for training, thus not requiring the patients to gather several heterogeneous features. Although this model requires a much smaller amount of data for training than others reported in the literature, it achieves better predictive performance. Chapter 4 reports an extension of the previous study by including a larger cohort of patients, one of whom suffers from T2D and two of whom performed physical activity during the monitored period, by comparing the results to those achieved using well-established methods in the literature, and by evaluating performance in terms of event detection. Chapter 5 presents an analysis of different learning techniques for training a neural network on data of T1D patients that regularly perform physical activity. Offline training, online training, and online training with a penalty are compared to assess what approach performs better when facing the particularly difficult task of forecasting glycemic levels during sports with a precision-medicine approach. It is observed that the improvement in performance generated by continuously updating the model with the most recent data is not so pronounced as to justify the notable increase in the computational burden. Chapter 6 presents a study that aims at the identification of the optimal amount of training data for the implementation of a glucose levels forecasting model on an edge device, exploiting up to 100 days of data from 6 virtual patients and taking into account both numerical and computational resources with a precision-medicine approach. The investigated model achieves state-of-the-art performance when trained with 1 day of data,

while the RMSE slowly decreases as the size of the training set increases; a plateau in predictive performance is observed when more than 60 days of data are used for training, and increasing the training set always produces better predictions than just re-training the model with the most recent available data. The time necessary for predictions on an edge-computing device is far below the time constraints imposed by the specific task. Chapter 7 presents a study focused on data of virtual pediatric T1D patients, consisting in the application of two state-of-the-art deep neural networks for time-series forecasting, and on the analysis of the performance achieved when running the neural networks on two edge-computing devices that utilize different data representations. The analyzed deep networks outperform models in the literature in terms of clinical accuracy, and the performance does not decrease considerably when running tests on edge devices.

With regard to the classification task, Chapter 8 presents a meta-learning approach based on a multi-expert predictive model (a CNN and a LSTM) relying on an event-based approach that is capable of achieving a trade-off between the number of predicted events and the number of false alarms. The model is validated with a Leave-1-Patient-Out-Cross-Validation on a public dataset composed of 12 patients, and on a private dataset composed of 5 patients who regularly perform physical activity. The approach is evaluated in terms of classification performance for predictions made with a time advance ranging from 5 to 30 minutes; it exploits techniques for imbalanced datasets and pursues a univariate approach by utilizing only CGM data. The proposed approach outperforms other models in the literature, and the meta-learning strategy improves performance considerably compared to using only one predictive system. Finally, the models are implemented on an edge-computing device in order to evaluate the real-life feasibility and applicability of the proposed approach.

With regard to the control task, Chapter 9 presents a fully-closed loop control based on a Dyna-Q RL algorithm that uses a predictive model as a model of the environment (i.e., the patient). Different learning techniques are investigated for the predictor. The whole model takes into account information concerning only CGM and injected insulin, which are automatically recorded by the sensor and the insulin pump; thus, the patient is never asked to manually provide information. A deep-Q-Network is trained as a controller to decide the optimal amount of insulin to be injected. The proposed system is capable of achieving a noteworthy TIR without exploiting information on meals, while producing a very limited amount of hypoglycemic events, and outperforms a realistic manual control.

In order to ease the understanding of the contribution of this thesis compared to the

literature, Table 2.1 reports the state-of-the-art methods for each of the tasks described in sections 1.3.1, 1.3.2, and 1.3.3, together with the dataset they utilized, their points of strength and main limitation.

Briefly, the main contributions of this thesis can be summarized as follows:

- Several models have been developed for the forecasting of glucose levels of adult and pediatric patients with a regression task. The investigation was extended to the identification of the optimal amount of data and training strategy for the predictive models;
- A meta-learning approach has been developed for the event-based prediction of hypoglycemia and hyperglycemia that outperforms pre-existing models;
- A model-based RL algorithm has been utilized for developing a fully closed-loop controller that does not require intervention from the patient.

Finally, it is worth stressing that the studies presented in Chapters 6, 7 and 8 investigated the feasibility and the applicability of the discussed models on edge-computing devices, tackling the drawbacks related to the inference at the edge using different techniques.

Table 2.1.: Methods at the state of the art for the tasks of regression, classification, and control related to T1D.

First author	Task	# patients and dataset	Model	Results	Strengths	Limitations
Kalita [33]	Regression	10, UVA/Padova	LSTM-GRU	RMSE=5.27 mg/dL for PH=15' RMSE=14.85 mg/dL for PH=30'	Good numerical results	Only tested on in silico patients
Zecchin [23]	Regression	20, Private	Jump NN	RMSE=16.6 mg/dL for PH=30'	Easy and intuitive model, Effective on real patients	Multivariate approach, Unable to catch sudden changes
Prendin [48]	Classification	112, Private	ARIMA	Hypoglycemia Recall and Precision: 82% and 64%	High recall, few false alarms, univariate	Doesn't consider hyperglycemia, Low time gain
Zhu [50]	Classification	49, Ohio T1DM, ARISES, and ABC4D	Bidirectional RNN and meta-learning	Hypoglycemia Recall and Precision = 84.1% and 65.6%	High recall, Few false alarms	Doesn't consider hyperglycemia, requires a lot of data, multivariate
Zhu [62]	Control	30, UVA/Padova	Q-learning with dilated RNN	85.6% Time in range	Good control for single- and double-hormone therapy	Dual-hormone delivery system, Multivariate

3. Early experience in forecasting blood glucose levels using a delayed and auto-regressive jump neural network

Many people with diabetes face the daily challenge of maintaining blood glucose levels in the physiological (euglycemic) range, avoiding adverse events such as hyperglycemia or hypoglycemia, which may lead to short- and long-term complications. To avoid their side effects and help maintain the blood glucose level as constant as possible, research efforts in the last decade have been directed towards the development of either machine learning-based algorithms [16, 70, 23, 15, 24] or physiological-based models [71, 14] capable of forecasting future glycemic levels via a regression task. Within this context, a study from Zecchin et al. [23] presented a jump neural network exploiting as input the CGM data and its recent trend, the glucose rate of appearance in blood, and its derivative. This multivariate model consists in a feed-forward neural network with the addition of direct connections from the inputs to the output, combining linear and nonlinear connections between inputs and outputs in a single structure. This allows the simulation of the linear and nonlinear dependencies of the output from the inputs using a rather simple model. Experiments predict future blood glucose levels up to 30 minutes ahead of time (RMSE $= 16.6 \pm 3.1\, mg/dL$).

It is interesting to point out that most methods in the literature exploit several features to perform a reliable prediction [16, 23], including emotional states and physical activity [70] which are difficult to quantify numerically, whilst others only exploit values collected from CGM devices as input [15, 24]. This choice is more versatile in a real-life application since it does not require the patient to monitor several physiological features.

In this chapter, we present a novel neural network [72] capable of predicting glycemic levels after having been trained on only 24 hours of CGM data of a single patient, without

requiring the gathering of additional features such as insulin and ingested CHO. The model is trained with a precision medicine approach which maximizes accuracy on the specific subject.

3.1. Dataset

The Unit of Diabetology and Endocrinology of the Campus Bio-Medico University Polyclin provided data of 12 subjects suffering from T1D who used CGM through the Medtronic $Enlite^{TM}$ glucose sensor. The dataset includes 7 females and 5 males, aged between 24 and 69 (average 40 $\pm$ 15) who have been diagnosed with T1D from 1 to 40 years ago (average 16.2 $\pm$ 12.9). Four patients present complications related to diabetes (e.g. neuropathy, dyslipidemia), while eight suffer from other autoimmune diseases such as systemic lupus erythematosus and hypothyroidism; four patients have no complications or further diseases.

Each patient was monitored for a period ranging from 15 to 30 days. However, each patient presents some days with discontinuities in the CGM track, which were not taken into account in order to avoid introducing a bias. The effective dataset includes a total of 111 days of continuous glucose monitoring, whereas the other 62 days in which some relevant disconnection occurred were discarded. This observation suggests using, when possible, a rather short monitoring time to develop a prediction algorithm, because long periods of recording are, in fact, not available in real applications.

3.2. Methods

Section 1.3 shows that patients need a method able to predict blood glucose levels using short training time and without any request for additional data. In this respect, we deem that a 24-hour training time is a reasonable choice since it would allow us to catch the glucose dynamics while requiring little effort from patients, using only CGM data as model input. Furthermore, we are interested in developing a personalized learning model able to describe the glucose dynamics of each subject, which is strongly tailored to the specific patient. The pipeline of the adopted approach is illustrated in Figure 3.1, where we distinguish between the training/validation and testing phases.

Among the models mentioned and described in section 1.3, the jump neural network [23] seems to be the most promising for the specific task of blood glucose levels forecasting. Indeed, the hidden neurons (with nonlinear activation functions) model the

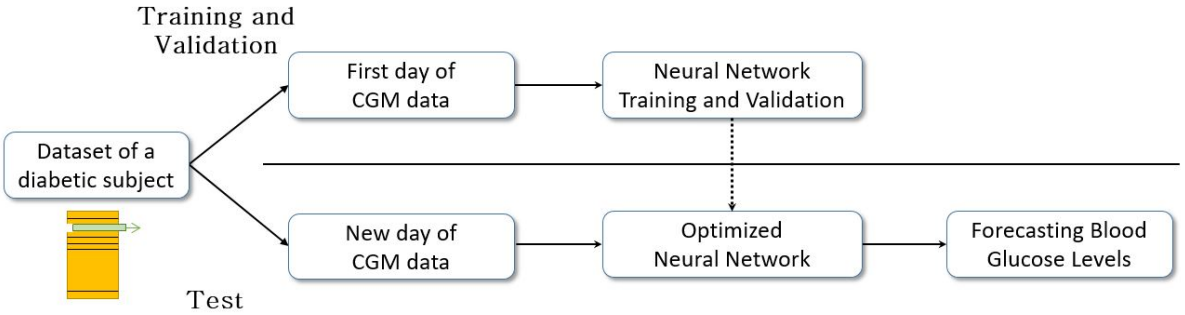


Figure 3.1.: Pipeline of the proposed approach to forecasting the blood glucose level.

nonlinear relationship between inputs and targets, while the output neurons (with linear activation functions) learn the linear relationship between inputs and targets. To formally present this neural network let us introduce the following notation:

- PH is the considered prediction horizon, expressed in minutes;
- $\mathbf{I}(t)$ is a row vector with M elements given by $M-1$ input signals at time t and by an input equal to 1 accounting for the bias;
- **IOW** is a row vector with M weight elements directly connecting every input to the output neuron;
- **HOW** is a row vector of L weights connecting every hidden neuron to the output neuron;
- **IHW** is a $L \times M$ matrix of weights connecting every input to every hidden neuron;
- f is the tangent-sigmoid activation function, computed element-wise on the results of $\mathbf{IHW} \cdot \mathbf{I}^T(t)$;

On this basis, the predicted signal of the jump neural network at time $t + PH$ is given by [23]:

$$\tilde{y}(t + PH|t) = \mathbf{IOW} \cdot \mathbf{I}(t)^T + \mathbf{HOW} \cdot f(\mathbf{IHW} \cdot \mathbf{I}(t)^T) \tag{3.1}$$

The first term models the linear relationship between the target and the inputs, whilst the second term models the nonlinear relationship. Furthermore, the hidden layer of the model proposed in [23] consists of just 5 hidden neurons, making it very quick both in the training and test phase. Despite its advantages, when applied to forecast the blood glucose concentration it did not perform better than other approaches in the literature,

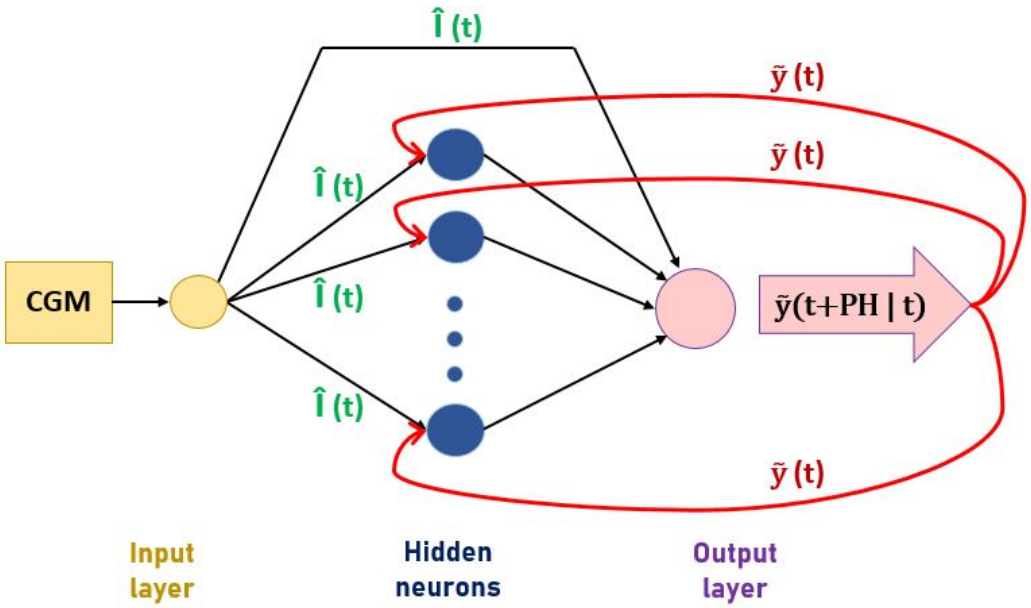


Figure 3.2.: Schematic illustration of the proposed ARTiDe jump neural network. The input is directly connected to the hidden neurons and the output layer and includes time delays (in green). Auto-regression (in red) is performed by directly connecting the output to the hidden layer.

mainly because it is not able to predict abrupt changes in the glucose trend and to detect hypoglycemic events.

To overcome the aforementioned limitations, in this work we extend the original jump neural network adding time delays and auto-regressive connections. For this reason, the proposed network is referred to as Auto-Regressive Time Delayed (ARTiDe) jump neural network, and its schema is shown in Figure 3.2. The time delays, shown using green symbols in Figure 3.2, are related to the input, and they are included from the input neuron to each hidden neuron and directly to the output layer. The rationale lies in observing that the prediction of future blood glucose levels can benefit from the knowledge of the recent glycemic history [24], since it supplies information on abrupt increase or decrease of blood glucose concentration in the last few minutes, e.g. due to the ingestion of sugars or due to an insulin bolus. The auto-regressive connections, represented by red connections in Figure 3.2, link the output neuron to the hidden layer, since this may reduce the error in the light of new incoming glucose values [26] thanks to the combination of auto-regressive feedback with weights that are modified during the learning process. Formally, assuming that η and ϕ denote the values of the input and auto-regressive delays respectively, these extensions modify as follows the parameters already introduced to describe (3.1):

- $\widehat{\mathbf{I}}(t)$ is a row vector of $H = M \cdot \eta$ elements that concatenates the last η input vectors so that $\widehat{\mathbf{I}}(t) = [\mathbf{I}(t),\ \mathbf{I}(t-1), \ldots, \mathbf{I}(t-\eta+1)]$;
- $\widehat{\mathbf{IOW}}$ is the row vector of H weights directly connecting every input to the output neuron;
- $\widehat{\mathbf{IHW}}$ is the $L \times H$ matrix of weights connecting each of the H inputs to every hidden neuron;
- $\tilde{\mathbf{y}}(t)$ is the row vector containing the last ϕ predicted values $\tilde{\mathbf{y}}(t) = [\tilde{y}(t-1), \tilde{y}(t-2), \ldots, \tilde{y}(t-\phi)]$;
- $\mathbf{OHW}$ is the $L \times \phi$ matrix of weights connecting the last ϕ predicted values to the hidden neurons;
- $\mathbf{HOW}$ and f are the same of (3.1).

Thus, the prediction $\tilde{y}(t+PH|t)$ of the ARTiDe jump neural network at the timestamp t can be expressed as:

$$\tilde{y}(t+PH|t) = \widehat{\mathbf{IOW}} \cdot \widehat{\mathbf{I}}(t)^T + \mathbf{HOW} \cdot f\left(\widehat{\mathbf{IHW}} \cdot \widehat{\mathbf{I}}(t)^T\right) + \mathbf{HOW} \cdot f\left(\mathbf{OHW} \cdot \tilde{\mathbf{y}}(t)^T\right) \quad (3.2)$$

In our case, $M = 1$ since we aim to forecast the blood glucose level $\tilde{y}(t+PH|t)$ using only CGM data as input.

3.2.1. Parameters search

The proposed model considers three structural parameters to be set, which are the number of neurons L in the hidden layer, the number of input delays η, and the number of feedback delays ϕ. To select their best combination, we perform a preliminary validation phase as follows. For each patient in the dataset, the initial 24 hours are used for training, and the following 24 hours are used as a validation set, where we evaluate the performances in terms of RMSE attained by carrying out a grid search of such parameters. The Bayesian regularization back-propagation is chosen as the training function since it allows the neural network to generalize well on data where the noise distribution on the training set and the prior distribution for the weights are Gaussian [73], as it could be expected in a physiological phenomenon. The back-propagation and the early

stopping processes ensure avoiding overfitting. The aforementioned grid search considers the parameters illustrated in Table 3.1.

Table 3.1.: Range of parameters investigated in the grid search to find the optimal combination.

	Minimum Value	Step	Maximum Value
Number L of hidden neurons	3	1	6
Number η of input delays	5	5	30
Number ϕ of feedback delays	5	5	30

The preliminary tests we performed confirm that a number of hidden neurons or hidden layers larger than the considered ranges would have resulted in overfitting of the network, confirming the results reported in [23], and longer delays would have caused considerable oscillations in the predicted values. The results of the grid search show that the most recurrent parameter combination among the different patient data in the validation set is given by 4 hidden neurons in the hidden layer, 10-minute input delays and 10-minute feedback delays, i.e. in both cases the values of the last 10 minutes are taken into account.

3.2.2. Experimental design

The proposed method was evaluated performing the three experiments described below. In all the cases, the prediction horizon was set equal to 15, 20 and 30 minutes, according to the medical practice [22]. Hereinafter, we used 80% of training data to train the network, whereas 20% of training data is used for detecting early stopping conditions.

3.2.2.1. Training: initial 24 hours - Test: all the following days

This first test aims to evaluate the performance of the proposed model when the first 24 hours of recorded data are used to train the network. For each patient, each group of consecutively monitored days was considered as a stand-alone group of data. All the following days were considered as test set (except for day 2 used as validation set as described in section 3.2.1).

3.2.2.2. Training: 24 hours - Test: next 24 hours

This experiment aims to investigate what happens if we retrain the learning model using data collected in the last 24 hours. Furthermore, it may reveal if taking into account the daily evolution of the glucose dynamics of the patient could be beneficial in some respect. In practice, we proceeded as follows. Each group of consecutive days was considered as a stand-alone group of data and, for each isolated group of days, the first day was used as training set, and the performance was evaluated on the second day. Hence, in this case, the network was trained from scratch. Next, the second day was used as training set to retrain the model from scratch, and performance was evaluated on the third day, and so on until the last recorded day of each group. Straightforwardly, the pairs of days used to determine the network configuration as described in section 3.2.1 were excluded from the experiment.

3.2.2.3. Training: incremental - Test: last day

This test aims to study what happens if we increase the amount of training data. For a fair comparison among the different trained models we fixed the test day to the last one. Hence, for every group of consecutive days of each patient, the last day was considered as test set. Different training sets were considered: first, only 24 hours of CGM data were used for training; second, 48 hours were used to train the network, and so on until the training set is composed by every day except the last one. Depending on the number of days available in each group, 2 to 5 days of CGM data were used to train each neural network used in this test.

Further to these three experiments, we performed a supplementary test resembling the traditional k-fold cross validation. For each patient, every single day was used as training set, and the performances were evaluated on all the other days, including the eventuality that the test set is composed by days prior to the day used for training. We designed this experiment to verify that the performances obtained in experiments A, B and C were not affected by the fact that the days used for training were more difficult to predict than those used for the test set. So, this further test also investigated these days, which otherwise would not have been included in any test set.

3.3. Results and Discussion

Table 3.2 reports the average results of the experiments described in the previous section for each patient and for each of the three prediction horizons considered. The results are expressed in terms of RMSE (equation 1.1) and, hence, the smaller the values the better the performances.

Let us turn the attention to the results attained in test A (*Training: initial 24 hours - Test: all the following days*). It permits us to investigate if 24 hours are enough for the network to learn the glycemic dynamics of the patient. The obtained results are promising, since the average RMSE with a 30-minute prediction horizon is in line with the state of the art, whereas the average value obtained for a 15-minute prediction horizon is slightly better, although the tests were performed on datasets composed of different patients. This suggests two observations: first, 24 hours are sufficient to properly train the proposed jump neural network for glucose levels forecasting. Second, the CGM data alone can be used as input, allowing to reduce as much as possible the burden of data collection on the patient. In particular, results with a 15-minute prediction horizon are promising because other algorithms at the state-of-the-art use much longer training periods to obtain similar or worse performances, even though they perform tests on datasets different from ours [23, 24, 26].

Let us now focus on the results attained in test B (*Training: 24 hours - Test: next 24 hours*), which permit us to assess if the performances improve by retraining the neural network with the latest available data. This test is very similar to the first one, except for the day used to train the model that varies every 24 hours, and it is always the day prior to the one used as test set. As can be observed in Table 3.2 the results get worse as the prediction horizon increases; however, the average results with a 30-minute prediction horizon are in line with those reported in the state of the art, and results with a 15-minute prediction horizon are slightly better [23, 24, 26]. Nevertheless, they are not significantly better than the results obtained in the first test, suggesting that both approaches could be used in a real-life application.

The third experiment, i.e. C (*Training: incremental - Test: last day*), shows that no significant performance improvement is observed increasing the training period of the model to several days, as illustrated in Figure 3.3. Specifically, we found that the most remarkable improvement concerns patient 10, for whom the RMSE performance improves from 11.8 to 10.9 mg/dL when 5 days are used to train the model, and the last available day is used as test set. However, the improvements concerning other patients

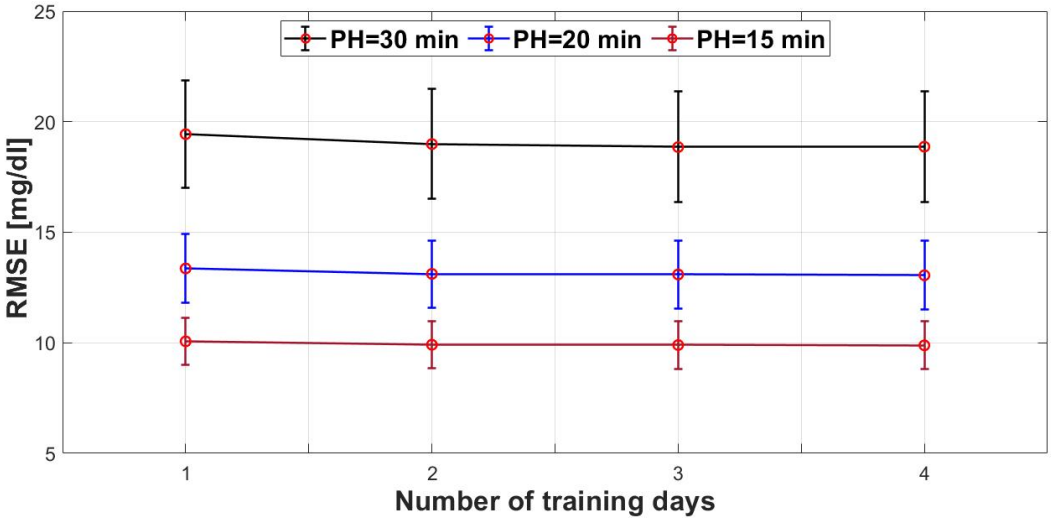


Figure 3.3.: Performance improvement for test C as the size of the training set increases. The average results in terms of $RMSE$ and related standard deviations are reported from 1 to 4 training days for the three tested prediction horizons.

Table 3.2.: Average results of the three tests with a 15-, 20- and 30-minute prediction horizon PH. Each tabular shows the average and the standard deviation of the RMSE $[mg/dL]$.

PH	Test A	Test B	Test C
$15'$	10.5 ± 1.5	9.9 ± 1.3	10.1 ± 1.9
$20'$	13.9 ± 2.1	13.5 ± 1.7	13.5 ± 2.6
$30'$	20.1 ± 2.8	19.7 ± 2.6	19.9 ± 4.4

are less notable, and they are smaller than 1 mg/dL. The average results are in line with those of the other tests, suggesting that 24 hours can be enough to properly train a predictive model with the proposed neural network.

The supplementary test proves the versatility of the proposed model. In terms of RMSE the average results in case of 15, 20 and 30-minute prediction horizons are equal to 10.3 ± 1.5, 13.4 ± 2.3, and $20.1 \pm 3.3\, mg/dL$ respectively. They are very similar to the results attained b previous tests, suggesting that the proposed neural network did not experience overfitting and that it is capable of generalizing well, even in presence of discontinuities of data recording.

Some graphical illustrations of the predictions from the three tests are offered in Figure 3.4. Considering a 20-minute prediction horizon, we report the plots for patients 3, 9, and 6 who are selected since the proposed neural network attains the worst, best

Table 3.3.: Average results achieved with the proposed ARTiDe jump neural network and with other methods in the state-of-the-art with a 15, 20 and 30-minute Prediction Horizon (PH), after 24-hour training. We tested a feed-forward neural network, a delayed feed-forward neural network, a jump neural network and a three-compartmental physiological-based model. Each tabular shows the average and the standard deviation of the RMSE $[mg/dL]$.

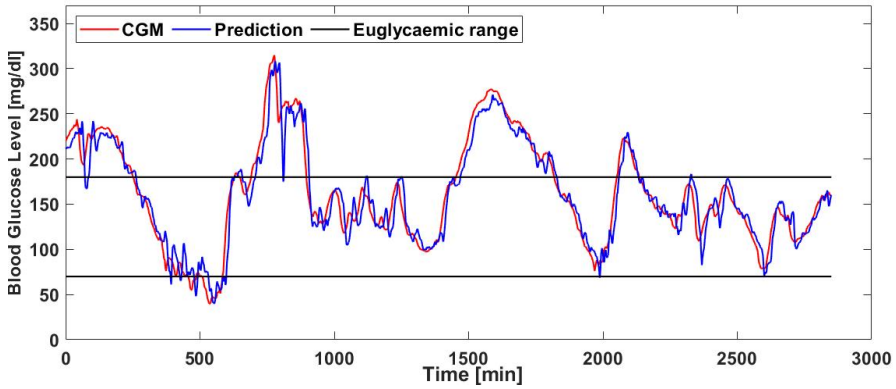
PH	ARTiDe Jump Neural Network	Feed-Forward Neural Network [70]	Delayed Feed-Forward Neural Network	Jump Neural Network [23]	3-Compartmental PB Model [71]
15'	10.5 ± 1.5	15.9 ± 2.9	17.3 ± 3.4	15.1 ± 4.7	44.1 ± 9.5
20'	13.9 ± 2.1	20.0 ± 3.7	21.7 ± 3.4	19.5 ± 3.5	/
30'	20.1 ± 2.8	28.1 ± 5.1	30.2 ± 4.7	29.9 ± 6.5	/

and average performance on their data during tests A, B and C. In detail, for patient 3 we have RMSE = 17.3, 16.2 and 18.3 mg/dL for tests A, B and C respectively. For patient 9 we have RMSE = 9.5, 9.5 and 9.6 mg/dL, whilst for patient 6 we have RMSE = 14.3, 13.6 and 13.8 mg/dL. Results with a 20-minute prediction horizon are not comparable to other works in the literature, since tests with this prediction horizon are not reported; however, the good results of forecasts 20 minutes ahead of time suggest to increase the prediction horizon in order to allow the patient to have more time to perform adjustments on their glucose levels in case of issues.

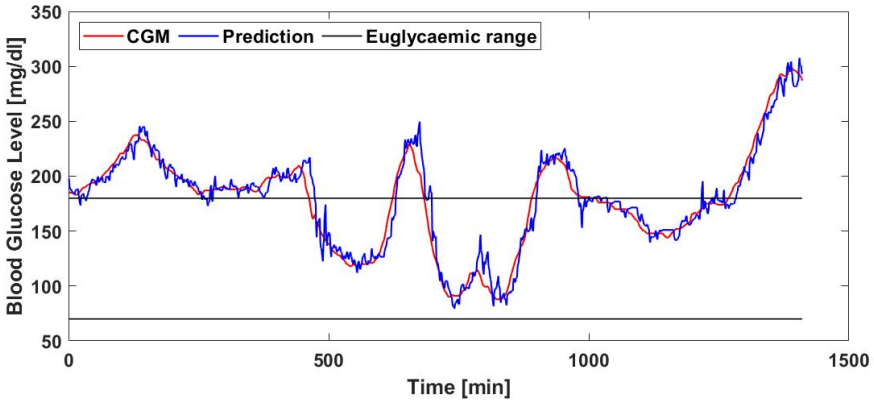
3.3.1. Comparison with other methods

In this section we show that results reported in the previous section are noteworthy if compared with those achieved using other methods in the literature. We tested the following competitors on test A (section 3.2.2.1), using CGM data as input and 24-hour training time for predictions 15, 20 and 30 minutes ahead of time, reporting the achieved results in Table 3.3:

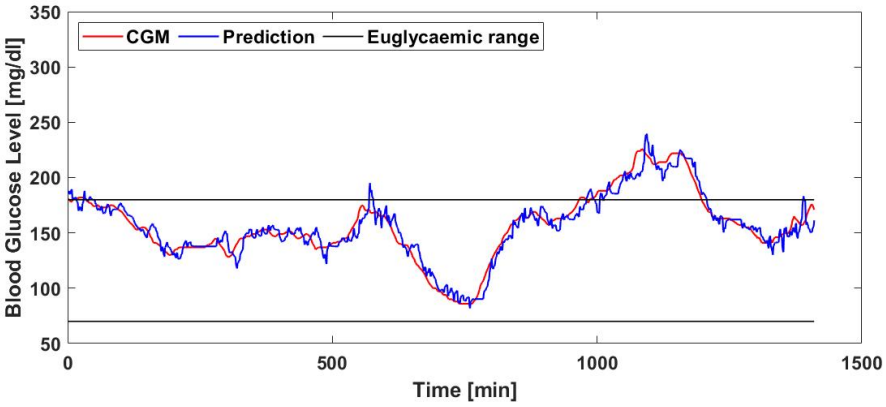
- *Feed-forward neural network*: we selected this architecture since many methods in the state-of-the-art use it [70, 74]. In particular we tested the implementation presented in [70], which makes use of 9 neurons with hyperbolic tangent transfer function in its hidden layer;
- *Delayed feed-forward neural network*: this method is tested for comparison with the proposed ARTiDe jump neural network, in order to measure the performance achieved using only time delays regarding the input data. To perform an adequate comparison, the number and type of hidden neurons is the same used in the proposed model;



(a) Test A on patient 3, average RMSE = 17.3 mg/dL



(b) Test B on patient 6, average RMSE = 13.6 mg/dL



(c) Test C on patient 9, average RMSE = 9.6 mg/dL

Figure 3.4.: Graphical examples of the three tests, performed with a 20-minute prediction horizon. The reported predictions are related to patients with the best, worst and average performance.

- *Jump neural network*: this method reproduces the jump neural network proposed in [23], which includes 5 hidden neurons in its hidden layer. This comparison permits us to investigate the benefits given by the introduction of time delays and auto-regressive feedbacks in ARTiDe architecture;
- *Three-compartmental physiological-based model*: the physiological model proposed in [71] is tested as well to compare the performances of physiological-based and machine learning-based approaches. Note that it does not perform a prediction ahead of time of the future glucose values, since it aims to simulate the specific glycemic trend of diabetic patients exploiting the initial glucose value and the real-time knowledge of the injected insulin and the ingested CHO. As a consequence, only one numerical value is reported in Table 3.3. Furthermore, differently from ARTiDe and other competitors, this model needs data on insulin and CHO in addition to CGM data.

Comparing the results reported in Table 3.3 with those shown in the second column of Table 3.2 (test A), it is straightforward to notice that ARTiDe attains better performances than all the competitors whatever the prediction horizon. Interestingly, these results also show that the introduction of time delays and auto-regressive feedbacks in the jump neural network is beneficial. The p-values of t-test confirm that a statistically significant difference exists between the performances achieved by the proposed and by the compared models: indeed, $p < 0.01$ when comparing ARTiDe and the pyhisiological-based model, the feed-forward and the delayed neural networks, while $p < 0.02$ when comparing ARTiDe and the original implementation of the jump neural network.

4. Auto-Regressive Time Delayed jump neural network for blood glucose levels forecasting

The forecasting of blood glucose levels based on data-driven models suffers from some open issues, including but not limited to the complications related to gathering and joining several heterogeneous features, and the large amount of data necessary to train a machine learning model. This is in contrast with what is observed in real data, where patients are unlikely to collect all the information related to their daily activity (e.g., amount of ingested CHO or physical activity) and many sensor disconnections occur. In this respect, this chapter extends the results achieved in the previous chapter [72]. First, to validate the proposed neural network, a larger cohort of patients is considered, including one suffering from T2D and two others who performed physical activity. Second, the results are compared with those achieved by other well-established methods from the state of the art of time series forecasting. Third, the *event detection* performance of the developed model is investigated, as proposed in [29] and described in section 4.3.5. Fourth, we perform tests on a public dataset [11] comparing our results with those already published for the same task [51].

4.1. Datasets

A wide-ranging analysis of the state of the art shows that tests are usually performed on private datasets, which makes it difficult to compare algorithms. However, a public dataset is available since 2018 [11], and data concerning the performance of other methods on this dataset exist. Hence, to improve the significance of our evaluation, in this work we consider both a private internal dataset, on which the algorithm is tested in cross-validation, and the aforementioned public external dataset [11] used to evaluate the performance in comparison with other methods in the literature.

4.1.1. Internal Validation Dataset

The Unit of Endocrinology and Diabetology of Campus Bio-Medico University Polyclinic provided data of 33 patients with T1D and using three different CGM devices (i.e. Dexcom G5®, Medtronic Guardian™ sensor 2 and Medtronic Guardian™ sensor 3). The dataset includes 16 females and 17 males, aged between 24 and 70 (average 43 ± 17) who were diagnosed with diabetes from 1 to 40 years ago (average 16 ± 13). The dataset includes a highly variable population: nine patients present complications related to diabetes (e.g. neuropathy, dyslipidemia), whereas fourteen suffer from other autoimmune diseases such as SLE and hypothyroidism; eleven patients have no complications or further diseases. One patient suffers from T2D. Some information about physical activity is available: patient 25 reported time and duration, whereas patient 17 regularly performs physical activity, although no specific events or their duration are reported in the available data.

Every patient was monitored for a period of time ranging from 8 to 30 days (average 15 ± 11). Nonetheless, each patient presents some days with several discontinuities or long interruptions in the CGM track; these days were excluded from this study to avoid introducing bias. Thus, the effective dataset includes a total of 296 days (average 9 ± 7 for each patient) of continuous glucose monitoring, and we discarded other 198 days (average 6 ± 5 for each patient) in which some relevant disconnection occurred. This observation suggests to use, when possible, a rather short monitoring time to train a predictive algorithm, because long recording periods are, in fact, not available in real applications.

4.1.2. External Validation Dataset

We use the Ohio T1DM dataset [11], which was initially available to participants in the Blood Glucose Level Prediction (BGLP) Challenge of the Third International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI 2018 in Stockholm, Sweden, and then became publicly available to other researchers. It contains eight weeks of data concerning CGM, insulin, physiological sensor, and self-reported life-events of six people suffering from T1D. The dataset includes 4 females and 2 males, aged between 40 and 60, each using the Medtronic $Enlite^{TM}$ CGM sensor. Table 4.4 reports state-of-the-art performance on this dataset.

4.2. Methods

Blood glucose levels are characterized by both linear and nonlinear components, as are many other physiological signals. For this reason, in order to achieve good performance, many methods that combine a linear and a nonlinear model (e.g. a generative model and a neural network) have been proposed in the literature [31, 75]. Conversely, here we propose a method capable of performing the same task using a single model. This goal is achieved using the structure of a classic multilayer perceptron with the addition of three main components altogether:

1. direct connections from the input to the output layer;
2. feedback connections from the output to the hidden layer;
3. time delays for each of the input-to-hidden, output-to-hidden and input-to-output connections.

Using a single model simplifies the training phase and the model buildup, without sensitively increasing the computational burden [23]; see section 4.4.4 for more details. The proposed model is capable of exploiting the previous knowledge of past input values. The rationale lies in observing that the prediction of many physiological phenomena, such as blood glucose levels, can benefit from the previous knowledge of the recent input variables history [16, 22, 24, 26, 29]. Moreover, the auto-regressive connections from the output to the hidden neurons introduce a dependence from the predicted future, which proved to be beneficial for prediction performance, as shown in Table 4.4. The model is the one presented in chapter 3 and is defined ARTiDe jump neural network; its schema has been shown in Figure 3.2 in the previous chapter.

Let us assume a Univariate Time Series (UTS) approach. If the past η values of the time series are used as input, then the input vector $\widehat{\mathbf{I}}(t)$ is a row vector of size $1 \times \eta$ in the form:

$$\widehat{\mathbf{I}}(t) = [I(t), I(t-1), \ldots, I(t-\eta+1)] \tag{4.1}$$

which concatenates the values of the η most recent timestamps (green symbols in Figure 3.2). They are transferred from the input to each of the L hidden neurons through an $L \times \eta$ weight matrix **IHW**, and directly to the output neuron through a vector **IOW** composed of η weights.

The auto-regressive connections, represented by red connections in Figure 3.2, link the output neuron to the hidden layer, since this may reduce the error in the light of new incoming input values [26] thanks to the combination of auto-regressive feedbacks

with weights that are modified during the learning process. That is, $\tilde{\mathbf{y}}(t)$ is a row vector containing the last ϕ predicted values $\tilde{\mathbf{y}}(t) = [\tilde{y}(t-1), \tilde{y}(t-2), \ldots, \tilde{y}(t-\phi)]$, and it is connected to the L hidden neurons through an $L \times \phi$ weight matrix **OHW**.

This leads to an equation for the prediction $\tilde{y}(t+PH|t)$ of the ARTiDe jump neural network at the timestamp t as the one introduced in the previous chapter, and reported in the following to facilitate the reader:

$$\tilde{y}(t+PH|t) = \mathbf{IOW} \cdot \widehat{\mathbf{I}}(t)^T + \mathbf{HOW} \cdot f\Big(\mathbf{IHW} \cdot \widehat{\mathbf{I}}(t)^T\Big) + \mathbf{HOW} \cdot f\Big(\mathbf{OHW} \cdot \tilde{\mathbf{y}}(t)^T\Big) \quad (4.2)$$

where **HOW** is a row vector of L weights connecting every hidden neuron to the output neuron and $f(\bullet)$ is the tangent-sigmoid activation function, computed element-wise on the results of $\mathbf{IHW} \cdot \widehat{\mathbf{I}}(t)^T$ and $\mathbf{OHW} \cdot \tilde{\mathbf{y}}(t)^T$.

The network training resorts to the Bayesian regularization back-propagation as a training function. It allows a neural network to generalize well on data where the distribution of noise on the training set is Gaussian and the prior distribution for the weights is Gaussian [73], as one could expect when dealing with a physiological phenomenon. Let us define e_i as the network error related to a prediction at the i-th timestamp t_i, computed as the subtraction between the true value $y(t_i + PH)$ and the predicted value $\tilde{y}(t_i + PH|t_i)$ given a prediction horizon PH. Straightforwardly, after k predictions have been run, the vector of network errors $\mathbf{e}_k$ is composed of the past k values of e_i. The Bayesian regularization aims to minimize a regularized objective function F on the whole time series of length N used for training. F is composed of two terms:

$$F = \alpha_1 \, SSE + \alpha_2 \, W \quad (4.3)$$

where $$SSE = \sum_{i=1}^{N-PH} {e_i}^2 \qquad \text{and} \qquad W = \frac{1}{\Omega} \sum_{j=1}^{\Omega} {w_j}^2$$

are the Sum-of-Squared-Errors computed on the total $(N - PH)$ predicted timestamps and the average of the squares of the Ω networks weights w_j, respectively. The scalars $\alpha_1 \in \mathbb{R}^+$ and $\alpha_2 \in \mathbb{R}^+$ are objective function parameters. Minimizing F consists in finding a trade-off between minimizing the SSE and W: the former generates a network response which is as close as possible to the ground truth values, whereas the latter produces a smoother network response due to smaller weight size. The weights and biases of the network are initialized randomly, and the Bayesian regularization updates their values according to the Levenberg-Marquardt optimization. This algorithm is faster

than standard back-propagation since it approximates[1] the Hessian matrix as $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ and computes the gradient as $\mathbf{G} = \mathbf{J}^T\mathbf{e}_k$, so that the series values are updated as

$$\tilde{\mathbf{y}}_{k+1} = \tilde{\mathbf{y}}_k - [\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}]^{-1}\mathbf{J}^T\mathbf{e}_k \tag{4.4}$$

where $\mu \in \mathbb{R}^+$ is a scalar learning coefficient and $\mathbf{J}$ is the Jacobian matrix that contains the first derivatives of the network errors with respect to the weights and biases. The Jacobian matrix can be computed through a standard back-propagation technique, which back-propagates the network error to every layer of the neural network so that at the end of the process the optimal values of weights and biases are determined. In addition, we set early stopping conditions by imposing a minimum gradient and by setting the maximum number of validation check fails. The back-propagation and the early stopping processes ensure to avoid overfitting: at each training iteration, the weights of all the connections are updated until an early stopping condition is reached or the maximum number of epochs has passed.

4.2.1. Parameters search

The proposed model considers three structural parameters to be set, which are the number of neurons L in the hidden layer, the number of input delays η, and the number of feedback delays ϕ. Preliminary tests limited the search interval for L between 3 and 6 hidden neurons, and between 5 and 30 for both η and ϕ. Exceeding such ranges results in overfitting of the network or in poor performance. In order to find the best combination possible, a validation phase is performed as follows. We randomly select 17 out of the 33 patients in the internal dataset described in section 4.1.1. For these 17 patients, the initial 24 hours of recorded data are used for training and the following 24 hours are used as validation set. The patient with T2D and those with information concerning physical activity are not considered in this phase. We evaluate performance on the validation set in terms of RMSE (Equation 1.1), attained carrying out a grid search of the structural network parameters. The results of the grid search show that the most recurrent parameter combination among the different patient data in the validation set is given by 4 neurons in the hidden layer, 10-minute input delays and 10-minute feedback delays, i.e. in both cases the values of the last 10 minutes are taken into account.

[1] The approximation is possible only if the performance function has the form of a sum of squares, as happens in this case.

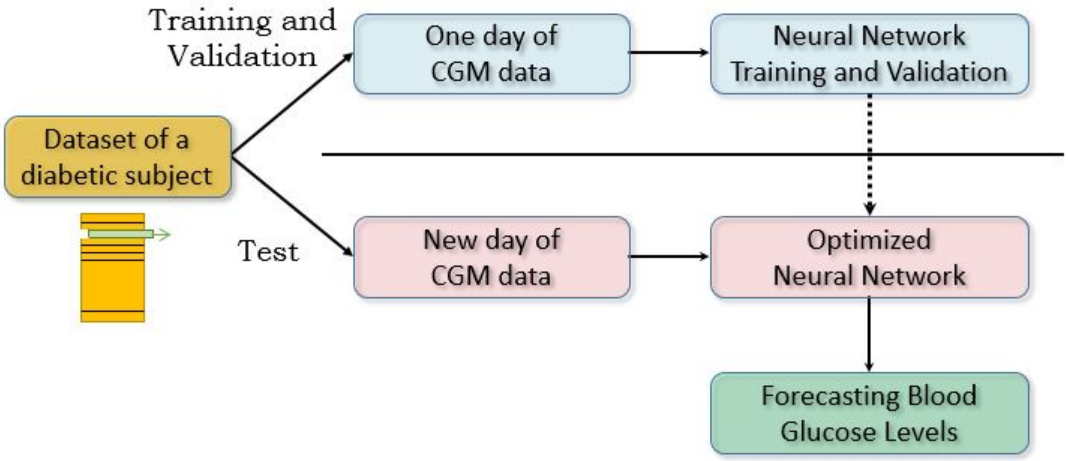


Figure 4.1.: Pipeline of the blood glucose levels forecasting task.

4.3. Experimental Design

As pointed out in section 4.2.1, only data of 17 out of the 33 patients from the internal dataset are used to set the optimal number of structural parameters. The second day of each group of consecutive days is used for validation. From now on, we will refer to these 17 patients as *group A*, and to the further 16 patients as *group B*. In order to evaluate the performance of the model, we implement the experiments described below. The prediction horizon is set equal to 15, 20 and 30 minutes, according to the medical practice [22]. Hereinafter we use data corresponding to 24 hours of CGM reading as training set for each test, unless differently specified. We use 80% of training data to train the network, whereas the remaining 20% is used for detecting early stopping conditions. A general pipeline of the performed task is illustrated in Figure 4.1. We evaluate the model performance by measuring the RMSE between the predicted and the actual glucose values. Such a measure permits an immediate comparison with the results achieved in other works, being a widespread performance evaluation metric. However, RMSE returns an error value which strongly depends on the order of magnitude of the analyzed quantity; as a consequence, we also evaluate performance in terms of SSGPE, as defined in equation 1.2.

4.3.1. Training: initial 24 hours - Test: all the following days

This first test aims to evaluate the performance of the proposed model when the first 24 hours of recorded data are used to train the network. All the following days are considered as test set (except for day 2 of patients in *group A* used as validation set).

For each patient, each group of consecutively monitored days is considered as a stand-alone group of data.

4.3.2. Training: 24 hours - Test: next 24 hours

This experiment aims to investigate what happens if we retrain the learning model using data collected in the latest 24 hours. Furthermore, it may reveal if taking into account the daily evolution of the glucose dynamics of the patient could be beneficial in some respect. In practice, we proceed as follows. Each group of consecutive days is considered as a stand-alone group of data. The first day of each group is used for training the network from scratch, and the performance is evaluated on the second day. Next, the second day is used as training set to retrain the model from scratch, and performance is evaluated on the third day, and so on until the last recorded day of each group. For patients in *group A*, the pairs of days used to determine the network configuration are excluded from the experiment.

4.3.3. Training: incremental - Test: last day

This test aims to study what happens if we increase the amount of training data. For a fair comparison between the models trained using different training sets, we fix the test day to the last one. Different training sets are considered: first, only 24 hours of CGM data are used for training; second, 48 hours are used to train the network, and so on until the training set is composed by every day except the last one. Depending on the size of each group of days, 2 to 5 days of CGM data are used to train the models used in this test.

4.3.4. Comparison with other methods

To further assess our method, we compare the results we achieve with those of well-established methods in the literature. The list of competitors includes one model in the frame of kernel machines (SVR), one in the frame of the forests of trees (Random Forest Regression), one symbolic model (SAX), one generative model (AR) and four artificial neural networks (RNN, NARX, Delayed FFNN, Jump neural network). Each method is optimized on the internal dataset described in section 4.1.1 through a validation set, so that each competitor achieves the best possible performance. According to the procedure we pursued to train our model, we consider for each competitor the possibility

of exploiting the knowledge of previous glucose levels, if this leads to better performance. The optimization procedure produced the following results:

- *Random Forest Regression*: we test a forest of regression trees, investigating the optimal amount of past values to be used in predictions, the number of learners and the depth of trees. The optimal combination turns out to be a forest of 70 regression trees exploiting the knowledge of the past 5 minutes of glucose dynamics.
- *SVR*: we investigate which between linear, gaussian and polynomial kernel ensures the best performance. The optimal model turns out to be an SVR with linear kernel function exploiting the past 5 minutes of CGM data, after standardization of the input.
- *SAX model*: the optimal solution turns out to be the utilization of an alphabet of 10 symbols, after dividing each test day in 288 segments (i.e. one segment every five minutes).
- *Auto-Regressive (AR) model*: we test this dynamic model to investigate the optimal number of past glucose values to use as input, which results in an input vector composed of the past 5 minutes.
- *Recurrent Neural Network*: the optimal combination includes 3 neurons in the hidden layer with tangent-sigmoid activation function, and it takes into account the past 30 predicted values, regardless of past input values.
- *NARX neural network*: we test the closed-loop configuration to find the optimal values of hidden neurons, input delays and feedback delays. The optimal combination turns out to be a neural network having 4 hidden neurons, and taking into account the past 30 minutes of CGM data and the latest 10 predicted values.
- *Jump neural network*: this method reproduces the jump neural network proposed in [23]. The optimal model includes 4 neurons in its hidden layer and only the last available glucose value as input. It is worth noting that the ARTiDe jump neural network represents a more general framework of this model, since when $\phi = 0$ and $\eta = 1$ ARTiDe does correspond to the model proposed in [23] with a Multivariate Time Series (MTS) approach.
- *Delayed feed-forward neural network*: this method is tested for comparison with the proposed ARTiDe jump neural network, in order to measure the performance

achieved using the same input with a more traditional network structure. The optimal combination turns out to be a neural network having 4 neurons in the hidden layer and taking into account the past 10 minutes of CGM data.

4.3.5. Event detection

To further assess the model prediction capability, we perform the *event detection* task defined in [29]. According to the timestamps of samples verifying particular conditions, the following *events* are defined:

- Severe Hypoglycemia: the blood glucose level falls below the 50 mg/dL threshold, i.e. $y(t-1) > 50$ and $y(t) \leq 50$;
- Hypoglycemia: the blood glucose level falls below the 70 mg/dL threshold, i.e. $y(t-1) > 70$ and $y(t) \leq 70$;
- Hyperglycemia: the blood glucose level exceeds the 180 mg/dL threshold, i.e. $y(t-1) < 180$ and $y(t) \geq 180$;
- Severe Hyperglycemia: the blood glucose level exceeds the 250 mg/dL threshold, i.e. $y(t-1) < 250$ and $y(t) \geq 250$;

where all measurement units are mg/dL. Four different sets are created according to this criterion, each containing all the events that meet one of the above defined conditions. Due to measurement noise, multiple consecutive events of the same type could be observed within a short time frame. Hence, if a specific event occurs, additional events of the same type are not considered for the following 30 minutes, in order to limit this unrealistic behavior [76]. With this purpose, both the predicted and the original glycemic time series are investigated to detect the events defined above. We use the same rationale of [29] and proceed as follows: let $\mathbb{A} = \{t_1, \ldots, t_a\}$ be the generic set containing the a timestamps associated with the actual events, extracted from the values in the actual CGM values $\{y(t_1), \ldots, y(t_N)\}$, and let $\mathbb{P} = \{t_1, \ldots, t_p\}$ be the set containing the p timestamps associated with the predicted events corresponding to the predicted CGM track $\{\tilde{y}(t_1), \ldots, \tilde{y}(t_N)\}$. Each element in $\mathbb{P}$ is analyzed according to its timestamp t_p and marked as either a true positive or a false positive, according to the following criterion. Let us define $\Delta t_{pa} = t_p - t_a$ as the distance between the timestamps of a predicted event and of an actual event. Let PH be the considered prediction horizon. For each $t_p \in \mathbb{P}$, if there exists $t_a \in \mathbb{A}$ so that $-k < \Delta t_{pa} < PH$ (where $k \in \mathbb{N}$ is a positive constant

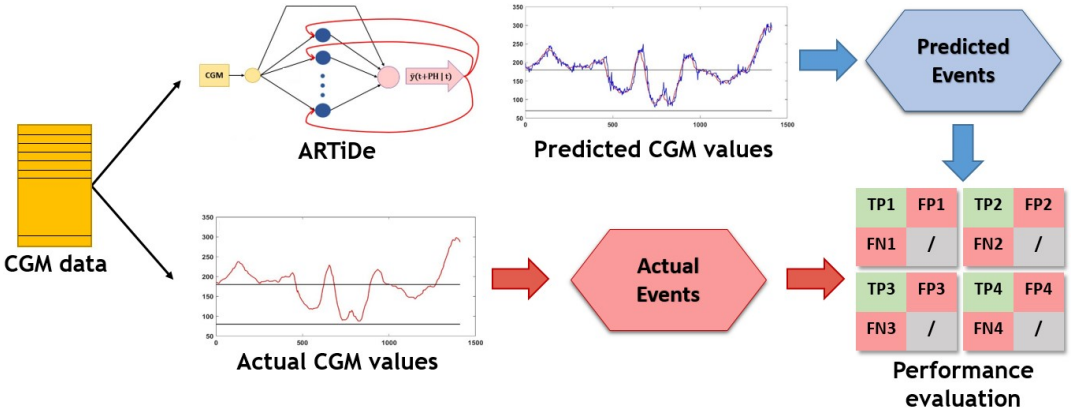


Figure 4.2.: Pipeline of the event detection task. Performance is evaluated in terms of recall, precision, and F1-Score for each class, according to the events detected in the actual and predicted time series.

that prevents the association of events that are too distant apart), then t_p is considered as a true positive and t_a is no longer considered. If the aforementioned condition is not met, t_p is considered to be a false positive, i.e. the predicted event will not occur or has already occurred. Following [29] we set $k = 30$ minutes. After having checked all the elements in set $\mathbb{P}$, the events in set $\mathbb{A}$ that are still not tagged are tagged as false negatives. Consequently, Precision, Recall, and F1-Score are evaluated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{4.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.7}$$

where TP, FP and FN are the total number of true positives, false positives and false negatives, respectively. A general pipeline of the event detection task is illustrated in Figure 4.2. Data of the 33 patients in the internal dataset include a total amount of 563 events which are 12 severe hypoglycemia, 97 hypoglycemia, 280 hyperglycemia and 174 severe hyperglycemia events.

4.3.6. Test on the External Dataset

Forecasting tests are also performed on the public Ohio T1DM dataset [11] to compare the results we achieve with those reported in the literature. The dataset has a given division between training and test sets for each patient. Following the rationale of

the test described in section 4.3.1, we use only the first 24 hours of data available in each training set to train and to optimize the neural network on the specific patient. Unfortunately, we found that the test set of every patient has several interruptions, as occurs in our private dataset. As a consequence, in order to assess the performance on the whole test set, we perform a test on every fraction of continuous data, and then merge all the fractions of the test set and of the predicted values into two respective time series, so that we can properly calculate the RMSE for every subject as described in equation (1.1). It is worth noting that the structural parameters of the neural network are those detected on patients from *group A* with the method described in section 4.2.1, so they are totally unbiased from this dataset.

Further to this test, we also increase the amount of training data to the whole training set to evaluate performance improvement while considering different prediction horizons.

4.4. Results and Discussion

Table 4.1 reports the average results of the experiments described in the previous section for all the patients in the internal dataset and for each of the three prediction horizons considered. The results are expressed in terms of RMSE (equation 1.1) and SSGPE (equation 1.2), hence, the smaller the values the better the performance.

Let us focus on the results attained in Test 1 (*Training: initial 24 hours - Test: all the following days*). It allows us to investigate if 24 hours are enough for the network to learn the glycemic dynamics of the patient. The obtained results are promising, since the average RMSE with a 30-minute prediction horizon is in line with the state of the art presented in Table1.1, whereas the average result obtained with a 15-minute prediction horizon is slightly better, although the tests are performed on different datasets. This suggests two observations: first, the CGM data alone can be used as input, and this reduces the burden of data collection as much as possible. Second, 24 hours are sufficient to properly train the proposed neural network: to the best of our knowledge, this is by far the smallest amount of data used in the literature to train a model for glucose level forecasting. In particular, results with a 15- and 30-minute prediction horizon are promising because other algorithms at the state of the art obtain similar or worse performance, even though they use much longer training periods and, in many cases, they exploit MTS approaches.

Let us now focus on the results attained in Test 2 (*Training: 24 hours - Test: next 24 hours*), which allows us to assess if the performance improves by retraining the neural

Table 4.1.: Average results of the three tests described in section 4.3 with a 15-, 20- and 30-minute prediction horizon PH. The performance is evaluated considering the internal dataset only. The reported results are the average of the RMSE $[mg/dL]$ and the SSGPE [%] scores on all the patients, with standard deviation.

PH	RMSE $[mg/dL]$			SSGPE [%]		
	$Test\ 1$	$Test\ 2$	$Test\ 3$	$Test\ 1$	$Test\ 2$	$Test\ 3$
$15'$	9.5 ± 1.9	9.3 ± 1.7	9.4 ± 1.8	5.3 ± 1.3	5.8 ± 1.4	5.7 ± 1.6
$20'$	12.6 ± 2.4	12.4 ± 2.2	12.5 ± 2.5	7.1 ± 1.7	6.8 ± 2.0	7.6 ± 2.1
$30'$	18.7 ± 3.5	18.4 ± 3.3	18.4 ± 3.9	10.6 ± 2.4	10.1 ± 2.8	11.1 ± 3.0

network with the latest available data. This test is very similar to the first one, except for the day used to train the model which varies every 24 hours, and it is always the day before the one used as the test set. We notice in Table 4.1 that the average results are in line with or slightly better than those reported in the state of the art. Nevertheless, they are not significantly better than the results achieved in the first test, suggesting that both approaches could be used in a real-life application.

The third experiment (*Training: incremental - Test: last day*) shows that no significant performance improvement is observed increasing the training period of the model to several days, as illustrated in Figure 4.3 where we report the average performance on all patients. Considering a 30-minute prediction horizon, we observe that the performance of only 4 out of the 33 patients improves more than 2 mg/dL, whereas other patients have modest benefits. The only remarkable improvement concerns patient 29, for whom the RMSE performance improves from 21.7 to 16.8 mg/dL when 5 days are used to train the model, and the last available day is used as the test set. This could be due to the prominent difference between the first (i.e. original training set) and the last (i.e. effective test set) recorded day of the patient: the former includes moderate glycemic levels, whereas the latter presents several hyperglycemic events, which are consequently underestimated with impact on the prediction quality. On the contrary, the days immediately preceding the last one are characterized by high glucose levels, being more similar to the tested day. Nonetheless, the improvements concerning other patients are less notable, and they are, in most cases, smaller than 1 mg/dL. The average results are in line with those of the other tests, suggesting that 24 hours can be enough to properly train a predictive model with the proposed neural network.

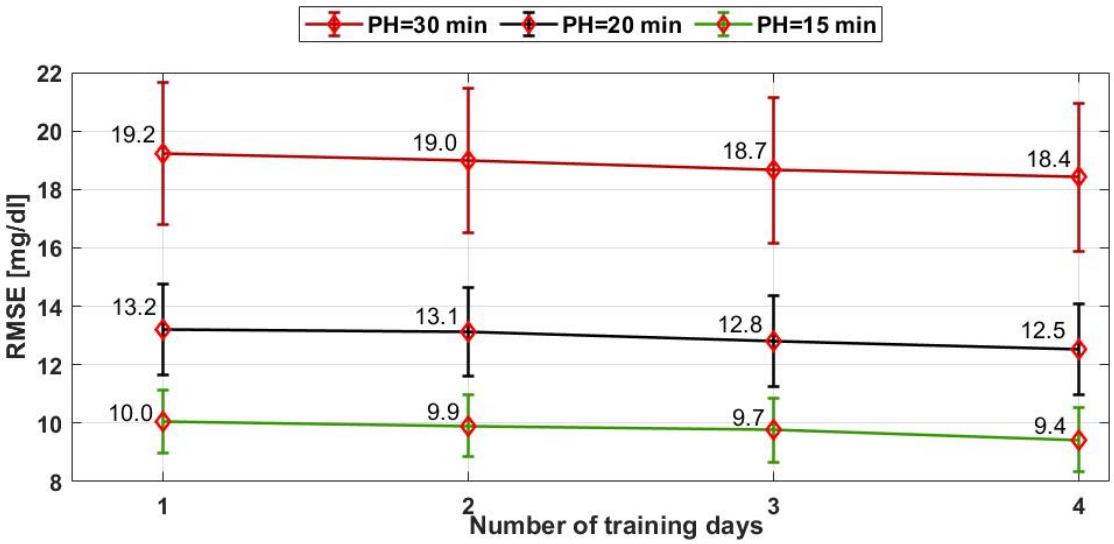


Figure 4.3.: Performance improvement for test 3 as the size of the training set increases. The average results in terms of RMSE and related standard deviations are reported from 1 to 4 training days for the three tested prediction horizons.

It is worth noting that RMSE results illustrated in Table 4.1 are slightly better than those reported in our previous work on the respective tests [72], proving that the model was fairly general despite the relatively low number of patients involved. Indeed, the larger dataset confirms the good performance with a more variable population. The SSGPE scores we achieved in the tests are noteworthy as well. Indeed, ARTiDe outperforms all the models that are investigated in the work of Gadaleta et al. [29].

The RMSE and SSGPE scores do not vary significantly among patients, considering that those suffering from diabetes complications or other diseases present the same average results as those with no complications. This proves that the model is properly capable of generalizing on a wide range of people with diabetes and that it performs accurate predictions regardless of disease severity. With regards to the patient suffering from T2D, we observe that the performance is in line with the total average since the RMSE on Test 1 results to be 18.7 mg/dL with a 30-minute prediction horizon. However, a single patient is not sufficient to assess the effectiveness of our method on T2D patients, and a specific and wide dataset would be necessary for this purpose. Turning the attention to the two patients who performed physical activity, we observe that the performance is slightly worse than the average for patient 17, whose RMSE is 19.9mg/dL for a 30-minute prediction horizon on Test 1; however, we do not exactly know when physical activity was performed. Performance is above average for patient 25 as well

Table 4.2.: Comparison between the results achieved on the internal dataset by the proposed ARTiDe jump neural network and by other well-established methods in the literature of time series forecasting, evaluated in terms of RMSE with a 15-, 20- and 30-minute Prediction Horizon (PH). The p-values from the t-test between the results of ARTiDe and the competitors are also reported.

Method	RMSE $[mg/dL]$			p-value
	PH=15'	PH=20'	PH=30'	
Random Forest Regression	27.4 ± 6.6	31.5 ± 6.6	39.4 ± 8.2	$< 10^{-15}$
SVR	62.8 ± 18.9	63.4 ± 19.2	64.0 ± 19.8	$< 10^{-15}$
AR model	12.2 ± 2.6	15.3 ± 2.7	20.9 ± 3.8	< 0.02
SAX model	70.7 ± 23.0	71.0 ± 22.3	71.4 ± 21.9	$< 10^{-15}$
Recurrent Neural Network	17.0 ± 8.4	22.0 ± 9.6	29.5 ± 9.3	< 0.01
NARX Neural Network	41.4 ± 36.2	50.6 ± 41.1	54.2 ± 28.1	$< 10^{-9}$
Jump Neural Network	14.9 ± 5.3	18.7 ± 3.7	24.1 ± 4.9	< 0.02
Delayed Feed-Forward Neural Network	17.3 ± 3.4	21.7 ± 3.4	30.2 ± 4.7	< 0.01
ARTiDe Jump Neural Network	$\mathbf{9.5 \pm 1.9}$	$\mathbf{12.6 \pm 2.4}$	$\mathbf{18.7 \pm 3.5}$	/

(RMSE = 19.4 mg/dL) since it deteriorates in the occurrence of the timestamps where physical activity is reported. This suggests that an *ad hoc* model for glucose levels prediction should be developed to perform predictions during physical activity, but a larger amount of information is necessary for this purpose, such as the use of MTS [77, 78].

4.4.1. Results of the comparison with other methods

In this section, we show that results reported in the previous section are noteworthy if compared with those of other well-established methods of time series analysis, which are described in Table 1.1. We tested the competitors on Test 1 (section 4.3.1), using only CGM data as input and considering 24 hours of data as training for predictions 15, 20, and 30 minutes ahead of time, reporting the achieved results in Table 4.2. The best-performing models are the AR model and three artificial neural network models, which are recurrent, jump, and delayed feed-forward neural networks. These models achieve performance in line with the state of the art. The performance achieved by the Random Forest is slightly worse than the state of the art, whereas the performance achieved by SVR, SAX, and NARX neural network is considerably worse.

Comparing the results reported in Table 4.2 with those of the proposed model, it is straightforward to notice that ARTiDe attains better results than all the competitors whatever the prediction horizon. Interestingly, the AR model outperforms many competitors, proving that the linear relation between the input and the predicted value

Table 4.3.: Event detection metrics for the proposed method tested on the internal dataset with a 30-minute prediction horizon.

	Recall %	Precision %	F1-Score %
Severe Hypoglycemia	83.3	83.3	83.3
Hypoglycemia	59.8	47.2	52.7
Hyperglycemia	86.4	58.0	69.4
Severe Hyperglycemia	72.4	54.5	62.2

plays a predominant role in forecasting blood glucose levels. In addition, ARTiDe improves the performance of the AR model by more than $2mg/dL$ in every test, proving that including the nonlinear component can further improve the results. Furthermore, the introduction of time delays and auto-regression confirms to be beneficial, according to the generally good results achieved by auto-regressive competitors. We performed the t-test between the results from the proposed method and from the competitors. The p-values reported in Table 4.2 confirm a statistically significant difference: indeed, $p < 0.02$ when comparing ARTiDe with the jump neural network and the AR model; in addition, the p-value is less than 0.01 when comparing ARTiDe with other methods.

4.4.2. Event Detection performance

Data from the 33 patients in the internal dataset include a total of 563 events, namely 12 severe hypoglycemia, 97 hypoglycemia, 280 hyperglycemia, and 174 severe hyperglycemia. Defined the Imbalance Ratio (IR) as the ratio between the number of instances in the majority class and the number of instances in the minority class, the dataset has $IR = 23.3$ and thus presents high imbalance according to the definition given in [79]. We evaluate the event detection performance considering the predictions generated from Test 1 (section 4.3.1), i.e. the regression outputs provided by ARTiDe are compared with the thresholds defined in section 4.3.5 to detect the corresponding event. Considering a 30-minute prediction horizon, the recall, precision, and F1-Score of each class are reported in Table 4.3.

A direct comparison with the results reported by Gadaleta [29] is not possible, due to the different tested datasets, the number of tested models, and the different aims of the works. However, a qualitative comparison is shown in Figure 4.4 using boxplots, which

provide a statistical and compact view of the results. The results we achieve are comparable to those of the methods in [29] that achieve the best performance, although the significance of this conclusion is purely qualitative since the datasets used for evaluation are different.

Due to the high IR of the dataset, F1-Score represents the most appropriate metric to evaluate performance between classes. It is worth noting that the F1-Score for hyperglycemia is sensitively higher than the one for hypoglycemia. A possible explanation for this finding may be the fact that data used in this work are gathered from patients in real-life conditions. Indeed, patient interventions may have occurred to prevent or mitigate hypoglycemia, making it more difficult for the model to learn the correct pattern preceding a hypoglycemic event, e.g. a prediction performed 30 minutes in advance may take place before the intervention by the patient. This generates false alarms and reduces the precision score of hypoglycemia. As a matter of fact, all the severe hypoglycemia events are preceded by the same pattern, i.e. missed intervention by the patient during hypoglycemia, which results in a high F1-Score despite the small number of instances. In addition, after comparing the forecasted data with the original events of hypoglycemia, we notice that some predictions take place after an event has already occurred and that some forecasts predict a glycemic value that is slightly higher than the 70 mg/dL threshold; this reduces the recall of hypoglycemia. On the other hand, for the reasons discussed in the Introduction, patient interventions are less likely to occur in the case of a hyperglycemic event. This is proven by the larger amount of such events and results in the higher F1-Score of this class.

4.4.3. Results on the External Dataset

The previous sections propose an immediate comparison with other methods tackling the same issue in the literature by directly analyzing the performance achieved in terms of RMSE. We also test our method on the public Ohio T1DM dataset [11], so that we can compare the performance of the proposed method with those of other works in the literature that tested the same dataset. The results of this comparison are illustrated in Table 4.4. We report the results of predictions 30 minutes ahead of time because all the listed works tested this prediction horizon. The third column of the table shows that the proposed method outperforms all the others in the literature, although the amount of training data utilized is considerably smaller. The best results among the listed competitors are achieved by Chen et al. [26], whose method lies on our same rationale since it exploits the knowledge of both past inputs and predicted values to

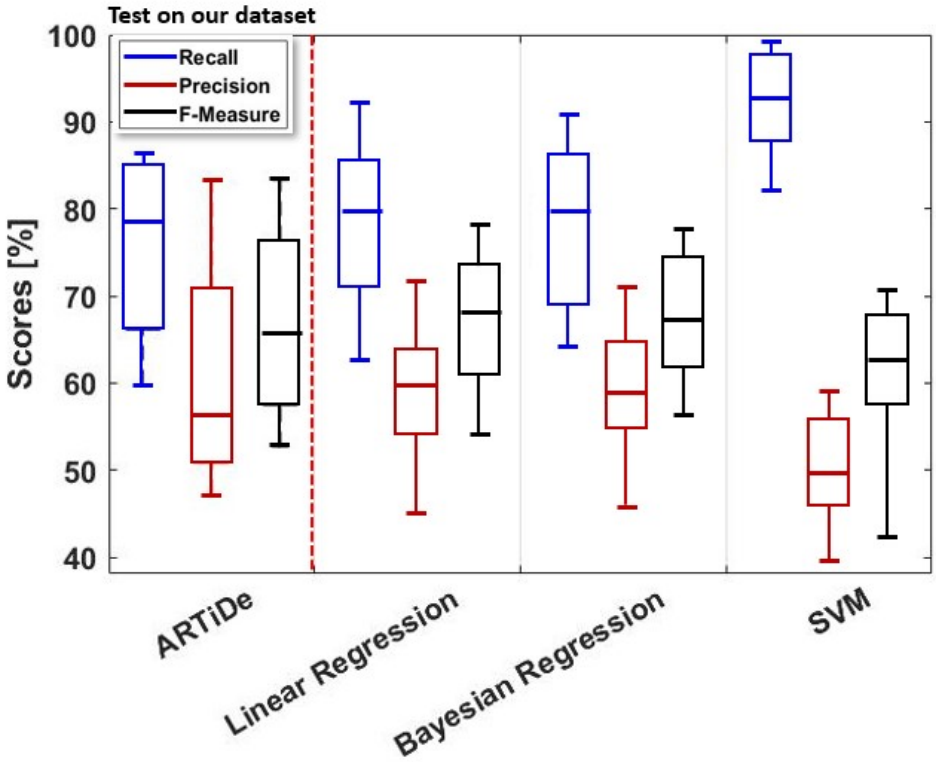


Figure 4.4.: Results of the Event Detection task in terms of Recall, Precision, and F1-Score. The left column reports the results achieved using the proposed method on our internal dataset; the remaining columns report the results of the 3 best-performing methods in [29] on their private dataset for a qualitative comparison.

Table 4.4.: Comparison between the results achieved on the Ohio T1DM dataset by the proposed method and others in the literature, considering a prediction horizon of 30 minutes. The methods and relative authors are illustrated, as well as the performance achieved in terms of RMSE (with standard deviation). Multivariate models are marked with an asterisk. We report both the results we achieve using 24 hours of data to train the model (Test 1) and the full training set (Test 3).

Study	Method	RMSE $[mg/dL]$
Midroni et al. [17]	XGBoost Random Forest*	20.4 ± 2.2
Contreras et al. [20]	Grammar Evolution approach*	21.2 ± 1.8
Martinsson et al. [24]	LSTM Recurrent Neural Network	20.1 ± 2.4
Zhu et al. [25]	Convolutional Neural Network*	22.2 ± 2.5
Chen et al. [26]	Dilated Recurrent Neural Network*	19.0 ± 2.4
Bertachi et al. [27]	Feed-Forward Neural Network*	19.3 ± 2.2
Li et al. [28]	GluNet*	19.3 ± 2.8
Proposed model	ARTiDe (full Training set)	18.4 ± 1.6
	ARTiDe (24-hour Training)	18.8 ± 1.3

improve performance. Nevertheless, the direct input-output connections of our method ensure slightly better performance, even though Chen et al. exploit information on CHO and insulin doses in addition to CGM values.

Figure 4.5 illustrates the results achieved by increasing the amount of training data until completing the training set. As mentioned in section 4.1.2, this dataset presents many discontinuities in data recording. Consequently, in order to consider the whole training set for each patient, we initially train the model from scratch on the initial part of the data, and then re-train the model on the remainder of the training data, updating the weights and bias values. We tested prediction horizons from 15 to 120 minutes, increasing by 15 minutes for every iteration. Each test is executed two times: 1) only the first 24 hours of CGM data are used to train the model, and 2) the complete training set is exploited for each patient. Figure 4.5 shows the average results, in terms of RMSE, of the predictions performed in both cases. Interestingly, no appreciable difference occurs when short-term predictions are operated; in particular, for the considered 30-minute prediction horizon the RMSE performance improves from 18.8 to $18.4 mg/dL$. Conversely, the enhancement of the training set produces more sensitive upgrades when dealing with longer prediction horizons. It is worth noting that, in every case, the average improvement is smaller than $3\, mg/dL$, proving that 24 hours of recorded data can be enough to properly train the ARTiDe jump neural network.

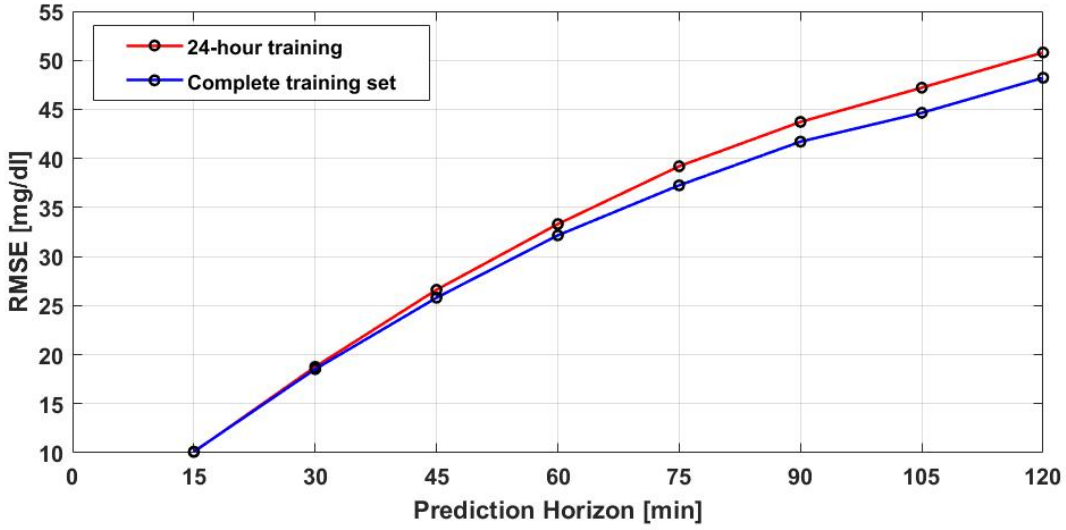


Figure 4.5.: Performance related to the Ohio T1DM dataset for different prediction horizons. The proposed model is trained using 24 hours of data or the complete training set.

This test also confirms the robustness of the proposed method, since comparable results are achieved on datasets composed of patients from different parts of the world (i.e. Italy and Ohio), proving that the structure of the proposed neural network is capable of generalizing results on a wide range of people with diabetes having different lifestyle and diet.

4.4.4. Computational complexity

In this section, we briefly investigate the computational complexity of the ARTiDe jump neural network, together with its training and testing time. According to the notation introduced in section 4.2, the computational complexity of the proposed model during the training phase can be approximated as being in the range of $O\Big(NL(\eta+\phi)\Big)$ per iteration, where N is the number of timestamps considered as the training set, L is the number of neurons in the hidden layer, η and ϕ are the numbers of input and feedback delays, respectively. For comparison, the computational complexity of a fully-connected multilayer perceptron with one hidden layer composed of L neurons and with η inputs can be approximated as $O(NL\eta)$. Straightforwardly, the computational complexities during the test phase for a single prediction can be approximated as $O\Big(L(\eta+\phi)\Big)$ and $O(L\eta)$, respectively. Considering that η and ϕ share the same order of magnitude in

practical cases, the computational complexity of ARTiDe is about twice the complexity of a multilayer perceptron. In practice, this is not an issue: with regard to Test 1 with a 30-minute prediction horizon, the average time (with standard deviation) for training the model on each training set is 0.28 ± 0.17 seconds, whereas the average time to perform a single prediction is $3.38 \cdot 10^{-5} \pm 4.36 \cdot 10^{-6}$ seconds. This is largely suitable for real-time predictions since a single value is predicted every few minutes. The training and testing times of other tests with different prediction horizons are in the same range and are not reported for brevity. All tests are coded in Matlab R2018b and performed on an HP Pavilion Notebook with Windows 10 Home 64-Bit Operating System, 2.6 GHz Intel Core i7 CPU, 16 GB DDR4-2133 SDRAM.

5. Blood Glucose Level Forecasting on Type-1-Diabetes Subjects during Physical Activity: A Comparative Analysis of Different Learning Techniques

Although regular physical activity is important for people living with T1D for a variety of health reasons [80], most subjects are unable to manage the consequences because the intensity [81], duration, and type of physical activity have a significant impact on glucose homeostasis [82, 83], which can lead to potential episodes of hypoglycemia or hyperglycemia. The former is mainly observed during aerobic physical activity, whereas the latter is observed during anaerobic exercise.

Despite the excellent performance obtained by predictive models, the prediction of abrupt changes in blood glucose values produced during sports remains one of the main challenges in this sector. Physical activity has rarely been addressed in the literature, mainly due to the difficulties in data collection. In addition, the works addressing this task usually resort to multivariate approaches that ask the patient to manually supply data concerning their state or integrate their data from several heterogeneous sensors [84, 85], or consider a dataset composed of virtual patients [86, 87].

In this study [88], we exploit the data of six adults suffering from T1D who regularly perform physical activity to develop a predictive model capable of effectively forecasting future glucose levels during sports. The experimental work has seen the application of a Jump Neural Network [23] to perform a regression task with a univariate approach. This latter choice was made to reduce as much as possible the burden of the patient without requiring them to supply data manually or to wear unnecessary sensors. A precision medicine approach was adopted, which proved to be the most effective from

the experimental results. Concerning the methods of application of the algorithm, three different configurations were developed, which include: a model with offline training, a model with online training, and a model with online training and a loss function that incorporates an increased penalty in case of a great difference between the predicted and real values. A comparative analysis was finally conducted, to determine whether the performance of a model with offline training could be exceeded by models with online updates, with or without penalty contribution.

5.1. Dataset

The Unit of Endocrinology and Diabetology of the Campus Bio-Medico Polyclinic of Rome supplied anonymized data of six subjects suffering from T1D, aged between 23 and 52 (average 39 ± 10), that held regular physical activity and exploited CGM during the period of observation. Data were accompanied by information provided by the patients regarding the days and times in which physical activity was performed and its type. In detail, three of the participating subjects performed anaerobic activities (gym, sailing, and home workouts for patients 1, 2, and 3, respectively), whereas the remaining performed aerobic activities (padel/bicycle, belly dance, and eight-a-side soccer for patients 4, 5, and 6, respectively).

For each patient, several sensor disconnections occurred during the monitored period, which makes it difficult for an AI model to learn the patterns of a time series [26]. As a consequence, we decided to include in this study only those days for which the monitoring was continuous for 24 hours. After excluding incomplete days, the amount of collected data ranged from a minimum of 6 days for patient 1 to a maximum of 81 for patient 4 (average 31 ± 28), and a number of physical activity events from a minimum of two for patient 2 to a maximum of eight for patient 4 (average 5 ± 2).

5.2. Methods

With the purpose to make a prediction to 30 minutes, in the labeling process, the CGM value of the timestamp $(t + 30)$ was considered as a target for the training of the CGM sample at timestamp t. The three configurations developed in this work used a Jump Neural Network, which is a particular feed-forward neural network whose inputs are connected not only to the first hidden layer but also to the output layer. This model was originally proposed by Zecchin et al. [23] for predicting the blood glucose levels of 10

subjects with T1D in daily-life conditions exploiting a MTS approach. The network used in this work presents only one hidden layer composed of four neurons with a sigmoidal activation function. In practice, it differs from the original model of Zecchin because it has a different number of hidden neurons and exploits a UTS approach. A schematic illustration of the proposed jump neural network is shown in Figure 5.1.

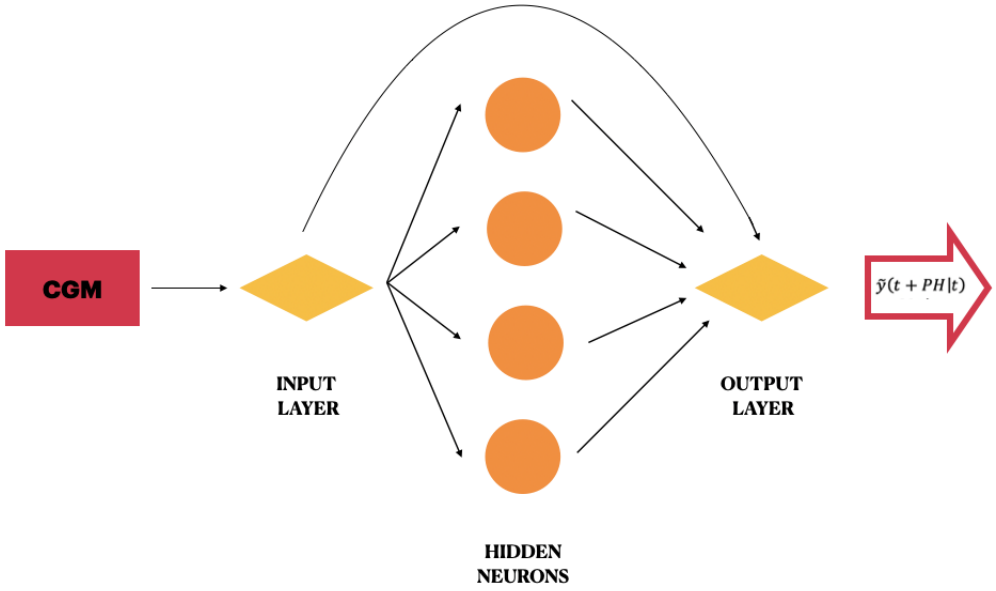


Figure 5.1.: Schematic illustration of the proposed Jump Neural Network.

The regression task was conducted following a precision medicine approach. At each prediction timestamp t, the network input consists of the most recent 10 minutes of CGM values. The network output is the future blood glucose value at timestamp $(t + PH)$. PH is the prediction horizon, i.e., how far forward in time a prediction is performed, and, in this work, it is set to be equal to 30 minutes. At each timestamp t, the jump neural network predicts a signal that can be expressed as

$$\tilde{y}(t + PH|t) = \mathbf{IOW} \cdot \mathbf{I}(t)^T + \mathbf{HOW} \cdot f(\mathbf{IHW} \cdot \mathbf{I}(t)^T) \tag{5.1}$$

where

- $\mathbf{I}$(t) is a row vector with M elements corresponding to the CGM $[t - M + 1, t]$. We experimentally found the optimal value for $M = 10$.
- **IOW** is a row vector with M weight elements directly connecting every input to the output neuron.

- **HOW** is a row vector of L weights connecting every hidden neuron to the output neuron.
- **IHW** is a $L \times M$ matrix of weights connecting every input to every hidden neuron.
- f is the tangent-sigmoid activation function, computed element-wise on the results of $\mathbf{IHW} \cdot \mathbf{I}(t)^T$.

The first term models the linear relationship between the target and the inputs, whereas the second term models the nonlinear relationship.

5.2.1. Experimental Design

For each patient, each group of continuous days is considered a stand-alone group of data. With regard to the network input, the CGM of each group of continuous days is used both as a vector of features and as a vector of labels. In detail, if a group of data is composed of N minutes of CGM, the timestamps ranging from 1 to $(N - PH)$ are considered as features, and the timestamps ranging from $(PH + 1)$ to N are considered as labels. The vectors are then reorganized in such a manner that, at each timestamp t, the network receives as an input 10 minutes of CGM data (i.e., the time window $[(t - 9), t]$) that is associated with the label corresponding to the value of CGM at the timestamp $(t + 30)$.

In each of the proposed configurations, and for each patient, in the beginning, offline training is carried out on the initial 80% of data of the first 24 hours of the first group of continuous days, and the model is validated on the remaining 20% of the data of that same day. The network performance during the offline training is evaluated in terms of Mean Squared Error (MSE). Defining $y(t)$ as the true CGM value at timestamp t and $\tilde{y}(t)$ as the related model prediction, we can define the error at a timestamp t as

$$e(t) = y(t) - \tilde{y}(t) \tag{5.2}$$

and the MSE as

$$MSE = \sum_{t=1}^{N} \frac{e(t)^2}{N} \tag{5.3}$$

where N is the number of timestamps of the predicted time series. After the first offline training, the process diversifies according to the configuration used:

- Offline training configuration: after training on the first available 24 hours of data, all the following days are considered as test sets of the offline trained model, and

the performance is evaluated. A schematic illustration of the first configuration behavior is shown in Figure 5.2;

- Online training configuration: after the training of the offline model, the jump neural network uses the online update mode. The weights obtained from the offline-trained model are used to initialize the online configuration. At each timestamp, the model makes a prediction of the blood glucose value at $(t + PH)$. Every five timestamps, the configuration of the network is updated using the 24 hours of CGM immediately preceding the test timestamp, and the training and test windows are moved forward by 5 min. The online training performance is evaluated in terms of the MSE. This process is iterated until all of the patient's CGM samples have been considered as test sets so that every prediction is performed after training the model with the most recent 24 hours of data. Therefore, this configuration presents a considerably greater computational burden compared with the previous one;

- Online training configuration with penalty: the model is trained online and works similarly to the second configuration; however, the performance during the online training phase is evaluated through the MSE product with a penalty. The latter was generated starting from the difference between the current value and the predicted value.

A schematic representation of the latter two configurations is illustrated in Figure 5.3. As introduced, the third configuration is characterized by a custom loss function built *ad hoc*, which is composed of the product between the squared errors and the penalty variable:

$$penalized\,loss = \sum_{t=1}^{N} \frac{e(t)^2 \cdot penalty(t)}{N} \tag{5.4}$$

where the latter is defined as follows:

$$penalty(t) = \begin{cases} 0, & \text{if } e(t) \leq 5 \\ 1, & \text{if } 5 < e(t) \leq 10 \\ 2, & \text{if } 10 < e(t) \leq 20 \\ 0.5 \cdot e(t), & \text{if } e(t) > 20 \end{cases} \tag{5.5}$$

The construction of the penalty is designed to make the model act more rigidly when the error is larger, trying to improve the prediction on those timestamps where the

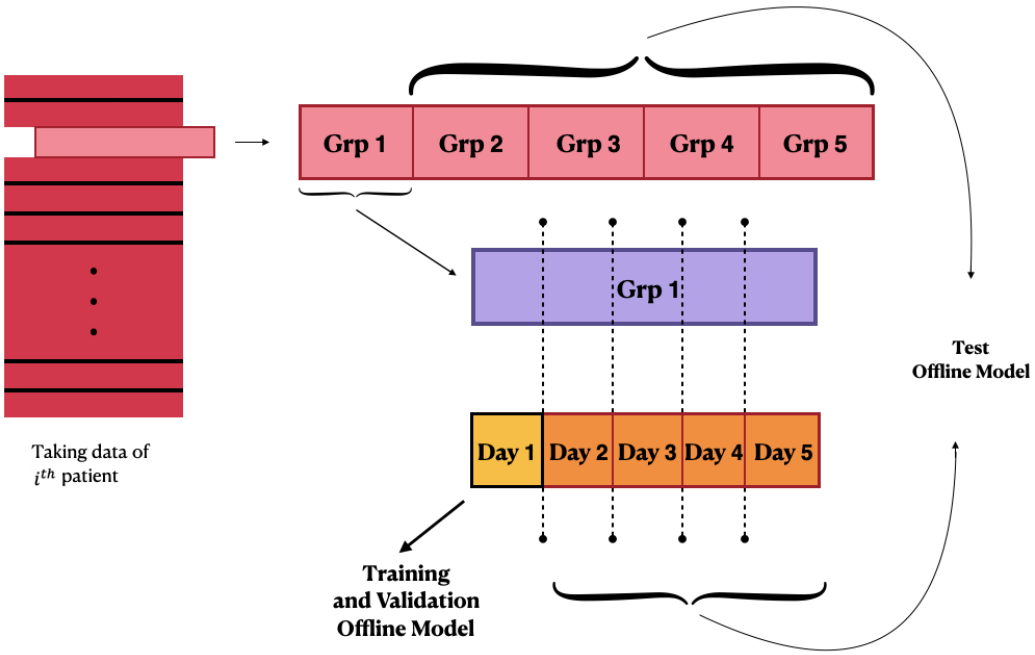


Figure 5.2.: Schematic illustration of the first configuration's behavior.

error is great while taking less into account those timestamps where the error is small; conversely, errors whose magnitude is under a set threshold are considered as correct predictions.

5.3. Results and Discussion

All the simulations were run using Google Colab and the open-source libraries of Keras and TensorFlow. In the offline training phase, a learning rate of 0.1 and a maximum number of epochs equal to 500 were set for the network training. In order to prevent overfitting, the early stopping technique was also used, i.e., the network finished its training if the validation set had no improvement in terms of loss for 10 consecutive validation checks carried out every four epochs. With regard to the configurations involving online training, gradient clipping was exploited to prevent the gradient explosion observed during preliminary tests, and its value was set to 0.3 from the experimental results. The learning rate was set to 0.01.

The final performance of the configurations was evaluated in terms of RMSE in order to provide an immediate comparison with other works in the literature. Different types of performance were considered for the three configurations: first, the average RMSE

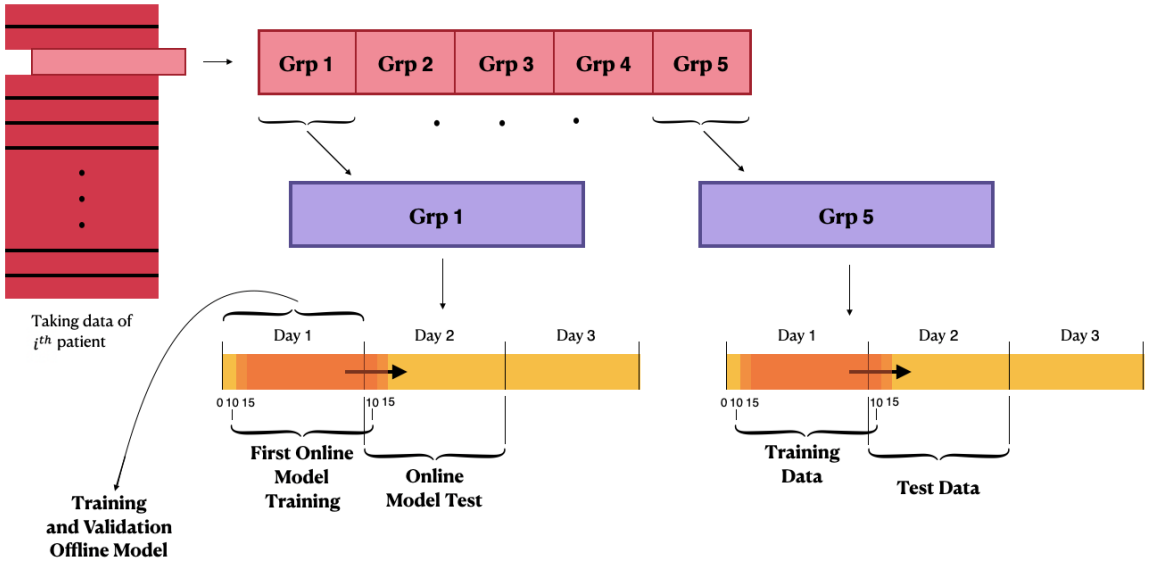


Figure 5.3.: Schematic illustration of the second and third configurations' behavior.

was calculated for each patient by taking into account the total of days in the test set; second, the RMSE associated only with the timestamps in which each patient performed physical activity was calculated; finally, the average RMSE of all patients and for each type of physical activity was computed.

5.3.1. Offline Training Configuration

In the offline configuration, the model was trained from scratch only once for each patient, exploiting the first 24 hours of data available, and tested on all the following days. Table 5.1 reports the results of this test for each patient. The second column reports the results referring to the total amount of recorded days, whereas the third refers only to the timestamps during which physical activity was performed. The bottom panel of the table reports the mean of the RMSEs concerning the total days and the timestamps associated with physical activity. In detail, the two bottom lines report the average RMSE referred only to the timestamps in which anaerobic or aerobic exercise was performed. Interestingly, the average RMSEs related to predictions associated with exercise were better than those on the total of days. The patients for which the best performance was achieved for predictions during exercise were not the same as those that provided the best predictions on the total of days, both concerning aerobic and anaerobic exercise. The average results regarding physical activity show that the model

Table 5.1.: Results of the tests of the offline configuration. The results are reported in terms of average RMSE (mg/dL). For each patient, we report whether they performed aerobic (AE) or anaerobic (AN) exercise. The second column reports the results referring to the total amount of recorded days, whereas the third refers only to the timestamps during which physical activity (PA) was performed. The average RMSEs are also reported. The bottom panel reports the average RMSE related only to anaerobic and aerobic exercise.

Patient ID	**RMSE Total Days**	**RMSE PA**	**PA Type**
Patient 1	22.0	23.2	AN
Patient 2	20.7	21.6	AN
Patient 3	25.1	17.8	AN
Patient 4	22.8	29.6	AE
Patient 5	29.0	23.7	AE
Patient 6	29.9	25.6	AE
Average RMSE	24.9	23.5	-
Average RMSE—AN	-	20.8	-
Average RMSE—AE	-	26.3	-

was better at predicting the anaerobic type of activity over aerobic.

5.3.2. Online Training Configuration

In the online configuration, the model was trained from scratch only once for each patient and then updated every time new data were available. Table 5.2 reports the results of this test for each patient. The structure and meaning of the elements shown in the table are the same as in Table 5.1. Again, the results concerning physical activity were better than those concerning the total amount of days. The average results were similar to those achieved using the offline configuration. The results regarding physical activity show that this model was also better at predicting anaerobic activity over aerobic.

5.3.3. Online Training Configuration with Penalty

In the online configuration with the penalty, the model was trained from scratch only once for each patient and then was updated every time new data were available, while considering a penalty that was greater as the forecast error increased. Table 5.3 reports the results of this test for each patient. The structure and meaning of the elements shown in the table are the same as in Table 5.1. The results concerning physical activity

Table 5.2.: Results of the test of the online configuration. The results of performed tests are reported in terms of the average RMSE (mg/dL). For each patient, we report whether they performed aerobic (AE) or anaerobic (AN) exercise. The second column reports the results referring to the total amount of recorded days, whereas the third refers only to the timestamps during which physical activity (PA) was performed. The average RMSEs are also reported. The bottom panel reports the average RMSE related only to anaerobic and aerobic exercise.

Patient ID	RMSE Total Days	RMSE PA	PA Type
Patient 1	22.0	23.3	AN
Patient 2	20.1	21.0	AN
Patient 3	24.2	18.9	AN
Patient 4	22.8	30.2	AE
Patient 5	29.7	24.7	AE
Patient 6	28.3	25.2	AE
Average RMSE	24.5	23.9	-
Average RMSE—AN	-	21.1	-
Average RMSE—AE	-	26.7	-

were better than those concerning the total amount of days, and, once again, the model was better at predicting the anaerobic type of activity over aerobic. The average results were similar to those achieved using the previous configurations.

5.3.4. Comparison between the Three Configurations and the State of the Art

Table 5.4 summarizes the results achieved with the three configurations, together with the average results on the total days and on days with aerobic and anaerobic exercise. All the configurations achieved similar results, both concerning the total days and the physical activity. All the models performed better when predicting glucose levels related to anaerobic exercise rather than aerobic; this may be a sign that the abrupt glycemia decreases that occur during aerobic physical activity are particularly difficult to predict accurately.

During aerobic exercise, glycemic variation can be influenced by the intensity and duration of exercise, insulin to glucagon ratio, fitness, nutrition, and initial glucose concentration [89]; moreover, most patients consume snacks immediately before aerobic exercise to prevent hypoglycemic events, and this causes an increase in glucose levels

Table 5.3.: Results of the test of the online configuration with the penalty. The results of performed tests are reported in terms of the average RMSE (mg/dL). For each patient, we report whether they performed aerobic (AE) or anaerobic (AN) exercise. The second column reports the results referring to the total amount of recorded days, whereas the third refers only to the timestamps during which physical activity (PA) was performed. The average RMSEs are also reported. The bottom panel reports the average RMSE related only to anaerobic and aerobic exercise.

Patient ID	**RMSE Total Days**	**RMSE PA**	**PA Type**
Patient 1	23.0	24.2	AN
Patient 2	21.1	22.2	AN
Patient 3	26.0	17.3	AN
Patient 4	21.7	28.7	AE
Patient 5	27.8	22.4	AE
Patient 6	28.5	28.5	AE
Average RMSE	24.6	23.9	-
Average RMSE—AN	-	21.2	-
Average RMSE—AE	-	26.5	-

followed by a decrease due to exercise [90]. A multivariate approach taking into account such heterogeneous features may improve glucose predictions during aerobic physical activity.

Although the offline configuration achieved better results on the total number of days for only one out of the six patients compared to the online configuration without penalty, it remains the best choice for physical activity. Indeed, the offline configuration achieved better average performance for both aerobic and anaerobic physical activity. On the other hand, the offline model was outperformed by the other configurations concerning the performance on the total days. The fact that better results of the configurations with online training on the total number of days did not translate into better results concerning only physical activity could indicate that these configurations have a better long-term adaptation.

However, the performance improvement is not great enough to justify the significant increase in the computational burden introduced by the online configurations; thus, the offline configuration would likely be most appropriate for use in a real-life application. With regard to the online configurations, the performance of the penalty and non-penalty configurations were very similar, indicating that the negative reinforcement did not present an advantage. With regard to the individual performance on the total number

Table 5.4.: Results of the test of all three configurations. The results of the performed tests are reported in terms of the average RMSE (mg/dL). For each patient, we report whether they performed aerobic (AE) or anaerobic (AN) exercise. The results were calculated both for the total number of days and only for the CGM forecasts associated with physical activity (PA). For the latter, in addition to the total average, the average of the RMSEs associated only with anaerobic and aerobic PA is also reported.

Patient ID and PA type	**Total Days offline**	**Total Days online**	**Total Days online penalty**	**PA Offline**	**PA Online**	**PA Online penalty**
Patient 1—AN	22.0	22.0	23.0	23.2	23.3	24.2
Patient 2—AN	20.7	20.1	21.1	21.6	21.0	22.2
Patient 3—AN	25.1	24.2	26.0	17.8	18.9	17.3
Patient 4—AE	22.8	22.8	21.7	29.6	30.2	28.7
Patient 5—AE	29.0	29.7	27.8	23.7	24.7	22.4
Patient 6—AE	29.9	28.3	28.5	25.6	25.2	28.5
Average RMSE	24.9	24.5	24.6	23.5	23.9	23.9
Average RMSE—AN	-	-	-	20.8	21.1	21.2
Average RMSE—AE	-	-	-	26.3	26.7	26.5

of days, we observed that some individual differences may affect the prediction of the three models.

The configuration with online training achieved the best performance for Patients 2, 3, and 6, whereas the online and offline training configuration equally achieved the best performance for Patient 1, and the online configuration with penalty achieved the best performance for Patients 4 and 5. As no clear pattern was observed using one configuration over another, we must conclude that the small variations in performance may be due to individual differences between patients. A similar analysis can be applied to the timestamps related to PA. Figure 5.4 shows a graphical comparison of the predictions of the three configurations on a sample day of Patient 3, who performed anaerobic physical activity, and Patient 5, who performed aerobic exercise; the timestamps in which physical activity was performed are also highlighted.

As mentioned in the Introduction, the forecasting of the blood glucose levels of T1D patients that perform physical activity has rarely been addressed in the literature mainly due to the difficulties in collecting data. Despite the reduced amount of published works, a partial comparison can be performed with the work of Hobbs et al. [85], which is likely the most remarkable work resorting to a regression task. They achieved an average RMSE of 29 mg/dL on 32 T1DM subjects who practiced skiing and snowboarding, which are considered to be mainly anaerobic types of exercise.

All the configurations proposed in this work achieved considerably better performance

on anaerobic physical activity, although the number of patients investigated is exiguous. Furthermore, differently from other works in the literature, the proposed model exploits only the previous knowledge of CGM data without requesting the patients to manually provide information on their status [77, 84] or integrating data from several heterogeneous sensors [85], making it a promising approach for future developments. However, we acknowledge that the results achieved cannot be conclusive nor exhaustive because they were achieved on a small number of patients, and therefore they need to be validated on a larger dataset to be considered definitive.

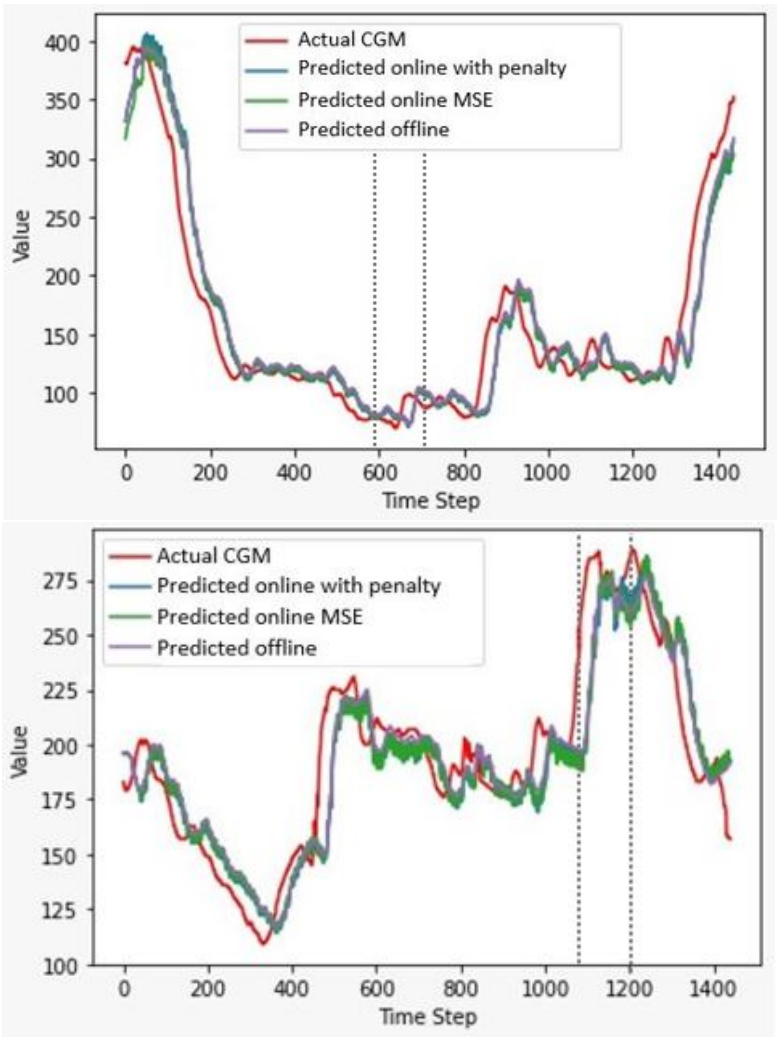


Figure 5.4.: Comparison between the predictions of the three configurations on two sample days. We report the actual CGM track (red line), the predictions of the online configuration with the penalty (blue line), the predictions of the online configuration (green line), and the predictions of the offline model (purple line). All the predictions are almost overlapped and slightly shifted from the original CGM values. The black dotted lines delimit the period during which physical activity was performed. **Top:** Predictions on a whole day of Patient 3, who performed anaerobic exercise (RMSEs between 24.2 and 26.0 mg/dL). **Bottom:** Predictions on a whole day of Patient 5, who performed aerobic exercise (RMSEs between 27.8 and 29.7 mg/dL).

6. Identification of Optimal Training for Prediction of Glucose Levels in Type-1-Diabetes Using Edge Computing

Although predictive models for blood glucose levels achieve very promising performance, their application is limited in practice by the fact that most studies have investigated only the predictive capability of such models, whereas few studies have focused on real-life applications. In the latter case, the CGM sensor should be able to continuously communicate with the predictive model, which can be on cloud services (i.e. on the internet) or an Edge Computing system (i.e., on a computational board). Edge Computing systems are based on a distributed computing model in which data processing takes place as close as possible to where the data are collected [91]; this approach leads to a solution that does not require continued connection to cloud services, reducing the risk of system malfunctioning. For the above-mentioned reasons, this study [92] aims to identify a model capable of predicting future blood glucose levels taking into account both numerical accuracy and computational resources.

6.1. Dataset and preprocessing

The UVA/Padova simulator [32], which was approved in 2008 by the FDA as a replacement for preclinical animal trials, is used to generate data from 6 adult T1D patients. For each patient, 100 days of data are generated with 1-minute sampling, in such a way that, during each day, patients can have 3 to 5 meals; each meal has a reference CHO intake taken from the Dietary Reference Intakes for Carbohydrate [93]. For each considered day, breakfast, lunch, and dinner are always present and set at reference times 8:00, 13:00, and 20:00, with reference carbohydrate intake of 45 grams, 70 grams, and

80 grams, respectively. One or two snacks at 11:00 and/or 17:00 are randomly included with a reference carbohydrate intake of 20 grams. To create more realistic data, the exact time of each meal is obtained by shifting the reference time by a random amount of time taken from a uniform discrete distribution in the interval [-60, +60] minutes, whereas the exact amount of CHO for each meal is obtained by adding a casual value extracted from a uniform discrete distribution in the interval [-20, +20] grams to the reference carbohydrate amount. As an example, each day patients have lunch at some point between 12:00 and 14:00, with an amount of CHO that varies between 50 and 90 grams.

Frequent hypoglycemic and hyperglycemic events are intentionally generated in order to investigate the behavior of the model to facing critical cases, and to make the generated dataset more similar to the observed clinical data [11, 12]. This is achieved by modifying the optimal insulin bolus value computed by the simulator itself in the occurrence of each meal, in order to simulate human error. The modified value is extracted in the same way as for CHOs, except that the interval from which the corrective value is randomly taken is [-3, +3] units of insulin. The dataset consists of two features: blood glucose and Insulin-On-Board (IOB). The latter is a combination of basal insulin and insulin bolus which represents an estimation of the amount of insulin still active in the subject's body after bolus injection. For the Insulet pump, which is the one considered during the simulations in this study, the active insulin time is equal to 3 hours and its action has a linear plot [94]. Thus, the value of IOB for each timestamp t is computed as:

$$IOB(t) = \sum_{k=0}^{179} a(k)I(t-k) \tag{6.1}$$

where $I(t-k)$ represents the value of insulin injected at timestamp $(t-k)$, and $a(k) = (180-k)/180$ is the coefficient corresponding to the insulin decay curve discretized in accordance with a 1-minute timestamp of insulin delivery, and $k = 0, 1, 2, ..., 179$ are the total timestamps for the 3 hours of active insulin. We intentionally decided not to consider further features provided by the simulator, such as the amount of ingested CHO, as an input of the predictive model: in this way, patients are not asked to manually provide any feature.

Following the precision medicine approach, each patient is considered separately. After raw data are created, a Z-score standardization is applied to the features of the dataset to obtain a normal distribution of the values with zero mean and unit standard deviation. Finally, the dataset is split into windows consisting of 30 minutes of data

(input of the model) and one sample as the label (output of the model), being so ready to be fed to the neural network. A PH is fixed to determine how forward in time the prediction is performed. Thus, at each timestamp t, the estimated future CGM value $\widehat{CGM}$ is a function of CGM and IOB values from the previous 30 minutes:

$$\widehat{CGM}(t + PH) = f(CGM[t - 29, \ldots, t]; IOB[t - 29, \ldots, t]) \tag{6.2}$$

where the PH is set to 30 minutes. This is a sensible choice because it is the most widely adopted value in the literature, as it would allow sufficient time advance to prevent an adverse event [42].

6.2. Methods

Recurrent neural networks represent the golden standard for time series forecasting, but in their classic implementation, they are afflicted by some limitations when facing long time series data [95]. Since the presented dataset consists of a 100-day time frame for each patient, an LSTM architecture is selected, i.e., a specific recurrent neural network variant that is more suitable to handle long-term dependencies [96]. An LSTM is in synthesis a recurrent neural network where the single cell at each considered timestamp t contains an internal memory vector, or state vector, c_t that defines its state as described in the following. The units of the network are composed of an input cell, an output cell and a forget gate. Considering the matrices $\mathbf{W}$ input weights, $\mathbf{R}$ recurrent weights, and $\mathbf{b}$ bias for each gate, the state of the cell at timestamp t defined as

$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{g_t} \tag{6.3}$$

whereas the hidden state is defined by

$$\mathbf{h_t} = \mathbf{o_t} \odot \sigma(\mathbf{c_t}) \tag{6.4}$$

where the terms, for a given input $\mathbf{x}(t)$, are referred to the equations governing the gates. In detail, for the input gate:

$$\mathbf{i_t} = \sigma(\mathbf{W_{in}}\mathbf{x}(t) + \mathbf{R_i}\mathbf{h_{t-1}} + \mathbf{b_i}) \tag{6.5}$$

for the forget gate:

$$\mathbf{f_t} = \sigma(\mathbf{W_f}\mathbf{x}(t) + \mathbf{R_f}\mathbf{h_{t-1}} + \mathbf{b_f}) \tag{6.6}$$

for the cell candidate:

$$\mathbf{g_t} = \sigma(\mathbf{W_g}\mathbf{x}(t) + \mathbf{R_g}\mathbf{h_{t-1}} + \mathbf{b_g}) \tag{6.7}$$

and, finally, for the output gate:

$$\mathbf{o_t} = \sigma(\mathbf{W_o}\mathbf{x}(t) + \mathbf{R_o}\mathbf{h_{t-1}} + \mathbf{b_o}) \tag{6.8}$$

where σ is the sigmoid activation function.

6.2.1. Experimental design

At first, to determine the optimal hyperparameters for the proposed LSTM, the first 10 days of data of each patient are taken and divided into three subsets, namely training (70%), validation (20%), and test set (10%). These days are excluded from the dataset for successive tests. A grid search on the validation set is performed, as an optimization stage, to identify the combination of hyperparameters resulting in a better and faster performance of the neural network over data from all the patients. As an evaluation criterion, RMSE between the true and the forecasted time series is used. The investigation of optimal parameters is performed over the learning rate, the number of cells of the LSTM layer, the optimizer, and the activation function. This preliminary search led to a Recurrent neural network characterized by a 3-layer architecture (16-cell LSTM layer, a dense layer, and an output layer), a learning rate of 0.01, Adam as an optimizer, and Rectified Linear Unit (ReLU) activation function for the dense layer. The training is run using a maximum of 200 epochs with a mini-batch size of 1400. Furthermore, the early stopping technique is implemented. To evaluate the performance of the different configurations, two training approaches are tested on the whole dataset, namely windowed and cumulative, which will be described in the next section.

All the following tests are performed on a dataset composed of 90 days for each patient to keep the analysis unbiased, as the first 10 days were leveraged in the optimization phase:

1. at first, to characterize the minimum training set size required to achieve promising performance, the model behavior is analyzed using training sets ranging in size from 1 to 10 days;

2. subsequently, the extent to which the performance improves as the number of days used for the training set increases is investigated. Two different approaches are considered, namely cumulative and windowed. In the cumulative approach, 7 different training sets are analyzed, the first consisting of 10 days, the second consisting of 20 days (including the previous 10), up to the training set consisting of 70 days. In the windowed approach, the 7 training sets are all composed of 10 consecutive, non-overlapping days. The comparison is carried out using test sets containing the days immediately following the last training day;

3. finally, an analysis is carried out on training sets ranging in size from 10 to 80 days, i.e., the cumulative approach was applied again by keeping the test set fixed to the final available days.

6.2.1.1. Edge system

The presented model is implemented in Python using the open-source libraries of TensorFlow and Keras. The optimization stage is performed utilizing the Google Colaboratory environment; the identified model is thus run on the edge device. Raspberry Pi 4 is used as hardware; it includes a quad-core Cortex-A72 1.5GHz processor with 8GB of RAM; additional information can be found in the datasheet [97]. The model is converted to the .tflite format to allow a faster prediction [98].

6.3. Results and Discussion

The results from the 3 tests can be summarized as follows:

1. with regard to the minimum size of the training set, it is observed that a wider training set leads to a larger decrease in the RMSE value. In detail, the system can provide reasonable predictions from day one, corresponding to an average RMSE of 14.5 mg/dL; the RMSE decreases down to 12 mg/dL if 10 days of training are leveraged. These results are comparable to those presented in the literature, where an RMSE of 15 to 20 mg/dL is usually achieved on data of real patients [13].

2. a comparison of the average performance of the windowed and the incremental approaches is reported in Figure 6.1. The cumulative approach always achieves better results, whatever the size of the training set. The results are equal for a training set of size 10 days because the two training sets coincide. This shows that

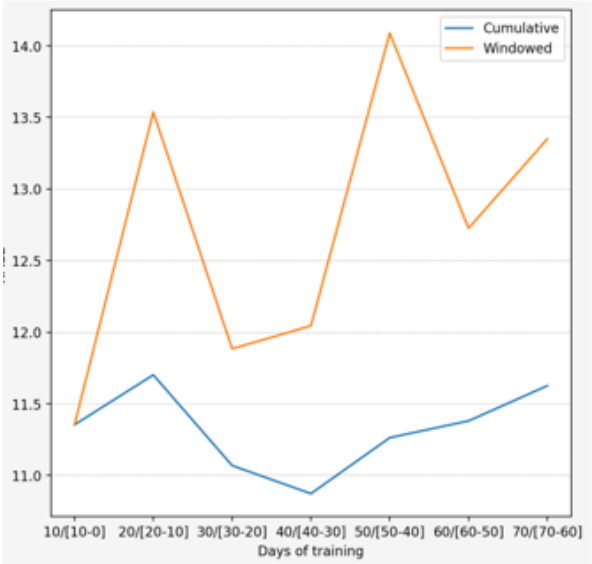


Figure 6.1.: Comparison of the average results of the proposed approaches.

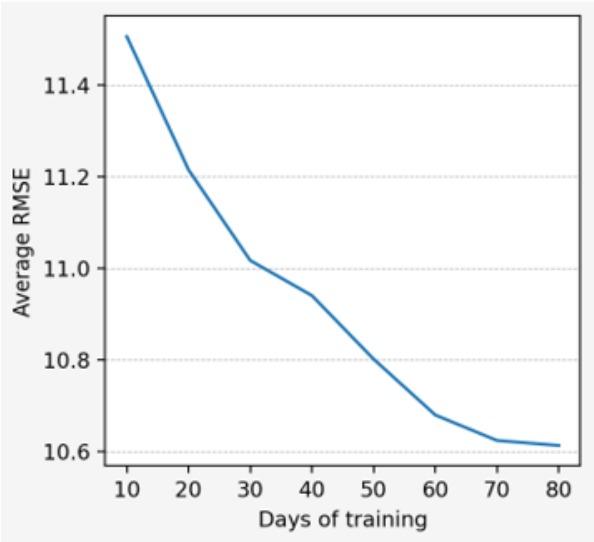


Figure 6.2.: Average test RMSE for different sizes of the training set.

Table 6.1.: Profiling of the required time to execute different parts of code as the size of the training set changes.

Training Days	Dataframe Acquisition	Preprocessing	Train	Overall
10	40.8 s	8.1 s	123.2 s	3.3 min
20	77.5 s	15.7 s	209.3 s	5.5 min
30	113.2 s	23.8 s	308.3 s	8.4 min
40	149.5 s	30.8	403.9 s	11.0 min
50	191.2 s	38.2 s	567.0 s	14.0 min
60	227.6 s	46.1 s	780.9 s	18.0 min
70	263.9 s	53.6 s	833.3 s	16.0 min
80	294.7 s	59.9 s	904.9 s	21.0 min

increasing the size of the training set leads to better results than just considering the 10 most recent days for training and updating the model.

3. the performance variation for larger sizes of the training set is reported in Figure 6.2. As it can be observed, training the network over 60 days leads to an improvement in numerical performance on a fixed test set, and RMSE decreases down to 10.7 mg/dL. Extending the training set further leads to a performance plateau. Indeed, the RMSE improves slightly, reaching an average value of 10.6 mg/dL after 80 days are used for training. The results are similar to those reported in the literature by models including more parameters that require a much longer training time [99].

The time necessary for these training procedures on the edge system is reported in Table 6.1; in addition to the total times, the times for dataframe acquisition, standardization, batch splitting, and model training are listed. The only atomic data not reported in Table 6.1 are the times for the conversion in tf.lite format, because they are not dependent on the dimensions of the dataset but only from the model size, which is the same for all the different training procedures. Therefore even if these were slightly different, these differences could be attributed to the daemon processes performed by the edge device.

As pointed out, the model performance achieves a plateau after a training size of 60 days. As shown in Table 6.1, training the model on 60 days of data requires 18 minutes; this makes the proposed approach feasible in real-life applications [100]. The

RAM used in each simulation is about 1GB, which is by a large margin under the 8GB available for the device, thus not leading to a decrease in performance. This is achieved also thanks to the shallow architecture chosen for the LSTM, which includes a limited number of parameters compared to deep neural networks which are usually exploited for glucose levels forecasting [34]. Future analysis may focus on the identification of the optimal hardware for running such algorithms, in terms of minimum computational capacity and resources; indeed, considering the global shortage of semi-conductors, resorting to cheaper and more available devices that do not decrease performance could be an effective strategy to implement the mass distribution of the system.

7. Prediction of Glucose Concentration in Children with Type 1 Diabetes Using Neural Networks: An Edge Computing Application

Despite the large number of works presented for the forecasting of future glycemic levels and the noteworthy results they achieve, all the aforementioned papers focus on the prediction of glycemic levels of adult subjects. Indeed, few works in the literature aim to predict blood glucose levels specifically in pediatric patients. Children represent the most challenging diabetic population because pediatric patients go through a period of rapid growth, and physiological and hormonal changes along with complex individualization and socialization processes. This often results in a significant decline in the quality of disease management, treatment adherence, and glycemic control [101, 102].

Normally, machine learning techniques are validated on laboratory setup, and, when they are applied in practice, they are performed directly on servers or centralized processing units. The task of future glycemic levels prediction makes no exception, as most systems performing real-time prediction exchange data between an edge device, only used to gather information, and the cloud, where the actual glucose level forecasting is performed [100, 103]. This is mainly due to the memory limits of edge-computing devices. Nonetheless, the drawback of such systems is that they constantly require an internet connection to work; this is not arguable about medical devices, because an interruption in the signal may result in missing decision support to the user. However, the increasing development of new, more powerful, and dedicated hardware, combined with the widespread use of IoT (Internet of Things) tools, is enabling the emergence of a branch of artificial intelligence known as inference at the edge [64, 65].

The contribution of this study [91] is twofold. On the one hand, two state-of-the-art models for the prediction of glycemic levels are implemented, and they are applied to

the specific task of blood glucose levels forecasting in pediatric patients; such models improve the performance of the models currently studied in this field. On the other hand, these models are implemented on an edge computing system, thus laying the foundations for the future creation of embedded devices capable of forecasting blood glucose levels to improve patients' quality of life and aid medical diagnosis; the feasibility of such a prediction-at-the-edge system is evaluated on two different boards in terms of prediction accuracy and execution time.

7.1. Materials

Data were produced for 10 pediatric patients by running several simulations in the UVA/Padova simulator [32]. Such a tool allows the generation of different scenarios for *in silico* patients by only providing a meal schedule. The simulator can determine the optimal insulin boluses to be injected for each meal of a specific patient and thus provide the glycemic evolution for each subject for a pre-set number of days. However, the tool allows the user to modify the insulin bolus value and include a sensor error in the CGM readings. Data are generated with a 1-minute sampling.

Two different datasets were generated on a scenario consisting of 30 days of simulation for each patient, with 5 meals per day. The first set consists in a scenario that has no errors in sensor reading and insulin administration, as automatically computed by the simulator, and thus corresponds to an ideal T1D management. Differently, we created the second scenario using the same meal schedule as the first scenario, but by including CGM sensor errors and by forcing the presence of hyperglycemic and hypoglycemic events. We were able to achieve such a goal by first allowing the UVA/Padova simulator to simulate its optimal bolus control; then, we extracted the vector of injected boluses and added random noise taken from a uniform distribution. The modifications were made using the same strategy described in Section 6.1. The modified bolus vector was given as an effective bolus vector to the UVA/Padova to run the simulations for this scenario. This makes such a scenario more realistic because in real life the increase or decrease in blood sugar levels occurs mainly due to an inaccurate estimate of the number of carbohydrates ingested, or to deviations in correction dosing [104]: noise was added on insulin boluses to simulate the human error.

The datasets consist of information on blood glucose levels and data on insulin (bolus, basal, and injection were added together and considered as one) and finally carbohydrate intake. Specifically, the final datasets consider IOB as an insulin feature, which was

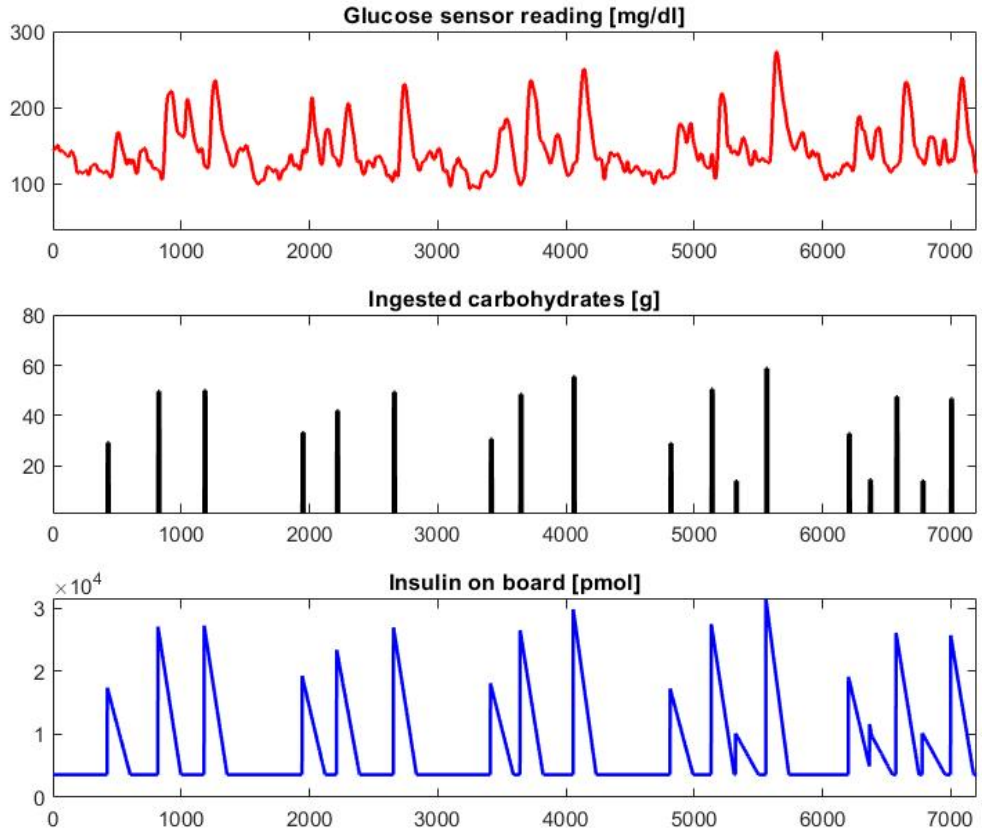


Figure 7.1.: Graphical example of 5 days of data generated for patient child#007. Many hyperglycemic (Blood Glucose Level $> 180\,mg/dL$) values can be observed due to the modification of the optimal bolus values.

manually generated by exploiting a mathematical model [94] as described in Section 6.1. IOB is a quantity that refers to the amount of rapid-acting insulin still active in the patient's body after bolus injection, and thus provides deeper information on the recent history of insulin injections compared to the punctual insulin values themselves. A graphical example of 5 days of data concerning the CGM sensor reading, the ingested carbohydrates, and the IOB of a sample patient, generated with a 1-minute sampling using the simulator and the pre-processing are reported in Figure 7.1.

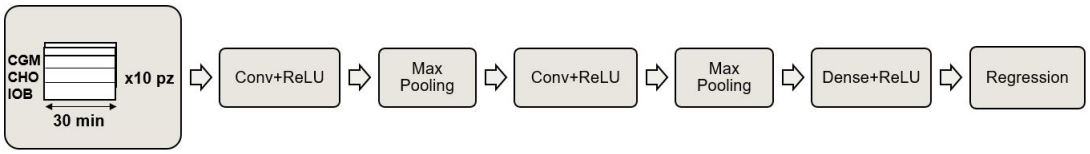


Figure 7.2.: Schematic representation of the proposed Convolutional Neural Network.

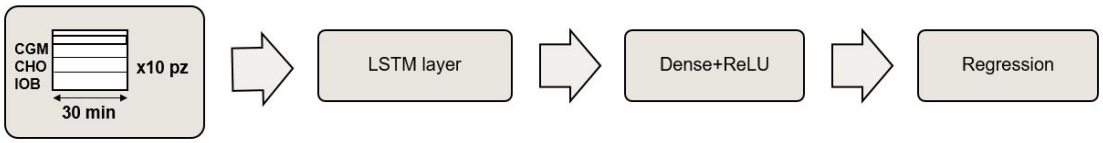


Figure 7.3.: Schematic representation of the proposed LSTM Recurrent Neural Network.

7.2. Methods

A precision medicine approach was used to tune the predictive models, which involves choosing the hyperparameters optimally and individually for each different subject. In this work, we implemented and optimized a CNN and an LSTM recurrent neural network, because such models achieve the most promising performance in the literature [105]. Both networks were trained using a subset of the available data and then tested on subsequent data of the same *in silico* patient without being updated again. The networks have a sequence-to-label architecture, as the expected output is a single value corresponding to the expected blood glucose value in 30 minutes. After splitting the data into Training (70%), Validation (20%), and Test set (10%), the models were built.

The proposed CNN is a 1D-CNN, with a one-dimensional kernel, consisting of two convolutional layers with ReLU activation function, each followed by a MaxPooling that cuts the parameters in half by taking, in pairs, only the largest value. To complete the model, the convolutional layers are followed by a dense layer with a ReLU activation function, and an output neuron that provides the final regression. A schematic representation of the proposed CNN model is reported in Figure 7.2. The choice of hyperparameters was made by performing a grid search on the validation set, based on a range of parameters including values identified through preliminary tests and parameters reported in the literature [105]. The optimization was done concerning the kernel size and the number of feature maps.

The proposed LSTM model consists of a first LSTM layer, a dense layer with a ReLU activation function, and an output layer that returns the predicted CGM value. Also in this case, the model was optimized in terms of the number of neurons in the first LSTM layer and the dense layer by investigating both parameters identified in preliminary

tests and parameters reported in the literature [105]. A schematic representation of the proposed LSTM model is reported in Figure 7.3.

Both models take as input a (3×30) matrix of values, corresponding to the last 30 minutes of the 3 feature values. Such a parameter was identified in preliminary tests, as it provides the models with enough information to capture the recent trend of the features. We found empirically that using longer monitoring periods did not improve performance. With regards to the strategy chosen to train both networks, the Stochastic Gradient Descent (SGD) optimizer is adopted, which requires a learning rate (0.0001), a momentum (0.9), and a clip Value (0.5), which is a necessary parameter to prevent the gradient explosion phenomenon in deep neural networks, improving the prediction quality. The training of both models was performed by splitting the data into mini-batches of 1400 samples (i.e., approximately one day of data) and setting the maximum number of epochs to 200. Finally, to prevent overfitting, the early stopping strategy was adopted, which stops training if the performance on the validation set does not improve within a fixed number of consecutive epochs.

Two different evaluation metrics are used to evaluate the performance of the models. RMSE (equation 1.1) is utilized to assess numerical accuracy, as it provides a numerical estimate of how close the predicted values are to the real ones. In addition, the CEGA is considered as a measure of the clinical accuracy of the predictions produced [30].

7.2.1. Edge system description

In order to test the feasibility of implementing and utilizing the predictive models on an edge system, we needed to identify the target hardware. Our choice fell on two different devices: a Raspberry Pi4, chosen for its low cost and high computational capability, and a Coral DevBoard, a developer kit containing a Tensor Processing Unit (TPU) processor wthat accelerates the execution of machine learning models. The Raspberry Pi4 has a Broadcom BCM2711 quad-core Arm Cortex A72 of 1.5GHz processor, with 4 GB of memory. Furthermore, to be able to carry out the tests, we chose to use Raspbian OS (a Debian-derived ISO) as the operating system. Python and Mendel Development Tool (MDT) were also installed. The former is necessary to perform tests directly on the Raspberry; the latter is used to give commands to the Coral DevBoard and therefore allows its set-up and use. The Coral Devboard has a quad Cortex-A53, Cortex-M4F CPU, with 1 GB LPDDR4 RAM, and it has a 4 TOPS (8bit) TPU accelerator for machine learning processes. The operating system running on the DevBoard is Mendel Linux. We installed and utilized all the dependencies necessary to run the model on the

board using the Py CoralAPI.

7.2.2. Edge system implementation

Both datasets were provided as input, as sequences of the last 30 minutes of values, for two models compared: CNN and LSTM. The models were implemented and trained on Google Colab through the use of the open-source libraries of Keras and TensorFlow. Through this API, the networks were trained, and the hyperparameters were optimized.

Although the single models were trained on two different datasets, topologically, the trained networks do not differ in terms of hyperparameters. Therefore, the number of algebraic operations performed by a single network is invariant to the dataset. Having made this consideration, we decided to implement on the edge device only the models trained on the dataset including more hypo/hyperglycemic events, as it is more similar to a real use case.

For the implementation of the models on edge computing architectures, it is necessary to perform a quantization step that differs depending on the architecture on which inference is going to be performed. In order to perform regression tasks on the Raspberry, we chose to use the quantization in *.tflite* format, that transforms the model keeping output variables in *float32* format. This optimization, namely dynamic range quantization, provides latency close to fully fixed-point inference. However, the outputs are still stored using a floating point so that the speedup with dynamic-range operations is less than a full fixed-point computation, as reported on the official TensorFlow web page [106]. From now on we will refer to the model obtained with this quantization as *.tflite*.

For the implementation on the Dev Board, it was necessary to transform the models in their 8-bit representation to execute them exploiting the full potential provided by Coral's TPU. In this case, the quantization method to be used is known as full integer quantization. Applying this approach requires providing a representative dataset to calibrate variable tensors such as model input, activation functions, outputs of intermediate layers, and model output. As a representative dataset, it would theoretically be sufficient to provide a set of 100-500 sample data, taken between the training and validation set. In our case, a dependence of the goodness of the quantization on the subset of data passed to the model as a representative dataset was noted. In fact, it was not sufficient to use data taken randomly from the training or validation set but it was necessary to use ordered data, given the time series forecasting nature of the task. At the end of this quantization procedure, all input and output values are taken to *uint8*. From now on we will refer to the model obtained with this quantization as *uint8*.

Due to the 8-bit nature of the quantization required to exploit the capabilities of the Coral Devboard TPU processor, a problem arose for the regression task. The range of values of the dataset varies between 10 and 600 mg/dL, whereas the values that can be represented with 8 bits are 256. Consequently, we pursued two approaches. The first consists in avoiding any pre-processing of the input data and then reconstructing the possible overflow cases obtained in the output through post-processing of the data, maintaining the granularity of the prediction at 1 mg/dL. The reconstruction was done following the procedure set out in the algorithm 1. It assumes that a decrease in glucose concentration of more than 50 mg/dL in a single minute is very unlikely or impossible. In this case, we post-process the prediction and sum 255 to the predicted value.

Algorithm 1 Output reconstruction algorithm

```
reconstructed_pred = []                                  ▷ initialization of variables
overflow = False
deltaY = 50
For i,x in enumerate (tflite_uint8_model_prediction):    ▷ Start of the for loop
if x >= 240 then
    if overflow and (x - tflite_uint8_model_prediction[i-1]) >= deltaY: then
        overflow = False
    else if  not overflow and (x - tflite_uint8_model_prediction[i+1]) >= deltaY: then
        overflow = True
delta = 255 if overflow else 0
reconstructed_pred.append(x + delta)                     ▷ End of the for loop
```

The second approach consists in the application of a normalization step in the pre-processing phase, remapping the data values between 0 and 255. Such an approach avoids overflow-related problems, but it takes the granularity of the prediction to approximately 2.33 mg/dL. Then, we de-normalized the predicted values to compute the evaluation metrics. This could introduce inaccuracy in the predictions.

The Raspberry and DevBoard were used for the calculation of inference times, to be compared with the performance limits that our application requires (less than the sampling period of the sensor, i.e. 1 minute). At each timestamp, the edge system takes as input the 30 most recent values of the features (i.e., the data of the *in silico* patient produced by the simulator), computes the latest value of the IOB, and performs a prediction of the future blood glucose level. A representative schematic of the experimental system can be seen in Figure 7.4.

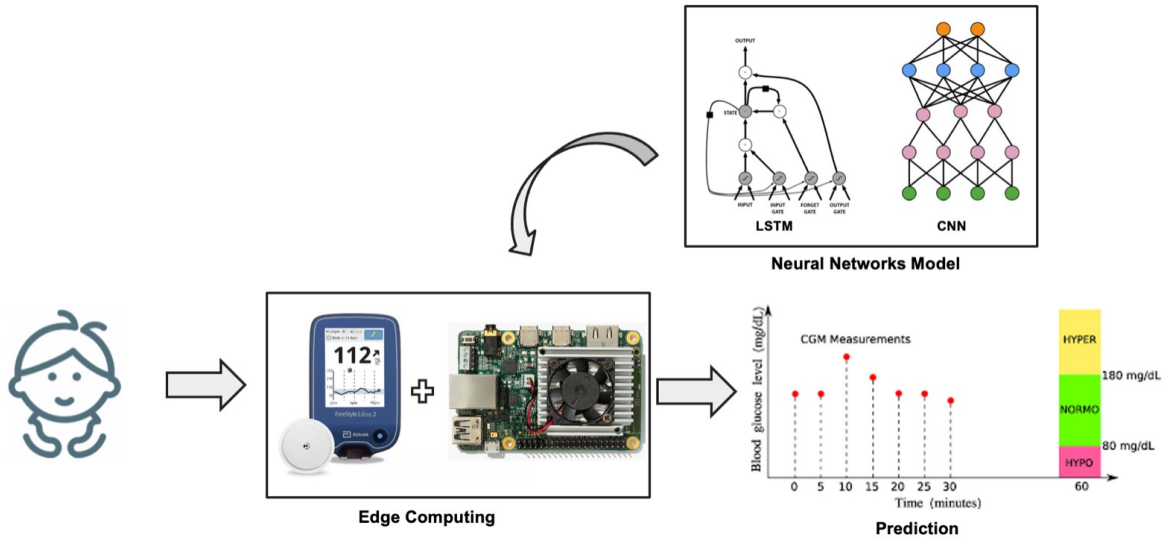


Figure 7.4.: Schematic representation of the experimental setup during the test phase with edge systems.

7.3. Results

As a result of the grid search performed on the discovery set, the optimal configuration of the CNN comprises 26 filters in the first convolutional layer, 20 filters in the second convolutional layer, and a Kernel size equal to 1x5 on both. Note that, due to the shape chosen for the filters and to the structure of the input matrix, in the first CNN layer the convolutions are performed on different timestamps of the same feature. With regards to the LSTM model, the optimal configuration resulted in 64 neurons for both the LSTM and the fully-connected layer. Once the models were optimized, predictions were performed on the Test set, and the RMSE and the CEGA were computed. With regards to the CEGA values, only those from the second dataset were evaluated, as they present more hypo- and hyperglycemic values and are thus more similar to scenarios observed in real life [12].

Table 7.1 reports the average values and their standard deviation of the tests performed using the different versions of the models. As expected, the results achieved by the baseline model on the standard dataset are better than those achieved on the dataset with outliers. The LSTM model outperforms the CNN on both datasets in terms of average RMSE and CEG results. In particular, concerning the realistic dataset, the LSTM achieves an RMSE of $16.3 \pm 4.7\, mg/dL$, which is noteworthy if compared to other studies presented in the literature concerning the prediction on pediatric T1D patients.

Table 7.1.: Results of the tests performed with the proposed models CNN and LSTM. In this test, the normalization step was not performed in the pre-processing phase. The results refer to the RMSE $[mg/dL]$ achieved on both the ideal (no-error) and the realistic (hypo-hyper) dataset. Such results are reported in terms of average RMSE ± standard deviation. The CEGA results are referred only to the realistic dataset, and its results are reported as percentages on the total dataset. For each neural network, we reported the results for the model implemented on Google Colab, for the model implemented on Raspberry (*.tflite float32* format), and for the model implemented on the Dev Board (.tflite *uint8*).

Model	RMSE (no-error)	RMSE (hypo-hyper)	CEGA (A;B;C;D;E)
CNN	22.2 ± 2.5	23.2 ± 2.3	87.0; 12.0; 0.0; 1.0; 0.0
LSTM	13.5 ± 3.4	16.3 ± 4.7	93.8; 5.2; 0.0; 1.0; 0.0
CNN *.tflite*	/	23.6 ± 2.0	85.7; 13.6; 0.0; 0.7; 0.0
LSTM *.tflite*	/	16.3 ± 4.7	93.7; 5.2; 0.0; 1.1; 0.0
CNN *uint8*	/	40.1 ± 11.1	75.4; 20.8; 0.0; 1.2; 2.5
LSTM *uint8*	/	35.0 ± 13.3	82.4; 12.5; 0.0; 1.5; 3.6

Also, 99.0% of its predictions fall in zones A and B of the CEGA and thus represent clinically accurate or acceptable predictions, whereas 1.0% of predictions fall in zone D. The latter mainly correspond to failures of predicting hypoglycemia. No predictions fall in zones C and E.

A comparison with the results achieved in the literature can be only partial because few studies are addressing the prediction task on pediatric patients, and only one of them exploits the UVA/Padova simulator. The model tested on data from 4 real pediatric patients by Mougiakakou et al. [107] that achieves an average of 22.1 mg/dL RMSE is outperformed by both the proposed models; however, it is known that forecasting glycemia of real patients is though compared to virtual patients because some unpredictable events might be present. De Bois et al. [108] tested the same 10 virtual children of the UVA/Padova simulator we utilized; they achieved an average RMSE of 5.2 mg/dL that outperforms both the proposed models in all configurations in terms of numerical accuracy; nonetheless, the clinical accuracy of their best model (zones A+B) is 97.5% and it is outperformed by our models, which both achieve accuracy above 99.0% in their best configuration. However, it must be considered that the two datasets have been generated with different meal and bolus schedules, so this comparison is qualitative.

7.3.1. Edge system results and discussions

The results reported in Table 7.1 refer to the models trained without having carried out the normalization of the input values. The expected increase in the RMSE values of the models implemented on the edge devices can be observed; however, this variation differs between the two quantized representations of the networks. With regards to models quantized using dynamic range quantization for implementation on the Raspberry, the RMSE values increase by a maximum of 0.4 mg/dL for the CNN, whereas there is no difference for the LSTM. Again, the LSTM model outperforms the CNN in terms of numerical accuracy, achieving an RMSE of $16 \pm 4.7\, mg/dL$, and 98.9% of its predictions fall in zones A and B of the CEGA. This result is of particular interest because it is similar to the performance achieved on datasets composed of data of adult T1D patients, and it is achieved on the edge device, without resorting to cloud computing. A graphical example of the predictions is reported in Figure 7.5, where we report as an example data of two patients for whom the best and the worst performance is achieved in terms of RMSE. The LSTM prediction is closer to the true CGM value compared to the CNN, which produces more oscillatory predictions; however, the LSTM tends to overestimate both hyperglycemic and hypoglycemic peaks.

Nonetheless, it is worth noting that only 0.7% of predictions of the CNN model fall outside the A and B zones of the CEGA, compared to 1.1% of the LSTM; conversely, the LSTM produces more predictions that fall in zone A (93.7% against 85.7% of the CNN). This may be explained considering that the LSTM is more capable of performing accurate predictions in the euglycemic range, which translates into better RMSE and a larger percentage of predictions in zone A, whereas it may miss some hypoglycemic events; on the contrary, the CNN has a larger RMSE and a larger number of predictions in zone B of the CEGA, corresponding to errors in the euglycemic range, whereas it is more capable to predict hypoglycemia. Examples of the CEGA are shown in Figure 7.6, where we report as an example data of two patients for whom the best and the worst performance is achieved in terms of CEGA percentage in zone A. In conclusion, the CNN may be more appropriate to predict critical hypoglycemic events when implemented in *.tflite*, although its average numeric accuracy is worse than that of LSTM. However, it should be taken into account that results achieved on virtual patients are, in general, slightly better than those obtained on real patients, thus performance may deteriorate when testing on a real dataset.

A different analysis applies to the models on which the full integer quantization was performed for implementation on the Coral DevBoard. Indeed, this quantization tech-

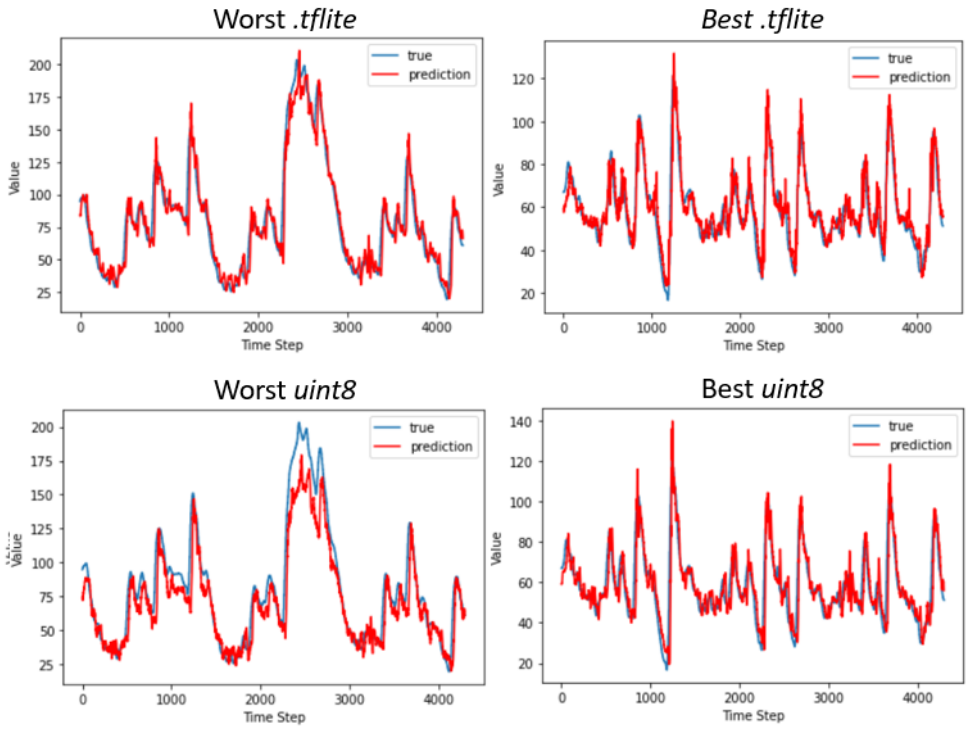


Figure 7.5.: Graphical examples of the best and worst predictions performed by the CNN (left) and LSTM (right) using different edge devices. We computed the confidence interval for the predicted values, which are 2.01 for the worst *.tflite*, 2.14 for the worst *uint8*, and 1.09 for either the best *.tflite* and *uint8*, respectively. Nonetheless, we do not report such an interval in the figure because its values are too small to be observed in the graphics. The glycemic index values shown in the figure are normalized between 0 and 255, thus, to obtain the real glycemic values, we need to multiply by 2.33.

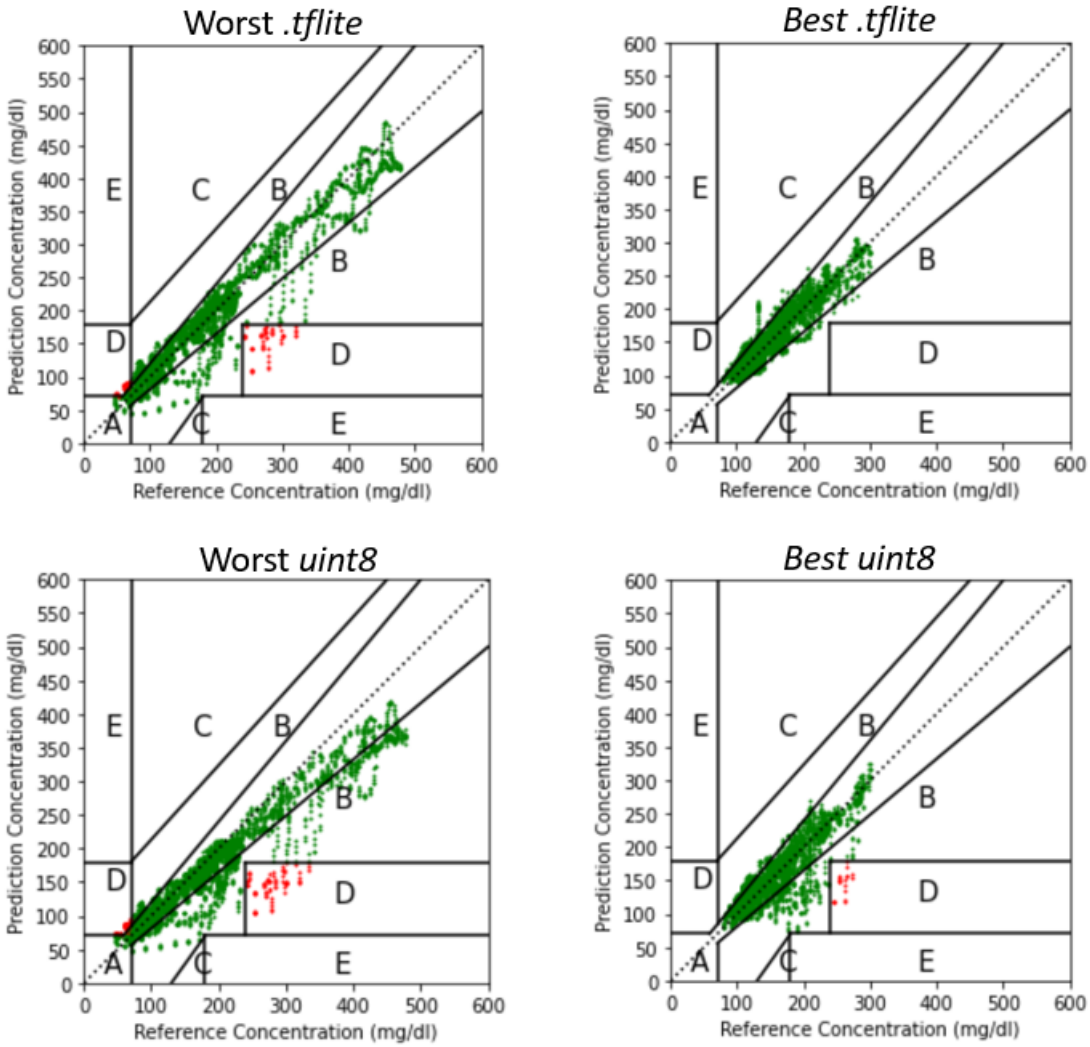


Figure 7.6.: Clarke Error Grids resulted by the best and worst predictions of the CNN (left) and LSTM (right) using different edge devices. Predictions falling in the safe zones A and B are plotted in green; predictions in zone C are plotted in yellow; predictions falling in the dangerous zones D and E are plotted in red.

Table 7.2.: Results of the tests performed with the proposed models CNN and LSTM, on which was carried the normalization step in the pre-processing phase. The results refer to the RMSE [mg/dL] achieved on the realistic (hypo-hyper) dataset. Such results are reported in terms of average RMSE ± standard deviation. The CEGA results are referred only to the realistic dataset, and its results are reported as a percentage of the total dataset. For each neural network, we reported the results for the model implemented on Google Colab, and for the model implemented on the Dev Board (.tflite uint8 format).

Model	RMSE (hypo-hyper)	CEGA (A;B;C;D;E)
CNN	21.8 ± 2.3	87.8; 10.9; 0.0; 1.1; 0.0
LSTM	16.0 ± 3.4	93.7; 5.5; 0.0; 0.8; 0.0
CNN *uint8*-normalized	24.7 ± 5.5	87.6; 9.8; 0.0; 0.9; 0.0
LSTM *uint8*-normalized	21.2 ± 8.6	87.4; 7.5; 0.0; 5.1; 0.0

nique, which casts the values from *float32* to *uint8*, has more significant effects on the goodness of prediction. In particular, the overflow that is observed when glycemic values are above 255 mg/dL considerably increases the RMSE scores and generates some predictions that fall in the dangerous E zone of the CEGA. For this reason, as explained in section 7.2.2, two different approaches were chosen. The second one, which involved an initial pre-processing of the data, gave considerably better results than the first one, and they are reported in Table 7.2. In particular, the results obtained for the models in Google Colab do not differ substantially from those achieved without the normalization; conversely, the *uint8* implementation of such models achieves considerably better performance than those obtained with the first approach. It must be considered that the granularity of the prediction increases from 1 mg/dL to 2.3 mg/dL. Despite this drawback, we can still consider this approach better than the first one, because the increase in granularity obtained is not critical from a clinical point of view. It is worth noting that, although the LSTM model outperforms the CNN in terms of RMSE (21.2 ± 8.6 and $24.7 \pm 5.5\, mg/dL$, respectively), 5% of the predictions produced by the LSTM fall in the D zone of the CEGA, corresponding to a failure of predicting dangerous events. This situation shows the LSTM model to be weaker than the *uint8* representation, which brings it a greater drop in accuracy. This is probably due to the narrowness of the model, which has only one LSTM plane. Given the limited number of mathematical operations required to achieve an output, the conversion step of the model to *uint8* fails to optimize the weights with the new integer values. On the contrary, only 0.9% of the predictions produced by the CNN fall in the D zone, proving that this latter model is

Table 7.3.: Maximum inference time obtained in the test phase in milliseconds. The inference times are reported for each model. They were calculated: for the models saved in TensorFlow saved model format over the Colab online TPU, for the *.tflite* model format over the Raspberry and the *.tflite* format quantized in *uint8* over the Coral DevBoard.

Model	Colab TPU (TF Saved Model)	Raspberry (*.tflite*)	Coral DevBoard (*.tflite uint8*)
CNN	0.085	101.56	18
LSTM	0.086	70.3	12

more clinically accurate and reliable when implementing the models in *uint8*, despite the better numerical accuracy achieved by the LSTM model.

A further comparison between the different implementations concerns the actual inference times obtained, which returned largely satisfying results. We reported in Table 7.3 the worst-case results for each model and hardware to show compliance with the time constraints posed by the application. The inference times for both models in all three representations are far below the limit imposed by the application, i.e. 1 minute. However, the total times in the case of a real application should also take into account the times necessary for: signal collection by the sensors, pre-processing of the raw data, and displaying the results on an appropriate Graphic User Interface (GUI). Nonetheless, the times for a single inference operation to be summed are, in the worst case, the ones of the CNN performed in *.tflite* format by the Raspberry, corresponding to 101.56 ms. We can therefore assert that inference times, covering at most 0.17% of the total time limit imposed by the application, are not one of the parameters to be optimized in the case of a real implementation of the system. Furthermore, looking at Table 7.3 and comparing the data obtained in the tests of the two Edge systems, a consistent acceleration can be observed with the use of the Coral DevBoard when compared to the Raspberry's performance, although it does not reach the performance of Google Colab TPU. This result is in line with Google's claims [109].

8. Layered meta-learning algorithm for predicting adverse events in Type 1 Diabetes

Works in the literature attempting to reduce the number of these events suffer from some open issues. First, most models only focus on predicting future blood glucose levels with a regression task [13, 58]. As such, regression predicts future glucose levels regardless of whether they are in the hypoglycemic or hyperglycemic range. It has been proven by recent works that predicting adverse glycemic events using classification rather than regression leads to improved performance [29, 38].

Second, the vast majority of studies focus only on the prediction of hypoglycemia [39, 40, 41, 42, 43, 44]. It is a sensible choice because this condition can arrive unannounced even in the most severe cases, leading to serious short-term complications. In this regard, in a recent review on machine learning techniques for hypoglycemia prediction, Mujahid et al. [42] stated that *"is important to understand that hypoglycemia prediction is blood glucose level prediction in essence"*. Nonetheless, most of such works mainly aim at maximizing the true positive rate at the expense of a considerably low precision score, which is often not reported [39, 40] or impossible to compute [45, 46, 38, 47, 41, 48]. Indeed, it is acknowledged that any prediction algorithm has to "decide" between raising a lot of alerts to detect all events (good recall, bad precision, a lot of false positives) or trying to minimize the nuisance of the patient (good precision, limited false positives, at the expense of a lower recall). Works focusing on hypoglycemia prediction usually choose the former approach [49], with few exceptions [47]. It reduces patient engagement with the technology.

Third, predicting glycemic excursions, and in particular incoming hypoglycemic events, is a very challenging task. Although a wide literature exists about the prediction of glycemic events, spanning from regressive models [48] to ensemble models [39] and cutting-edge technologies such as deep neural networks [38], none of such models can

fully represent the complex rules lying behind the different glucose dynamics of T1D patients. It also happens because the datasets utilized to build such models are usually limited in size. Recently, meta-learning has proven to solve and improve the generalization of few-shot tasks that would be unsolvable by training from scratch [110]. A new study from Zhu et al. [50] successfully used model-agnostic meta-learning to enable fast adaptation of a neural network for forecasting future glycemic levels of T1D patients. However, this approach requires a second, patient-personalized fine-tuning phase, which could require weeks of data gathering and manual labeling from the physicians.

Finally, some works focus only on the sample-based approach [46, 40, 47, 41]. This is a limitation, because such an approach may lead to overestimating the performance, generating high recall scores because correctly predicted continuous hypo/hyperglycemic samples count as several true positives, whereas the event may have not been predicted in advance.

For the reasons above, we propose [111] a meta-learning system based on a multi-expert predictive model relying on an event-based approach. The experts consist of either Recurrent LSTM or CNN. We aim to develop a model capable to achieve a good trade-off between the amount of correctly predicted events (i.e., high recall per class) and the number of false alarms (i.e., high precision per class) while evaluating performance on a public dataset. We consider a 30-minute (6-timestamp) PH since it would be a sufficient time to warn patients about incoming adverse events [22]. We evaluate the effective advance by which predictions are performed by introducing a parameter α, evaluating performance as α varies. Due to the strong imbalance between the classes, we use a Leave-1-Patient-Out Cross-Validation approach to maximize the number of samples from the minority classes in the discovery set. Such an approach would also provide users with a ready-to-use model which does not require a fine-tuning period on patient-specific data. In addition, we aim to develop a univariate approach to make the predictive models more suitable for real-life applications. By not requiring the user to utilize different devices for data recording, it could be usable by patients that exploit only CGM for therapy while reducing the computational burden required to combine several heterogeneous data. Moreover, previous works have shown that using several input features besides CGM does not improve performance sensitively without a computationally expansive preprocessing [112, 16], which is likely to be avoided when performing tasks on edge devices [91]. Finally, we implement the proposed system on an edge-computing device to evaluate the real-life feasibility and applicability of the proposed approach.

8.1. Materials

8.1.1. Public Validation dataset (Ohio)

The Ohio T1DM dataset was initially available to participants in the first and second Blood Glucose Level Prediction (BGLP) Challenge in 2018 and 2020 and then became publicly available to other researchers. In this work, the original format [11] and its expansion [12] are considered as a single dataset. It contains eight weeks of data concerning continuous glucose monitoring, insulin, physiological sensor, and self-reported life-events of twelve adults suffering from T1D (five females and seven males, aged between 20 and 60, each using the Medtronic $Enlite^{TM}$ CGM sensor and a fitness band), all following a Continuous Subcutaneous Insulin Infusion therapy(CSII). More detailed information about the dataset can be found in [11, 12].

The dataset is already split into a training and a test set for each patient; however, since we aimed to perform a Leave-1-Patient-Out Cross Validation, we joined the training and the test sets of each patient to make a single fold. The recorded data report many interruptions; plus, two different fitness bands were used in the first and second releases to record physical data.

We decided to pursue a univariate approach, so CGM sensor data is used alone as an input feature of the proposed model. In order to test the multivariate variant of the models, and provide a fair comparison between different approaches, we utilized only the features that are in common between the datasets; furthermore, in order to develop a system as autonomous as possible and to reduce the burden on the patient, we only considered the features collected by sensors and without the direct involvement of the user. After this selection, the four considered features are CGM sensor read values, injected insulin, skin temperature, and galvanic skin response.

8.1.2. Private Validation Dataset (UCBM)

The Unit of Endocrinology and Diabetology of Campus Bio-Medico University (UCBM) Polyclinic provided anonymized CGM data of five T1D patients (all males), all using Dexcom G5 CGM sensor, aged between 32 and 43 (average 38.6±5), glycated hemoglobin (HbA1c) between 5.7 and 8.4, weight between 67 and 95 kg, daily insulin requirement per kg between 0.07 and 0.85 UI/Kg/die (average 0.49 ± 0.29). Three patients use CSII, whereas two follow Multi-Injection Therapy. Every patient was monitored for a period ranging from 3 to 14 days (average 8 ± 3.8), for a total of 40 days, during which they

regularly performed physical activity. Predicting glucose levels of T1D patients during physical activity is particularly tough due to quick variations occurring [88]. It is worth noting that the patients from the UCBM dataset utilize a different CGM sensor than patients from the public dataset.

8.1.3. Data Preprocessing

As aforementioned, many disconnections occurred during the data recording period concerning both the CGM sensor and the fitness band. In general, this leads to complications when training a time-series model. To minimize complications and allow a comparison between the performance of the UTS and the MTS approach, we included in the dataset only the timestamps in which all the considered features were available at the same time for at least 12 consecutive timestamps (60 minutes). Indeed, in this work, we found that the size of the input sequence of 6 timestamps (i.e. the latest 30 minutes) provides optimal results. Since a PH of 30 minutes is being considered, consecutively recorded sequences shorter than 60 minutes would not provide a ground truth value to evaluate the effectiveness of the prediction. Also, we excluded from the analysis the 6 timestamps preceding and following a sensor calibration or disconnection, since huge variations of glycemia were present during such events, resulting in noisy data for the model training. Next, we composed a different feature matrix for each patient by joining all the portions of data obtained in this way. No further preprocessing was performed on raw data; the only exception concerns the amount of injected insulin: we added the bolus values to the basal insulin rate at the corresponding timestamps. In this way, we joined the basal insulin and the injected boluses into a single insulin feature.

8.1.4. Data Labeling

Data labeling is essential to perform a classification task and properly evaluate the model. Different approaches have been pursued in the literature for the prediction of glycemic events, spanning from binary classification problems [39, 40, 41] to 4-class problems [29]. In this study, we approached a three-class classification task, considering classes hypoglycemia, hyperglycemia, and normoglycemia (euglycemia). We chose well-established thresholds to define classes based on CGM values, considering the following

formal definition:

$$\begin{cases} \text{Hypoglycemia} & \text{if } CGM \leq 70\ mg/dL \\ \text{Normoglycemia} & \text{if } 70\ mg/dL < CGM < 180\ mg/dL \\ \text{Hyperglycemia} & \text{if } CGM \geq 180\ mg/dL \end{cases}$$

For each sample in the dataset, we observe the subsequent 6 timestamps (30 minutes) and act differently according to the values in that time window:

- if a hypo/hyperglycemic value is in the considered time window, then the sample under observation is labeled as either hypoglycemia or hyperglycemia.
- if the sample under observation falls within the hypo- or hyperglycemic ranges, the sample is labeled as either hypoglycemia or hyperglycemia regardless of the values in the following time window.
- if the sample under observation and all the samples in the considered time window are in the euglycemic range, then the sample is labeled as normoglycemia.

Note that this labeling strategy generates "alarms" every time an adverse event is forthcoming or is already happening, whereas it considers as "normal" all the other timestamps. It is also why, differently from other works [29], we decided not to consider severe hypo- or hyperglycemia as classes: the proposed model generates an alarm every time an event is predicted or present, regardless of its severity.

In the sample-based approach, after the labeling step, the public dataset includes 5866 hypoglycemia, 67972 euglycemia, and 38175 hyperglycemia samples, corresponding to about 389 days of data. The Imbalance Ratio, defined as the ratio between the number of samples of the most and the least represented class, is $IR = 11.6$. Thus, the dataset presents a high imbalance ($IR \geq 9$) according to the definition given in [79]. The event-based approach presents 413 events of hypoglycemia, 66786 samples of euglycemia, and 1417 events of hyperglycemia, with a consequent $IR = 161.7$. It indicates a strongly imbalanced dataset [79]. Euglycemia cannot be considered an event. According to the physiological meaning and the labeling strategy we chose, we consider all the normoglycemia samples (every single timestamp) as independent observations (events) in the event-based approach. Following this strategy, the number of observations is slightly smaller due to data rearrangement during the event-based performance evaluation.

The private dataset includes 819 hypoglycemia, 7113 normoglycemia, and 3221 hyperglycemia samples ($IR = 8.7$), corresponding to 55 events of hypoglycemia, 7044 samples

of normoglycemia, and 72 events of hyperglycemia ($IR = 128$).

8.1.5. Edge Devices

The increasing development of new, more powerful, dedicated hardware enables the emergence of a branch of artificial intelligence known as inference at the edge [64, 65]. It involves the machine learning models being run directly from a proximity device using data collected from associated sensors. With the growing interest in the telemedicine approach [66, 67], the inference at the edge can enable predictive models that work in real-time with patient data to improve both medical quality and efficiency. For this reason, to date, several works exploit the potential of edge computing not only from a more methodological and general point of view (e.g., [68]) but also in the field of glycemic level prediction. Zhu et al. [69], for example, proposed an Embedded Edge Evidential Neural Network to predict future glycemic levels of adult T1D patients in real-time by exploiting CGM sensor readings and an edge-computing device.

To test the feasibility of the predictive model implementation and utilization on an edge system, we needed to identify the target hardware. Because of its low cost and high computational capabilities, our choice fell on the Raspberry Pi4. The Raspberry Pi4 presents a Broadcom BCM2711 quad-core Arm Cortex A72 of 1.5 GHz processor, with 4 GB of random access memory. Furthermore, we used Raspbian OS (a Debian-derived operating system) as the operating system to carry out the tests. To limit the experimental time, we chose to carry out these tests using three identical devices. We standardized the data collected during testing and installed the dependencies required to carry out the tests only on one device. Then, the operating system image was copied over two different memory cards and inserted into the other devices to make them clones of the previous one.

8.2. Methods

We propose a meta-learning approach based on a multi-expert system. In particular, we resort to layered meta-learning, in which a base learner models task-specific characteristics while a meta-learner models the features shared by the tasks [110]. As the base learner, we utilized a multi-expert system based on a deep neural network architecture. We evaluated two different architectural approaches, one based on recurrent neural networks (LSTM) and the other based on convolutions (CNN). We selected these

models because they achieve state-of-the-art performance on tasks related to time series, including T1D management [42, 49]. The softmax layer output of each expert is passed to a decision tree (the meta-learner). Figure 8.1 reports the architectural schemes of the two implemented base learners, while Figure 8.2 reports the scheme of the entire system.

8.2.1. Base learner

The base learner is a multi-expert system consisting of three deep neural networks, either Recurrent with LSTM units or with three convolutional layers. We will refer to these multi-expert models as **ME-LSTM** and **ME-CNN**, respectively. The rationale lies in observing that the overall performance on a skewed dataset may be improved by combining the decisions of three different models [113], each specialized in detecting one of the three classes under examination. In other words, in this phase, the original three-class problem is decomposed into three binary classification problems, and, straightforwardly, a binary relabeling was performed before training each expert. During the training of the single expert, a weighted classification layer provides the final decision. We optimized the LSTM and CNN models through a grid search on the number of hidden layers and the number of nodes for each layer. We report further details in paragraph 8.2.3.

LSTM In general, recurrent layers of RNNs consist of recurrent cells which are affected by both past states and current inputs. Almost all the exciting results achieved in the latest years with RNNs have been achieved by the LSTM. Thanks to its ability to learn long- and short-term sequence patterns, it is nowadays considered the state-of-the-art model for time-series forecasting and sequence classification [114]. Each LSTM cell consists of three gates. The first two have a role when updating the cell state: the *input* gate decides what part of the new information will be stored, while the *forget* one what information will be thrown away. The third gate, the *output* one, decides what information can be output based on the cell state.

In this work, a single expert consists of the succession of the following layers: a sequence-input layer, which takes as input an $m \times n$ matrix of features, where m is the number of features and n is the number of recent timestamps to be input; a first LSTM layer of n_h hidden units; a second LSTM layer of $\frac{1}{2}n_h$ hidden units; a fully-connected layer of two units (i.e., one for each class investigated by the expert); a two-neurons softmax layer, which takes the network output values between 0 and 1. We report the schematic representation of the expert structure in Figure 8.1. The proposed model exploits only CGM as an input feature, thus $m = 1$ (univariate approach). In this work,

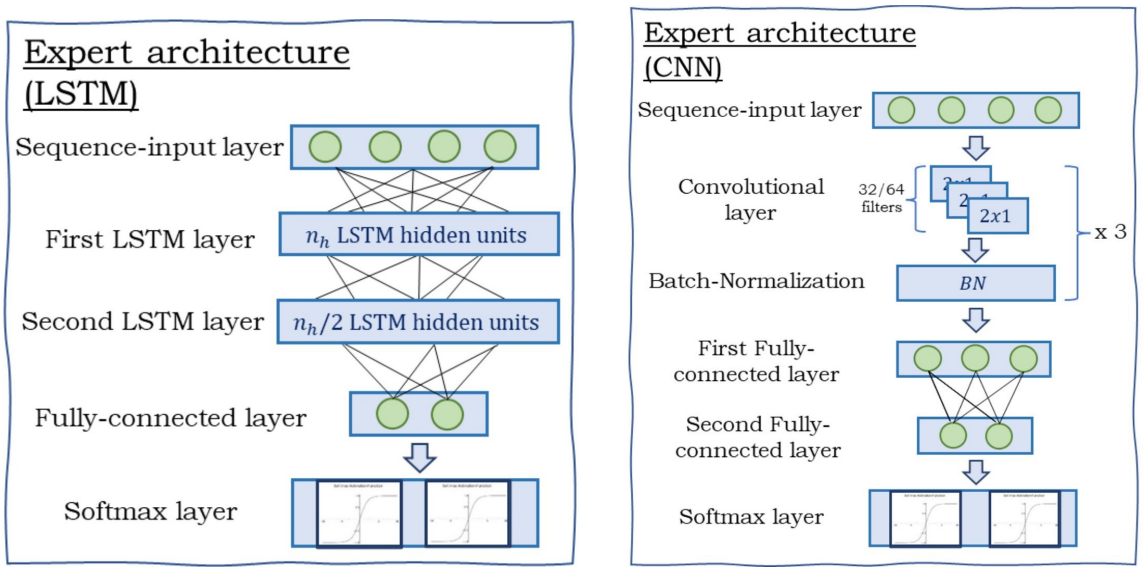


Figure 8.1.: Schematic representation of the expert architectures. Left: the architecture based on the LSTM. Right: the architecture based on the CNN.

we found that a value of $n = 6$ (i.e., the latest 30 minutes) provided optimal results. The value of n_h for each expert was empirically determined as described in section 8.2.3.

CNN Apart from a sequence-input layer (the same as in the LSTM case), each CNN expert involves three convolutional layers with different numbers of filters, also called kernels. In the univariate approach, we fix the filter size equal to 1×2. For each layer, each filter slides (with a stride equal to 1) along one direction (the temporal dimension). At each step, a convolution of the samples (time instants) covered by the filter window is applied. In the multivariate approach, we fix the filter size equal to 2×2, and each filter slides along the two dimensions.

Given the small size of the kernels, we have chosen not to include pooling layers. We applied, instead, a batch normalization layer [115] after each convolutional layer to standardize their inputs among the samples in each batch.

After the last convolutional layer, a dense layer of 64 nodes with the ReLU activation function and a 2-node dense layer with a softmax activation function provide the expert output.

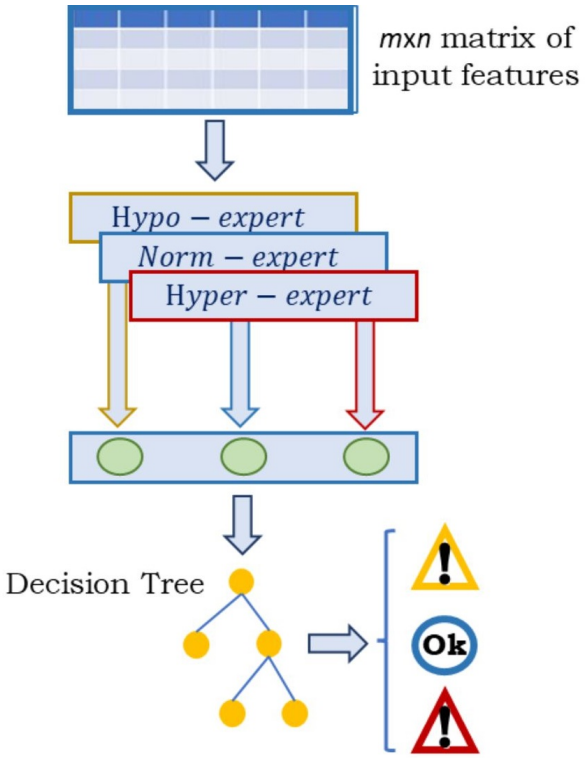


Figure 8.2.: Schematic representation of the meta-learning algorithm and the single experts' architecture.

8.2.2. Meta-learner

Given the outputs of the three experts for an input sample, a straightforward decision strategy could be to compare them and select the class for which its expert model shows the greatest value. We adopt this strategy to evaluate the performance of the base learners (**ME-LSTM** and **ME-CNN**) models. However, given that each expert is trained separately, it is not ensured that just picking the greatest value between the experts' outputs would provide the best choice for assigning the final label. Looking at the proposed architecture in terms of layered meta-learning, each expert in the base learner is utilized to model the characteristics that are specific to its binary classification task. This knowledge is exploited by the meta-learner to model the features shared between the binary classification tasks and the 3-class classification task.

The meta-learner utilized in this study is a CART decision tree, a powerful graph-based method used in machine learning. It is a successive model that unites a series of basic tests (nodes) cohesively, where a numeric feature is compared to a threshold value in each node [116]. Although it can be prone to overfitting, it is highly interpretable compared to artificial neural networks, and overfitting can be limited using pruning. It is characterized by hyperparameters such as the split criterion for nodes (we utilized the Gini diversity index as the split criterion) and a set of parameters optimized during training. The decision tree meta-learner automatically learns the optimal threshold from the outputs of the three experts. As will be discussed in the following sections, we proved that this meta-learner achieved better performance compared to other algorithms. We will refer to the complete systems (base learner and meta-learner) as **ME-LSTM-DT** and **ME-CNN-DT** (Figure 8.2).

8.2.3. Parameter search

Before performing the tests, it is necessary to determine the optimal number of parameters of the base learners, i.e., the number of hidden units n_h of the first LSTM layer of each expert (the number of hidden units of the second LSTM layer is always set equal to $n_h/2$), and the number of filters and kernel size for the CNN. With regard to the meta-learner, we investigated whether or not using pruning or class weights would improve performance. In this phase, we use only the public dataset. Straightforwardly, taking apart data from one patient in each turn, we consider 12 different folds as the discovery set. Then, each discovery set is randomly split into a training (70%) and validation (30%) set.

About the LSTM, we investigate a variable number of hidden units n_h for each expert, ranging from 10 to 100, and evaluate the combination which guarantees the best performance through the medium of a grid search. For the CNN, we investigate the combinations with 32, 64, and 128 channels, considering all the parameter combinations by performing a grid search. During this phase, we train each binary expert on each training set and evaluate its performance with a sample-based approach on the corresponding validation set. Then, we evaluate all the possible combinations of experts to determine the optimal configuration.

As mentioned, this work aims to develop a model capable of achieving high scores for both recall and precision per class. Straightforwardly, to maximize precision and recall per class at the same time, we considered as the evaluation metric the F1-Score: F1-Score=$2 \cdot Precision \cdot Recall/(Precision + Recall)$. In particular, we evaluated the quality of the predictions by measuring the geometric mean G of the F1-Scores per class: G = $\sqrt[K]{\prod_{i=1}^{K} F1\text{-}Score_i}$, considering $K = 3$ classes. The utilization of functions for the parameter selection that takes into account a combination of metrics, e.g., a combination of recall and specificity, has already proven to be effective for the prediction of nocturnal hypoglycemia, even for longer prediction horizons [117].

Since several combinations of parameters generate similar results for each validation set, we take the best 10 combinations from each fold and then check which of these was the most recurrent combination of parameters. Following this analysis, we select the triplet of 30-80-70 hidden units for the hypoglycemia-euglycemia-hyperglycemia experts for the ME-LSTM, and the triplet of 32-64-64 filters for the three subsequent convolutional layers for the ME-CNN. For the grid search routine, as well as for all the successive training phases described in the next sections, we set the mini-batch size equal to 1/10 of the size of the training set. To avoid overfitting, we set the maximum number of epochs to 1500 and stop the training phase by early stopping if the performance on the validation set does not improve for 10 consecutive checks. We check the validation performance every 25 training iterations and shuffle training and validation data after every epoch.

8.3. Experimental Design

As widely mentioned in the previous sections, the sample-based approach presents several limitations. Consequently, we evaluate the performance using the event-based approach, as it provides a more realistic overview of the algorithm's capability to predict an adverse

event compared to the sample-based approach. Nonetheless, taking into account the strong imbalance related to the event-based approach, we train the model with a sample-based approach. Then, we evaluate performance on event prediction in the aftermath according to the definition of event-based prediction. We use this strategy as we assume that such training would improve performance because the model could see more samples belonging to the minority classes during the training and validation phase [118, 119].

8.3.1. Event detection

Event-based performance evaluation requires preprocessing. According to the most widely used definition [29], we consider a true positive an event correctly predicted in advance, and a false positive an event predicted without an actual counterpart. We consider false negatives the events not predicted. Straightforwardly, we consider consecutive timestamps of hypo/hyperglycemia as a single event. In our approach, we use this definition for the events of classes hypoglycemia and hyperglycemia.

For the reasons reported in section 8.1.4, we use a sample-based approach for class normoglycemia, instead. As a consequence, during the event-based performance evaluation, we follow a well-established strategy and consider consecutive misclassified samples as a single false-positive event when the actual observation is normoglycemia. Conversely, we consider each misclassified sample belonging to a minority class (either hypo- or hyperglycemia) a false negative for its class and a false positive for the wrongly assigned class.

Moreover, in order not to consider fluctuations in the read CGM signal nor the predictions, we consider an event or a prediction as such if it lasts for at least 10 minutes, i.e., if it lasts for at least 3 consecutive timestamps. It is worth noting that our approach increases the imbalance of the dataset, making the classification task more difficult.

In most works, an event is considered correctly predicted if the prediction is supplied with any advance with respect to the actual event [29, 45]. Furthermore, fixed a prediction horizon PH, a parameter k is set so that a prediction is considered correct if performed from 1 to $PH + k$ minutes in advance. In the literature, values of k range from 10 to PH minutes. In this work, we considered $k = 10$ minutes. The standard approach provides no clue as to the actual advance of the prediction.

For this reason, here we introduce a parameter α ranging from 1 to 6 (i.e., from 5 to 30 minutes) to evaluate the number of correct predictions performed with a fixed advance in terms of timestamps. In particular, for classes hypoglycemia and hyperglycemia, we classify the events according to the following rules: **True Positive (TP)** if a correct

prediction is performed in the time window $[-(PH+k),-\alpha]$ before the actual event; **False Positive (FP)** if an event is predicted and no actual counterpart is present in the $(k+PH)$ timestamps following the prediction; **False Negative (FN)** if an actual event is not predicted in the time window $[-(PH+k),-\alpha]$ before the actual event. It makes our approach differ from the standard approach, as it allows us to evaluate how many events are effectively detected at least α timestamps in advance.

Figure 8.3 reports a graphical comparison between the proposed and the standard event prediction approaches and some examples of correct and wrong predictions. The figure refers to the prediction of adverse events, i.e., hypo- and hyperglycemia, whereas the prediction of class normoglycemia exploits a sample-based approach. In this example, we consider α=3 for the proposed approach. In practice, the standard approach corresponds to our approach with $\alpha = 1$.

We performed three different tests, utilizing the public dataset and the private dataset, and implementing the proposed architecture on an edge device. The tests are described below and a schematic representation is shown in Figure 8.4.

8.3.2. Test 1: evaluation on the public dataset

We test the proposed approach on the Ohio T1DM dataset with a Leave-1-Patient-Out Cross-Validation (Fig. 8.4a). We fix, at each turn, data from one subject as the test set, and data from all the other subjects as the discovery set, randomly split into training (70%) and validation (30%) sets for the training of the base learners. The outputs of the softmax layers of the three experts are passed as training data to the decision tree meta-learner, together with the corresponding target label. At inference time, we classify all the samples in the test set. We then compute for each subject the event detection performance and a confusion matrix; then, we derive the final results from the total confusion matrix calculated by summing all the confusion matrices of all subjects.

8.3.2.1. Comparison with other methods

To further assess the proposed method, we compare the results we achieve on the public dataset to those of other state-of-the-art methods. The list of competitors that we test on the Ohio T1DM includes:

- A Support Vector Machine (SVM) with both polynomial (**SVM-poly**) and radial-basis-function (**SVM-rbf**) kernel. The latter model is the best classifier proposed

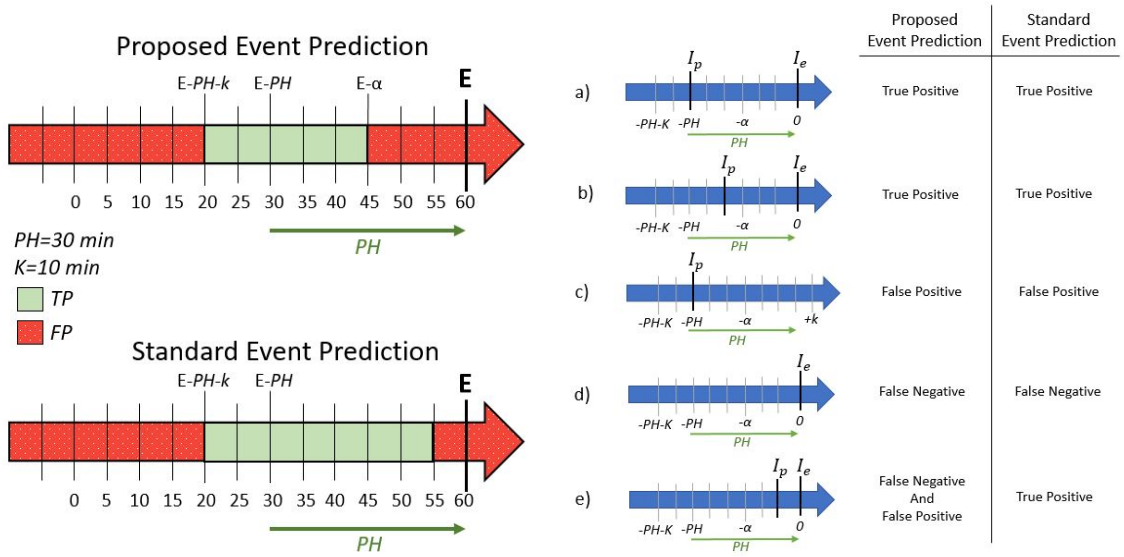
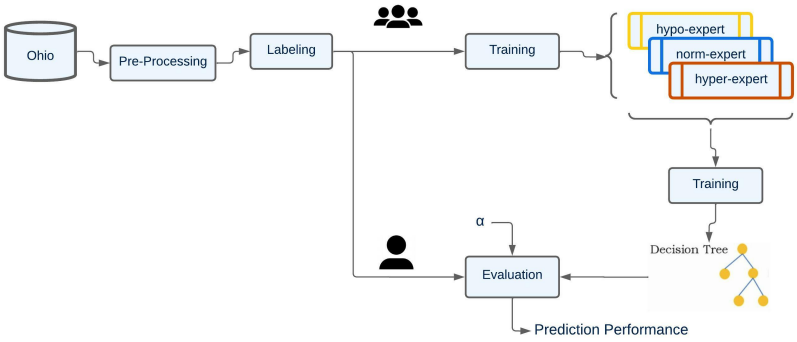
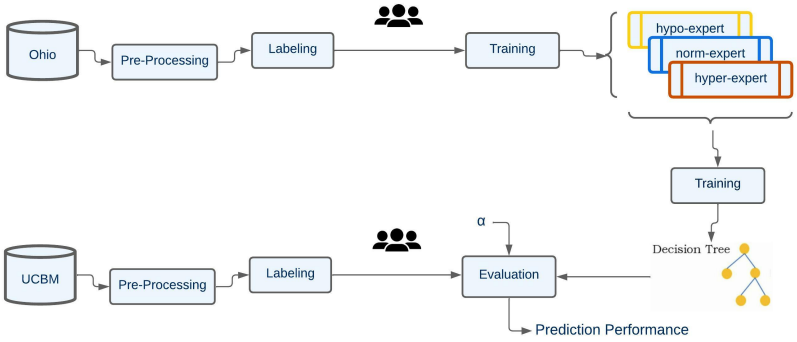


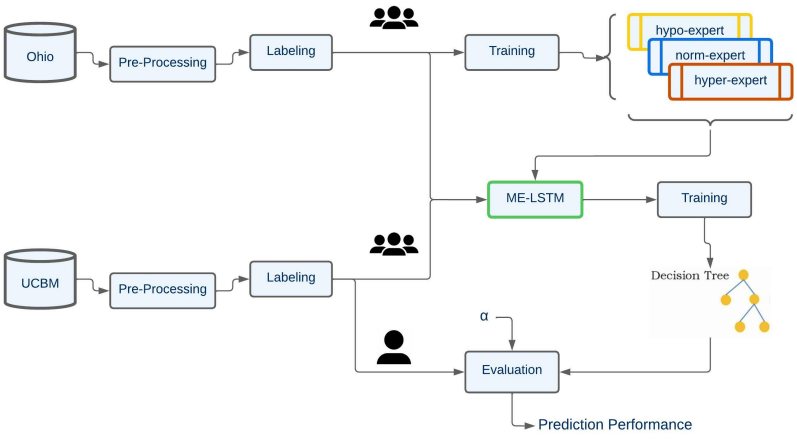
Figure 8.3.: Comparison and differences between the proposed and the standard [29] event prediction approach. Left: example of how a predicted event is classified whether as a true positive or a false positive depending on the advance by which the prediction is performed. Given an actual event E beginning after 60 minutes, bright cells indicate when a prediction would produce a true positive, whereas dark cells indicate when a prediction would produce a false positive. Right: examples of predictions and relative classification with the proposed and the standard approach. a) An actual event I_e occurs at $t = 0$. The event is predicted (I_p) exactly PH timestamps in advance. Both approaches consider I_p as a true positive. b) The prediction is performed less than PH but more than α timestamps in advance. Both approaches consider I_p as a true positive. c) I_p is predicted without an actual counterpart. Both approaches consider I_p as a false positive. d) An actual event occurs, but it is not predicted at least $(PH + k)$ minutes in advance. Both approaches consider I_e as a false negative. e) I_e occurs and it is predicted less than α timestamps in advance. The proposed approach considers I_p both as a false negative and a false positive, whereas the standard approach considers it as a true positive.



(a) Test 1



(b) Test 2.1



(c) Test 2.2

Figure 8.4.: Schematic representations of the experimental tests.

by Gadaleta et al. [29]. Similar to our model, the learners were trained and tested with one-vs-all decomposition for the classification task.

- A Random Forest (**RF**), which was proposed by Seo et al. [39] and Dave et al. [40]. We performed a grid search on our data to detect the optimal number of learners, resulting in 100. We used the same weights as our proposed models to tackle the data imbalance. It is worth noting that this model consists of an ensemble of decision trees, i.e., the model utilized as a meta-learner in the proposed approach.

- Two different configurations of LSTM neural networks. We performed a grid search on our dataset to determine the optimal amount of LSTM hidden units for both models. The first presents a multi-expert architecture like the one proposed but includes simpler and lighter neural networks with only one hidden layer for each expert. The grid search returned a value of 10, 100, and 1 hidden units for the hypoglycemic, euglycemic, and hyperglycemic experts, respectively (**ME-LSTM 10/100/1**). The second setup consists of a single neural network that presents the same architecture as a proposed expert, performing a three-class classification task. The grid search returned an optimal value of 70 units in the first and 35 units in the second LSTM layers (**LSTM 3-class**).

- CNN as a three-class classifier (**CNN 3-class**). To keep the framework comparable with the multi-expert model, we implement an analogous architecture as in the ME-CNN system.

Furthermore, we optimized and tested additional meta-learners following the optimal ME-LSTM and ME-CNN architecture already found as described in section 8.2.3:

- A SVM (**ME-LSTM-SVM** and **ME-CNN-SVM**) whose optimal configuration resulted in a polynomial kernel with one-vs-one decomposition and no class weights.

- A Naive-Bayes classifier (**ME-LSTM-NB** and **ME-CNN-NB**) whose optimal configuration resulted in normal Kernel smoothing and class weights for each class.

- A feedforward neural network (**ME-LSTM-NN** and **ME-CNN-NN**) whose optimal configuration resulted in one hidden layer with 3 neurons, each having ReLU activation function, and a size of 256 for the mini-batches.

We considered as additional competitors the **ME-LSTM** and the **ME-CNN**, i.e., the presented base learners, in which the final decision on the label to assign to every sample

is taken based on the greatest softmax output between the three experts. Finally, to assess if performance improves when including injected insulin and physiological features, we evaluated the proposed models, as well as every competitor, using all the four available input features (*Model*-**4F**).

8.3.3. Test 2: evaluation on the private dataset

We further validate the proposed approach on a private (UCBM) dataset. To implement a realistic evaluation approach, we train the ME-LSTM using only data from the Ohio T1DM dataset, using data of all patients as a discovery set and adopting a 70/30% split for training and validation set. Then, we perform tests on the five patients from the private dataset one by one. Before conducting these tests, we train the meta-learner following two different approaches:

1. utilizing only data from the public dataset (Fig. 8.4b). This approach consists of the application of a model trained using all the data available during test 1 to a different test set, consisting of patients that use different CGM sensors;

2. utilizing all the data from the public dataset and, at each turn, data from the four patients of the private dataset that are not the test patient (Fig. 8.4c). This approach is particularly suited for meta-learning because only the light meta-learner is updated with new data, while the base learners remain unchanged.

8.3.4. Test 3: edge implementation

To date, there are many devices capable of improving the lives of people with T1D [120], but there are still no devices capable of predicting the onset of hypo- or hyperglycemic episodes without the aid of a doctor. To investigate the possibility of integrating our system on edge and evaluate the time performance due to the utilization of the proposed solution in real applications, we perform an edge implementation test on the edge devices presented in section 8.1.5. We aim to obtain data on the training, transformation, and inference times of the proposed models and thus be able to discover their application scenarios and their possible limitation. We carry out the edge tests following a precise workflow. First, we train the classifiers, then we perform the transformation in *.tflite* to speed up the inference on the edge devices. Afterward, we run the classification process and feed the data to the decision trees downstream.

Regarding the number of operations accomplished:

- we train the base learners 30 times each for each patient, for a total of 360 training for each classifier;
- we perform the transformations in *.tflite* 100 times for each classifier and each patient, for a total of 1200 transformations for each classifier;
- we calculate the inference times $100 \times \ N\_test\_samples$ times for each classifier and each patient, following the leave-1-patient-out approach.

Finally, for calculating the training and inference times of the decision trees downstream of the three base learners, 1000 pieces of training were carried out and $1000 \times N\_test\_samples$ inference tests were calculated, always following the leave-1-patient-out approach. After that, we compute the mean and standard deviations for all the collected data.

8.4. Results and Discussion

In this section, we present and discuss the results achieved with the proposed meta-learning models. For compactness purposes, we use the abbreviations Hypo (hypoglycemia), Norm (normoglycemia), and Hyper (hyperglycemia) in the result tables.

8.4.1. Test 1: Results and Performance analysis

With regard to the event-based evaluation approach, we report the results achieved on the Ohio T1DM dataset with the proposed models in terms of recall per class, precision per class, and F1-Score per class. Table 8.1 reports the total results computed by summing all the confusion matrices of the patients, thus providing the performance on the whole dataset for the proposed models. The average results on the 12 patients are similar to the total results. We do not report them for brevity purposes.

Let us focus on the results achieved by the ME-LSTM-DT for different values of α. Recall, precision, and F1-Scores per class tend to become smaller as α increases. It indicates that the models are not fully capable of predicting adverse events with greater advance. The scores of class normoglycemia tend to remain high due to the strong imbalance of the dataset and the sample-based approach considered for this class. We can observe that more than half of the adverse events are predicted at least 30 minutes in advance; at the same time, the amount of FPs is very limited. In detail, the model can predict more than 81% hypoglycemic events and 83% hyperglycemic events at least 15 minutes in advance, while producing a small number of false alarms. Such a time

advance could be sufficient to avoid or considerably mitigate the complications [42]. More in detail, the average time gain, defined as the time between an alert and a real event (where the time gain is 0 in the case of an FN), is 22.8 minutes for hypoglycemia and 24.0 minutes for hyperglycemia. It is a good improvement compared to the literature, where a time gain of 15-20 minutes is usually achieved [48, 50].

It is worth noting that the decrease in the precision-per-class scores is due to the events predicted less than α timestamps in advance. In this case, they are considered false positives although a real event occurs; for this reason, the most appropriate precision scores to take into consideration are those obtained considering α=1, which express to what extent a wrongly predicted event is not going to occur.

It is also interesting to focus on the number of false alarms produced per day by the proposed method. Indeed, a 79.3% precision for hypoglycemia means that, on average, only 2 out of 10 alarms generated by the model are false alarms; in total, the amount of FPs for this class is 201, corresponding to an average of 0.45 false alarms per day. Some of these false alarms might be due to hypoglycemic events which would have actually occurred without a patient intervention [121], or that have not been detected by the CGM sensor [41, 121]. Similarly, a total of 202 FPs is observed for hyperglycemia, corresponding to an average of 0.46 false alarms per day. Such values are small enough not to stress patients with constant alarms that would generate a nuisance.

With regard to the results of the ME-CNN-DT, the F1-scores are always slightly greater than those achieved by the ME-LSTM-DT, except hypoglycemia for $\alpha \geq 5$. In particular, this model performs better on hyperglycemia prediction, as the recall scores are always slightly greater, while the precision scores are very similar. Taking into account hypoglycemia performance, this model presents greater precision (fewer false alarms) at the expense of a lower ability to detect events with greater advance, corresponding to values of $\alpha \geq 4$. It corresponds to an average time gain of 21.7 minutes for hypoglycemia and 25.0 minutes for hyperglycemia. The 87% precision achieved with $\alpha = 1$ corresponds to 1.3 false alarms every 10 alarms; in total, the amount of FPs for this class is 34, corresponding to an average of 0.087 false alarms per day. A total of 134 FPs are observed for hyperglycemia, corresponding to an average of 0.34 false alarms per day. Although the performance of the ME-CNN-DT model is better in general, the ME-LSTM-DT model would probably provide greater help to T1D patients, due to its improved ability to predict hypoglycemic events with greater advance while keeping small the number of false alarms. However, the ME-CNN-DT would be very helpful as well and would provide better performance in the prediction of hyperglycemia.

Table 8.1.: Total results of the proposed meta-learning systems with the event-based approach, extracted from the total confusion matrix for Test 1. Results are reported in terms of recall [%], precision [%], and F1-Score [%] per class for the different values of α investigated.

Model	α	Hypoglycemia			Normoglycemia			Hyperglycemia		
		Recall	Precision	F1-Score	Recall	Precision	F1-Score	Recall	Precision	F1-Score
ME-LSTM-DT	1	95.0	79.3	86.4	92.5	99.6	95.9	91.9	89.2	90.5
	2	88.3	78.0	82.9	92.5	99.6	95.9	89.0	88.5	88.8
	3	81.0	76.6	78.8	92.5	99.6	95.9	83.9	86.7	85.3
	4	73.3	75.0	74.1	92.5	99.6	95.9	78.6	85.2	81.1
	5	65.9	73.1	69.3	92.5	99.6	95.9	72.1	82.6	77.0
	6	54.8	69.1	61.1	92.5	99.6	95.9	62.9	79.2	70.2
ME-CNN-DT	1	92.3	87.3	89.7	92.5	99.9	96.0	94.8	89.0	91.8
	2	83.9	86.0	85.0	92.5	99.8	96.0	91.1	87.9	89.5
	3	75.8	84.8	80.0	92.5	99.8	96.0	87.3	86.4	86.9
	4	67.5	83.2	74.5	92.5	99.8	96.0	83.2	84.9	84.0
	5	59.4	80.8	68.5	92.5	99.7	96.0	77.9	82.7	80.3
	6	48.5	77.8	59.7	92.5	99.6	95.9	66.8	79.5	72.6

8.4.1.1. Qualitative comparison with the literature

In this section, we provide a comparison with the results presented by other works. Straightforwardly, we focus on the total results we achieve considering α=1 because they correspond to the approach pursued in the literature [29]. The comparison is qualitative because works that performed event detection used different datasets.

For hypoglycemia, the best recall score is 95%, proving that almost all hypoglycemic events are predicted at least 5 minutes in advance, while precision is strictly greater than 79%. Of the models listed in section 1.2, only our previous work [51] achieves a better precision (86.4%), which is lower than that of the ME-CNN-DT model, while achieving a sensitively lower recall (59.8%). The second best precision score is achieved by Zhu et al. [50] (65.6%) while achieving 84.1% recall. They proposed a bidirectional recurrent neural network refined with patient-specific model agnostic meta-learning for regression on three datasets (including the Ohio T1DM dataset), obtaining on average 0.48 false alarms per day. Similarly, the model proposed by Prendin et al. [48] achieves a good precision (64%), which also results in a smaller amount of 0.5 false alarms per day; however, the recall reported in that study is lower (82%). We outperform by more than 40% the remaining hypoglycemia precision scores. Daskalaki et al. [45] achieve 100% recall for both hypoglycemia and hyperglycemia; nonetheless, their work only aims at predicting events regardless of the precision per class. They report that their model

generates on average 1.6 false alarms per day, but there is no clue on the number of events in the test set, so the computation of the precision per class is not possible. The same applies to the work from Yang et al. [38].

For hyperglycemia, the recall score is noteworthy as well, being about 92%, whereas precision is above 89%. It is worth pointing out that, although the prediction of hyperglycemia may seem of reduced practical impact because most patients experience hyperglycemia after a meal, the proposed models do not exploit carbohydrate information to perform such a prediction, in the view of a fully-automated system that does not require the patient to provide meal data manually. We outperform by more than 33% the only ones who reported hyperglycemia precision (Gadaleta et al. [29], 56%), although the same study outperforms our hyperglycemia recall (95%). Nonetheless, their proposed SVM model produces many false alarms (hypo/hyperglycemia precision equal to 36/57%). In general, the proposed meta-learning approaches outperform the previously presented ones. However, these comparisons are qualitative because tests are performed on different datasets.

The F1-Score per class, which can be interpreted as the ability of the model to perform accurate predictions while generating few false alarms, is greater than 86% for every class. It proves that the proposed approach could be reliable in a real-life application without stressing patients with many false alarms, which is rarely achieved in the literature. However, a value of α=1 means that predictions are performed at least 5 minutes in advance, which may not be a sufficient time to prevent adverse events. It is the reason why we investigated the performance with different values of α.

For sake of completeness, we report in Table 8.2 the performance of the proposed meta-learning models with the sample-based approach, albeit it is not fully indicative of a model's real performance, as widely discussed in the previous sections. The results achieved are highly competitive compared to those reported by the models listed in Table 1.2 that pursue a sample-based approach, since only the study from Dave et al. [40], who proposed a model composed of two Random Forests, one day-specific and one night-specific, achieves better hypoglycemia recall (93.7%) but at the expense of a considerably lower precision (15.1%). The opposite approach was pursued by Marcus et al. [47], who aimed to reduce as much as possible the number of false alarms per day, achieving a 4% false-positive rate; nonetheless, their recall is considerably lower than ours (64% and 61% for hypo- and hyperglycemia).

Finally, we report in Table 8.3 the results achieved by the proposed models when a longer PH of 60 or 120 minutes is considered. The performance worsens sensitively for

Table 8.2.: Results with a sample-based approach.

Model	Class	Recall [%]	Precision [%]	F1-Score [%]
ME-LSTM-DT	Hypo	90.6	71.2	79.7
	Norm	91.1	96.0	93.5
	Hyper	94.7	90.2	92.4
ME-CNN-DT	Hypo	78.2	77.6	77.9
	Norm	91.8	92.2	92.0
	Hyper	89.5	88.9	89.2

Table 8.3.: Average percentage results over the 12 Ohio T1DM patients with the event-based approach of the two proposed models with a PH of 60 and 120 minutes.

Model	Class	Recall	Precision	F1-Score
ME-LSTM-DT PH = 60 min	Hypo	29.1	44.7	35.2
	Norm	80.1	97.6	87.9
	Hyper	41.6	47.4	42.9
ME-CNN-DT PH = 60 min	Hypo	25.3	43.8	31.2
	Norm	83.4	98.0	90.1
	Hyper	42.9	56.5	47.9
ME-LSTM-DT PH = 120 min	Hypo	26.3	21.9	21.7
	Norm	60.3	92.8	73.0
	Hyper	55.8	31.0	39.3
ME-CNN-DT PH = 120 min	Hypo	24.6	24.8	24.7
	Norm	65.7	92.9	76.9
	Hyper	55.3	37.2	43.9

both models. Although a longer PH would provide patients with more time to react to an incoming adverse event, a prediction over such a long temporal horizon necessarily increases the uncertainty in the predictions, for example, due to the attempt of the algorithm to maximize the performance for the minority classes, which leads to the generation of many false alarms, as demonstrated by the considerably lower recall scores for class normoglycemia. In light of this analysis, a 30-minute PH seems appropriate for event detection. However, the results achieved by the proposed model are comparable to those of other recent studies that investigate a longer PH for the prediction of nocturnal hypo- or hyperglycemia [43, 122], which also suffer from a lower recall or precision score.

8.4.1.2. Results of the comparison with other methods on the Ohio T1DM dataset

In this section, we compare our performance to the performance of the competitors listed in section 8.3.2.1. The results are referred to the event-based approach and are computed on the total confusion matrix with a Leave-1-Patient-Out Cross Validation approach. All the competitors have undergone a grid search to select the optimal model parameters. To provide a compact overview of the performance for different values of α, we report the results of each model in terms of the F1-Scores per class and of the geometric mean G of the F1-Scores per class, because they provide an overview of the model capability to achieve good performance for each class.

Table 8.4 reports the results of the comparison with the other methods when exploiting only CGM as an input feature. Results are reported in terms of the F1-Score for classes hypoglycemia (F_{Hypo}), normoglycemia (F_{Norm}), and hyperglycemia (F_{Hyper}), together with the geometric mean G of the F1-Scores per class. Considering all values of α, both the proposed models outperform all the competitors by a large margin, except for class normoglycemia for which the SVM with radial basis function always achieves better results. However, this is the majority class and is less important to predict accurately. The best competitors are the other CNN-based models for $\alpha \leq 2$, and the ME-LSTM for greater values.

Let us focus on the comparison between the results achieved with and without resorting to meta-learning. With regard to hyperglycemia, a small improvement is observed for F1-scores, as the slight precision increase is balanced by the slight recall decrease. The major advantage of the meta-learning is observed with regard to hypoglycemia, where an increase of 10 to 15% is observed for all the F1-scores. In detail, although the recall is slightly decreased by 3 to 7%, a considerable improvement of about 20% is observed for the precision, resulting in a much lower amount of false alarms. We can conclude that using a meta-learner considerably improves the capability of predicting adverse events while producing a low amount of false alarms. We also tested two other meta-learners (Naive-Bayes classifier and SVM) which returned very high recall scores (above 99%) for both hypo- and hyperglycemia, at the expense of very low precision (below 15%). We do not report these results for the sake of brevity. From a comparison with the ME-LSTM, the ME-CNN, and the Random Forest, it is clear that the utilization of the meta-learning approach as whole guarantees sensitively better performance than any of the models it is composed of. It is also interesting to note that the multi-expert systems ME-LSTM and ME-CNN outperform the correspondent three-class model, suggesting

Table 8.4.: Results of the proposed models and the competitors with the event-based approach, extracted from the total confusion matrix, for the different values of α investigated. All the competitors are tested using only CGM as an input feature. Results are reported in terms of the F1-Score for classes hypoglycemia (F_{Hypo}), normoglycemia (F_{Norm}), and hyperglycemia (F_{Hyper}), together with the geometric mean G of the F1-Scores per class. We investigated the models listed in section 8.3.2.1, which are an SVM with radial-basis-function (SVM-rbf) [29] and with polynomial (SVM-poly) kernel, a Random Forest (RF) [39, 40], and variations of LSTM and CNN models. The results in the bottom panel refer to the proposed base learners followed by different meta-learners. The best score of each column is highlighted in red.

Model	$\alpha = 1$				$\alpha = 2$				$\alpha = 3$				$\alpha = 4$				$\alpha = 5$				$\alpha = 6$			
	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G
ME-LSTM-DT	86.4	95.9	90.5	90.9	82.9	95.9	**88.8**	89.0	78.8	95.9	85.3	86.4	74.1	95.9	81.8	83.4	**69.3**	95.8	77.0	80.0	**61.1**	95.8	70.2	74.3
ME-CNN-DT	**89.7**	96.0	**91.8**	**92.4**	**85.0**	96.0	85.9	**90.0**	**80.0**	96.0	**86.9**	**87.3**	**74.5**	96.0	**84.0**	**84.4**	68.5	96.0	**80.3**	**80.7**	59.7	95.9	**72.6**	**74.6**
ME-LSTM	74.0	95.9	90.2	86.2	72.1	95.9	86.1	84.1	70.1	95.9	82.1	82.0	64.8	95.9	79.0	78.9	58.8	95.9	75.9	75.4	52.8	95.8	71.1	71.1
ME-LSTM 10/100/1	64.3	92.5	86.2	80.1	63.4	92.4	81.8	78.3	62.0	92.4	77.9	76.4	60.3	92.4	74.9	74.7	57.7	92.4	72.2	72.7	52.3	92.4	64.7	67.9
LSTM 3-class	46.0	92.0	62.8	64.3	45.0	92.0	59.8	62.8	42.4	92.0	56.8	60.5	39.3	92.0	52.7	57.6	35.8	92.0	47.8	54.0	31.3	92.0	43.3	50.0
ME-CNN	87.6	97.9	90.3	91.8	76.2	97.9	85.6	86.1	64.8	97.9	81.9	80.3	57.7	97.9	78.5	76.2	48.9	97.9	74.2	70.8	41.5	97.9	66.5	64.6
CNN 3-class	86.0	97.7	90.2	91.2	77.3	97.7	86.6	86.8	69.5	97.7	82.4	82.4	61.2	97.7	78.6	77.2	53.8	97.7	73.9	72.9	43.8	97.6	66.2	65.6
SVM-rbf	80.6	**99.5**	85.0	88.0	68.6	**99.5**	80.0	81.8	59.8	**99.5**	77.2	77.2	51.0	**99.5**	73.8	72.1	46.5	**99.5**	71.7	69.2	37.9	**99.5**	63.6	62.2
SVM-poly	76.4	99.1	78.3	84.9	65.9	99.1	77.0	79.5	59.8	99.1	74.3	76.1	53.9	99.0	71.4	72.5	46.0	99.0	67.8	67.6	39.8	99.0	59.5	61.7
RF	61.5	96.3	81.6	78.4	58.8	96.3	78.1	76.0	55.2	96.3	74.2	73.4	50.0	96.3	70.2	69.7	45.8	96.3	66.3	66.4	38.3	96.3	59.4	60.3
ME-LSTM-SVM	88.1	99.2	94.5	93.8	63.4	99.2	78.4	79.0	46.2	99.2	62.2	65.8	34.2	99.2	51.3	55.8	27.9	99.2	43.9	49.5	24.2	99.1	39.1	45.4
ME-CNN-SVM	69.3	95.0	82.9	81.7	64.0	94.9	80.4	78.7	59.9	94.9	76.9	75.9	54.2	94.9	72.6	72.0	51.2	94.9	67.2	68.9	47.0	94.9	61.1	64.8
ME-LSTM-NB	60.9	93.1	86.3	78.8	60.7	93.1	84.6	78.2	59.2	93.1	78.4	75.6	56.0	93.1	69.0	71.1	52.3	93.1	61.4	66.9	44.6	93.1	52.1	60.0
ME-CNN-NB	50.7	82.9	86.2	71.3	50.6	82.9	85.6	71.0	50.2	82.9	83.1	70.2	50.0	82.9	80.2	69.3	49.4	82.9	77.1	68.1	48.6	82.9	72.4	66.3
ME-LSTM-NN	89.6	97.5	90.2	92.4	84.5	97.5	84.3	88.5	76.1	97.4	80.1	84.0	65.4	97.4	77.8	79.1	58.3	97.4	73.7	74.8	49.5	97.4	67.4	68.7
ME-CNN-NN	87.8	97.8	90.2	91.8	79.6	97.8	85.6	87.3	69.8	97.8	81.5	82.2	61.2	97.8	77.9	77.5	53.5	97.8	73.6	72.3	45.2	97.8	65.4	66.1

Table 8.5.: Results of the proposed models and the competitors with the event-based approach, extracted from the total confusion matrix, for the different values of α investigated. The top panel reports the results of the proposed models with a univariate approach; results in the central panel refer to the proposed models and the competitors tested using all 4 available features (Model-4F); results in the bottom panel refer to the proposed base learners followed by different meta-learners. Results are reported in terms of the F1-Score for classes hypoglycemia (F_{Hypo}), normoglycemia (F_{Norm}), and hyperglycemia (F_{Hyper}), together with the geometric mean G of the F1-Scores per class. The best score of each column is highlighted in red.

Model	$\alpha = 1$				$\alpha = 2$				$\alpha = 3$				$\alpha = 4$				$\alpha = 5$				$\alpha = 6$			
	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G	F_{Hypo}	F_{Norm}	F_{Hyper}	G
ME-LSTM-DT	86.4	95.9	90.5	90.9	82.9	95.9	88.8	89.0	78.8	95.9	85.3	86.4	74.1	95.9	81.8	83.4	**69.3**	95.8	77.0	80.0	**61.1**	95.8	70.2	74.3
ME-CNN-DT	89.7	96.0	91.8	**92.4**	**85.0**	96.0	85.9	**90.0**	**80.0**	96.0	86.9	**87.3**	**74.5**	96.0	84.0	**84.4**	68.5	96.0	80.3	**80.7**	59.7	95.9	72.6	**74.6**
ME-LSTM-DT-4F	88.5	95.6	91.9	91.8	82.4	95.6	88.5	88.3	73.4	95.6	83.0	83.5	67.6	95.5	79.1	78.2	55.2	95.5	70.0	70.5	46.0	95.4	62.0	63.9
ME-CNN-DT-4F	87.2	96.4	**93.7**	92.3	82.6	96.4	**91.6**	**90.0**	77.9	96.4	**88.6**	87.2	71.9	96.3	**85.0**	83.7	66.8	96.3	**80.9**	80.4	57.6	96.3	**72.9**	73.9
ME-LSTM-4F	73.7	95.9	89.9	86.0	70.7	95.9	84.6	83.1	64.0	95.9	74.4	77.0	56.3	95.9	64.2	70.2	46.9	95.8	57.1	63.5	40.3	95.8	49.9	57.8
LSTM 3-class-4F	44.1	92.2	61.2	62.9	41.9	92.2	58.4	60.9	39.5	92.1	54.7	58.4	36.1	92.1	51.0	55.4	31.8	92.1	46.8	51.6	28.1	92.1	43.1	48.1
ME-LSTM 10/100/1-4F	43.7	92.5	73.5	66.7	42.2	92.4	63.7	62.9	38.1	92.4	49.9	56.0	34.4	92.3	37.6	49.2	30.1	92.3	30.8	44.1	25.6	92.2	24.1	38.5
ME-CNN-4F	90.9	98.1	92.3	**93.6**	76.9	98.1	87.0	86.8	66.7	98.1	82.1	81.2	57.6	98.1	78.6	76.2	50.9	98.0	74.5	71.9	42.2	98.0	66.6	65.0
CNN 3-class-4F	**91.0**	97.9	91.0	93.2	79.3	97.9	86.4	87.5	69.2	97.9	82.6	82.4	61.7	97.8	79.0	78.1	54.3	97.8	74.2	73.3	46.7	97.8	66.2	67.1
SVM-rbf-4F	63.6	**99.2**	68.6	75.6	44.1	99.1	55.6	62.4	29.5	99.1	46.0	51.3	22.7	99.1	37.9	44.0	12.0	99.0	31.5	38.4	13.6	99.0	27.0	33.1
SVM-poly-4F	75.5	**99.2**	83.4	85.5	58.7	**99.2**	70.9	74.5	44.1	**99.2**	56.0	62.6	34.3	**99.2**	41.1	51.9	26.7	**99.1**	32.9	44.3	21.4	**99.1**	26.4	38.3
RF-4F	71.7	97.4	82.4	83.3	63.6	97.4	72.1	76.5	55.3	97.4	72.1	67.8	47.1	97.4	46.5	59.7	37.9	99.3	36.3	51.2	30.0	97.3	28.0	43.4
ME-CNN-SVM-4F	84.4	98.6	89.1	90.5	61.8	98.5	75.8	77.3	44.3	98.5	63.1	65.0	36.1	98.5	53.3	57.4	29.8	98.5	46.9	51.7	26.9	98.5	41.2	47.8
ME-CNN-SVM-4F	90.3	97.9	90.8	92.9	77.6	97.9	86.3	86.9	67.9	97.9	82.4	81.8	59.2	97.9	78.9	77.0								
ME-LSTM-NB-4F	57.1	91.5	86.1	76.6	56.4	91.5	82.8	75.3	54.1	91.5	75.7	72.0	50.3	91.5	66.1	67.2	46.0	91.5	58.6	62.7	41.5	91.4	51.1	57.9
ME-CNN-NB-4F	48.8	79.1	89.8	70.3	48.7	79.1	88.2	69.8	48.5	79.1	85.0	68.8	47.9	79.1	81.4	67.6	46.7	79.1	78.3	66.1	45.5	79.1	73.7	64.2
ME-LSTM-NN-4F	57.2	91.9	74.6	73.2	49.5	91.9	71.8	68.9	46.3	91.9	68.8	66.4	40.3	91.9	64.7	62.1	36.1	91.8	59.8	58.3	31.8	91.8	55.8	54.8
ME-CNN-NN-4F	82.7	98.1	91.7	90.6	75.8	98.1	86.2	86.2	64.1	98.1	81.7	80.0	55.7	98.1	78.0	75.2	48.6	98.1	73.9	70.6	42.6	98.0	65.9	65.0

that the ensemble strategy is more effective for this task.

Finally, we tested our models and the competitors using a multivariate approach, i.e., using all four available features as input (Model-4F); these results are reported in the bottom panel of Table 8.5, whereas the top panel reports the results of the proposed univariate approach. The reported results are extracted from the total confusion matrix computed by adding the confusion matrices of all patients. In general, all the competitors perform better when using CGM alone as an input feature. The proposed models outperform all the competitors. The only exception concerns class normoglycemia, for which the SVM with a polynomial kernel always achieves better results. The analysis is very similar to that provided for the models which exploit only CGM. An interesting behavior is observed for hyperglycemia prediction, for which the ME-CNN-DT-4F outperforms all the other models, including its univariate counterpart. This is probably due to the information concerning insulin boluses, which allows an easier prediction of postprandial hyperglycemia; however, such a feature complicates the data management, and the improvement compared to the univariate model is not very marked (3-4%).

In conclusion, by testing different models on the same dataset we observed that:

1. resorting to multi-expert systems with a majority-based decision policy provides better performance compared to utilizing a single model for a 3-class classification task;

2. using meta-learning considerably improves the performance of multi-expert base learners.

8.4.2. Test 2: results and performance analysis

We tested a private dataset to evaluate the capability of the proposed approach to adapt to the data of new patients. The UCBM dataset includes patients that utilize a different CGM sensor than the patients enrolled in the Ohio T1DM dataset, and who regularly perform physical activity. This test was performed twice: 1) by training the meta-learner only on the Ohio patients, and 2) by training the meta-learner on the Ohio dataset joined with the UCBM dataset with a leave-1-patient-out approach. Table 8.6 reports the results of these tests (we do not report the results for the normoglycemia class, which are all above 95%).

Let us focus on the results of the first implementation of the test, in which only the Ohio T1DM dataset was used to train the meta-learner. The performance worsens considerably, particularly for larger values of α. The main worsening concerns the

hyperglycemia prediction of the ME-CNN-DT; however, also the ME-LSTM-DT model is able to predict only a few more than half hyperglycemic events with any advance. This suggests that the different cohort of patients, with different habits and lifestyles, joined with a different CGM sensor, presents completely different patterns preceding hyperglycemia. Conversely, the worsening for class hypoglycemia is less pronounced, suggesting that common patterns exist between the two datasets.

Let us now focus on the results achieved including part of the UCBM dataset in the training set. It is worth stressing that data from the UCBM dataset were used only to train the meta-learners, whose training requires a very small amount of time; differently, only the public dataset was used (once) for the more onerous training of the base learners. Again, the performance is considerably worse than Test 1; nonetheless, a pronounced improvement is observed for all classes and for all values of α, with the exception of class hypoglycemia of the ME-LSTM-DT model, which already achieved the best performance in the first configuration. The improvement is particularly noticeable for larger values of α and for the ME-CNN-DT, whose F1-scores increase by up to 4 times.

Although the results achieved with the second experimental setup are in line with those presented in previous works (e.g. an F1-score of 72% for hypoglycemia is presented in [48]), these results are considerably worse than those achieved in Test 1. This could be expected in light of the huge difference between the two datasets under observation and considering the limited size of the UCBM dataset for training. In addition, it has been widely investigated how the prediction of T1D events and glycemic levels is particularly challenging on patients that perform physical activity [88, 123]. In conclusion, the take-home message of this test is that the predictive performance of the proposed meta-learning approach can be considerably improved using a very limited amount of data from the new dataset. Such an improvement is achievable in the time required to train the meta-learner, which is far less than a second, as discussed in the next subsection.

8.4.3. Test 3: results of the edge implementation

The tests on the edge system were carried out following the pipeline described in subsection 8.3.4. The results concerning training, conversion and inference time are shown in Table 8.7.

From the data collected, on the one hand, it can be observed that the training of CNNs is more onerous in terms of time required when compared to that of LSTMs; on the other hand, the transformation times of the CNN models are less time-consuming,

Table 8.6.: Total results of the tests performed over the private dataset. Results are reported for the ME-LSTM-DT (left) and the ME-CNN-DT (right) in terms of recall [%], precision [%] and F1-Score [%] per class for the different values of α investigated. The top panel reports the results of the tests performed using only the Ohio dataset to train the meta-learner, whereas the results in the bottom panel are referred to the model in which the meta-learner is updated using data from the UCBM dataset using a leave-1-patient-out approach.

Training dataset	α	**ME-LSTM-DT**						**ME-CNN-DT**					
		Hypoglycemia			Hyperglycemia			Hypoglycemia			Hyperglycemia		
		Recall	Precision	F1-Score	Recall	Precision	F1-Score	Recall	Precision	F1-Score	Recall	Precision	F1-Score
Ohio	1	81.3	97.5	88.7	51.4	79.7	62.5	70.8	80.1	75.1	32.2	80.0	45.9
	2	67.9	96.7	79.8	39.9	74.0	51.9	38.8	55.8	45.7	25.9	76.7	38.8
	3	60.5	96.7	74.5	33.5	70.1	45.3	27.5	48.4	35.0	16.3	45.0	24.0
	4	46.8	95.0	62.7	21.5	65.0	32.3	19.5	41.4	26.5	14.3	45.0	21.7
	5	39.2	93.3	55.2	12.4	52.7	20.1	11.5	29.6	16.6	10.3	43.3	16.7
	6	34.6	90.0	50.0	8.4	46.0	14.2	11.5	29.6	16.6	7.3	23.3	11.2
Ohio + UCBM	1	91.8	90.9	91.3	84.6	91.2	87.8	88.4	66.3	75.7	63.3	73.6	68.1
	2	82.1	89.8	85.8	70.3	89.7	78.9	74.2	62.5	67.8	59.9	70.1	64.6
	3	64.6	87.7	74.4	47.1	87.8	61.3	74.2	62.5	67.8	52.8	65.3	58.4
	4	56.2	86.6	68.2	32.7	82.0	46.7	65.5	59.8	62.5	43.5	59.0	50.1
	5	40.6	77.1	53.2	23.8	78.0	36.5	53.0	50.2	51.5	39.1	56.9	46.4
	6	39.2	76.7	51.9	14.9	74.7	24.9	42.4	42.2	42.3	36.8	54.7	44.0

by a factor of 5, with respect to the LSTM ones. This is due to the steps needed for the conversion into *.tflite*; in fact, in order to transform an LSTM, or in general an RNN, into *.tflite* it is necessary to build the graph of the model itself, an operation that can be performed through the use of the concrete functions of Tensor Flow. This operation, which is not required for the CNN transformation, results in a longer transformation time for this type of model. In all cases, no appreciable loss in performance was observed.

As far as inference times are concerned, it can be observed that, regardless of the model under consideration, they are around values of less than a tenth of a millisecond. We can therefore state that the time required to perform this operation has little or any influence on the total time count, thus allowing both the considered models to work effectively in real-time when considering the 5-minute sampling window typical of CGM sensors. Moreover, the training and transformation times of the networks are in both cases greater than the single window required for prediction, but considerably shorter for LSTM. Therefore, in case of a possible implementation of an online learning system, i.e. a system capable of updating itself directly on the edge device using new incoming data, the use of multi-expert LSTMs would be preferable due to their speed in the training phase. The only data collected not shown in table 8.7 are those concerning

Table 8.7.: Average time required with standard deviation for the edge implementation of the multi-expert architecture. The results for both individual experts and the two multi-expert approaches are reported.

Model	Training time (s)	Transformation time (s)	Inference time (s)
LSTM hypo	51.4 ± 19.4	55.2 ± 2.6	$1 \cdot 10^{-4} \pm 5 \cdot 10^{-5}$
LSTM norm	181.5 ± 62.9	55.2 ± 2.7	$2 \cdot 10^{-4} \pm 8 \cdot 10^{-5}$
LSTM hyper	147.7 ± 43.6	55.1 ± 2.8	$2 \cdot 10^{-4} \pm 6 \cdot 10^{-5}$
CNN hypo	1133.4 ± 415.2	10.6 ± 1.1	$3 \cdot 10^{-4} \pm 8 \cdot 10^{-4}$
CNN norm	1358.3 ± 610.0	10.6 ± 1.0	$2 \cdot 10^{-4} \pm 5 \cdot 10^{-5}$
CNN hyper	1467.3 ± 456.3	10.6 ± 0.9	$2 \cdot 10^{-4} \pm 5 \cdot 10^{-5}$
ME-LSTM	380.7 ± 125.9	165.5 ± 8.2	$5 \cdot 10^{-4} \pm 2 \cdot 10^{-4}$
ME-CNN	3958.9 ± 1481.6	31.8 ± 3.0	$6 \cdot 10^{-4} \pm 9 \cdot 10^{-4}$

the training and inference time of the decision trees. We made this choice because, for both the ME-LSTM-DT and the ME-CNN-DT, the results obtained are overlapping with a mean time for training the decision tree of $0.055 \pm 0.002\,s$ and inference time of $9.86 \cdot 10^{-8} \pm 1.86 \cdot 10^{-8}\,s$ and therefore, similarly to the inference times of the models, negligible for a real application scenario. This suggests that updating the meta-learners on the edge with new incoming data would have a very limited impact on the device in terms of computational time.

9. A New Glycemic closed-loop control based on Dyna-Q for Type-1-Diabetes

Current T1D management based on CGM devices is based on hybrid closed-loop control, i.e., the patient is asked to close the control loop with the CGM sensor and the insulin pump by taking the final decision on the amount of insulin to be administered, and thus providing the actual control on their glycemia, while the automated system limits to give suggestions on the amount of bolus based on additional information provided by the patients themselves, such as the amount of ingested CHO. This should be avoided in the light of a fully closed-loop artificial pancreas system capable of properly controlling glycemic levels without the patient's intervention. In this respect, RL has shown to be promising in many recent studies [124, 62], but two main concerns prevent the application of such control systems on real artificial pancreas devices. First, the vast majority of the studies use model-free RL algorithms [62] which directly perform their actions on the patients. This is a limitation because, in the light of the exploration-exploitation approach which is typical of model-free RL, many incorrect insulin boluses would be injected into the patients with tremendous consequences before the agent has learned the correct control policy. Moreover, to achieve such a control, the most performing methods to date utilize large amounts of data, ranging from several months to years [57], which is not feasible in real life. The second limitation concerns the input features utilized for the control models. Most of the existing methods, including the only previously presented model-based approach [63], utilize input features that the patient is asked to supply manually, such as the amount of ingested CHO. As mentioned, this should be avoided in the light of a fully-automated artificial pancreas system capable of properly controlling the glycemic levels. We aim to utilize only direct measures that depend exclusively on the reading from sensors, without utilizing indirect measures that depend on the input from the patient and are thus error-prone. Recent studies [9] demonstrated that a

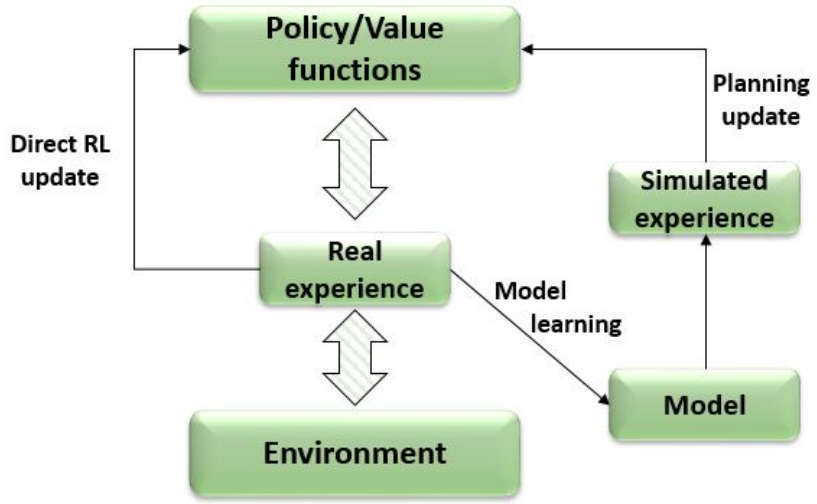


Figure 9.1.: General Dyna architecture.

fully closed-loop artificial pancreas can achieve performance non-inferior to commercial hybrid closed-loop devices which depend on carbohydrate counting, which is a burden for patients and an error-prone task, with an estimation error of around 20% in adults [125]. In such an approach, planning, acting, learning, and direct RL happen continually. A model is learned from real experience with a precision medicine approach and generates a simulated experience, while planning is made by the application of RL methods to the simulated experiences as if they had happened. The interaction with real and simulated experience improves the value function and the policy. A schematic representation of the general Dyna architecture is shown in Figure 9.1.

For the reasons above, in this study [126] we present a model-based Dyna-Q RL algorithm, i.e., that uses an environment-patient model on which to train the agent. After having tested the safety of the proposed action, it is carried out in the real environment, that is, the patient themselves. In our case, the actions are insulin boluses and are chosen by the agent implemented via a deep Q-network (DQN). The agent's hypothesized bolus of insulin is tested on the simulated environment which, in the presented model, is a recurrent neural network that, trained on patient-specific data, learns the relationships underlying the patient's glucose-insulin system dynamic. Furthermore, in our model, CHO information is never taken into consideration, as the neural network is trained based on CGM and insulin data only, and the agent considers CGM alone as a state variable of the environment. Finally, we aim to train the whole system with a reasonable amount of data, as we acknowledge that using years of monitoring data is

hardly achievable in practice.

9.1. Dataset

Since the proposed model-based control is based on a glucose levels predictor that must be trained, in this study we considered data from both a private dataset and data of virtual patients generated using a simulator.

9.1.1. Simulated dataset and preprocessing

10 adult subjects with T1D have been produced by running different simulations in the UVA/Padova simulator. Such software was approved in 2008 by the FDA as a substitute for animal trials for preclinical tests of insulin treatments and it can describe the glucose-insulin dynamic model of an *in silico* population, including adults, adolescents and children [32]. In particular, two different datasets have been generated, both consisting of 30 days of simulation and obtained by setting *GuardianRT* and *Insulet* as simulated CGM sensor and insulin pump, respectively. In fact, the UVA/Padova simulator allows the generation of in silico data of both ideal (without any error) and realistic sensors, such as GuardianRT and Dexcom. Our choice to generate data from the GuardianRT - a sensor used in the MiniMed system by Medtronic for diabetes therapeutic automation - arises from the attempt to make the simulated data as similar as possible to the real-world ones, in order to take into account the bias derived from a real sensor. To run the simulations, a scenario defining meal schedules is required. Information to be specified to create a scenario include the amount of CHO to be ingested at each meal, the corresponding time of the day, and the insulin doses to be injected. In the scenarios used for the generation of the two datasets, the first two settings are the same; conversely, the insulin administrations differ. In particular, data regarding meal schedules (CHO intake and meal times) are returned by a Matlab function defined in such a way that each day consists in (at least) three meals: breakfast, lunch, and dinner. In addition to these standard meals, morning and afternoon snacks are randomly introduced to generate variability between days. In order to make the artificially created datasets even more realistic, the variability between simulation days is not only given by the different number of meals per day, but also by changing the CHO amount in daily meal schedules. This was achieved by first defining a vector with fixed CHO quantity per meal, according to the Dietary Reference Intakes for Carbohydrate [93]; then, we modified such standard vector

by adding random noise taken from uniform distributions, each one chosen properly for the different meals involved. The whole algorithm implemented to generate the standard meal schedule for each individual is reported in Algorithm 2.

Algorithm 2 Generate standard scenario

Require: Input Data

$num\_day = 30$ ▷ Simulation duration

$cho\_st = [45, 20, 70, 20, 80]$ ▷ Standard CHO quantities

$a\_cho = [5, 2.5, 10, 2.5, 10]$ ▷ Right-hand limit of the uniform distribution linked to each meal

$hours\_st = [8, 10.5, 13, 17, 20]$ ▷ Standard meal time

$a\_hours = [1, 0.5, 1, 0.5, 1]$ ▷ Right-hand limit of the uniform distribution linked to each mealtime

▷ CHO VECTOR

$b\_cho = -a\_cho$ ▷ Left-hand limit of each uniform distribution for cho quantities

1: $i = 0$ to length(cho_st)

2: $CHO(:, i) = cho\_st(i) + [b\_cho(i) + (a\_cho(i) - b\_cho(i)) * rand(num\_day, 1)]$ ▷ CHO modified

▷ Generating random logical vector to decide when morning and afternoon snacks are present

$snk\_1 = logical(randi(2, [1 num\_day]) - 1)$ ▷ Morning snack

$snk\_2 = logical(randi(2, [1 num\_day]) - 1)$ ▷ Afternoon snack

▷ Multiply the logical vector with the column corresponding to the snack to obtain the snacks CHO per each day

$CHO(:, 2) = snk_1 * CHO(:, 2)$

$CHO(:, 4) = snk_2 * CHO(:, 4)$

$Ameals = []$ ▷ Initialize the vector *Ameals* that will contain the final CHO

3: $i = 0$ to num_day

4: **if** $CHO(i, 2) == 0$ **then**

5: $CHO(i, 2) = 0.000001$;

6: **if** $CHO(i, 4) == 0$ **then**

7: $CHO(i, 4) = 0.000001$;

$Ameals = [Ameals, CHO(i, :)]$; ▷ Final CHO vector to set in the scenario file

▷ TIME VECTOR

$b\_hours = -a\_hours$ ▷ Left-hand limit of each uniform distribution for timing

8: $i = 0$ to length($hours\_st$)

9: $timing(:, i) = hours\_st(i) + [b\_hours(i) + (a\_hours(i) - b\_hours(i)) * rand(num\_day, 1)]$ ▷ Time vector modified

$day\_in\_hours = 0 : 24 : (24 * (num\_day - 1))$ ▷ Vector with days expressed in hours (24, 48, 72...)

$Tmeals = []$ ▷ Initialize the vector *Tmeals* that will contain the final meal timing

10: $i = 0$ to num_day

11: $meal\_hour\_per\_day = [day\_in\_hours(i) + timing(i, :)]$;

12: $Tmeals = [Tmeals, meal\_hour\_per\_day]$ ▷ Final time vector to set in the scenario file

The scenario for the first dataset (*standard dataset*) involves the built-in control algorithm of the UVA/Padova simulator, which automatically computes the insulin dosages to be administered, thus simulating an ideal T1DM management. Differently, in the second scenario, the presence of hyperglycemic and hypoglycemic events is forced in order to test the predictor robustness with a more realistic dataset. In fact, in real life, abrupt increases or decreases in blood sugar levels happen mostly due to incorrect calculation of the ingested CHO amount. Nonetheless, to reproduce such situations, the

algorithm simulates the wrong insulin bolus administrations. To achieve this goal, we first exploited the UVA/Padova simulator with its own optimal bolus control to extract the vector of injected boluses; then, the ideal bolus vector was manually modified by adding random noise sampled from a uniform distribution in the interval $[-3, 3]$ [127]. This vector was then customized for each patient by taking into account their specific *carbo-to-insulin ratio* (CR) according to the following formula:

$$ib_{CR}(i) = ib(i) + 10/CR \tag{9.1}$$

where ib_{CR} is the personalized insulin bolus vector, ib is the bolus vector modified through the normal distribution, and CR is the specific carbo-to-insulin ratio for the specific patient. According to its definition, a CR = 10 means that for 10g of CHO, the subject needs 1 unit of insulin. Therefore, for a fixed CHO quantity, the higher the CR, the fewer units of insulin will be needed to cover that meal. As a consequence, according to equation 9.1, the variability added to the normal insulin bolus is higher when CR is low and vice versa. In particular, the adverse event is induced only once per day. The specific daily meal in which the error that will cause the abnormal blood glucose will be introduced by increasing/decreasing is chosen randomly through a logical vector with 1×5 dimension, where 5 is the number of meals per day, containing only one nonzero value, corresponding to the position that will contain the erroneous bolus. Finally, the modified bolus vector was given as an effective bolus vector to the UVA/Padova to run the simulations for this scenario. As a result, the final dataset (*dataset with outliers*) is characterized by more occurrences of hypo/hyperglycemia, which makes it more similar to real data.

Both the generated datasets contain information extracted directly from the results of the UVA/Padova simulations and, among these, the ones of interest are CGM and insulin data (bolus and basal insulin were added together and considered as one). Moreover, a third manually-generated feature has been added to the datasets: the Insulin-On-Board (IOB), which represents an estimate of the amount of insulin still active in the patient's body after a bolus injection. For the Insulet pump, which is the one considered by the simulator, the active insulin time is equal to 3 hours and the shape of the insulin action plot is linear [94]. Thus, the value of IOB for each timestamp t, as defined in previous chapters and reported here to facilitate the reader is computed as:

$$IOB(t) = \sum_{k=0}^{179} a(k)I(t-k) \tag{9.2}$$

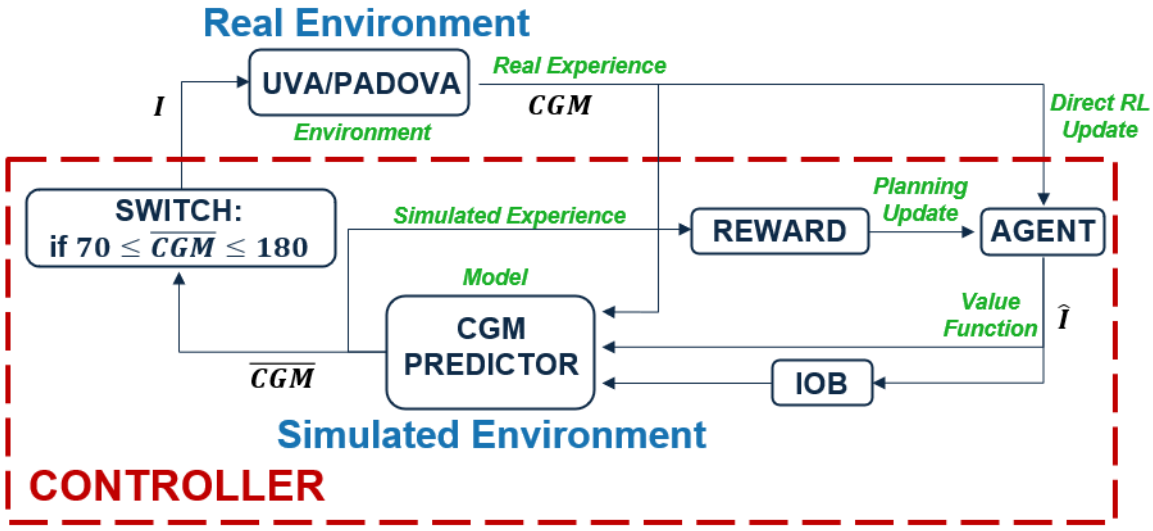


Figure 9.2.: Architecture of the Reinforcement Learning model for closed-loop glycemic control. The green text shows how blocks of the proposed model and their respective connections to each other reflect the general Dyna-Q algorithm architecture shown in Figure 9.1.

where $I(t-k)$ represents the value of insulin injected at timestamp $(t-k)$, $a(k) = (180-k)/180$ is the coefficient corresponding to the insulin decay curve discretized in accordance with a 1-minute timestamp of insulin delivery, and $k = 0, 1, 2, ..., 179$ are the total timestamps for the 3 hours of active insulin chosen for the Insulet pump. Therefore, the final patient-specific datasets, needed for the neural networks offline training, collect information on 3 time series for 30 days of simulation, 70% of which is used for training, 10% for validation, and the remaining 20% for testing. Before being given as input to the neural networks, the features undergo a pre-processing phase, consisting in a Z-score standardization that brings all the values to a normal distribution with zero mean and unit standard deviation. The forecasted value is then processed through a denormalization step to bring the CGM back into its normal range of variation using the mean and standard deviation values computed on the training set.

It is worth noting that the 30 days of data generated for each patient were utilized only for the training, validation, and test of the predictors to evaluate their performance retrospectively. Conversely, the simulations concerning glycemic control are performed on new data that are independent of the previous ones, in order not to introduce bias, and to evaluate the feasibility of the proposed approach for different scenarios.

9.1.2. Real validation dataset

It is known in the literature that prediction performance achieved on data from in silico patients could lead to an overestimation of the prediction capability. In order to evaluate the performance of the predictive model utilized in this study, we collected anonymized data from 12 real patients that were supplied from the Unit of Endocrinology and Diabetology of the Campus Bio-Medico University Hospital in Rome, Italy. All patients (8 females and 4 males, aged between 24 and 69) used a Medtronic GuardianTM sensor 3 with Medtronic Minimed 640G insulin pump and were monitored for a time ranging from 5 to 25 days (average 10.6 ± 5.1), for a total of 127 days of monitoring. Relevant clinical information of such patients, including gender, age, duration of T1D, average glycemic level, time in the target glycemic range of 70-180 mg/dL, C-peptide amount, HbA1c, comorbidities and T1D-related complications are reported in Table 9.1.

We performed tests on such patients twice: first, we trained the predictor only on patient-specific data; second, we trained the predictor on data from all the virtual UVA/Padova patients, and then fine-tuned the model on patient-specific data. In all tests, we maintained the normalization procedure and the 70/10/20 split utilized for the training, validation, and test of the virtual data; this ensures that at least an entire day of data is considered as a test set for each real patient.

9.2. Methods

The system proposed for the closed-loop control, whose architecture is shown in Figure 9.2, combines two Artificial Intelligence branches: a CGM predictor, implemented through a neural network, and an insulin controller, performed via a reinforcement-learning algorithm (DQN Agent + Reward). Unlike all the model-free RL algorithms presented in the literature, the architecture here developed is a model-based RL algorithm. In this case, the decision is made based on the interaction between an agent and a simulated environment. In practice, we use a Dyna-Q model in which the interaction is made with both a real and a simulated environment. As indicated in Figure 9.2, the simulated environment is the CGM predictor, which has the role of simulating the dynamics of the patient's insulin-glucose system. With reference to the same figure, the action I chosen by the actor is evaluated and optimized through a reward function calculated on the CGM predicted value ($\overline{CGM}$). Only when the insulin dosage hypothesized by the agent is considered to be adequate, in the sense that it will lead

Table 9.1.: Relevant clinical information concerning the real patients. For each patient are reported gender, age, duration of T1D in years, average blood glucose level with standard deviation (Avg BG) during the period of observation, time in range (TIR), C-peptide amount, HbA1c, basal insulin per day, and eventual comorbidities or complications.

Adults	Gender	Age	Duration	Avg BG $[mg/dL]$	TIR [%]	C-peptide $[ng/mL]$	HbA1c [%]	Basal insulin [U/die]	Comorbidities/Complications
R1	F	24	13	164 ± 52	63.8	<0.01	6.8	17.7	-
R2	M	40	34	170 ± 49	59.7	<0.01	5.7	25.9	Retinopathy, hypothyroidism
R3	F	26	20	157 ± 64	62.4	<0.01	7.8	24.8	PCOS
R4	M	69	1	193 ± 66	48.6	<0.01	7.6	13.2	Dyslipidemia, thyroiditis
R5	F	32	8	150 ± 42	73.7	<0.01	7.7	15.5	Hypothyroidism
R6	F	27	23	161 ± 63	61.0	<0.01	10	18.9	Thyroiditis
R7	M	46	32	167 ± 55	62.4	<0.01	7.5	15.6	Hypercholesterolemia
R8	F	46	17	195 ± 80	43.9	0.05	8.1	15.1	LES, Sjorgens, retinopathy
R9	F	45	4	148 ± 47	72.3	0.05	7.0	28.8	Basedow's disease
R10	F	58	16	183 ± 64	49.7	0.04	7.2	24.6	-
R11	F	28	23	235 ± 55	15.5	<0.01	8.6	22.4	Thyroiditis, pityriasis rosea
R12	M	62	40	146 ± 40	79.7	<0.01	7.7	60	-
Average	-	42 ± 15	19 ± 12	172 ± 26	57.7 ± 16.2	-	7.6 ± 1.0	23.5 ± 12.5	-

to a safe blood glucose level, the bolus is actually injected into the real environment, i.e. the UVA/Padova patient. In particular, from Figure 9.2 it can be seen that when the predicted CGM falls within the target range (TR) of 70-180 mg/dL, a switch is activated allowing the bolus to actually be delivered to the in silico patient. In this way, the information on insulin is fed back into the dynamic system of the simulator which returns the real CGM that, in turn, is given to the agent as information on the environment state, thus closing the control loop. The green text in Figure 9.2 shows how different parts of the proposed method mirror the general Dyna architecture. It is worth noting how the "model learning" part of the general Dyna architecture is not reported in this figure because the environment model is only trained "offline" on previous data before the closed-loop RL control begins.

9.2.1. Simulated Environment: CGM Predictors

The neural networks developed to simulate the environment have a *sequence-to-label* architecture: the input layer receives a sequence of 30 timestamps; it is followed by an LSTM layer with 30 units and a dense layer with 15 units, with Softsign and ReLU transfer functions, respectively; the output layer has only one regression neuron for the prediction of the CGM punctual value 30 minutes later. In fact, as pointed out in [128], an appropriate prediction horizon is required considering the timing of insulin action on blood glucose levels. In particular, in the study carried on by Cichosz et al. [128], by analyzing prediction lead times from 10 to 60 min on a large amount of data, it is shown how a PH of 30 minutes can lead to rather accurate and precise predictions when combined with an advanced machine learning model such as neural networks, because of their ability to model non-linear and non-stationary problems. Therefore, the neural network implemented forecasts the CGM punctual value 30 minutes later exploiting a 30-minute time window in input; in fact, in the UVA/Padova Simulator, the CGM is sampled at 1 minute (differently from what happens in real life where the CGM sampling happens every 5 minutes). The specific neural network architecture chosen is presented in Figure 9.3. The UVA/Padova simulator is implemented in Simulink; thus, after the offline training, the patient-specific tuned predictive model is included in a single block, i.e., an element that is used to build models in Simulink. In order to schematize the whole control, such block and the other required components were included in a *.slx* file, which was then set as the *Controller* of the UVA/Padova simulator to run the validation protocol.

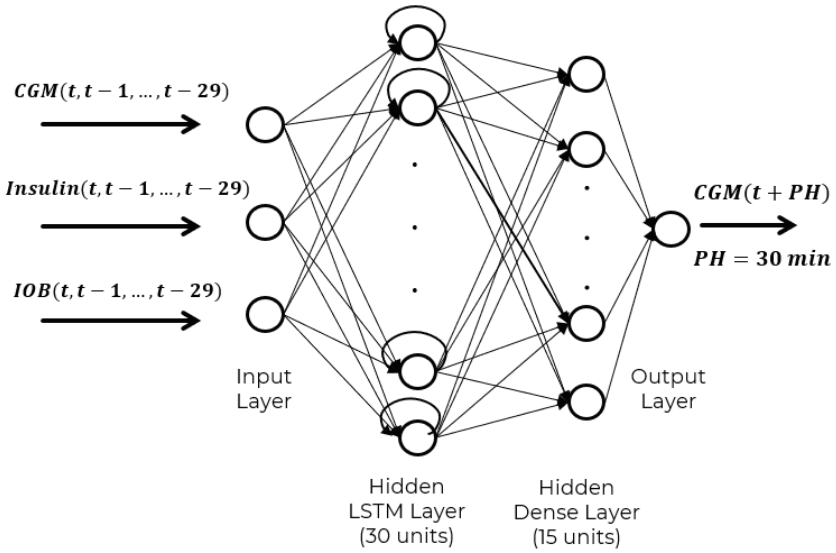


Figure 9.3.: Architecture of the Recurrent Neural Network developed: the model takes in input three time series, each with 30 timestamps, and returns as output the CGM punctual value with a prediction horizon of 30 minutes.

9.2.2. DQN Agent and Reward

The RL algorithm chosen for our model is a deep Q-Network (DQN) with 256 units implemented in Matlab R2022b. The DQN is an algorithm that allows the description of the space of environment state variables as continuous and the space of the possible actions as discrete. As a consequence, this algorithm seems quite suited for the diabetes problem, since the state variable of the CGM needs to be continuous, whereas the actions can be chosen as a finite set. In our case, the action set chosen is [0, 1, 2, 3, 4, 5] units of possible bolus to be injected. Since the goal of a glycemic controller is to maximize the TIR of blood glucose levels in the TR, the reward function implemented to train the agent is built in such a way that it returns a positive value only if the action, tested on the predictor, leads to a CGM within the TR. In particular, we selected a reward function that aims to maintain the glycemic level as close as possible to the center of the euglycemic range, while decreasing quickly as the glucose level differs from the target. The implemented reward function, reported in Figure 9.4, is expressed by the formula:

$$Reward = 1000 - (125 - CGM)^2 \tag{9.3}$$

It is worth noting that the first negative values of such a function correspond to CGM values of 93 and 157 mg/dL, in such a way that the reward value is already sensitively

negative when experiencing a blood glucose value of 70 or 180 mg/dL.

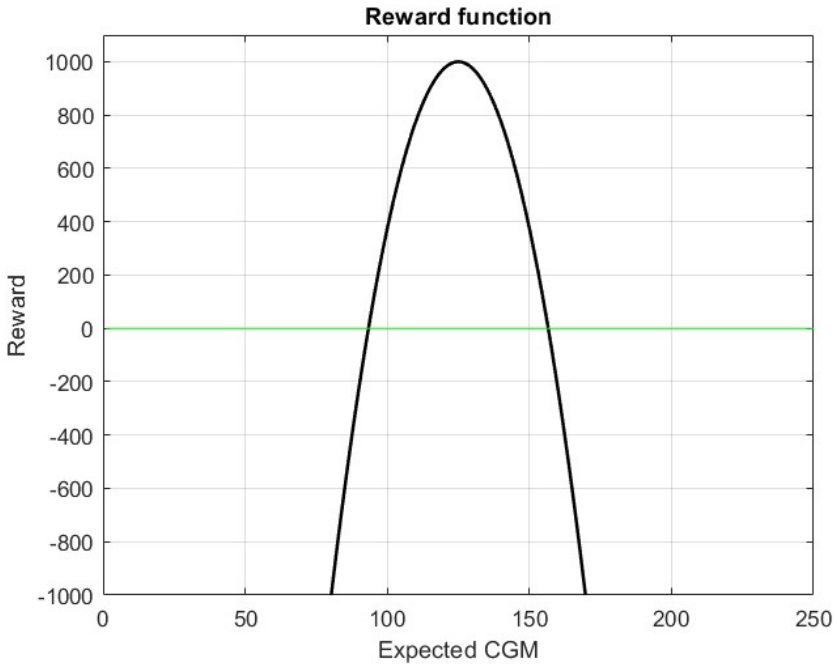


Figure 9.4.: The reward function chosen to train the agent such that hypoglycemic and hyperglycemic events are penalized severely, in order to minimize as much as possible the percentage time spent far from the center of the euglycemic range.

9.3. Results

The validation protocol adopted to test the proposed method consists of a first part to validate the performance of the CGM predictors and a second one to test the control algorithm; metrics adopted to evaluate the performance are distinct for each phase.

9.3.1. CGM Predictors

All the simulations regarding the neural networks have been implemented in Python using the Google Colab environment through the open source libraries of Keras and TensorFlow. We trained the model for virtual patients three times, first by using a pure precision-medicine approach (i.e. each model is trained only on patient-specific data), then by using a Leave-1-patient-out approach (i.e. each model is trained on all the data from patients different from that used for test), and finally by fine-tuning the models

pre-trained with a Leave-1-patient-out approach on the same patient-specific data used in the first approach, all by keeping the test set fixed. In addition, two different trials were performed to assess the predictors' performance: in both cases, the networks were trained on the standard dataset; then, in the first trial, the testing was performed on the test set of the standard dataset itself, whereas in the second trial on the test set of the dataset with outliers, in order to verify the CGM predictor robustness against sudden hypoglycemic and hyperglycemic events. The rationale behind this approach is to evaluate the robustness of the model to unexpected events of hypoglycemia and hyperglycemia. Similarly, we performed tests on patients from the real dataset twice, first by training the predictor only on patient-specific data, and then by training the predictor on data from all the virtual UVA/Padova patients, and then fine-tuning the model on patient-specific data; the latter approach ensures a larger training set also for patients that have been monitored for a short time.

All datasets have been divided as follows: 70% training, 10% validation, and 20% test. In particular, after tuning and optimizing the model on the validation set, the final performance was evaluated on the test set in terms of RMSE (equation 1.1) and MARD (1.3). In addition, we considered the CEGA as a measure of the clinical accuracy of the predictions. Table 9.2 reports the average results in terms of RMSE, MARD, and percentage of samples in the C-D-E zones of the CEGA for the 3 different training strategies adopted for the virtual patients, on both the standard dataset and on the dataset with outliers. Table 9.3 reports the detailed results of the approach which includes fine-tuning on patient-specific data after an initial leave-1-patient-out training, which from a comparison between the results appears to be the safest approach as it minimizes the MARD and the number of predictions in the C-D-E zones of the Clarke Error Grid. The reported results are referred to the realistic virtual dataset with outliers, whereas the detailed results on the standard dataset are omitted for brevity purposes.

With regard to the real dataset, a similar behavior can be observed, as the fine-tuned model achieves better performance than the model trained using a pure precision-medicine approach in terms of average RMSE, MARD, and CEGA. This could be due to the small amount of data available for single patients, which could be not sufficient to train a neural network for regression. Table 9.4 reports the detailed results of the predictions for the real patients using the fine-tuning approach, and the average results achieved using a precision-medicine approach.

Table 9.2.: Average results of the CGM predictors in terms of RMSE, MARD, and percentages of samples in the C-D-E zones of the CEGA, both with the dataset standard and with outliers. The results are reported for the 3 training approaches investigated, namely precision medicine (PM), leave-1-patient out (L1PO), and fine-tuning (FT) on patient-specific data after an initial L1PO training.

Training	Standard	Outliers	Standard	Outliers	Standard	Outliers
Approach	RMSE $[mg/dL]$		MARD [%]		C-D-E zone [%]	
PM	10.4	14.6	6.8	8.0	0.5	1.8
L1PO	10.0	12.9	6.3	7.4	0.2	1.2
FT	9.9	13.2	6.3	6.9	0.5	1.1

Table 9.3.: Results on the dataset with outliers of the CGM predictors shown in Figure 9.3 for the dataset with outliers using a fine-tuning approach on patient-specific data after initial leave-1-patient-out training.

Adults	RMSE $[mg/dL]$	MARD [%]	CEGA [%]				
			A	B	C	D	E
#001	14.2	7.6	94.4	4.7	0.0	0.9	0.0
#002	14.3	7.6	93.3	4.4	0.0	2.3	0.0
#003	12.5	6.6	95.6	3.9	0.0	0.5	0.0
#004	13.6	7.9	94.4	4.8	0.0	0.8	0.0
#005	8.6	0.9	98.9	1.1	0.0	0.0	0.0
#006	10.7	6.7	96.1	3.0	0.0	0.9	0.0
#007	21.8	9.5	90.5	7.8	0.0	1.7	0.0
#008	11.1	6.0	97.8	2.1	0.0	0.2	0.0
#009	14.2	8.8	92.2	5.8	0.0	2.0	0.0
#010	11.2	7.8	94.8	3.6	0.0	1.6	0.0
average	13.2	6.9	94.8	4.1	0.0	1.1	0.0

Table 9.4.: Results of the CGM predictors for the dataset composed of real patients using a fine-tuning (FT) approach on patient-specific data after initial leave-1-patient-out training. The bottom line of the table reports the average results when using a precision medicine (PM) approach.

Adults	RMSE $[mg/dL]$	MARD [%]	CEGA [%]				
			A	B	C	D	E
R1	15.6	9.7	87.0	13.0	0.0	0.0	0.0
R2	14.5	6.7	95.4	4.6	0.0	0.0	0.0
R3	9.3	5.1	99.8	0.2	0.0	0.0	0.0
R4	16.3	6.1	97.2	2.8	0.0	0.0	0.0
R5	13.3	8.3	94.7	4.6	0.0	0.7	0.0
R6	25.3	13.2	78.3	19.8	0.0	1.9	0.0
R7	6.2	3.2	99.6	0.4	0.0	0.0	0.0
R8	25.5	10.0	90.1	9.8	0.1	0.0	0.0
R9	18.6	9.5	89.4	9.7	0.0	0.9	0.0
R10	11.3	5.3	96.3	3.7	0.0	0.0	0.0
R11	12.1	3.3	99.8	0.2	0.0	0.0	0.0
R12	12.6	6.1	97.2	2.8	0.0	0.0	0.0
average FT	15.0	7.2	93.7	6.0	<0.01	0.3	0.0
average PM	17.0	8.5	90.9	7.9	0.0	1.2	0.0

9.3.2. Control

The simulations implemented to test the control model proposed has been conduced on Matlab R2022b, since it is the environment in which the UVA/Padova Simulator is implemented. In particular, for each subject, the patient-specific control was included in the overall Simulink block diagram of the simulator. Specifically, to evaluate the robustness of the controller, it has been tested on 4 different scenarios, that differ in the simulation length (from 6 to 24h) and the number of meals (from 1 to 3), as shown in Table 9.5. The reason behind this choice is to test first whether or not our model is able to control a single meal, over a more or less long scenario (6h and 12h). After that, in order to level up to a more realistic simulation, the number of meals was increased, to 2 in 12h and, finally, to 3 in 24h.

The results obtained for each of the scenarios presented are listed in Table 9.6. Since the glycemic control goal is to keep blood glucose within the physiological range and avoid performing actions that may lead to hypoglycemic or hyperglycemic events, an effective strategy to evaluate the soundness of the controller is to measure the percentage of time that blood glucose is maintained in the TR, which is calculated as defined in

Table 9.5.: Validation protocol utilized to validate the control algorithm performance. The proposed four scenarios are aimed to test the control with a gradually increasing number of hours and meals.

	Duration [h]	NÂ° Meals
Scenario 1	6	1
Scenario 2	12	1
Scenario 3	12	2
Scenario 4	24	3

Equation 1.5. This is why the performance of the implemented controller is evaluated by measuring the percentage time during a simulation during which blood glucose is in hypoglycemia, hyperglycemia, and TR. Also, the use of TR as a metric for assessing controller performance allows us to compare the results obtained with those reported in the literature. In addition, we report in Table 9.7 a detailed analysis of the 24-hour scenario, which also reports the percentage of time spent in severe hypoglycemia ($\leq 50\, mg/dL$) and in severe hyperglycemia ($\geq 300\, mg/dL$) for each patient, in order to provide a more in-depth analysis for the simulation on the longest scenario.

9.4. Discussion

The main objective of the closed-loop control model proposed in this experimental work is to decide, on the basis of the current glycemic value, the units of insulin bolus to be injected into the patient. This insulin value is given as input to the predictor, which tests its correctness by predicting the CGM value after 30 minutes. It follows that the accuracy of the prediction returned by the neural network is a key point in order to obtain a reliable control. In the following, we discuss separately the results achieved for prediction and for control.

9.4.1. Predictor performance analysis

The results in Table 9.3 show that the RMSE for each patient on the dataset with outliers varies from a minimum of 8.6 mg/dL for patient #005 to a maximum of 21.8 mg/dL for patient #007. The trends of the CGM predicted by the best and the worst neural networks are plotted in Figure 9.5, considering both the real and the virtual datasets with outliers.

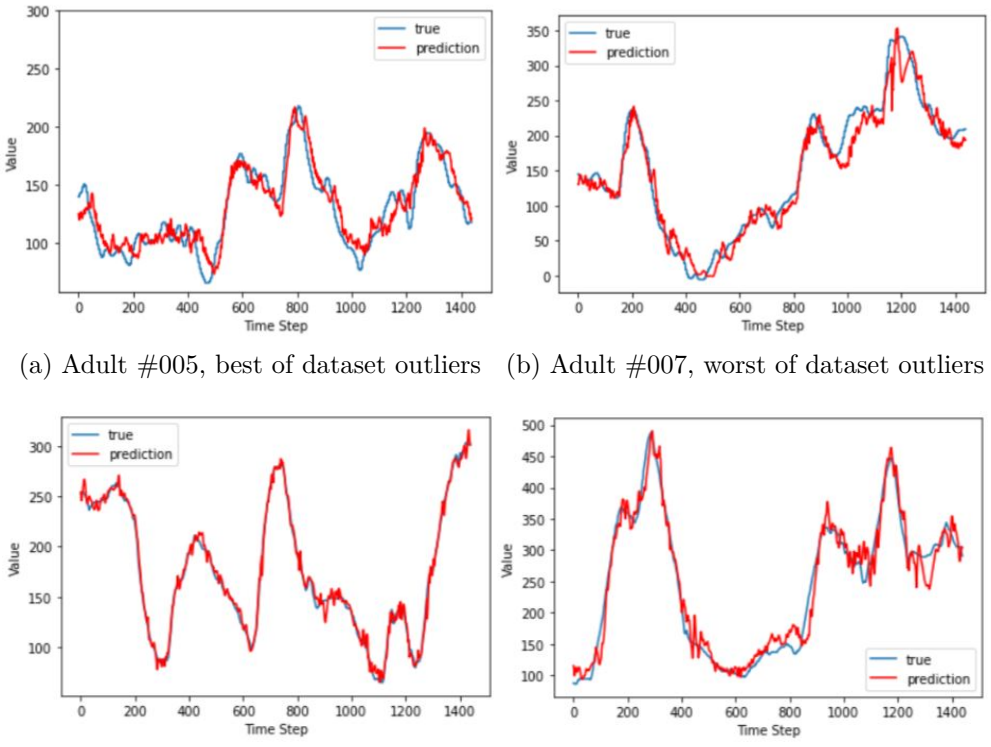
Considering the RMSE relative to each standard subject, it results in an average of

Table 9.6.: Percentage of time spent in the hypoglycemia (HYPO), hyperglycemia (HYPER), and target range (TR) for different scenarios using the proposed model-based reinforcement learning control.

	SCENARIO 1			SCENARIO 2			SCENARIO 3			SCENARIO 4		
Adult	HYPO [%]	TR [%]	HYPER [%]	HYPO [%]	TR [%]	HYPER [%]	HYPO [%]	TR [%]	HYPER [%]	HYPO [%]	TR [%]	HYPER [%]
#001	0	100	0	0	100	0	0	89	11	0	79	21
#002	0	100	0	0	100	0	0	89	11	0	78	22
#003	5	95	0	3	97	0	3	87	10	13	53	31
#004	0	100	0	0	100	0	0	91	9	0	79	21
#005	0	71	29	0	79	21	0	73	27	0	28	72
#006	0	53	47	0	75	25	0	70	30	0	62	38
#007	32	68	0	16	84	0	16	75	9	6	16	78
#008	0	52	48	0	55	45	0	44	56	0	47	53
#009	0	71	29	0	82	18	0	73	27	0	81	19
#010	0	58	42	0	79	21	0	75	25	0	84	16
average w/ adult#007	3.7	76.8	19.5	1.9	85.1	13	1.9	76.6	21.5	1.9	60.7	37.1
average w/o adult#007	0.5	75.2	24.3	0.3	82.2	17.5	0.3	73.5	26.2	1.3	63.8	34.6

Table 9.7.: Detailed and average results of the glycemic control for scenario 4 (24 hours of simulation) for the 10 adult patients, including the percentage of time spent in severe hypo- and hyperglycemia.

Adults	Severe HYPO	HYPO	TR	HYPER	Severe HYPER
#001	0	0	79	21	0
#002	0	0	78	22	0
#003	0	13	53	31	0
#004	0	0	79	21	0
#005	0	0	28	62	10
#006	0	0	62	38	0
#007	4	2	16	56	22
#008	0	0	47	53	0
#009	0	0	81	19	0
#010	0	0	84	16	0
average	0.4	1.5	60.7	33.9	3.2



(a) Adult #005, best of dataset outliers (b) Adult #007, worst of dataset outliers

(c) Adult R7, best of real dataset (d) Adult R8, worst of real dataset

Figure 9.5.: Predicted and real CGM trends obtained for the best (left) and worst (right) patient for the virtual with outliers (top) and the real (bottom) dataset, over 24 hours of prediction.

9.9 mg/dL, which is a good and competitive score compared to what has been reported in the literature. In fact, in [129] the average RMSE is equal to 22.0 mg/dL for a model involving an LSTM neural network and trained on in silico data from the UVA/Padova simulator. Results closer to ours were obtained in [130], where a mean RMSE of 12.6 mg/dL is obtained with an LSTM trained on a higher number of in silico patients tough (100 against the 10 exploited in our work). The results concerning MARD and CEGA are noteworthy as well. In addition, it is worth noting that the implemented CGM predictors also perform well in the presence of sudden hypoglycemic and hyperglycemic events. This consideration comes from the comparison with the mean RMSE, MARD, and CEGA obtained on the test of the virtual dataset with induced adverse events: the little difference between the two values (about 3.3 mg/dL) indicates how the generated neural networks are robust to outliers. Also, RMSE values similar to ours are also obtained in [131], even though only qualitative comparisons are possible given the large differences in the methodology adopted. In their study, the time window of input data is 90 minutes and the amount of data used to train the neural network is extremely large compared to ours, but it should be emphasized that their model has been tested on a large population of real patients.

We validated the performance of the CGM predictors using different approaches and by testing on real data. With regard to the real patients, we achieved RMSE scores of 17.0 mg/dL and 15.0 mg/dL using the precision-medicine and the fine-tuning approaches, respectively. With regard to the virtual subjects, we achieved RMSE scores of 14.6 mg/dL, 12.9 mg/dL, and 13.2 mg/dL using the three precision-medicine, the leave-1-patient-out, and the fine-tuning approaches, respectively. This shows a difference of 1.8 mg/dL between the average results achieved on virtual and real patients. Similarly, the MARD and the number of predictions in the A+B zones of the Clarke Error Grid are similar. The MARD scores are 8.5% and 7.0% for real patients on the two tests, whereas the scores are 7.9%, 7.4%, and 6.9% for the virtual patients. The number of predictions in the A and B zones of the Clarke Error Grid are 98.8% and 99.7% for real patients, and 98.2%, 98.8%, and 98.9% for simulated patients. These results demonstrate that fine-tuning the models using a precision-medicine approach provides sensitively better results compared to using a general-purpose model trained only on data from patients different from the one under consideration. This could be expected based on previous studies in the literature which demonstrated the major effectiveness of the patient-specific approach in T1D [21, 13]. In addition, the similar results achieved for virtual patients with outliers and real patients confirm the robustness of the proposed

predictor. In particular, it is worth noting how only 0.3% of the predictions on the real dataset fall outside of the safe A+B zone of the Clarke Error Grid.

9.4.2. Controller performance analysis

Moving to the actual control part, from the average percentage results shown in Table 9.6, it can immediately be observed that the time spent in the TR decreases as the number of meals in the scenario increases. This can be expected because T1D patients often experience postprandial hyperglycemia even in conditions of proper insulin therapy. The proposed control algorithm is able to maintain blood glucose in the TR for an average percentage time of 76.8 and 85.1, in scenarios 1 and 2, respectively. In both scenarios, CGM is maintained in the physiological range by delivering a single bolus that, in the presence of a single meal, prevents hyperglycemia, causing 0 cases in 8 out of 10 patients. Notably, even mild hyperglycemia is avoided for adults #001, #002 and #004. Nonetheless, the algorithm produces a notable percentage of time spent in hypoglycemia for adult #007, because the insulin bolus injected is too large, whereas adult #003 spends 5% of the time in mild hypoglycemia, which is then adjusted by the delayed effect of the CHO absorption. Furthermore, passing from scenario 1 to scenario 2, it can be observed that the time spent in hyperglycemia and hypoglycemia diminishes for all patients in favor of the time in the TR, and the average TIR passes from 76.8% to 85.1%; again, the control for adult #007 is not optimal. This stresses the fact that on a simulation lasting twice (12 hours compared to 6 hours in scenario 1), with the same amount of CHO ingested, the algorithm is able to control well the glycemic trend during the fasting phase too. The performance begins to worsen as the number of meals during the simulation increases. In fact, in scenario 3, sometimes the control continues to deliver a single bolus even though the simulation contains 2 meals, causing an increase in the percentage of time spent in hyperglycemia and leaving hypoglycemic events completely unaffected, while the TIR decreases to 76.6%. This result could be due to the fact that the DQN implementing the agent, in only 12 hours of simulation, cannot train so well as to be able to control multiple meals. In contrast, in the 24h scenario, the agent begins to release more boluses and this occurs just as the CGM predicted by the network exceeds the 180 mg/dL limit. The average time spent in the TR over all the patients is 60.7% (63.8% if we do not consider adult #007). The majority of the remaining time is spent in postprandial hyperglycemia; conversely, only 2 patients out of 10, namely adults #003 and adult #007, experience hypoglycemia during the day. The latter result is particularly important, as hypoglycemia is the event that should be avoided by any

controller; in our model, the average time spent in hypoglycemia is 1.9%, all due to adults #003 and #007. With regard to the last scenario, it is also interesting to look at the percentage of time spent in severe hypoglycemia and hyperglycemia for each patient. As it can be observed in Table 9.7, the vast majority of hypoglycemic and hyperglycemic events are mild. Adult #005 experiences 10% of the time in severe hyperglycemia, due to a missing bolus in the occurrence of the lunch, whereas adult #007 experiences both severe hypoglycemia and severe hyperglycemia. All the other subjects do not experience extreme conditions.

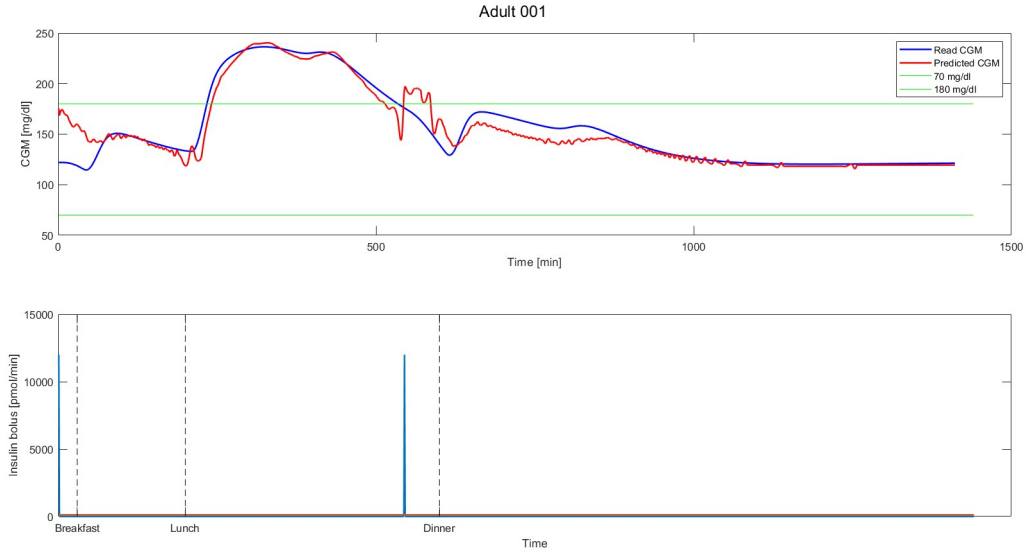
A separate discussion is necessary about adult #007. The proposed system is not able to provide appropriate control, whatever the length of the scenario. It is interesting to note that this is also the virtual patient for which the worst performance is achieved in terms of future glucose prediction accuracy, as reported in Table 9.3. In order to understand the optimal control strategy for this patient, we have run a simulation on scenario 4 using the UVA/Padova simulator built-in control system, which exploits knowledge of the ingested CHO and on all the physiological features of the patient. In the occurrence of the different meals, this control supplies 0.7, 3.3, and 2.7 units of insulin, by keeping the glucose in the TR for 95% of the simulation time. The control system proposed in this study is not able to provide fractions of units of insulin bolus. In an attempt to improve the performance on this patient, we included half-unit boluses in the action space of the DQN agent, which did not improve performance. Finally, we investigated if some considerably different feature exists for this patient compared to the others, which could explain such a marked difference in performance. We found out that a considerably lower body weight (47 kg against an average of 76 kg for the other patients) and a much lower value of the Michaelis-Menten constant for the computation of the insulin-dependent utilization of glucose (184.7 against an average of 224.5 mol/m^3 for the other patients) might influence in a drastic way the glucose oscillations of this patient. This can be observed also in Figure 9.5b, where a slight deviation of the bolus, such as the one introduced by Equation 9.1, from its optimal value causes a dramatic increase or decrease in the glycemic level up to 350 mg/dL or down to 0 mg/dL. It should also be taken into consideration that, in real life conditions, a patient would hardly be able to make an estimate of the ingested CHO and inject corresponding fragmented units of insulin as accurately as done by the simulator, which utilizes a deterministic system that is unrealistic to be used as a mobile device.

In Figure 9.6, the trends regarding CGM and bolus injections are reported for one of the best patients (Adult #001) and for the worst patient in terms of total time

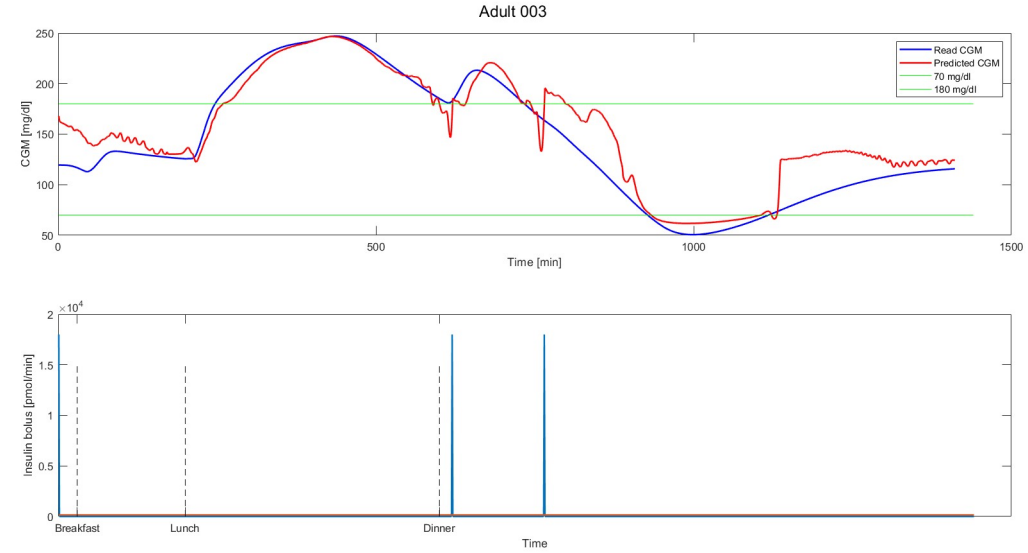
spent in hypoglycemia (Adult #003); these plots are referred to scenario 4 (24h and 3 meals). The predicted CGM track has been shifted by 30 minutes in order to overlap the prediction with the corresponding observed value. It is worth noting that the real and predicted CGM tracks should not perfectly overlap in this plot, because the latter reports, timestamp by timestamp, the expected glucose level based on a free glucose variation. Regarding Adult #001, as it can be observed in Figure 9.6a, the first prediction by the CGM predictor indicates high blood glucose value and, consequently, the controller releases a bolus equal to 12000 $pmol/min$ (2 units). This insulin injection prevents the previously predicted hyperglycemia and leads to a blood glucose level in the TR. Then, following lunch, there is a mild hyperglycemia lasting 11% of the total simulation time. Although this event is predicted by the neural network, another bolus is not injected; this could be due to a prediction of hypoglycemia in the eventuality of a further insulin bolus. However, the effect of basal insulin alone takes the glycemic level back into the target range. Another hyperglycemia is predicted in the proximity of the dinner, and a further bolus is injected preventing further hyperglycemia. Finally, during the last fasting phase, the glycemic level is taken very close to the target level of 125 mg/dL. Instead, with regard to Adult #003, as it can be seen in Figure 9.6b, there are 3 boluses released against 3 meals simulated by the scenario. Although a total time of 31% of the day is spent in mild hyperglycemia, an additional bolus of 18000 $pmol/min$ supplied after dinner generates a hypoglycemic event for 13% of the total simulation time, with the blood glucose level going down to 51 mg/dL. However, the delayed effect of the CHO absorption takes the glucose level close to the target value for the final hours of the simulation.

9.4.3. Contributions and Limitations

In this section, the main contributions of the proposed approach are summarized, and a comparison with different control approaches applied to the generated scenarios is proposed; the main limitations are also analyzed. A significant contribution of the proposed experimental work is the fact that the controller does not receive any information about ingested CHO. Instead, this information is given as a state variable of the environment in [62] (model-free DQN, 80.94% in TR) and [63] (model-based RL, 66.7% in TR). However, it should be noted that the lack of a standard scenario, to which the results can be uniformly referred, makes possible only qualitative comparisons with studies in the literature. We can perform a comparison with the study of Yamagata et al. [63], as it was the only one to propose a model-based approach and thus applicable in real life. They



(a) Adult #001 specific controller



(b) Adult #003 specific controller

Figure 9.6.: CGM trends and corresponding insulin boluses obtained by simulating the 24h and 3 meals scenario using the UVA/Padova software. The specific controllers of Adult #001 and Adult #003 are reported in (a) and (b), respectively, in order to present one of the best and worst in silico patients.

tested the first 3 adult patients, achieving an average TIR of 66.3%; for comparison, our model achieves a TIR of 70.0% on the same 3 patients in the longest scenario. Thus, the proposed approach can achieve better performance in terms of TIR without exploiting information on CHO. A further comparison can be made with the performance of the UVA/Padova simulator controller during the generation of the dataset with outliers. Indeed, during that procedure, the optimal bolus value computed by the simulator was randomly modified according to Algorithm 2 with the action of Equation 9.1 in order to include the effect of human errors inherent in the manual calculation of ingested CHO, and thus mimicking as much as possible the effect of a hybrid closed-loop control. Table 9.8 reports the detailed performance of this control strategy, computed by counting how many hypoglycemic and hyperglycemic events are observed for each patient in the dataset with outliers. As it can be observed, the percentage of TIR is sensitively greater compared to the proposed approach, whereas the time spent in hyperglycemia diminishes to a total of 6.9%, and severe hypoglycemia is experienced only by adult #007; however, the control on this patient is considerably more effective. Conversely, all but one patient experience episodes of hypoglycemia, and 7 of them experience severe hypoglycemia. Such behavior is observed also in real data, where, besides hypoglycemia, the percentage of time spent in hyperglycemia is also sensitively higher. From a comparison with the results reported in Table 9.7, the proposed control reduces by 1.1% the average time spent in hypoglycemia, and achieves considerably better performance in the prevention of severe hypoglycemic episodes. It is worth stressing that also this type of control, differently from the proposed one, takes into account CHO information. As a further comparison, we tested the proposed approach using a different reward function, which is the step reward function adopted in [58] and has the following mathematical formula:

$$\begin{cases} R = 0.5 & 70 < CGM \leq 180 \\ R = -0.8 & 180 < CGM \leq 300 \\ R = -1 & 300 < CGM \leq 350 \\ R = -1.5 & 30 < CGM \leq 70 \\ R = -2 & \text{else} \end{cases} \tag{9.4}$$

We report the results achieved using this function in scenario 4 in Table 9.9. Although the average TIR is comparable to that of the proposed approach reported in Table 9.6, the percentage of time spent in hypoglycemia is sensitively larger (7.3%) and due to 4 different patients. Since hypoglycemia is considerably more dangerous in the short term

Table 9.8.: Control results of the built-in glycemic controller of the UVA/Padova simulator with the addition of noise on the optimal amount of bolus, computed from the virtual dataset with outliers.

Adults	Severe HYPO	HYPO	TR	HYPER	Severe HYPER
#001	1	1	91	7	0
#002	1	2	94	3	0
#003	0	1	94	5	0
#004	1	2	91	6	0
#005	1	2	90	7	0
#006	0	1	92	7	0
#007	3	2	76	18	1
#008	0	0	90	10	0
#009	3	4	88	5	0
#010	1	4	95	0	0
average	1.1	1.9	90.1	6.8	0.1

than hyperglycemia, we conclude that the proposed reward function is more reliable than the step function in a real-life application.

A second benefit is given by the absence of a real training of the agent that, instead, is trained during the simulation period itself. This makes the proposed system feasible in reality; classic RL algorithms, on the contrary, require long training periods and would act directly on the patient. For comparison, the model proposed by Fox et al. [57] utilizes 2 years of data for model-free training and achieves an average TIR of 72%. Nonetheless, the presented study has some limitations. First, the controller is not able to provide effective control for a patient with very specific features such as adult #007, also due to its inability to supply fractions of units of insulin bolus. Second, the available version of the simulator does not allow taking into consideration real-life conditions such as physical activity, illness, or stress, which play a key role in glycemic management, and would make glycemic control more challenging.

Table 9.9.: Control results of the proposed approach using the step reward function utilized in [58].

Adults	HYPO [%]	TR [%]	HYPER [%]
#001	9	71	20
#002	0	76	24
#003	6	73	21
#004	0	63	37
#005	0	49	51
#006	0	65	35
#007	24	48	28
#008	0	61	39
#009	34	66	0
#010	0	79	21
average	7.3	65.1	27.6

10. Conclusions

This manuscript aims to provide a significant contribution in all the fields of application of AI methodologies to T1D management.

Studies concerning the regression task have introduced a novel neural network for the forecasting on adult patients during daily-life activity. This neural network outperforms state-of-the-art models although it exploits a much smaller amount of data for training. A further study presented an analysis of different learning techniques for the training of a neural network on data of T1D patients that regularly perform physical activity, investigating offline training, online training, and online training with a penalty, and it was observed that the improvement in performance generated by continuously updating the model with the most recent data is not as large as to justify the notable increase of the computational burden. Another study investigated the optimal amount of data for training an AI algorithm for the application on an edge-computing device, achieving a plateau in predictive performance when more than 60 days of data are used for training, whereas the time necessary for predictions on the edge-computing device is far below the time constraints imposed by the specific task. In addition, an edge-computing application has been developed for the forecasting of glycemic levels of pediatric patients; the analyzed deep networks outperform models in the literature in terms of clinical accuracy; the models have been implemented on two different edge devices using two different prediction reconstruction approaches, and no considerable performance decrease has been observed when running the tests.

A study has investigated the classification task. A layered meta-learning approach has been presented for the prediction of hypoglycemic and hyperglycemic events of adult patients during daily-life activities and during sports, and the system has been implemented on an edge-computing device. It exploits techniques for imbalanced datasets and pursues a univariate approach by utilizing only CGM data. It was observed that using meta-learning improves performance considerably compared to using the baseline model alone.

With regard to the control task, a new glycemic closed-loop control based on Dyna-Q

has been presented that does not necessitate information on CHO, not requiring any human intervention and thus providing a fully closed-loop controller. The proposed system is capable of achieving a noteworthy TIR while producing a very limited amount of hypoglycemic events, and outperforms a realistic manual control.

Despite the notable advances introduced by the application of AI models to the management of T1D, some major concerns still remain that limit the real-life application of such technologies. First, it should be kept in mind that the prediction of future glycemic levels, and of hypoglycemia in particular, is a challenging task [42]. On the one hand, using a univariate approach to model these dynamics may result oversimplified because important and influential features such as insulin, CHO, physical activity, and stress are not taken into account; on the other hand, gathering several heterogeneous features including unstructured data complicates considerably the data management and integration; this task results even more complicated when running the prediction on edge devices that need to communicate with all the data recording devices. In practice, a trade-off has to be identified between performance and efficiency. Second, further background and psychological factors should be taken into account when developing a medical decision support system. Indeed, some patients that are not familiar with the utilization of technological devices may find it difficult to utilize an AI tool; moreover, anxiety related to T1D and fear of hypoglycemia has been observed in adolescent patients [132]. These factors could lead to the abandonment of the technological system, with consequent detrimental effects on health outcomes. Bearing this in mind, a good solution may consist in a digital helper that combines the predictions of the AI system with psychological considerations and natural language processing techniques to help patients cope with the disease [133], or, alternatively, a fully-automated glycemic controller that the patient can completely trust, without the need to perform several therapy adjustments. Third, the application of a fully closed-loop controller is limited by the potentially severe consequences of malfunctioning. Indeed, although a single-hormone, fully closed-loop system could achieve performances that are comparable to hybrid closed-loop systems [9], the injection of an excessive bolus of insulin could lead to catastrophic consequences, as it could not be counteracted automatically as it would happen in a dual-hormone system; nonetheless, evidence on the effectiveness of the latter has not yet been assessed, and it could increase hyperglycemia during daytime [134]. In this frame, an optimal fully closed-loop controller could consist in a single-hormone control that relies on an effective meal detection module, capable to accurately estimate the amount of ingested CHO without a meal announcement [125].

For these reasons, future works will be directed towards the validation of the already developed predictive models on real patients to evaluate the improvement in their glycemic management through a clinical trial; several heterogeneous features will be utilized for the predictions with the goal of utilizing the minimum necessary amount of features to achieve the desired performances, also including a comparison with more deep-learning strategies. A digital helper based on a chatbot developed using natural language processing could be implemented to make the patients more aware of their condition based on the prediction of the AI models, and by understanding their sentiment expressed in a text [135]. With regard to the control task, different agents could be developed to investigate a continuous action space, and parallel training of the predictor and the controller could be investigated; finally, it would be interesting to study how the performance of the proposed control varies when using a meal-detector system to inject a bolus, or taking into account pediatric patients.

Bibliography

[1] FD Martini and Judi L Nath. Anatomia & fisiologia. *EdiSES, Napoli*, 1994.

[2] Henry Gray and Susan Standring. *Gray's anatomy: the anatomical basis of clinical practice*. Churchill Livingstone, 2008.

[3] J. Karry Jameson, Anthony S. Fauci, Dennis Kasper, Stephen Hauser, Dan Longo, and Joseph Loscalzo. *Harrison's principles of internal medicine*. McGrow Hill, 2018.

[4] Sarah K Lewis and Susan B Promes. Diabetic emergencies. *Prehospital Emergency Medicine Secrets E-Book*, page 65, 2021.

[5] Irene M Stratton, Amanda I Adler, H Andrew W Neil, David R Matthews, Susan E Manley, Carole A Cull, David Hadden, Robert C Turner, and Rury R Holman. Association of glycaemia with macrovascular and microvascular complications of type 2 diabetes (ukpds 35): prospective observational study. *Bmj*, 321(7258):405–412, 2000.

[6] World Health Organization. Diabetes. *https://www.who.int/news-room/fact-sheets/detail/diabetes accessed 2022-12-20*, 2022.

[7] Senseonic. Senseonics announces fda approval of the eversense e3 continuous glucose monitoring system for use for up to 6 months. *https://www.senseonics.com/investor-relations/news-releases/2022/02-11-2022-120033959*, 2022.

[8] Medtronic®. *MiniMed^{TM} 670G*, 2017. https://www.medtronic.com/us-en/healthcare-professionals/products/diabetes/insulin-pump-systems/minimed-670g.html.

[9] Michael A Tsoukas, Dorsa Majdpour, Jean-François Yale, Anas El Fathi, Natasha Garfield, Joanna Rutkowski, Jennifer Rene, Laurent Legault, and Ahmad Haidar. A fully artificial pancreas versus a hybrid artificial pancreas for type 1 diabetes: a single-centre, open-label, randomised controlled, crossover, non-inferiority trial. *The Lancet Digital Health*, 3(11):e723–e732, 2021.

[10] Omar Diouri, Monika Cigler, Martina Vettoretti, Julia K Mader, Pratik Choudhary, Eric Renard, and HYPO-RESOLVE Consortium. Hypoglycaemia detection and predic-

tion techniques: A systematic review on the latest developments. *Diabetes/Metabolism Research and Reviews*, 37(7):e3449, 2021.

[11] Cindy Marling and Razvan Bunescu. The OhioT1DM dataset for blood glucose level prediction. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 60–63, 2018.

[12] Cindy Marling and Razvan Bunescu. The ohiot1dm dataset for blood glucose level prediction: Update 2020. *5th International Workshop on Knowledge Discovery in Healthcare Data, ECAI*, 2020.

[13] Silvia Oviedo, Josep Vehí, Remei Calm, and Joaquim Armengol. A review of personalized blood glucose prediction strategies for T1DM patients. *International journal for numerical methods in biomedical engineering*, 33(6):e2833, 2017.

[14] Razvan Bunescu, Nigel Struble, Cindy Marling, Jay Shubrook, and Frank Schwartz. Blood glucose level prediction using physiological models and support vector regression. In *12th Int. Conf. on Machine Learning and Applications*, volume 1, pages 135–140. IEEE, 2013.

[15] Takoua Hamdi, Jaouher Ben Ali, Véronique Di Costanzo, Farhat Fnaiech, Eric Moreau, and Jean-Marc Ginoux. Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm. *Biocybernetics and Biomedical Engineering*, 38(2):362–372, 2018.

[16] Eleni I Georga, Vasilios C Protopappas, Demosthenes Polyzos, and Dimitrios I Fotiadis. A predictive model of subcutaneous glucose concentration in type 1 diabetes based on random forests. In *Int. Conf. of the IEEE Eng. in Medicine and Biology Society*, pages 2889–2892, 2012.

[17] Cooper Midroni, Peter J Leimbigler, Gaurav Baruah, Maheedhar Kolla, Alfred J Whitehead, and Yan Fossat. Predicting glycemia in type 1 diabetes patients: Experiments with XGBoost. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 79–84, 2018.

[18] Antonio Galicia, R Talavera-Llames, A Troncoso, Irena Koprinska, and Francisco Martínez-Álvarez. Multi-step forecasting for big data time series based on ensemble learning. *Knowledge-Based Systems*, 163:830–841, 2019.

[19] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[20] Iván Contreras, Arthur Bertachi, Lyvia Biagi, Josep Vehí, and Silvia Oviedo. Using grammatical evolution to generate short-term blood glucose prediction models. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 91–96, 2018.

[21] Jaques Reifman, Srinivasan Rajaraman, Andrei Gribok, and W Kenneth Ward. Predictive monitoring for improved management of glucose levels. *Journal of Diabetes Science and Technology*, 1(4):478–486, 2007.

[22] Giovanni Sparacino, Francesca Zanderigo, Stefano Corazza, Alberto Maran, Andrea Facchinetti, and Claudio Cobelli. Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series. *IEEE Transactions on Biomedical Engineering*, 54(5):931–937, 2007.

[23] Chiara Zecchin, Andrea Facchinetti, Giovanni Sparacino, and Claudio 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*, 113(1):144–152, 2014.

[24] John Martinsson, Alexander Schliep, Björn Eliasson, Christian Meijner, Simon Persson, and Olof Mogren. Automatic blood glucose prediction with confidence using recurrent neural networks. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 64–68, 2018.

[25] Taiyu Zhu, Kezhi Li, Pau Herrero, Jianwei Chen, and Pantelis Georgiou. A deep learning algorithm for personalized blood glucose prediction. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 74–78, 2018.

[26] Jianwei Chen, Kezhi Li, Pau Herrero, Taiyu Zhu, and Pantelis Georgiou. Dilated recurrent neural network for short-time prediction of glucose concentration. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 69–73, 2018.

[27] Arthur Bertachi, Lyvia Biagi, Iván Contreras, Ningsu Luo, and Josep Vehí. Prediction of blood glucose levels and nocturnal hypoglycemia using physiological models and artificial neural networks. In *3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI-ECAI*, pages 85–90, 2018.

[28] Kezhi Li, Chengyuan Liu, Taiyu Zhu, Pau Herrero, and Pantelis Georgiou. Glunet: A deep learning framework for accurate glucose forecasting. *IEEE journal of biomedical and health informatics, 2019*, 2019.

[29] Matteo Gadaleta, Andrea Facchinetti, Enrico Grisan, and Michele Rossi. Prediction of adverse glycemic events from continuous glucose monitoring signal. *IEEE Journal of Biomedical and Health Informatics*, 23(2):650–659, 2018.

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

[31] Juan Qiu, Qingfeng Du, Wei Wang, Kanglin Yin, and Liang Chen. Short-term performance metrics forecasting for virtual machine to support anomaly detection using hybrid ARIMA-WNN model. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 330–335. IEEE, 2019.

[32] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The UVA/PADOVA type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[33] Deepjyoti Kalita and Khalid B Mirza. Ls-grunet: Glucose forecasting using deep learning for closed-loop diabetes management. In *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, pages 1–6. IEEE, 2022.

[34] Mehrad Jaloli and Marzia Cescon. Long-term prediction of blood glucose levels in type 1 diabetes using a cnn-lstm-based deep neural network. *Journal of Diabetes Science and Technology*, page 19322968221092785, 2022.

[35] Grazia Aleppo, Katrina J Ruedy, Tonya D Riddlesworth, Davida F Kruger, Anne L Peters, Irl Hirsch, Richard M Bergenstal, Elena Toschi, Andrew J Ahmann, Viral N Shah, et al. Replace-bg: a randomized trial comparing continuous glucose monitoring with and without routine blood glucose monitoring in adults with well-controlled type 1 diabetes. *Diabetes care*, 40(4):538–545, 2017.

[36] Xiang Lu and Ruizhuo Song. A hybrid deep learning model for the blood glucose prediction. In *2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 1037–1043, 2022.

[37] Diabetes Research Studies. Rt_cgm dataset. *Journal of Community Health Research*, accessed 2023, March.

[38] Mu Yang, Darpit Dave, Madhav Erraguntla, Gerard L Cote, and Ricardo Gutierrez-Osuna. Joint hypoglycemia prediction and glucose forecasting via deep multi-task learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1136–1140. IEEE, 2022.

[39] Wonju Seo, You-Bin Lee, Seunghyun Lee, Sang-Man Jin, and Sung-Min Park. A machine-learning approach to predict postprandial hypoglycemia. *BMC Medical Informatics and Decision Making*, 19(1):210, 2019.

[40] Darpit Dave, Daniel J DeSalvo, Balakrishna Haridas, Siripoom McKay, Akhil Shenoy, Chester J Koh, Mark Lawley, and Madhav Erraguntla. Feature-based machine learning model for real-time hypoglycemia prediction. *Journal of Diabetes Science and Technology*, 15(4):842–855, 2021.

[41] Simon Lebech Cichosz, Jan Frystyk, Ole K Hejlesen, Lise Tarnow, and Jesper Fleischer. A novel algorithm for prediction and detection of hypoglycemia based on continuous glucose monitoring and heart rate variability in patients with type 1 diabetes. *Journal of diabetes science and technology*, 8(4):731–737, 2014.

[42] Omer Mujahid, Ivan Contreras, and Josep Vehi. Machine learning techniques for hypoglycemia prediction: Trends and challenges. *Sensors*, 21(2):546, 2021.

[43] Morten H Jensen, Claus Dethlefsen, Peter Vestergaard, and Ole Hejlesen. Prediction of nocturnal hypoglycemia from continuous glucose monitoring data in people with type 1 diabetes: a proof-of-concept study. *Journal of diabetes science and technology*, 14(2):250–256, 2020.

[44] Virginie Felizardo, Diogo Machado, Nuno M. Garcia, Nuno Pombo, and Pedro Brandão. Hypoglycaemia prediction models with auto explanation. *IEEE Access*, 10:57930–57941, 2022.

[45] Elena Daskalaki, Kirsten Nørgaard, Thomas Züger, Aikaterini Prountzou, Peter Diem, and Stavroula Mougiakakou. An early warning system for hypoglycemic/hyperglycemic events based on fusion of adaptive prediction models. *Journal of diabetes science and technology*, 7(3):689–698, 2013.

[46] Giacomo Cappon, Andrea Facchinetti, Giovanni Sparacino, Pantelis Georgiou, and Pau Herrero. Classification of postprandial glycemic status with application to insulin dosing in type 1 diabetes-an in silico proof-of-concept. *Sensors*, 19(14):3168, 2019.

[47] Yonit Marcus, Roy Eldor, Mariana Yaron, Sigal Shaklai, Maya Ish-Shalom, Gabi Shefer, Naftali Stern, Nehor Golan, Amit Zeev Dvir, Ofir Pele, et al. Improving blood glucose level predictability using machine learning. *Diabetes/Metabolism Research and Reviews*, page e3348, 2020.

[48] Francesco Prendin, Simone Del Favero, Martina Vettoretti, Giovanni Sparacino, and Andrea Facchinetti. Forecasting of glucose levels and hypoglycemic events: head-to-

head comparison of linear and nonlinear data-driven algorithms based on continuous glucose monitoring data only. *Sensors*, 21(5):1647, 2021.

[49] Virginie Felizardo, Nuno M Garcia, Nuno Pombo, and Imen Megdiche. Data-based algorithms and models using diabetics real data for blood glucose and hypoglycaemia prediction–a systematic literature review. *Artificial Intelligence in Medicine*, page 102120, 2021.

[50] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Personalized blood glucose prediction for type 1 diabetes using evidential deep learning and meta-learning. *IEEE Transactions on Biomedical Engineering*, 2022.

[51] Federico D'Antoni, Mario Merone, Vincenzo Piemonte, Giulio Iannello, and Paolo Soda. Auto-regressive time delayed jump neural network for blood glucose levels forecasting. *Knowledge-Based Systems*, page 106134, 2020.

[52] Jeremy Beauchamp, Razvan Bunescu, Cindy Marling, Zhongen Li, and Chang Liu. Lstms and deep residual networks for carbohydrate and bolus recommendations in type 1 diabetes management. *Sensors*, 21(9):3303, 2021.

[53] Sahar Zadeh Birjandi, Seyed Kamal Hosseini Sani, and Naser Pariz. Insulin infusion rate control in type 1 diabetes patients using information-theoretic model predictive control. *Biomedical Signal Processing and Control*, 76:103635, 2022.

[54] Gianni Marchetti, Massimiliano Barolo, Lois Jovanovic, Howard Zisser, and Dale E Seborg. An improved pid switching control strategy for type 1 diabetes. *ieee transactions on biomedical engineering*, 55(3):857–865, 2008.

[55] Katrin Lunze, Tarunraj Singh, Marian Walter, Mathias D Brendel, and Steffen Leonhardt. Blood glucose control algorithms for type 1 diabetic patients: A methodological review. *Biomedical signal processing and control*, 8(2):107–119, 2013.

[56] Elena Daskalaki, Peter Diem, and Stavroula G Mougiakakou. Model-free machine learning in biomedicine: Feasibility study in type 1 diabetes. *PloS one*, 11(7):e0158722, 2016.

[57] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. Deep reinforcement learning for closed-loop blood glucose control. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 5th Machine Learning for Healthcare Conference*, volume 126 of *Proceedings of Machine Learning Research*, pages 508–536. PMLR, 07–08 Aug 2020.

[58] Taiyu Zhu, Kezhi Li, Lei Kuang, Pau Herrero, and Pantelis Georgiou. An insulin bolus advisor for type 1 diabetes using deep reinforcement learning. *Sensors*, 20(18):5058, 2020.

[59] Zihao Wang, Zhiqiang Xie, Enmei Tu, Alex Zhong, Yingying Liu, Jichang Ding, and Jie Yang. Reinforcement learning-based insulin injection time and dosages optimization. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[60] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[61] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[62] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation. *IEEE Journal of Biomedical and Health Informatics*, 25(4):1223–1232, 2020.

[63] Taku Yamagata, Aisling O'Kane, Amid Ayobi, Dmitri Katz, Katarzyna Stawarz, Paul Marshall, Peter Flach, and Raúl Santos-Rodríguez. Model-based reinforcement learning for type 1diabetes blood glucose control. In *ECAI 2020 SP4HC Workshop*, 2020.

[64] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen. Edge ai: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications*, 19(1):447–457, 2020.

[65] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

[66] William R Hersh, Mark Helfand, James Wallace, Dale Kraemer, Patricia Patterson, Susan Shapiro, and Merwyn Greenlick. Clinical outcomes resulting from telemedicine interventions: a systematic review. *BMC Medical Informatics and Decision Making*, 1(1):1–8, 2001.

[67] Clemens Scott Kruse, Priyanka Karem, Kelli Shifflett, Lokesh Vegi, Karuna Ravi, and Matthew Brooks. Evaluating barriers to adopting telemedicine worldwide: a systematic review. *Journal of telemedicine and telecare*, 24(1):4–12, 2018.

[68] Mario Merone, Alessandro Graziosi, Valerio Lapadula, Lorenzo Petrosino, Onorato d'Angelis, and Luca Vollero. A practical approach to the analysis and optimization of neural networks on embedded systems. *Sensors*, 22(20), 2022.

[69] Taiyu Zhu, Lei Kuang, John Daniels, Pau Herrero, Kezhi Li, and Pantelis Georgiou. Iomt-enabled real-time blood glucose prediction with deep learning and edge computing. *IEEE Internet of Things Journal*, 2022.

[70] Scott M Pappada, Brent D Cameron, Paul M Rosman, Raymond E Bourey, Thomas J Papadimos, William Olorunto, and Marilyn J Borst. Neural network-based real-time prediction of glucose in patients with insulin-dependent diabetes. *Diabetes technology & therapeutics*, 13(2):135–141, 2011.

[71] Vincenzo Piemonte, Mauro Capocelli, Luca De Santis, Anna Rita Maurizi, and Paolo Pozzilli. A novel three-compartmental model for artificial pancreas: development and validation. *Artificial organs*, 41(12):E326–E336, 2017.

[72] Federico D'Antoni, Mario Merone, Vincenzo Piemonte, Paolo Pozzilli, Giulio Iannello, and Paolo Soda. Early experience in forecasting blood glucose levels using a delayed and auto-regressive jump neural network. In *2019 IEEE 18th Int. Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC19)*, pages 394–402. IEEE, 2019.

[73] F Dan Foresee and Martin T Hagan. Gauss-newton approximation to bayesian learning. In *Proceedings of the 1997 International Joint Conference on Neural Networks*, volume 3, pages 1930–1935, 1997.

[74] Javier Fernandez de Canete et al. Artificial neural networks for closed loop control of in silico and ad hoc type 1 diabetes. *Computer methods and programs in biomedicine*, 106(1):55–66, 2012.

[75] S de O Domingos, João FL de Oliveira, and Paulo SG de Mattos Neto. An intelligent hybridization of ARIMA with machine learning models for time series forecasting. *Knowledge-Based Systems*, 175:72–86, 2019.

[76] Andrea Facchinetti, Simone Del Favero, Giovanni Sparacino, Jessica R Castle, W Kenneth Ward, and Claudio Cobelli. Modeling the glucose sensor error. *IEEE Transactions on Biomedical Engineering*, 61(3):620–629, 2013.

[77] Ravi Reddy, Navid Resalat, Leah M Wilson, Jessica R Castle, Joseph El Youssef, and Peter G Jacobs. Prediction of hypoglycemia during aerobic exercise in adults with type 1 diabetes. *Journal of diabetes science and technology*, 13(5):919–927, 2019.

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

[79] Alberto Fernández, Salvador García, María José del Jesus, and Francisco Herrera. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, 159(18):2378–2398, 2008.

[80] Sheri R Colberg, Ronald J Sigal, Jane E Yardley, Michael C Riddell, David W Dunstan, Paddy C Dempsey, Edward S Horton, Kristin Castorino, and Deborah F Tate. Physical activity/exercise and diabetes: a position statement of the american diabetes association. *Diabetes care*, 39(11):2065–2079, 2016.

[81] Kenneth Robertson, Michael C Riddell, Benjamin C Guinhouya, Peter Adolfsson, and Ragnar Hanas. Exercise in children and adolescents with diabetes. *Pediatric diabetes*, 15(S20):203–223, 2014.

[82] Ryan A Williams, Simon Cooper, Karah J Dring, Lorna Hatch, John G Morris, Caroline Sunderland, and Mary E Nevill. Effect of acute football activity and physical fitness on glycaemic and insulinaemic responses in adolescents. *Journal of Sports Sciences*, pages 1–9, 2021.

[83] Sam N Scott, Matt Cocks, Rob C Andrews, Parth Narendran, Tejpal S Purewal, Daniel J Cuthbertson, Anton JM Wagenmakers, and Sam O Shepherd. High-intensity interval training improves aerobic capacity without a detrimental decline in blood glucose in people with type 1 diabetes. *The Journal of Clinical Endocrinology & Metabolism*, 104(2):604–612, 2019.

[84] Kamuran Turksoy, Elif Seyma Bayrak, Lauretta Quinn, Elizabeth Littlejohn, and Ali Cinar. Multivariable adaptive closed-loop control of an artificial pancreas without meal and activity announcement. *Diabetes technology & therapeutics*, 15(5):386–400, 2013.

[85] Nicole Hobbs, Iman Hajizadeh, Mudassir Rashid, Kamuran Turksoy, Marc Breton, and Ali Cinar. Improving glucose prediction accuracy in physically active adolescents with type 1 diabetes. *Journal of diabetes science and technology*, 13(4):718–727, 2019.

[86] Navid Resalat, Joseph El Youssef, Ravi Reddy, and Peter G Jacobs. Design of a dual-hormone model predictive control for artificial pancreas with exercise model. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2270–2273. IEEE, 2016.

[87] Chiara Dalla Man, Marc D Breton, and Claudio Cobelli. Physical activity into the meal glucose-insulin model of type 1 diabetes: In silico studies, 2009.

[88] Benedetta De Paoli, Federico D'Antoni, Mario Merone, Silvia Pieralice, Vincenzo Piemonte, and Paolo Pozzilli. Blood glucose level forecasting on type-1-diabetes sub-

jects during physical activity: A comparative analysis of different learning techniques. *Bioengineering*, 8(6):72, 2021.

[89] Lia Bally, Markus Laimer, and Christoph Stettler. Exercise-associated glucose metabolism in individuals with type 1 diabetes mellitus. *Current Opinion in Clinical Nutrition & Metabolic Care*, 18(4):428–433, 2015.

[90] Michael C Riddell, Ian W Gallen, Carmel E Smart, Craig E Taplin, Peter Adolfsson, Alistair N Lumb, Aaron Kowalski, Remi Rabasa-Lhoret, Rory J McCrimmon, Carin Hume, et al. Exercise management in type 1 diabetes: a consensus statement. *The lancet Diabetes & endocrinology*, 5(5):377–390, 2017.

[91] Federico D'Antoni, Lorenzo Petrosino, Fabiola Sgarro, Antonio Pagano, Luca Vollero, Vincenzo Piemonte, and Mario Merone. Prediction of glucose concentration in children with type 1 diabetes using neural networks: An edge computing application. *Bioengineering*, 9(5):183, 2022.

[92] Federico D'Antoni, Lorenzo Petrosino, Andrea Velieri, Daniele Sasso, Onorato D'Angelis, Tamara Boscarino, Luca Vollero, Mario Merone, and Vincenzo Piemonte. Identification of optimal training for prediction of glucose levels in type-1-diabetes using edge computing. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–5. IEEE, 2022.

[93] Joanne R Lupton, JA Brooks, NF Butte, B Caballero, JP Flatt, SK Fried, et al. Dietary reference intakes for energy, carbohydrate, fiber, fat, fatty acids, cholesterol, protein, and amino acids. *National Academy Press: Washington, DC, USA*, 5:589–768, 2002.

[94] Howard Zisser, Lauren Robinson, Wendy Bevier, Eyal Dassau, Christian Ellingsen, Francis J Doyle III, and Lois Jovanovic. Bolus calculator: a review of four "smart" insulin pumps. *Diabetes technology & therapeutics*, 10(6):441–444, 2008.

[95] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[96] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[97] Copyright 2019 Raspberry Pi (Trading) Ltd. All rights reserved. Link to the official datasheet:. *https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf*, 2019.

[98] Tensorflow. Link to the official guide:. *https://www.tensorflow.org/lite/guide*, Last updated 2022-06-28.

[99] M.Munoz-Organero. Deep physiological model for blood glucose prediction in t1dm patients. *Sensors*, 20(14):3896, 2020.

[100] Ahmed R Nasser, Ahmed M Hasan, Amjad J Humaidi, Ahmed Alkhayyat, Laith Alzubaidi, Mohammed A Fadhel, José Santamaría, and Ye Duan. Iot and cloud computing in health-care: A new wearable device and cloud-based deep learning algorithm for monitoring of diabetes. *Electronics*, 10(21):2719, 2021.

[101] Thomas Danne, T Battelino, P Jarosz-Chobot, O Kordonouri, E Pánkowska, Johnny Ludvigsson, E Schober, Eero Kaprio, Tero Saukkonen, M Nicolino, et al. Establishing glycaemic control with continuous subcutaneous insulin infusion in children and adolescents with type 1 diabetes: experience of the pedpump study in 17 countries. *Diabetologia*, 51(9):1594–1601, 2008.

[102] Chris Worth, Mark Dunne, Arunabha Ghosh, Simon Harper, and Indraneel Banerjee. Continuous glucose monitoring for hypoglycaemia in children: perspectives in 2020. *Pediatric Diabetes*, 21(5):697–706, 2020.

[103] Guruprasad M Bhat and Nayana G Bhat. A novel iot based framework for blood glucose examination. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pages 205–207. IEEE, 2017.

[104] Kathleen M Dungan, Colleen Sagrilla, Mahmoud Abdel-Rasoul, and Kwame Osei. Prandial insulin dosing using the carbohydrate counting technique in hospitalized patients with type 2 diabetes. *Diabetes Care*, 36(11):3476–3482, 2013.

[105] Touria El Idrissi and Ali Idri. Deep learning for blood glucose prediction: Cnn vs lstm. In *International Conference on Computational Science and Its Applications*, pages 379–393. Springer, 2020.

[106] TensorFlow. Post-training quantization. *https://www.tensorflow.org/lite/performance/post_training_quantization*, 2022.

[107] Stavroula G Mougiakakou, Aikaterini Prountzou, Dimitra Iliopoulou, Konstantina S Nikita, Andriani Vazeou, and Christos S Bartsocas. Neural network based glucose-insulin metabolism models for children with type 1 diabetes. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3545–3548. IEEE, 2006.

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

[109] Google. Frequently asked questions. *available online at: https://coral.ai/docs/edgetpu/faq/#how-is-the-edge-tpu-different-from-cloud-tpus*, 2022.

[110] Huimin Peng. A comprehensive overview and survey of recent advances in meta-learning. *arXiv preprint arXiv:2004.11149*, 2020.

[111] Federico D'Antoni, Lorenzo Petrosino, Alessandro Marchetti, Luca Bacco, Silvia Pieralice, Luca Vollero, Paolo Pozzilli, Vincenzo Piemonte, and Mario Merone. Layered meta-learning algorithm for predicting adverse events in type 1 diabetes. *IEEE Access*, 2023.

[112] William PTM van Doorn, Yuri D Foreman, Nicolaas C Schaper, Hans HCM Savelberg, Annemarie Koster, Carla JH van der Kallen, Anke Wesselius, Miranda T Schram, Ronald MA Henry, Pieter C Dagnelie, et al. Machine learning-based glucose prediction with use of continuous glucose and physical activity monitoring data: The Maastricht Study. *Plos one*, 16(6):e0253125, 2021.

[113] Jianping Li, Jun Hao, QianQian Feng, Xiaolei Sun, and Mingxi Liu. Optimal selection of heterogeneous ensemble strategies of time series forecasting with multi-objective programming. *Expert Systems with Applications*, page 114091, 2020.

[114] Madiha Bukhsh, Muhammad Saqib Ali, Muhammad Usman Ashraf, Khalid Alsubhi, and Weiqiu Chen. An interpretation of long short-term memory recurrent neural network for approximating roots of polynomials. *IEEE Access*, 10:28194–28205, 2022.

[115] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[116] Irfan Sudahri Damanik, Agus Perdana Windarto, Anjar Wanto, Sundari Retno Andani, Widodo Saputra, et al. Decision tree optimization in c4.5 algorithm using genetic algorithm. In *Journal of Physics: Conference Series*, volume 1255, page 012012. IOP Publishing, 2019.

[117] Arthur Bertachi, Clara Viñals, Lyvia Biagi, Ivan Contreras, Josep Vehí, Ignacio Conget, and Marga Giménez. Prediction of nocturnal hypoglycemia in adults with type 1 diabetes under multiple daily injections using continuous glucose monitoring and physical activity monitor. *Sensors*, 20(6):1705, 2020.

[118] Yun-Chun Wang and Ching-Hsue Cheng. A multiple combined method for rebalancing medical data with class imbalances. *Computers in Biology and Medicine*, page 104527, 2021.

[119] Nonso Nnamoko and Ioannis Korkontzelos. Efficient treatment of outliers and class imbalance for diabetes prediction. *Artificial Intelligence in Medicine*, 104:101815, 2020.

[120] Shaker El-Sappagh, Farman Ali, Samir El-Masri, Kyehyun Kim, Amjad Ali, and Kyung-Sup Kwak. Mobile health technologies for diabetes mellitus: Current state and future challenges. *IEEE Access*, 7:21917–21947, 2019.

[121] Zeinab Mahmoudi, Morten Hasselstrøm Jensen, Mette Dencker Johansen, Toke Folke Christensen, Lise Tarnow, Jens Sandahl Christiansen, and Ole Hejlesen. Accuracy evaluation of a new real-time continuous glucose monitoring algorithm in hypoglycemia. *Diabetes technology & therapeutics*, 16(10):667–678, 2014.

[122] Amparo Güemes, Giacomo Cappon, Bernard Hernandez, Monika Reddy, Nick Oliver, Pantelis Georgiou, and Pau Herrero. Predicting quality of overnight glycaemic control in type 1 diabetes using binary classifiers. *IEEE Journal of Biomedical and Health Informatics*, 24(5):1439–1446, 2019.

[123] Mert Sevil, Mudassir Rashid, Iman Hajizadeh, Minsun Park, Laurie Quinn, and Ali Cinar. Physical activity and psychological stress detection and assessment of their effects on glucose concentration predictions in diabetes management. *IEEE Transactions on Biomedical Engineering*, 68(7):2251–2260, 2021.

[124] Miguel Tejedor, Ashenafi Zebene Woldaregay, and Fred Godtliebsen. Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial Intelligence in Medicine*, 104:101836, 2020.

[125] Anas El Fathi, Emilie Palisaitis, Benoit Boulet, Laurent Legault, and Ahmad Haidar. An unannounced meal detection module for artificial pancreas control systems. In *2019 American Control Conference (ACC)*, pages 4130–4135. IEEE, 2019.

[126] Silvia Del Giorno, Federico D'Antoni, Vincenzo Piemonte, and Mario Merone. A new glycemic closed-loop control based on dyna-q for type-1-diabetes. *Biomedical Signal Processing and Control*, 81:104492, 2023.

[127] Chiara Zecchin, Andrea Facchinetti, Giovanni Sparacino, Giuseppe De Nicolao, and Claudio Cobelli. Neural network incorporating meal information improves accuracy of short-time prediction of glucose concentration. *IEEE Transactions on Biomedical Engineering*, 59(6):1550–1560, 2012.

[128] Simon Lebech Cichosz, Thomas Kronborg, Morten Hasselstrøm Jensen, and Ole Hejlesen. Penalty weighted glucose prediction models could lead to better clinically usage. *Computers in Biology and Medicine*, 138:104865, 2021.

[129] Qingnan Sun, Marko V Jankovic, Lia Bally, and Stavroula G Mougiakakou. Predicting blood glucose with an lstm and bi-lstm based deep neural network. In *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, pages 1–5. IEEE, 2018.

[130] Eleonora Maria Aiello, Giuseppe Lisanti, Lalo Magni, Mirto Musci, and Chiara Toffanin. Therapy-driven deep glucose forecasting. *Engineering Applications of Artificial Intelligence*, 87:103255, 2020.

[131] Simon Lebech Cichosz, Morten Hasselstrøm Jensen, and Ole Hejlesen. Short-term prediction of future continuous glucose monitoring readings in type 1 diabetes: development and validation of a neural network regression model. *International Journal of Medical Informatics*, 151:104472, 2021.

[132] Kaitlyn Rechenberg, Robin Whittemore, and Margaret Grey. Anxiety in youth with type 1 diabetes. *Journal of pediatric nursing*, 32:64–71, 2017.

[133] Cristina Bianchi, Teresa Mezza, Fabrizio Febo, Basilio Pintaudi, and Giuliano Caggiano. *AIDA chatbot*, 2020. https://www.aidachatbot.it/.

[134] Alezandra Torres-Castaño, Amado Rivero-Santana, Lilisbeth Perestelo-Pérez, Andrea Duarte-Díaz, Analia Abt-Sacks, Vanesa Ramos-García, Yolanda Álvarez-Pérez, Ana M Wäagner, Mercedes Rigla, and Pedro Serrano-Aguilar. Dual-hormone insulin-and-pramlintide artificial pancreas for type 1 diabetes: A systematic review. *Applied Sciences*, 12(20):10262, 2022.

[135] Luca Bacco, Andrea Cimino, Luca Paulon, Mario Merone, and Felice Dell'Orletta. A machine learning approach for sentiment analysis for italian reviews in healthcare. In *CLiC-it*, 2020.

A. Contributions in Computer Science and Bioengineering

Artificial Intelligence in Low Back Pain

Reference D'Antoni, F., Russo, F., Ambrosio, L., Vollero, L., Vadalà, G., Merone, M., Papalia, R., & Denaro, V. (2021). Artificial Intelligence and Computer Vision in Low Back Pain: A Systematic Review. International journal of environmental research and public health, 18(20), 10909. https://doi.org/10.3390/ijerph182010909

Abstract Chronic Low Back Pain (LBP) is a symptom that may be caused by several diseases, and it is currently the leading cause of disability worldwide. The increased amount of digital images in orthopaedics has led to the development of methods related to artificial intelligence, and to computer vision in particular, which aim to improve diagnosis and treatment of LBP. In this manuscript, we have systematically reviewed the available literature on the use of computer vision in the diagnosis and treatment of LBP. A systematic research of PubMed electronic database was performed. The search strategy was set as the combinations of the following keywords: "Artificial Intelligence", "Feature Extraction", "Segmentation", "Computer Vision", "Machine Learning", "Deep Learning", "Neural Network", "Low Back Pain", "Lumbar". Results: The search returned a total of 558 articles. After careful evaluation of the abstracts, 358 were excluded, whereas 124 papers were excluded after full-text examination, taking the number of eligible articles to 76. The main applications of computer vision in LBP include feature extraction and segmentation, which are usually followed by further tasks. Most recent methods use deep learning models rather than digital image processing techniques. The best performing methods for segmentation of vertebrae, intervertebral discs, spinal canal and lumbar muscles achieve Sørensen-Dice scores greater than 90%, whereas studies focusing on localization and identification of structures collectively showed an accuracy greater than 80%. Future advances in artificial intelligence are expected to increase sys-

tems' autonomy and reliability, thus providing even more effective tools for the diagnosis and treatment of LBP.

Reference D'Antoni, F., Russo, F., Ambrosio, L., Bacco, L., Vollero, L., Vadalà, G., Merone, M., Papalia, R., & Denaro, V. (2022). Artificial Intelligence and Computer Aided Diagnosis in Chronic Low Back Pain: A Systematic Review. International Journal of Environmental Research and Public Health, 19(10), 5971. https://doi.org/10.3390/ijerph19105971

Abstract Low Back Pain (LBP) is currently the first cause of disability in the world, with a significant socioeconomic burden. Diagnosis and treatment of LBP often involve a multidisciplinary, individualized approach consisting of several outcome measures and imaging data along with emerging technologies. The increased amount of data generated in this process has led to the development of methods related to artificial intelligence (AI), and to computer-aided diagnosis (CAD) in particular, which aim to assist and improve the diagnosis and treatment of LBP. In this manuscript, we have systematically reviewed the available literature on the use of CAD in the diagnosis and treatment of chronic LBP. A systematic research of PubMed, Scopus, and Web of Science electronic databases was performed. The search strategy was set as the combinations of the following keywords: "Artificial Intelligence", "Machine Learning", "Deep Learning", "Neural Network", "Computer Aided Diagnosis", "Low Back Pain", "Lumbar", "Intervertebral Disc Degeneration", "Spine Surgery" etc. The search returned a total of 1536 articles. After duplication removal and evaluation of the abstracts, 1386 were excluded, whereas 93 papers were excluded after full-text examination, taking the number of eligible articles to 57. The main applications of CAD in LBP included classification and regression. Classification is used to identify or categorize a disease, whereas regression is used to produce a numerical output as a quantitative evaluation of some measure. The best performing systems were developed to diagnose degenerative changes of the spine from imaging data, with average accuracy rates $> 80\%$. However, notable outcomes were also reported for CAD tools executing different tasks including analysis of clinical, biomechanical, electrophysiological, and functional imaging data. Further studies are needed to better define the role of CAD in LBP care.

Reference Bacco, L., Russo, F., Ambrosio, L., D'Antoni, F., Vollero, L., Vadalà, G., Dell'Orletta, F., Merone, M., Papalia, R., & Denaro, V. (2022). Natural language

processing in low back pain and spine diseases: A systematic review. Frontiers in surgery, 997. https://doi.org/10.3389/fsurg.2022.957085

Abstract Natural Language Processing (NLP) is a discipline at the intersection between Computer Science (CS), Artificial Intelligence (AI), and Linguistics that leverages unstructured human-interpretable (natural) language text. In recent years, it gained momentum also in health-related applications and research. Although preliminary, studies concerning Low Back Pain (LBP) and other related spine disorders with relevant applications of NLP methodologies have been reported in the literature over the last few years. It motivated us to systematically review the literature comprised of two major public databases, PubMed and Scopus. To do so, we first formulated our research question following the PICO guidelines. Then, we followed a PRISMA-like protocol by performing a search query including terminologies of both technical (e.g., natural language and computational linguistics) and clinical (e.g., lumbar and spine surgery) domains. We collected 221 non-duplicated studies, 16 of which were eligible for our analysis. In this work, we present these studies divided into sub-categories, from both tasks and exploited models' points of view. Furthermore, we report a detailed description of techniques used to extract and process textual features and the several evaluation metrics used to assess the performance of the NLP models. However, what is clear from our analysis is that additional studies on larger datasets are needed to better define the role of NLP in the care of patients with spinal disorders.

Decision Support Systems

Reference Biggio, M., Caligiore, D., D'Antoni, F. et al. Machine learning for exploring neurophysiological functionality in multiple sclerosis based on trigeminal and hand blink reflexes. Scientific Reports 12, 21078 (2022). https://doi.org/10.1038/s41598-022-24720-6

Abstract Brainstem dysfunctions are very common in Multiple Sclerosis (MS) and are a critical predictive factor for future disability. Brainstem functionality can be explored with blink reflexes, subcortical responses consisting in a blink following a peripheral stimulation. Some reflexes are already employed in clinical practice, such as Trigeminal Blink Reflex (TBR). Here we propose for the first time in MS the exploration of Hand Blink Reflex (HBR), which size is modulated by the proximity of the stimulated hand to the face, reflecting the extension of the peripersonal space. his work

is to test whether Machine Learning (ML) techniques could be used in combination with neurophysiological measurements such as TBR and HBR to improve their clinical information and potentially favour the early detection of brainstem dysfunctionality. HBR and TBR were recorded from a group of People with MS (PwMS) with Relapsing-Remitting form and from a healthy control group. Two AdaBoost classifiers were trained with TBR and HBR features each, for a binary classification task between PwMS and Controls. Both classifiers were able to identify PwMS with an accuracy comparable and even higher than clinicians. Our results indicate that ML techniques could represent a tool for clinicians for investigating brainstem functionality in MS. Also, HBR could be promising when applied in clinical practice, providing additional information about the integrity of brainstem circuits potentially favouring early diagnosis.

Reference Conte, F., D'Antoni, F., Natrella, G., & Merone, M. (2022). A new hybrid AI optimal management method for renewable energy communities. Energy and AI, 10, 100197. https://doi.org/10.1016/j.egyai.2022.100197

Abstract In this study, we propose a hybrid AI optimal method to improve the efficiency of energy management in a smart grid such as Renewable Energy Community. This method adopts a Time Delay Neural Network to forecast the future values of the energy features in the community. Then, these forecasts are used by a stochastic Model Predictive Control to optimize the community operations with a proper control strategy of Battery Energy Storage System. The results of the predictions performed on a public dataset with a prediction horizon of 24 hours return a Mean Absolute Error of 1.60 kW, 2.15 kW, and 0.30 kW for photovoltaic generation, total energy consumption, and common services, respectively. The model predictive control fed with such predictions generates maximum income compared to the competitors. The total income is increased by 18.72% compared to utilizing the same management system without exploiting predictions from a forecasting method.

Reference D'Amico, N. C., Merone, M., Sicilia, R., Cordelli, E., D'Antoni, F., Zanetti, I. B., ... & Soda, P. (2019). Tackling imbalance radiomics in acoustic neuroma. International Journal of Data Mining and Bioinformatics, 22(4), 365-388. https://doi.org/10.1504/IJDMB.2019.101396

Abstract Acoustic neuroma is a primary intracranial tumor of the myelin-forming cells of the 8^{th} cranial nerve. Although it is a slow growing benign tumor, symptoms

in the advanced phase can be serious. Hence, controlling tumor growth is essential and stereotactic radiosurgery, which can be performed with the CyberKnife robotic device, has proven effective for managing this disease. However, this approach may have side effects and a follow-up is necessary to assess its efficacy. To optimize the administration of this treatment, in this work we present a machine learning-based radiomics approach that first computes quantitative biomarkers from MR images routinely collected before the CyberKnife treatment and then predicts the treatment response. To tackle the challenge of class imbalance observed in the available dataset we present a cascade of cost-sentitive decision trees. We also experimentally compare the proposed approach with several approaches suited for learning under class skew. The results achieved demonstrate that radiomics has a great potential in predicting patients response to radiosurgery prior to the treatment that, in turns, can reflect into great advantages in therapy planning, sparing radiation toxicity and surgery when unnecessary.

Reference Sicilia, R., Merone, M., Valenti, R., Cordelli, E., D'Antoni, F., De Ruvo, V., Dragone, P. B., Esposito, S., & Soda, P. (2018, December). Cross-topic rumour detection in the health domain. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 2056-2063). IEEE. https://doi.org/10.1109/BIBM.2018.8621580

Abstract Nowadays information diffusion has become more and more immediate and fast thanks to social media and its services. However, lack of controls and moderation in resources as social microblogs often leads to spread unverified information, such as rumours, which can become a threat to the society. To improve life quality and good information diffusion, various automatic systems have been studied for rumour detection in microblogs at level of aggregation of posts, whereas a few effort has been tried to the most challenging scenario where the rumour has to be recognized at level of each single post. In this work, we direct our efforts towards individual post rumour detection: we investigate how features describing influence potential, personal interest and network characteristic perform on two different datasets of posts collected from Twitter using two different health-related keywords. As a further contribution, we study what happens in cross-topic tests, i.e. when the rumour detection system is trained with posts with an hashtag and tested on samples with a different one.

Reference Infantino, M., Merone, M., Manfredi, M., Grossi, V., Landini, A., Alessio, M. G., ... & Bizzaro, N. (2021). Positive tissue transglutaminase antibodies with nega-

tive endomysial antibodies: Unresolved issues in diagnosing celiac disease. Journal of Immunological Methods, 489, 112910. https://doi.org/10.1016/j.jim.2020.112910

Abstract Background: The serological screening for celiac disease (CD) is currently based on the detection of anti-transglutaminase (tTG) IgA antibodies, subsequently confirmed by positive endomysial antibodies (EMA). When an anti-tTG IgA positive/EMA IgA negative result occurs, it can be due either to the lower sensitivity of the EMA test or to the lower specificity of the anti-tTG test. This study aimed at verifying how variation in analytical specificity among different anti-tTG methods could account for this discrepancy. Methods: A total of 130 consecutive anti-tTG IgA positive/EMA negative samples were collected from the local screening routine and tested using five anti-tTG IgA commercial assays: two chemiluminescence methods, one fluoroimmunoenzymatic method, one immunoenzymatic method and one multiplex flow immunoassay method. Results: Twenty three/130 (17.7%) patients were diagnosed with CD. In the other 107 cases a diagnosis of CD was not confirmed. The overall agreement among the five anti-tTG methods ranged from 28.5% to 77.7%. CD condition was more likely linked to the positivity of more than one anti-tTG IgA assay (monopositive = 2.5%, positive with $\geq$ three methods = 29.5%; p = 0.0004), but it was not related to anti-tTG IgA antibody levels (either positive or borderline; p = 0.5). Conclusions: Patients with positive anti-tTG/negative EMA have a low probability of being affected by CD. Given the high variability among methods to measure anti-tTG IgA antibodies, anti-tTG-positive/EMA-negative result must be considered with extreme caution. It is advisable that the laboratory report comments on any discordant results, suggesting to consider the data in the proper clinical context and to refer the patient to a CD reference center for prolonged follow up.