ID N. 21

UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

DEPARTMENT OF ENGINEERING

UNIVERSITÀ DI CHIETI-PESCARA

Italian National Ph.D. in Artificial Intelligence
Health and Life Sciences
XXXVII Cycle

Deep Reinforcement Learning and Edge Computing for Type-1 Diabetes Management

Supervisors
Prof. Maurizio Parton
Prof. Francesco Morandin
Ing. Dott. Mario Merone

Candidate
Alessandro Marchetti

January, 2025

To my father Alfiero and my mother Elisa,
who taught me to walk through life and enjoy its pleasures.

Abstract

Type 1 Diabetes (T1D) is an autoimmune disorder characterized by the destruction of pancreatic β-cells, leading to an absolute deficiency of insulin production and a consequent inability to regulate blood glucose levels. Globally, T1D affects a smaller proportion of individuals compared to Type 2 Diabetes (T2D), yet both types significantly contribute to the global health burden. T1D requires lifelong insulin therapy, while T2D which accounts for over 90% of diabetes cases is primarily driven by lifestyle factors such as obesity and physical inactivity. Together, diabetes contributes to a rise in associated complications, including cardiovascular disease and kidney failure, with the International Diabetes Federation estimating that approximately 1 in 10 adults worldwide live with some form of diabetes, highlighting the urgent need for effective management and prevention strategies.

To mitigate acute complications such as hypoglycemia and hyperglycemia, as well as prevent long-term organ damage, modern therapeutic approaches often include continuous glucose monitoring (CGM) sensors and insulin pumps to provide real-time insights and precision in insulin delivery. The rapid proliferation of machine learning (ML) and artificial intelligence (AI) techniques, particularly deep learning and deep reinforcement learning, has opened new avenues for personalized and proactive diabetes management. By leveraging these techniques, researchers aim to develop data-driven models capable of predicting short-term and long-term fluctuations in blood glucose levels, thereby enabling early intervention and tailored insulin dosing. This capability is crucial because accurate blood glucose level (BGL) forecasting not only allows for improved glycemic control but also reduces the risk of severe complications through preemptive actions, thus transforming the clinical paradigm from a reactive to a preventive approach.

Nonetheless, deploying AI solutions in T1D care is not without challenges: the need for high predictive accuracy, real-time computational efficiency, and robust data privacy protections underlines the complexity of constructing scalable and trustworthy systems. Consequently, integrating AI-driven forecasting and control mechanisms into the existing framework of CGM sensors and insulin pumps holds significant promise in improving patient outcomes, but it necessitates concerted efforts in algorithm development, hardware

optimization, and ethical data governance.

This thesis integrates advanced predictive modeling and control frameworks to address the challenges of managing Type 1 Diabetes, combining state-of-the-art algorithms with personalized solutions. The contributions begin with a layered meta-learning approach that introduces a multi-expert architecture for predicting adverse glycemic events. By leveraging continuous glucose monitoring data and specializing in hypoglycemia, hyperglycemia, and normoglycemia detection, this method not only anticipates critical events with high precision but also enhances generalization through a meta-learner trained on limited patient-specific data. Complementing this, a Federated Online Extreme Learning Machine framework demonstrates the efficacy of decentralized learning in blood glucose level forecasting. This system achieves computational efficiency while maintaining robust data privacy, offering a scalable and secure solution for personalized diabetes management across distributed devices.

Then this thesis advances the control of blood glucose levels through a novel dual Deep Reinforcement Learning (DRL) framework. This approach introduces a hybrid closed-loop control system for optimizing insulin delivery in real-time, achieving superior glycemic stability with minimal patient intervention. A safe-control mechanism and adaptive insulin caps ensure the mitigation of hypo- and hyperglycemic risks. Moreover, as a natural extension, the thesis investigates the application of Multi-Agent Reinforcement Learning (MARL), emphasizing inter-agent communication and collaboration to optimize decision-making in complex scenarios. This framework enables agents to dynamically share information and coordinate strategies, effectively capturing cooperative dynamics while enhancing the system's capability to deliver highly personalized and adaptive medical interventions tailored to the unique physiological and pathological profiles of individual patients.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Medical Background

Type 1 diabetes mellitus (T1D) is a chronic autoimmune disorder marked by the destruction of insulin-producing β-cells in the pancreas [8]. This destruction typically leads to a near-total deficiency in insulin production, although some residual production may persist in certain cases, particularly in the early stages of the disease.

In the physiological state, insulin and glucagon work as counterregulatory hormones to maintain blood glucose homeostasis. Insulin, secreted by pancreatic β-cells, facilitates glucose uptake by tissues and inhibits hepatic glucose production, thereby lowering blood glucose levels. Conversely, glucagon, produced by pancreatic α-cells, stimulates glycogenolysis and gluconeogenesis in the liver, raising blood glucose levels when they fall too low [114].

In T1D patients, the absence of endogenous insulin necessitates exogenous insulin administration, which must be carefully calibrated to match varying metabolic demands throughout the day. Unlike Type 2 diabetes mellitus (T2D), which is primarily associated with insulin resistance and a relative insulin insufficiency [109], T1D presents unique challenges due to the complete dependency on external insulin. Patients with T1D experience significant glycemic variability, characterized by frequent episodes of hyperglycemia (blood glucose > 180 mg/dL) and hypoglycemia (blood glucose < 70 mg/dL) [60]. This variability significantly complicates disease management and increases the risk of both acute and chronic complications.

Acute complications include diabetic ketoacidosis (DKA) and severe hypoglycemia, both of which can be life-threatening. DKA arises from prolonged and severe insulin deficiency, leading to excessive lipolysis and ketogenesis, which result in metabolic acidosis, dehydration, and electrolyte imbalances with symptoms such as nausea, vomiting, abdominal pain,

rapid breathing, and altered mental status, requiring immediate medical intervention [37]. Severe hypoglycemia is characterized by critically low blood glucose levels, which impair normal brain function due to inadequate glucose availability. Symptoms range from confusion, weakness, and palpitations to seizures, loss of consciousness, or even death if untreated [24]. Hypoglycemic episodes are particularly concerning in individuals with impaired hypoglycemia awareness, increasing the risk of recurrent crises.

The paradoxical occurrence of hypoglycemia in insulin-deficient patients primarily results from iatrogenic causes—specifically, the administration of exogenous insulin without precise correlation to physiological needs [21]. Factors such as excessive insulin dosing, delayed or missed meals, unplanned physical activity, alcohol consumption, and impaired counter-regulatory hormone responses all contribute to hypoglycemic episodes.

Chronic hyperglycemia, on the other hand, even at subacute levels, poses a long-term risk for vascular damage, which underpins the development of microvascular complications such as diabetic retinopathy, nephropathy, and neuropathy. It also accelerates the progression of macrovascular conditions, including coronary artery disease and cerebrovascular events.

The global burden of diabetes, encompassing both T1D and T2D, continues to rise. According to the World Health Organization (WHO), 14% of adults aged 18 years and older were living with diabetes in 2022, a significant increase from 7% in 1990. Alarmingly, more than half (59%) of adults aged 30 years and over with diabetes were not receiving any medication for their condition, with treatment coverage being lowest in low- and middle-income countries [92]. In 2021, diabetes was directly responsible for approximately 1.6 million deaths, with 47% of these deaths occurring before the age of 70 years. Additionally, diabetes contributed to 530,000 deaths due to kidney disease, and elevated blood glucose levels were implicated in around 11% of cardiovascular deaths [92]. These figures underscore the urgent need for improved management strategies and equitable access to care globally.

These outcomes highlight the critical importance of stringent blood glucose management in individuals with T1D [121].

1.2 Technical Background

Effective glycemic control is paramount for mitigating the risks of both acute and long-term complications. The goal of therapy is to maintain blood glucose levels within the euglycemic range (70–180 mg/dL), as recommended by international clinical guidelines [24]. Achieving this target is challenging due to the multifaceted nature of glycemic regulation, which requires the delicate balancing of insulin dosing, dietary intake, physical activity, and psychological stress [84].

The advent of advanced technologies, including Continuous Glucose Monitoring (CGM) devices and automated insulin delivery systems, has significantly improved the quality of life and clinical outcomes for individuals with T1D. CGM systems have revolutionized diabetes management by providing real-time glucose level tracking [35]. A typical CGM device consists of a subcutaneous sensor that measures glucose in interstitial fluid, a transmitter, and a display device or smartphone application. These systems offer significant advantages over traditional fingerstick blood glucose monitoring, which provides only intermittent data [14].

Modern CGM systems feature wireless connectivity, low lag times (typically 8–10 minutes), and advanced analytical capabilities, enabling users to better understand glycemic trends [2]. In particular, next-generation sensors with a Mean Absolute Relative Difference (MARD) below 10%, allowing for highly accurate readings [50].

The integration of CGM with insulin pumps, often referred to as hybrid closed-loop or artificial pancreas systems, has further enhanced glycemic control by automating insulin delivery [19]. Furthermore, developments in artificial pancreas systems aim to fully automate glucose management through advanced control algorithms and adaptive insulin delivery [113].

However, even with these advancements, challenges remain, such as improving sensor accuracy, reducing device costs, and addressing data privacy concerns, while many patients struggle to achieve optimal glycemic control.

Recent developments in artificial intelligence (AI) have shown immense potential to revolutionize diabetes management, particularly through personalized predictive algorithms, insulin dosing recommendations, and decision-support tools [132]. These advancements aim to alleviate the cognitive burden on patients and enhance their ability to achieve stable glycemic control.

1.3 State of the Art

Despite the widespread adoption of Continuous Glucose Monitoring (CGM) systems among individuals with Type 1 Diabetes (T1D), the occurrence of hypoglycemic and hyperglycemic events remains a significant concern [36, 77, 78]. This challenge persists despite the primary objective of T1D management, which is to minimize, if not entirely eliminate, such adverse events.

The development of data-driven models has leveraged the time-series nature of CGM data, utilizing it either as a standalone feature (univariate approach) or in combination with other factors such as insulin doses and carbohydrate intake (multivariate approach). The univariate approach focuses exclusively on the temporal sequence of CGM data, simplifying the model while achieving significant insights into glycemic trends. Conversely, the multivariate

approach integrates additional features to provide a more comprehensive representation of patient physiology, though it may introduce complexity and noise in data interpretation.

Regression Regression is the most widely employed method [94]. It involves forecasting the exact future glucose levels within a specified prediction horizon. Accurate prediction of glycemic levels for the next 15–30 minutes enables patients to take preventive measures if glucose levels are expected to exceed the target range. This method directly supports proactive glycemic management, providing actionable insights to patients and healthcare providers.

Classification Classification focuses on determining whether a patient is likely to experience an adverse glycemic event, such as hypoglycemia or hyperglycemia, within a given time frame. Instead of forecasting exact glucose values, this approach categorizes future states, offering a simplified yet effective method for early warnings and preventive interventions. Classification models are particularly beneficial for alerting patients to potential risks without requiring precise numerical forecasts.

Control Control strategies aim to automate and optimize insulin infusion to maintain glucose levels within the euglycemic range. These methods include the development of fully automated insulin delivery systems that adapt dynamically to patient needs. Control models often employ advanced algorithms, including Reinforcement Learning (RL), to create closed-loop systems that minimize patient intervention while maximizing glycemic stability.

Personalized vs. statistical approach The methodologies described above can be evaluated using different validation procedures:

Precision Medicine Precision medicine develops predictive models tailored to individual patient data. Data from each patient are split into training and test sets, with the training set used to fit the model and the test set employed to evaluate performance on unseen data from the same patient. This approach ensures that models are fine-tuned for individual variability, enhancing their predictive accuracy for personalized management.

k-Fold Cross-Validation This statistical technique divides the dataset, comprising data from multiple patients, into k subsets. Model performance is assessed iteratively by using one subset as the test set and the remaining $k - 1$ subsets for training and validation. A common variant, Leave-1-Patient-Out Cross-Validation, uses all data from a single subject as a fold, enabling robust evaluation across patient-specific datasets.

All predictive models must define a *Prediction Horizon* (PH), which specifies the temporal extent of the forecast. Most studies adopt a PH of 30 minutes, as this interval typically provides sufficient time for patients to take corrective actions to avert adverse events.

The following sections delve into the methodologies, strengths, and limitations of these approaches in greater detail.

1.4 Prediction Strategies for T1D Management

Regression Regression tasks in Type 1 Diabetes (T1D) management are focused on forecasting blood glucose levels to help patients maintain stable glycemic control. Time series forecasting techniques for this purpose span kernel machines, forests of trees, symbolic representation, generative models, and artificial neural networks. Each approach has been explored extensively, with many notable contributions documented over the past decades.

Evaluation Metrics Performance evaluation is critical in forecasting tasks. Metrics such as Root Mean Square Error (RMSE), Sum of Squared Geometric Percent Error (SSGPE), and Mean Absolute Relative Difference (MARD) are widely used. Additionally, the Clarke Error Grid Analysis (CEGA) [18] assesses clinical relevance by categorizing predictions into zones A through E.

Early Contributions Kernel machines and forests of trees were among the first methods applied to this problem. For instance, Bunescu et al.[11] employed a three-compartmental physiological model of glucose dynamics to extract features for a Support Vector Regressor (SVR) trained on patient-specific data. Using CGM, insulin, and meal data from 5 private T1D patient datasets, their model achieved RMSE values of $22.6; mg/dL$ and $35.8; mg/dL$ for 30- and 60-minute prediction horizons, respectively. Later, Hamdi et al.[49] optimized an SVR using differential evolution to predict glucose levels for 12 T1D patients, achieving an RMSE of $9.4 \pm 3.7; mg/dL$ for 15-minute and $10.8 \pm 3.9; mg/dL$ for 30-minute horizons.

Generative models emerged as a complementary strategy, leveraging the intra-subject variability of glucose dynamics. Reifman et al.[100] proposed a time-invariant auto-regressive (AR) model with regularized least squares for CGM data from 9 private datasets. Subject-specific models outperformed generalized models, achieving an RMSE of $22.3 \pm 3.9; mg/dL$. Sparacino et al.[107] extended this work by introducing a first-order AR model with time-varying parameters, demonstrating RMSE values of $18.3 \pm 11.8; mg/dL$ for 30-minute horizons across 28 private datasets.

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 [11]	SVR	KM	MTS	5/private
Hamdi [49]	SVR + DE	KM	UTS	12/ private
Georga [46]	Random Forest Regression	FT	MTS	27/private
Midroni [83]	XGBoost Random Forest	FT	MTS	6/Ohio T1DM [77]
Contreras [20]	Search-based algorithm	SR	MTS	6/Ohio T1DM [77]
Reifman [100]	Auto-regressive model	GM	UTS	9/private
Sparacino [107]	Auto-regressive model	GM	UTS	28/private
Zecchin [126]	Jump NN	NN	MTS	20/private
Martinsson [79]	Long-Short Term Memory	NN	UTS	6/Ohio T1DM [77]
Zhu [133]	CNN	NN	MTS	6/Ohio T1DM [77]
Chen [15]	Dilated Recurrent NN	NN	MTS	6/Ohio T1DM [77]
Bertachi [6]	Feed-Forward NN	NN	MTS	6/Ohio T1DM [77]
Li [66]	Dilated Convolutional NN	NN	MTS	16/ private, Ohio T1DM [77]
Kalita [62]	LSTM-GRU	Multi-NN	MTS	UVA/Padova simulator [22]
Jaloli [55]	CNN-LSTM	Multi-NN	MTS	Replace-BG [1], DIAdvisor
Lu [69]	Stacked MLP, Bi-GRU, RNN +AM	Multi-NN	UTS	RT_CGM [110]

Symbolic Representation and Forests of Trees Symbolic representation approaches, such as Symbolic Aggregate approXimation (SAX) and Bag-of-Words (BoW), were later utilized for regression tasks. Contreras et al.[20] introduced a grammar-based feature generation system tested on the Ohio T1DM dataset[77]. Their model achieved RMSE values of 21.2, 31.3, and 36.3; mg/dL for 30-, 60-, and 90-minute prediction horizons, respectively.

Forest-based methods also advanced significantly during this period. Georga et al.[46] used Random Forest Regression with multivariate inputs including CGM data, physiological features, and lifestyle information, achieving an RMSE of $6.6 \pm 1.3 : mg/dL$ for 15-minute horizons. Midroni et al.[83] applied XGBoost with feature engineering on the Ohio T1DM dataset, reporting an RMSE of $20.4 \pm 2.4 : mg/dL$ for 30-minute horizons.

Artificial Neural Networks The introduction of artificial neural networks (ANNs) marked a paradigm shift in T1D forecasting. Zecchin et al.[126] proposed a jump neural network that combined linear and nonlinear dependencies to predict 30-minute glucose levels with an RMSE of $16.6 \pm 3.1 : mg/dL$. Martinsson et al.[79] demonstrated the effectiveness of Long Short-Term Memory (LSTM) networks in univariate time series (UTS) forecasting, achieving an RMSE of $20.1 \pm 2.5 : mg/dL$ on the Ohio T1DM dataset.

Further refinements included convolutional neural networks (CNNs) and hybrid architec-

tures. Zhu et al.[133] developed a CNN model incorporating CGM, carbohydrate (CHO) intake, and insulin events, achieving an RMSE of $22.1 \pm 2.5 : mg/dL$. Chen et al.[15] utilized dilated recurrent neural networks to capture long-term dependencies, resulting in an RMSE of $19.0 \pm 2.6 : mg/dL$. Bertachi et al. [6] implemented a feed-forward ANN that incorporated physical activity, CHO, and insulin data, producing RMSE values of $19.3 : mg/dL$ and $31.7 : mg/dL$ for 30- and 60-minute horizons, respectively.

Hybrid Models and Precision Medicine Recently, hybrid models have combined multiple neural network types for deeper feature extraction. Kalita et al.[62] fused LSTM with Gated Recurrent Units (GRU), achieving remarkable RMSE values of $5.27; mg/dL$ and $14.85; mg/dL$ for 15- and 30-minute horizons. Jaloli et al.[55] developed a CNN-LSTM hybrid model validated on Replace-BG [1] and DIAdvisor datasets, achieving robust predictions for up to 90-minute horizons. Lastly, Lu et al.[69] proposed a model combining MLP, bidirectional GRU (Bi-GRU), and attention mechanisms to achieve an RMSE of $11.76 : mg/dL$ on real-world CGM data[110].

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.

Classification An alternative to regression in blood glucose forecasting is classification, which predicts whether a patient will experience an adverse glycemic event (e.g., hypoglycemia or hyperglycemia) within a given prediction horizon (PH), regardless of the exact blood glucose level. This approach has gained traction due to its improved performance in predicting adverse events compared to regression-based methods [45, 123].

Evaluation Metrics As with other classification tasks, performance is evaluated using metrics derived from the confusion matrix: Recall and Precision. These are defined as:

$$Recall = TP/(TP + FN) \qquad Precision = TP/(TP + FP) \tag{1.1}$$

where TP, FP and FN are the total numbers of true positives, false positives, and false negatives per class.

The focus in glycemic event classification is typically on two primary classes—hypoglycemia and hyperglycemia—defined by CGM thresholds [104, 30, 16, 88, 58].

Hypoglycemia Prediction The majority of classification studies focus on hypoglycemia prediction due to its sudden onset and potential for severe short-term complications. Mujahid et al.[88] highlighted that "hypoglycemia prediction is blood glucose level prediction

in essence." However, many studies aim to maximize true positive rates (Recall) at the expense of Precision. This trade-off often results in a high number of false positives, which can discourage patient engagement with technology[104, 30, 13, 123].

For example, Felizardo et al.[41] prioritized Recall to detect more events but acknowledged the challenges of achieving balanced performance. Conversely, Marcus et al.[76] focused on improving Precision, demonstrating that an approach limiting false positives can enhance user acceptance.

Classification Strategies: Sample-Based vs. Event-Based Two main strategies are employed in glycemic event classification:

Sample-Based Prediction: Each timestamp is treated as an independent sample, and predictions are made for all timestamps within the dataset. Performance is evaluated based on these individual predictions [104, 30, 16, 76].

Event-Based Prediction: Consecutive timestamps indicating hypo- or hyperglycemia are grouped as a single event. A prediction is considered correct (true positive) if it aligns with an actual event occurring in the following time window [13, 45, 97].

Early Works on Classification Early studies primarily focused on sample-based approaches. Daskalaki et al.[29] applied a hybrid model combining cARX and recurrent neural networks (RNNs) to event-based classification. Despite achieving perfect Recall for hypoglycemia and hyperglycemia events, Precision metrics were not reported. Similarly, Cichosz et al.[16] used linear logistic regression for hypoglycemia prediction but omitted Precision scores.

Advancements in Sample-Based Models Seo et al.[104] employed Random Forests for predicting hypoglycemia, achieving a Recall of 89.6% but with low Precision (38.9%). Dave et al.[30] improved on this by incorporating separate daytime and nighttime Random Forest classifiers, achieving high Recall (93.7%) but significantly lower Precision (15.1%) during hypoglycemia detection.

Marcus et al. [76] explored kernel ridge regression and achieved more balanced performance by focusing on improving Precision, demonstrating its impact on user engagement.

Event-Based Innovations Event-based models gained traction due to their ability to reduce false positives by aggregating timestamps. Gadaleta et al.[45] applied an SVM with Leave-One-Patient-Out validation, achieving high Recall for hyperglycemia (95%) but low Precision (56%). Prendin et al.[97] introduced an ARIMA model, emphasizing the benefits

of event-based aggregation but still facing challenges in Precision for hypoglycemia detection (64%).

Recent works, such as Yang et al. [123], leveraged advanced architectures like LSTMs to achieve significant improvements in Recall (92.6%) for hypoglycemia prediction, albeit without detailed Precision scores.

Table 3.2 summarizes the state of the art in glycemic event classification. It highlights the variety of models, validation strategies, and their respective performance metrics for hypoglycemia (Hypo), normoglycemia (Norm), and hyperglycemia (Hyper).

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 [45]	89	Leave-1-Patient-Out	SVM	Event-Based	R	86	n/a	95
					P	36	n/a	56
Daskalaki [29]	23	Precision Medicine	cARX + RNN	Event-Based	R	100	n/a	100
					P	n/a	n/a	n/a
Cappon [13]	100 in silico	Precision Medicine	XGBoosted Tree	Sample-Based	R	92	76	86
					P	n/a	n/a	n/a
Seo [104]	104	5-fold Cross Validation	Random Forest	Sample-Based	R	89.6	91.3	n/a
					P	38.9	n/a	n/a
Dave [30]	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 [76]	11	Precision Medicine	Kernel Ridge Regression	Sample-Based	R	64	96	61
					P	n/a	n/a	n/a
Cichosz [16]	10	Precision Medicine	Linear Logistic Regression	Sample-Based	R	79	99	n/a
					P	n/a	n/a	n/a
Yang [123]	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 [97]	112	Precision Medicine	Autoregressive Integrated Moving Average (ARIMA)	Event-Based	R	82	n/a	n/a
					P	64	n/a	n/a
Jensen [58]	463	5-fold Cross Validation	Linear Discriminant Analysis (LDA) classifier	Sample-Based	R	73	75	n/a
					P	22	97	n/a
Zhu [135]	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 [25]	33	Precision Medicine	ARTiDe Jump NN	Event-Based	R	59.8	n/a	47.2
					P	86.4	n/a	58.0

1.5 Control Strategies for T1D Management

While regression and classification models offer reliable predictions for Type 1 Diabetes (T1D) patients by advising interventions such as carbohydrate (CHO) intake or insulin

administration [41, 4], they are limited when the goal is a fully automated artificial pancreas system. These models require patient involvement, which restricts their utility in achieving closed-loop control.

Performance Metrics in Control Systems The primary performance metric for glycemic control is Time in Range (TIR) percentage, defined as:

$$TIR = \frac{t_{\text{euglycemic}}}{t_{\text{tot}}} \tag{1.2}$$

where $t_{\text{euglycemic}}$ is the time for which $70 \leq \text{CGM} \leq 180$.

A higher TIR (Time in Range) indicates better glucose control, as it reflects the proportion of time a patient's glucose levels remain within the target range of 70 to 180 mg/dL. However, TIR is just one aspect of comprehensive glucose monitoring. Other critical metrics include:

- **Time Below Range (TBR)**: The percentage of time glucose levels are below the target range, typically split into:
  - **Level 1 Hypoglycemia**: Glucose levels between 50–70 mg/dL.
  - **Level 2 Hypoglycemia (Severe)**: Glucose levels below 50 mg/dL.

  Monitoring TBR is essential, as prolonged or frequent hypoglycemia can lead to acute complications such as seizures, loss of consciousness, and, in extreme cases, death.

- **Time Above Range (TAR)**: The percentage of time glucose levels exceed the target range, often categorized into:
  - **Level 1 Hyperglycemia**: Glucose levels between 181–250 mg/dL.
  - **Level 2 Hyperglycemia (Severe)**: Glucose levels greater than 250 mg/dL.

  Elevated TAR is associated with increased risks of long-term complications, including cardiovascular disease, neuropathy, and retinopathy.

- **Glycemic Variability**: While TIR, TBR, and TAR provide insights into glucose levels over time, glycemic variability (the fluctuations in glucose levels) is another important metric. High variability may indicate poor glucose control and has been linked to oxidative stress and vascular damage.

- **Composite Metrics**: Metrics such as the *Glucose Management Indicator (GMI)*, derived from continuous glucose monitoring data, provide an estimate of the patient's average glucose level, helping to contextualize TIR and related metrics.

These additional measures complement TIR to provide a more nuanced picture of glucose control. While improving TIR is a critical goal, minimizing time spent in hypoglycemic or hyperglycemic states, particularly severe episodes, is equally important for ensuring patient safety and long-term health outcomes.

Early Control Strategies Traditional T1D control strategies relied heavily on mathematical models. Proportional-Integral-Derivative (PID) controllers [75] and lookup-table or rule-based control systems [70] were common approaches. These methods depended on output error dynamics to determine insulin dosing, achieving moderate success in maintaining glycemic control. However, these rule-based systems lacked adaptability and could not optimize insulin delivery in complex, real-time scenarios.

Emergence of Reinforcement Learning (RL) in Glycemic Control In recent years, Reinforcement Learning (RL) has gained prominence as an alternative to traditional methods for blood glucose control [28]. RL systems are designed to autonomously determine the optimal action (e.g., insulin bolus or glucagon injection) to maximize a predefined reward. In this context the environment represents the patient or a simulator. The action typically involves insulin dosing, and in dual-hormone systems, glucagon administration. The reward is derived from maintaining blood glucose levels within the target range, avoiding adverse events such as hypoglycemia or hyperglycemia. However, most commercially available insulin pumps are limited to insulin administration, making dual-hormone systems impractical in many settings.

Model-Free RL Algorithms Nearly all RL applications for T1D control have utilized model-free RL algorithms [44, 136, 119]. These algorithms train agents through trial-and-error processes, directly interacting with the environment without prior knowledge of its dynamics [111, 112].

While model-free approaches are simple to implement and optimize, they face significant challenges. Training typically requires extensive data, which may span several years in simulated environments and trial-and-error learning directly on real patients poses unacceptable risks, making these models unsuitable for real-world deployment.

Zhu et al. [134] introduced a deep RL framework for basal insulin delivery using double deep Q-learning and recurrent neural networks. Their system captured the glucose-insulin-glucagon dynamics and utilized CGM, CHO intake, and insulin data as state variables. Despite achieving a Time in Range (TIR) of 80.9 %, their reliance on CHO intake as an input variable limited the system's feasibility for closed-loop control.

Fox et al. [44] addressed this limitation by designing 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.

Model-Based RL Given the limitations of model-free approaches, model-based RL algorithms have emerged as a safer and more efficient alternative. These algorithms integrate a predictive model of the environment to guide decision-making, minimizing the need for trial-and-error learning on real patients [112].

Yamagata et al. [122] pioneered a model-based RL framework combining Echo State Networks (ESNs) and Model Predictive Control (MPC). The ESNs predicted blood glucose levels, while the MPC optimized insulin dosing based on these predictions. This hybrid approach demonstrated performance comparable to model-free RL while significantly reducing risks. However, the system's reliance on CHO intake as an input variable limited its practical application in closed-loop artificial pancreas systems.

As we can see, achieving a fully automated artificial pancreas requires addressing the following key challenges:

- **Eliminating Carbohydrate (CHO) Dependency**: Current systems rely on manual meal announcements, which reduce autonomy. Future systems should detect and respond to glucose changes automatically.

- **Optimizing Training Methods**: Reducing training times and utilizing transfer learning can make reinforcement learning (RL)-based systems practical for real-world use.
- **Integrating Model-Based RL**: Predictive modeling of glucose-insulin dynamics can enhance safety and performance, enabling real-time decision-making without trial-and-error learning.

1.6 Motivation

Type 1 Diabetes (T1D) management requires constant monitoring and precise regulation of blood glucose levels to prevent complications such as hypoglycemia and hyperglycemia. While Continuous Glucose Monitoring (CGM) devices have transformed diabetes care by enabling real-time tracking of glycemic trends, several limitations in both hardware and software continue to hinder their effectiveness and accessibility, facing critical issues that affect their reliability and usability.

For instance, the measurement of glucose levels in interstitial fluid introduces inaccuracies and delays, particularly during rapid glycemic fluctuations. Furthermore, the reliance on battery-powered devices and their susceptibility to interference from other wireless systems often reduce their dependability.

Additionally, the sensitive personal data collected by CGM devices raise serious concerns about privacy and security, further complicating their adoption. The user experience with these devices also remains suboptimal, as discomfort, frequent maintenance, and calibration requirements discourage consistent use.

On the software side, Artificial Intelligence (AI) systems designed to enhance diabetes management are constrained by several technical and practical challenges. A major obstacle is the lack of sufficient and high-quality data for training these systems, as T1D patients must often manually record their blood glucose levels and insulin doses. The inherent variability in disease progression among individuals further complicates the development of generalized models capable of accurate predictions and tailored recommendations.

Effective insulin dosing, which must account for factors such as diet, exercise, and stress, presents additional complexities that current AI systems struggle to integrate comprehensively. Furthermore, regulatory approval processes for AI applications in healthcare remain slow and arduous, delaying the deployment of these technologies, while the performance of AI systems is often limited by the underlying hardware, as wearable devices with constrained computational capabilities cannot efficiently implement complex algorithms required for ad-

vanced analytics.

Finally, the lack of explainability of many AI models poses challenges for adoption in clinical settings, where transparency and interpretability are essential for building trust among clinicians and patients. Developing explainable AI methodologies is a necessary step toward embedding interpretability into predictive models and supporting informed decision-making. In addition, the sensitive nature of healthcare data necessitates stringent privacy and security measures. Federated Learning frameworks offer a scalable solution for decentralized data processing while preserving confidentiality and maintaining performance.

This thesis is motivated by the potential of AI-driven systems to address these interdisciplinary challenges and create transformative solutions for T1D management. By leveraging layered meta-learning and federated learning approaches for glycemic prediction, as well as Reinforcement Learning (RL) for adaptive insulin delivery, this research aims to bridge the gap between theoretical advancements and practical implementations. The integration of these methodologies is designed to provide accurate, reliable, and personalized care, while also addressing concerns related to computational efficiency, interpretability, and privacy.

Chapter 2

Contributions

This thesis aims to advance the application of Artificial Intelligence (AI) to diabetes management, primarily focusing on two critical and interrelated domains: *prediction* and *control*.

Prediction Forecasting of glycemic trends and adverse events is explored in detail through innovative methodologies aimed at addressing real-time applicability, precision, and privacy. Chapter 3 focuses on the application of meta-learning architectures for glycemic event prediction, emphasizing the advantages of multi-expert models and their scalability across diverse patient cohorts. The layered structure described in this chapter integrates outputs from specialized neural networks, achieving robust predictions with minimal false positives. Prediction plays a vital role in anticipating glycemic trends and adverse events, such as hypoglycemia and hyperglycemia, enabling proactive measures to mitigate potential risks. Recent advancements in predictive modeling have leveraged cutting-edge approaches like layered meta-learning and federated learning, which have significantly enhanced both the precision and applicability of glycemic forecasting. Layered meta-learning frameworks capitalize on multi-expert neural networks and meta-learners to provide robust, real-time predictions, tailored to individual patient profiles.

Complementing this is the federated learning approach presented in Chapter 4, which focuses on data privacy and computational efficiency. Federated Online Extreme Learning Machines (FedROS-ELMs) enable decentralized learning across devices while maintaining high predictive accuracy. This framework addresses security concerns by ensuring sensitive clinical data never leaves the device, using only mathematical gradients for model updates. The results demonstrate that this approach achieves comparable or superior performance to centralized models, with an emphasis on computational simplicity and privacy. The conclusion underscores the paradigm shift towards edge-based, privacy-preserving AI in healthcare, positioning federated learning as a critical tool for future applications.

Control Consequently, this manuscript investigates the domain of control, which aims to automate and optimize insulin delivery systems to maintain glycemic levels within a healthy range.

Chapter 5 presents the development of a Dual Proximal Policy Optimization (Dual PPO) system, which employs two specialized and independent PPO agents, each trained and optimized for different glycemic regions, to deliver personalized insulin doses. The dual-agent architecture enables precise control across hyperglycemic and euglycemic states while incorporating safety mechanisms to prevent hypoglycemia. Experimental evaluations using the UVA/Padova simulator demonstrate significant improvements in time-in-range metrics compared to conventional approaches, highlighting the robustness and adaptability of the dual PPO framework. The study concludes with a discussion on the scalability of reinforcement learning algorithms and their potential for real-world applications.

Building on this, Chapter 6 introduces a Multi-Agent Reinforcement Learning (MARL) framework called GLUMARL for cooperative insulin delivery. This chapter explores the decentralized execution and centralized training paradigm, illustrating the framework's effectiveness in managing glycemic variability across dynamic patient states. These agents operate in decentralized environments but are trained using shared frameworks to optimize collective performance, addressing the complexities of dynamic and partially observable patient conditions.

Through the integration of these advancements, this manuscript highlights a paradigm shift in diabetes management, where predictive insights derived from advanced AI models are seamlessly combined with adaptive, automated interventions. This synergy represents a transformative step toward the realization of a fully AI-driven diabetes management ecosystem, offering personalized, real-time solutions for improved patient care.

Chapter 3

Meta-Learning Architectures for Glycemic Event Prediction

The prediction of glycemic events, including hypoglycemia and hyperglycemia, is a critical area of research in diabetes management, with the aim of preventing adverse outcomes and improving patient quality of life. Despite significant advancements, current approaches face several limitations that hinder their effectiveness and widespread adoption. These challenges include reliance on regression tasks, an overemphasis on hypoglycemia prediction, and trade-offs between recall and precision that affect patient trust and engagement.

First, most models only focus on predicting future blood glucose levels with a regression task [94, 136]. 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 [45, 123].

Second, the vast majority of studies focus only on the prediction of hypoglycemia [104, 30, 16, 88, 58, 42]. 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. [88] 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 [104, 30] or impossible to compute [29, 13, 123, 76, 16, 97]. 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 [41],

with few exceptions [76]. 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 [97] to ensemble models [104] and cutting-edge technologies such as deep neural networks [123], 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 [96]. A new study from Zhu et al. [135] 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 [13, 30, 76, 16]. 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 [26] a meta-learning system based on a multi-expert predictive model relying on an event-based approach. The experts consist of either Recurrent Long Short-Term Memory (LSTM) or Convolutional Neural Network (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 [107]. 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 [115, 46], which is likely to be avoided when performing tasks on

edge devices [27]. Finally, we implement the proposed system on an edge-computing device to evaluate the real-life feasibility and applicability of the proposed approach.

3.1 Data and Preprocessing

3.1.1 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 [77] and its expansion [78] 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 [77, 78].

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.

3.1.2 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 [33]. It is worth noting that the patients from the UCBM dataset utilize a different CGM sensor than patients from the public dataset.

3.1.3 Preprocessing and Labeling

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.

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 [104, 30, 16] to 4-class problems [45]. 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 [45], 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 [43]. 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 [43]. 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 ($IR = 128$).

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, 130]. 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 [51, 103], 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., [82]) but also in the field of glycemic level prediction. Zhu et al. [131], 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.

3.2 Model Architecture

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 [96]. 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 [88, 41]. The softmax layer output of each expert is passed to a decision tree (the meta-learner). Figure 3.1 reports the architectural schemes of the two implemented base learners, while Figure 3.2 reports the scheme of the entire system.

3.2.1 Base Learner: Deep Neural Networks

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 [65], 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 3.2.2.

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 [10]. 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 3.1. The proposed model exploits only CGM as an input feature, thus $m = 1$ (univariate approach). In this work, 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 3.2.2.

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 [54] 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.

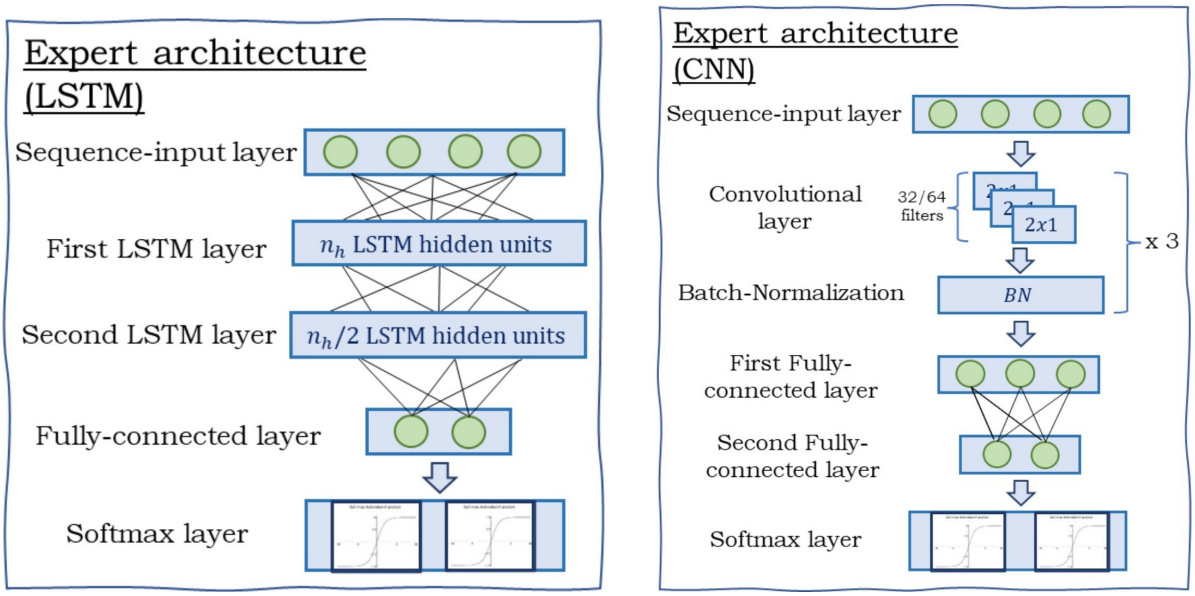


Figure 3.1: Schematic representation of the expert architectures. Left: the architecture based on the LSTM. Right: the architecture based on the CNN.

3.2.2 Meta-Learner: Decision Tree

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

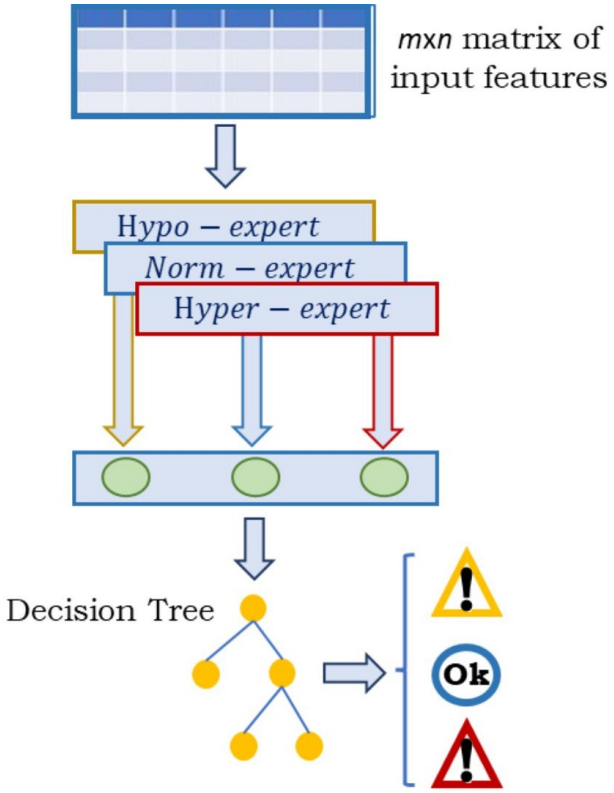


Figure 3.2: Schematic representation of the meta-learning algorithm and the single experts' architecture.

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 [23]. 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 3.2).

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 [7].

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.

3.3 Experimental Setup

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, 91].

Event detection Event-based performance evaluation requires preprocessing. According to the most widely used definition [45], 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 3.1.3, 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 [45, 29]. 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

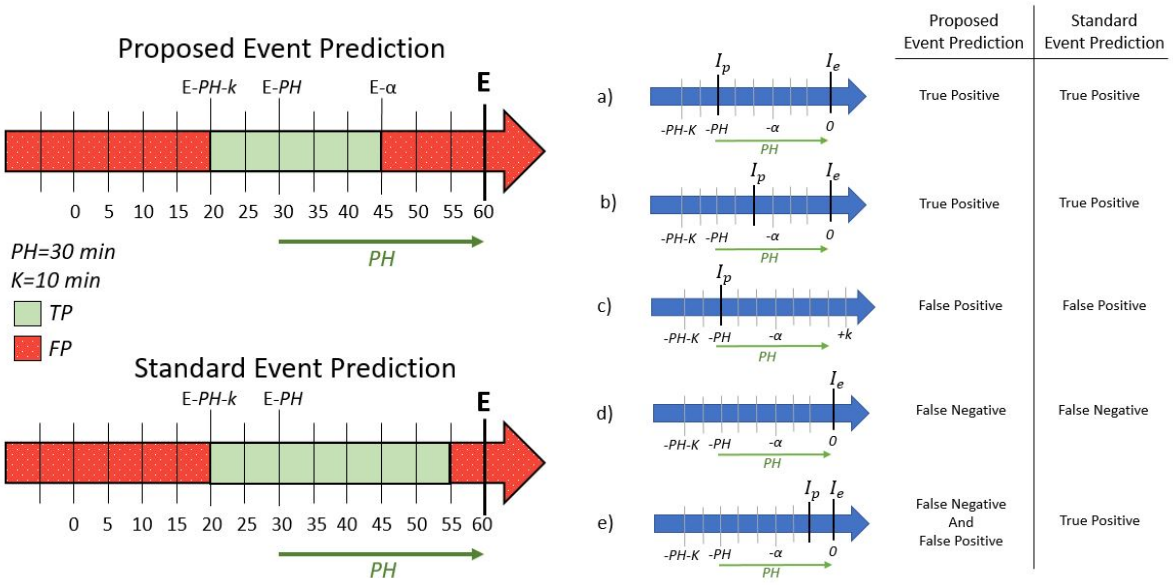


Figure 3.3: Comparison and differences between the proposed and the standard [45] event prediction approach. Left: example of how a predicted event is classified whether as a true positive or a false positive depending on the advance by which the prediction is performed. Given an actual event E beginning 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.

the standard approach, as it allows us to evaluate how many events are effectively detected at least α timestamps in advance.

Figure 3.3 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 3.4.

3.3.1 Evaluation on the public dataset

We test the proposed approach on the Ohio T1DM dataset with a Leave-1-Patient-Out Cross-Validation (Fig. 3.4a). 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.

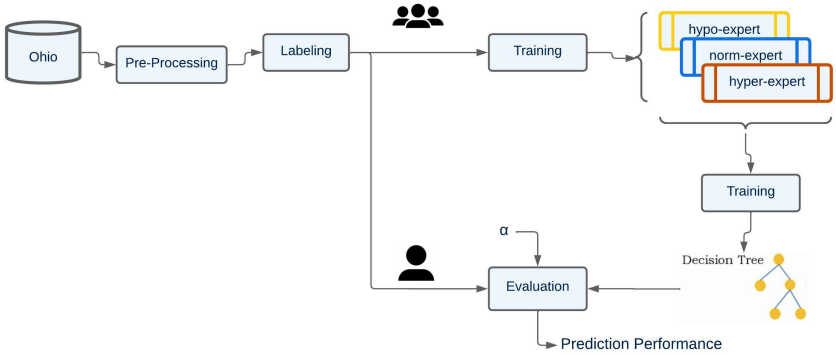
Comparison with other methods

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:

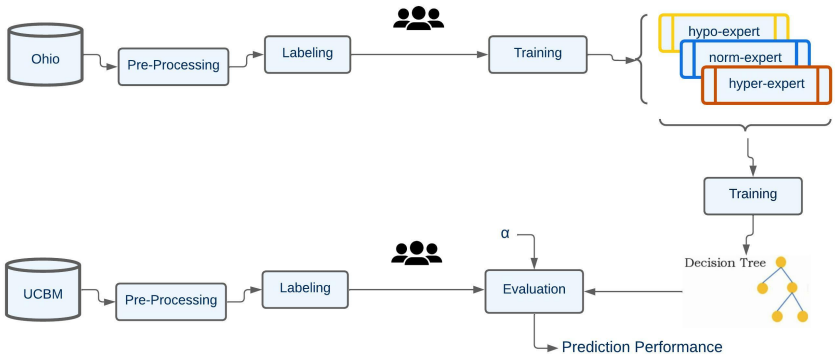
- A Support Vector Machine (SVM) with both polynomial (**SVM-poly**) and radial-basis-function (**SVM-rbf**) kernel. The latter model is the best classifier proposed by Gadaleta et al. [45]. Similar to our model, the learners were trained and tested with one-vs-all decomposition for the classification task.

- A Random Forest (**RF**), which was proposed by Seo et al. [104] and Dave et al. [30]. 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.

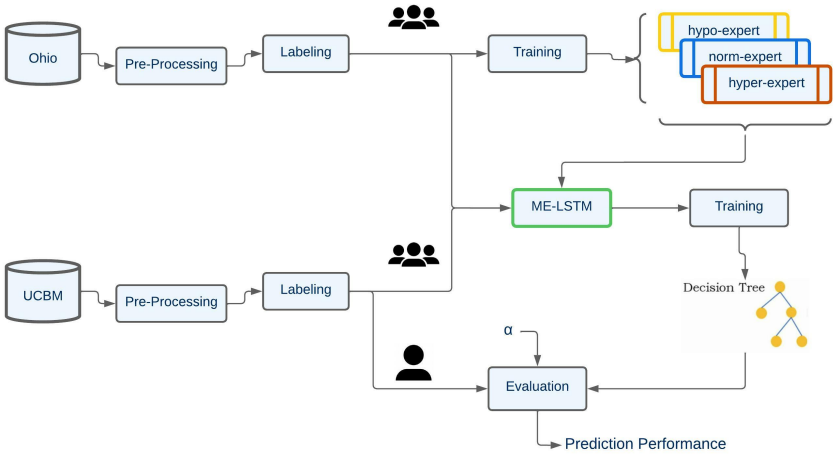
- Two different configurations of LSTM neural networks. We performed a grid search on our dataset to determine the optimal amount of LSTM hidden units for both models. The first presents a multi-expert architecture like the one proposed but includes simpler



(a) Test 1



(b) Test 2.1



(c) Test 2.2

Figure 3.4: Schematic representations of the experimental tests.

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 3.2.2:

- 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**).

3.3.2 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. 3.4b). 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. 3.4c). 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.

3.3.3 Edge implementation

To date, there are many devices capable of improving the lives of people with T1D [39], 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 3.1.3. 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.

3.4 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.

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 3.1 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 [88]. 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 [97, 135].

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 [72], or that have not been detected by the CGM sensor [16, 72]. 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 3.1: 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

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 [45]. 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%.

Table 3.2: 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 [45]	89	Leave-1-Patient-Out	SVM	Event-Based	R	86	n/a	95
					P	36	n/a	56
Daskalaki [29]	23	Precision Medicine	cARX + RNN	Event-Based	R	100	n/a	100
					P	n/a	n/a	n/a
Cappon [13]	100 in silico	Precision Medicine	XGBoosted Tree	Sample-Based	R	92	76	86
					P	n/a	n/a	n/a
Seo [104]	104	5-fold Cross Validation	Random Forest	Sample-Based	R	89.6	91.3	n/a
					P	38.9	n/a	n/a
Dave [30]	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 [76]	11	Precision Medicine	Kernel Ridge Regression	Sample-Based	R	64	96	61
					P	n/a	n/a	n/a
Cichosz [16]	10	Precision Medicine	Linear Logistic Regression	Sample-Based	R	79	99	n/a
					P	n/a	n/a	n/a
Yang [123]	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 [97]	112	Precision Medicine	Autoregressive Integrated Moving Average (ARIMA)	Event-Based	R	82	n/a	n/a
					P	64	n/a	n/a
Jensen [58]	463	5-fold Cross Validation	Linear Discriminant Analysis (LDA) classifier	Sample-Based	R	73	75	n/a
					P	22	97	n/a
Zhu [135]	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 [25]	33	Precision Medicine	ARTiDe Jump NN	Event-Based	R	59.8	n/a	47.2
					P	86.4	n/a	58.0

Of the models listed in section 3.2, only our previous work [25] 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. [135] (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. [97] 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. [29] 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. [123].

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. [45], 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 3.3 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 3.2 that pursue a sample-based approach, since only the study from Dave et al. [30], 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. [76], 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 3.4 the results achieved by the proposed models when a longer PH of 60 or 120 minutes is considered. The performance worsens sensitively for 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 [58, 48], which also suffer from a lower recall or precision score.

Table 3.3: 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 3.4: 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

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 3.3.1. 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 3.5 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 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 3.6, 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

Table 3.5: 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. 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 3.3.1, which are an SVM with radial-basis-function (SVM-rbf) [45] and with polynomial (SVM-poly) kernel, a Random Forest (RF) [104, 30], 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 3.6: 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

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.

Results and performance analysis on private dataset 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 3.7 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 [97]), 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 [33, 105]. 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.

Table 3.7: 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

Results of the edge implementation The tests on the edge system were carried out following the pipeline described in subsection 3.3.3. The results concerning training, conversion and inference time are shown in Table 3.8.

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, 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 3.8 are those concerning 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.

Table 3.8: 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	$6 \cdot 10^{-4} \pm 9 \cdot 10^{-4}$

Chapter 4

Federated Learning for Glycemic Level Forecasting

This study proposes a system for Online Learning (OL) of specialized models that are part of a federation dedicated to managing a privacy secure learning paradigm. The framework is based on a triple NN approach, in which each NN is a submodel that specializes in a particular glycemic condition. It leverages a decentralized and distributed learning paradigm called Federated Learning (FL), known for its excellent data privacy management capabilities. FL allows knowledge extracted from private data to be shared across a large network of devices that are part of a federation, without the need to share the data itself. Furthermore, the use of Extreme Learning Machine (ELM) [52] models within the FL environment enables the attainment of excellent generalization capabilities in conjunction with a very low computational burden. A variant of ELM, known as Online Sequential Extreme Learning Machine (OS-ELM), has been further enhanced to increase robustness, resulting in the Robust Online Sequential Extreme Learning Machine (ROS-ELM) [52], which is particularly well-suited for time series forecasting tasks. Therefore, models can be trained online with the minimum necessary computational resources and preserving privacy of sensitive data. The aforementioned elements are unified by a single methodology, which will be referenced here as *FedROS-ELM*, represented in Figure 4.1.

4.1 Overview

4.1.1 Federated Learning

Since its debut, FL has been extensively utilized in numerous domains of machine learning, applied to a multitude of tasks [128], including those in the clinical [86] and financial sectors.

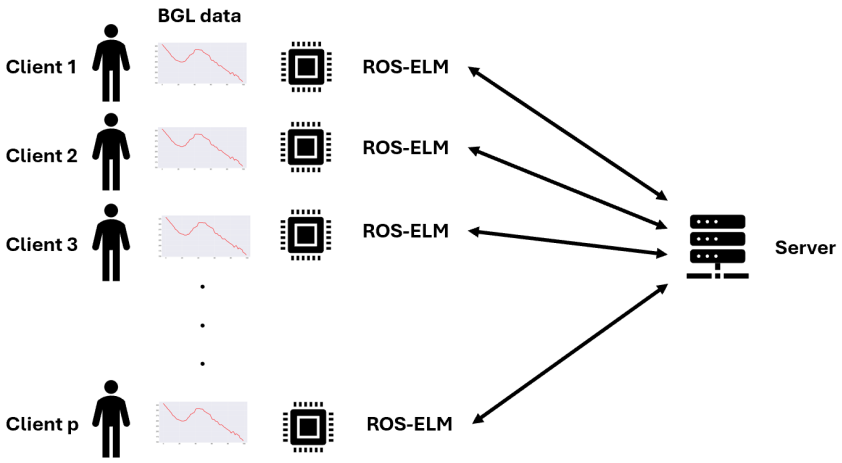


Figure 4.1: Representation of the FedROS-ELM framework in which each *client* represents a device connected to the individual patient for monitoring the glycemic level and training a local ROS-ELM model on it. The knowledge extracted from each local model is sent to the *server*, a centralized computing unit used for aggregating the knowledge shared within the federation.

FL, introduced by Google in 2016 [81], employs a decentralized and distributed learning paradigm, whereby sensitive data is never transferred outside of the device on which it is stored. Only training-related data like gradients or other mathematical structures containing knowledge is exchanged. For this reason, FL is particularly well-suited for the management of data that requires protection of its privacy, such as in the case of clinical data [73, 67]. In recent years, this approach has been demonstrated to be a viable option for the efficient and secure exchange of useful information between medical institutions [90]. This would facilitate a more quick flow of information, enhancing the quality of life for patients, accelerating the spread of knowledge derived from new data, and potentially making clinical solutions more prepared for novel challenges that arise. The case study in [32] illustrates a federated approach being successfully implemented in the field of BGL monitoring. A seven-class classification was performed, demonstrating the excellent generalising capabilities inherent to FL. The state of the art provides clear evidence of the effectiveness of the approach. The review in [86] presents the main applications of disease treatment using federated environments for predictions in the fields of electronic health records, heart rate, blood pressure, electrocardiogram, cardiovascular magnetic resonance images, diabetes and cancer. It is important to note that the number of studies applying FL to BGL regression tasks is relatively limited, which provides a motivation for the research supporting the present study. FL currently faces a number of challenges related to training time. A distributed environment introduces latency, which, while often negligible, becomes evident when numerous training

iterations are required. Additionally, the federated environment is temporally constrained by the computation time of the lowest-performing device. To address these issues, action can be taken to either reduce the latency of communication between servers and clients or to enhance the computational speed of the various clients.

4.1.2 ELM for Online tasks

From a detailed review of the literature on BGL prediction techniques, the topic of OL seems to be gaining in popularity. The reason for this lies in the phenomenon of interest. BGL, as we have already mentioned, is a quantity that depends on a multitude of factors linked to the lifestyle of the person in question, so there is a need to learn characteristics that vary very frequently. The availability of models that can be trained iteratively on new data makes it possible to deal with this variability. To do this, it is necessary to use specific architectures that not only guarantee the possibility of retaining the knowledge acquired in previous instances, but also have a low time complexity in order to be ready for frequent data updates. The choice of ELM is motivated by the above. ELM is a very versatile Pseudo-Linear model that makes it possible to learn from data very quickly, avoiding the complexity associated with the backpropagation process [31]. This learning model is particularly well suited to compensate the previously mentioned limitations of FL while retaining the advantages; in fact, it succeeds in being trained very quickly even on underperforming devices, since each learning iteration consists of solving a linear system. There are many examples in the literature of the application of ELM in its many variants, as in the case of [47], which highlights the main differences between ELM and three of its variants in the context of glycemic regression: K-ELM, OS-ELM and KOS-ELM. In [40] Vanilla ELM is used for the classification of the onset of diabetes achieving high levels of predictive accuracy. Unfortunately, only a few papers in the literature deal with applications of ELM to diabetes and even fewer deal with BGL regression tasks. An example of a hypoglycemia state detection task is given in [85]. The paper highlights the excellent generalisation capabilities of the model by comparing Vanilla ELM with one of its regularized variants: Regularized ELM. A minimal difference between the two variants and a common difficulty in handling OL tasks emerge. Difficulties solved by OS-ELM, designed specifically for OL tasks.

Comparison with the literature In order to highlight the innovation brought by this study, reference performance had to be selected from the literature. To do this, works containing deep learning models applied to the glycemic level regression task were selected. The comparison with the models was made in terms of tradeoff between prediction performance and computational complexity required for training. For this reason, we provisionally ex-

cluded from the selection excessively complex models which, although performing very well in terms of prediction error, belong to a category of prediction models that are not suitable for implementation in edge devices. These models include the Transformer and other large Deep Neural Networks architectures, which perform very well on time series forecasting tasks [129], [68], but are too complex for our purposes. Also relatively simple models, such as linear models, have been excluded from further consideration: while they require minimal computational effort, they typically exhibit lower levels of predictive accuracy than those found in the present study. Six deep learning approaches were selected from the literature, referred to as "comparison models", each of which was originally applied to Ohio T1DM Dataset. A summary overview of the approaches used can be observed in Table 4.1. The first selected work [89], based on ensemble learning, includes three different approaches (1), (2), (3) that differ in the criteria by which the predictions of three separate models (Linear Model, Vanilla-LSTM: VLSTM and Bidirectional-LSTM: BiLSTM) are aggregated: Stacking (1), Multi-variate (2) and Subsequences (3). The combined output of the three models is given as input to a Meta-Learner, which, depending on the approach, looks like a Linear Model (1), a Multivariate-LSTM (2) and a Conv-LSTM Encoder-Decoder (3). In the case of [63] we also find an ensemble approach, but this time it is the integration of the predictions made by MLPs and LSTM networks at different levels. Again, the work gives us more than one prediction model, in fact we find either an MLP network or an LSTM network as Meta-Learners. In [12] is shown the application of a BiLSTM network to three different feature sets (uni-variate and two multi-variate), of which only the uni-variate was considered for consistency. Also in [80] we find the application of Recurrent Neural Network (RNN) to the glucose level regression task, in particular LSTM networks. In all the papers presented the prediction error values are provided directly by the authors of the works, this is not the case for the computational complexity associated with training the models. The latter must be estimated from the architecture. To this end, reference may be made to [106], in which a consistent method is provided for calculating the Floating Point Operations (FLOPs), performed by the algorithm during training. However, this aspect will be described in more detail later.

4.2 Dataset and pre-processing

The proposed predictive system is structured in a federated macrostructure, comprising a *server* and several *clients*. Each *client*, which represents the physical device used to monitor the individual patient, is equipped with three ROS-ELM regressor models that are trained on distinct subsets of the dataset. At the conclusion of each training iteration, each *client*

Model	Approach	Based on	Computation	Dataset
Proposed	Triple NN	FedROS-ELM	Decentralized	Ohio T1DM
[89](1)	Ensamble	Linear+RNN	Centralized	Ohio T1DM
[89](2)	Ensamble	Linear+RNN	Centralized	Ohio T1DM
[89](3)	Ensamble	Linear+RNN	Centralized	Ohio T1DM
[63]	Ensamble	MLP+RNN	Centralized	Ohio T1DM
[12]	Single NN	RNN	Centralized	Ohio T1DM+Private
[80]	Single NN	RNN	Centralized	Ohio T1DM

Table 4.1: Summary of the proposed approach and the comparison models. The term Triple NN refers to the simultaneous use of three sub-models specializing in three different situations (euglycemia, hypoglycemia, and hyperglycemia).

transmits knowledge to the *server*, where it is aggregated with knowledge from all other *clients*. The information extracted from the data and aggregated is used to generate a global model containing knowledge from the entire federation, this model will then be responsible for making future predictions.

Dataset In clinical studies, excellent results are often reported on private datasets. Although this phenomenon allows researchers to have maximum control over the quality of the acquired data, it makes the methods proposed difficult to compare with other methods. A widely used dataset for diabetes-related research in recent years was selected for this work to facilitate comparison and maximize the reproducibility. Specifically, the Ohio T1DM Dataset [77], made available for a research challenge, the Blood Glucose Level Prediction Challenge, was chosen. The Ohio T1DM dataset is a collection of data regarding the main variables of interest in monitoring a patient with T1D. The version used for this study is the most recent and includes 12 patients on insulin pump therapy who are monitored by CGM devices. Specifically, patients are monitored over a period of 8 weeks with the Medtronic Enlite CGM sensors. There are 20 variables in the dataset, of which only the time series relative to the value read by the CGM sensor was taken into account to perform a uni-variate analysis. The variable read by the CGM sensor was sampled at a frequency of one sample every 5 minutes. This variable has not been processed but is reported as raw data, is affected by a slight noise and is about 7 minutes out of phase with the true BGL value, however, for the sake of simplicity, we will ignore this delay, the proportionality factor between the BGL and the value read by the CGM sensor, and the noise introduced by the latter. We will

refer to this variable as "BGL". It can be processed stochastically and theoretically it has a support: $S \in [0, \infty)$ mg/dL, in practice: $S \in [30, 400]$ mg/dL (from the data available). Samples acquired during the 8 week monitoring period are provided already divided into a training portion and a test portion. We decided to maintain this distinction.

Pre-processing In order to simulate the worst case scenario, no processing operation was applied to the BGL time series, the predictive model would then take as input the raw data as provided by the sensor. However, it was necessary to verify that the values were all temporally spaced by 5 minutes and that they were all positive real numbers ($\mathbb{R}^+$). For each sample with missing values or not spaced 5 minutes apart from the previous or next sample (with a tolerance of 10 seconds), it is labeled as "compromised". All compromised samples plus the (6 + PH) samples following the compromised were eliminated. PH is the prediction horizon, it represents how far ahead in time you want to perform the prediction and is expressed in number of samples. It was decided that a linear interpolation of the missing data would not be performed in order to avoid introducing a bias in the inference process. This would ensure that the model would not over-perform on the interpolated samples. Furthermore, no higher order interpolation was conducted as the majority of the missing or unusable data were isolated. The introduction of such a solution would have necessitated the local increase in sample density in the temporal neighbourhood of the sample to be interpolated. This, although effective in many contexts, would have deviated from the real-world application context in which sampling occurs at a fixed frequency.

The single training example has been generated as follows. A number of observation samples was chosen to be 6 (25-minute observation), i.e. the size of the input vector. This choice was motivated by two factors: firstly, this is the observation time span that maximizes the performance of our approach; secondly, a literature review shows that 6 observation samples is one of the most popular choices. Subsequently, the target sample for prediction must be selected, i.e. the one the model must learn to predict from the input. As previously stated, the time interval separating the last sample of the input vector from the prediction target is PH and this value was also chosen to be 6 samples for a prediction horizon of 30 minutes. Figure 4.2 provides a schematic representation of the generation of the single training sample. The entire dataset is transformed into training examples by treating the aforementioned process as a moving window. The step or resolution at which this window moves is variable and it was called SWstep.

4.3 Methodology

4.3.1 Expert model

With the aim of generate specialized models for different glycemic conditions, we first needed to define these conditions and find a way to differentiate them. This was achieved by defining a threshold, called $Hypo$, which separates euglycemic conditions from hypoglycemia ($Hypo$ = 100 mg/dL) and a second threshold, called $Hyper$, which separates euglycemic conditions from hyperglycemia ($Hyper$ = 180 mg/dL). Subsequently, $\hat{M}$, the mean value in mg/dL over the input vector, for each example was calculated. This value was then assigned to each example to identify its relevance to the areas of hypoglycemia, euglycemia and hyperglycemia. This operation permitted the training of three distinct models on three different *curriculum*: the first was designed for hypoglycemic conditions, or those that may lead to hypoglycemia, the second was designed for hyperglycemic conditions or those that may lead to hyperglycemia. Finally the third *curriculum* was designed for euglycemyc conditions. The calculation of $\hat{M}$ is precisely intended to manage transition conditions between the areas bounded by $Hypo$ and $Hyper$.
Although the dataset has been divided into the sections just seen, the order in which the global model is trained and thus with which the different local models view the time series has not been altered. For each training example, the glycemic condition indicated by $\hat{M}$ of the input vector is checked, on the basis of which the most appropriate regressor to be trained is chosen.

4.3.2 Extreme Learning

The idea behind ELM training underlies each of the local models in the proposed framework. This architecture is based on a pseudolinear model, which learns in a non-iterative manner: it simultaneously look at all the training examples given to it, on which it solves a linear system to extract knowledge. ELM is essentially a Neural Network *single hidden layer*, where the first layer performs a mapping of the input in a random feature space and has neurons that after being initialized, assigning to their weights $\mathbf{w}$ and biases $\mathbf{b}$ values sampled from a generally arbitrary distribution, are no longer updated. The second layer, on the other hand, performs predictions. An illustrative diagram of the network can be found in Fig. 4.3. The random mapping performed by the first layer can be formalized through the $\boldsymbol{H}$ matrix,

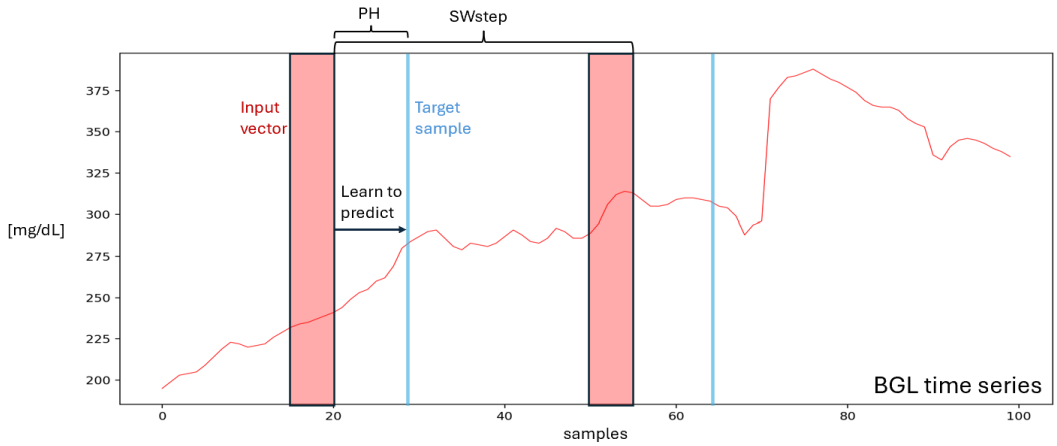


Figure 4.2: This image illustrates the process of generating training samples for the model. The red portion highlights the input vector formed by concatenating six consecutive samples, while the blue line identifies the target sample, which the model is tasked with predicting. 'PH' denotes the prediction horizon, expressed as the number of samples, and 'SWstep' refers to the step size by which the time window, used to generate training samples, shifts along the BGL series, also measured in sample units.

which takes the following form:

$$\boldsymbol{H}(\mathbf{X}, \mathbf{w}, \mathbf{b}) = \begin{pmatrix} g(X_1 w_1 + b_1) & \cdots & g(X_1 w_{\hat{N}} + b_{\hat{N}}) \\ \vdots & \ddots & \vdots \\ g(X_N w_1 + b_1) & \cdots & g(X_N w_{\hat{N}} + b_{\hat{N}}) \end{pmatrix} \tag{4.1}$$

The variable $\mathbf{X}$, which represents the training batch, is itself a matrix of dimensions (N, m), where N represents the number of examples and m represents the number of samples within the single example (in the present case, m is the size of the input vector and is therefore equal to 6), g represent a generic activation function. $\hat{N}$, on the other hand, represents the number of neurons in the hidden layer, $\mathbf{w}$ and $\mathbf{b}$ are respectively a matrix of dimension $(m, \hat{N})$ and a vector of dimension $(\hat{N})$. The output of the network can then be expressed as:

$$\boldsymbol{HB} = \boldsymbol{Out} \tag{4.2}$$

Where $\boldsymbol{B}$ represents the transfer function of the hidden layer and has dimensions $(\hat{N}, N_{out})$, where N_{out} is precisely the number of output neurons. By providing the network with targets $\boldsymbol{T}$ (having the same size as $\boldsymbol{Out}$), it is possible to calculate $\boldsymbol{B}$ by solving a simple linear problem:

$$\boldsymbol{B} = \boldsymbol{H}^{-1}\,\boldsymbol{T} \tag{4.3}$$

The calculated $\boldsymbol{B}$ will then be used to make predictions. The knowledge extracted from the data will therefore be contained within the values of $\boldsymbol{B}$. Such a learning process is limited: it has no memory of the past, since at each iteration $\boldsymbol{B}$ is recalculated and the information extracted during the previous iteration is lost. To overcome these limitation, it is possible to use a regularized variant of ELM specialized for online learning: ROS-ELM. In essence, our objective is to solve the following optimization problem:

$$Minimize : \frac{1}{2}\|\boldsymbol{B}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\|e_i\|^2 \tag{4.4}$$

$$subject\,to : \boldsymbol{H}(\mathbf{X})\boldsymbol{B} = \boldsymbol{T} - \boldsymbol{e}$$

Where $\boldsymbol{e}$ represents the vector of errors committed by the model for each training sample ($\boldsymbol{e} = \boldsymbol{T} - \boldsymbol{Out}$), C is a parameter that assigns a weight to the regularisation L_2 for the elements of $\boldsymbol{B}$.
The problem can be solved by deriving a closed-form:

$$\boldsymbol{B} = \left(\frac{1}{C}\boldsymbol{I} + \boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{T} \tag{4.5}$$

Assuming we increase the number of examples contained in $\mathbf{X}$ ($\mathbf{X_0}$ to $\mathbf{X_{0\cup1}} : \mathbf{X_0} \cup \mathbf{X_1}$), the number of rows of $\boldsymbol{H}$ ($\boldsymbol{H_0}$ to $\mathbf{H_{0\cup1}} : \boldsymbol{H_0} \cup \boldsymbol{H_1}$) will also increase. The overall $\boldsymbol{H}$ matrix takes the following form:

$$\boldsymbol{H_{0\cup1}} = \begin{pmatrix} \boldsymbol{H_0} \\ \\ \boldsymbol{H_1} \end{pmatrix} \tag{4.6}$$

it is then easy to verify that $\boldsymbol{B_1}$ (i.e. the $\boldsymbol{B}$ relative to $\mathbf{X_{0\cup1}}$) can be calculated from $\boldsymbol{B_0}$ (i.e. the $\boldsymbol{B}$ relative to $\mathbf{X_0}$):

$$\boldsymbol{B_1} = \boldsymbol{B_0} + \boldsymbol{K_1}\boldsymbol{H_1}^T(\boldsymbol{T_1} - \boldsymbol{H_1}\boldsymbol{B_0}) \tag{4.7}$$

$$\boldsymbol{K_1} = \boldsymbol{K_0} + \boldsymbol{H_1}^T\boldsymbol{H_1} \tag{4.8}$$

where $\boldsymbol{K_1} = \left(\frac{\boldsymbol{I}}{C} + \begin{pmatrix} \boldsymbol{H_0} \\ \\ \boldsymbol{H_1} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{H_0} \\ \\ \boldsymbol{H_1} \end{pmatrix}\right)^{-1}$. It is then possible to generalize these update laws with respect to the k-th iteration:

$$\boldsymbol{B_{k+1}} = \boldsymbol{B_k} + \boldsymbol{K_{k+1}}\boldsymbol{H_{k+1}}^T(\boldsymbol{T_{k+1}} - \boldsymbol{H_{k+1}}\boldsymbol{B_k}) \tag{4.9}$$

$$\boldsymbol{K_{k+1}} = \boldsymbol{K_k} + \boldsymbol{H_{k+1}}^T \boldsymbol{H_{k+1}} \tag{4.10}$$

The update process represented by equations 4.9 and 4.10 is employed within the ROS-ELM models utilized in the proposed system.

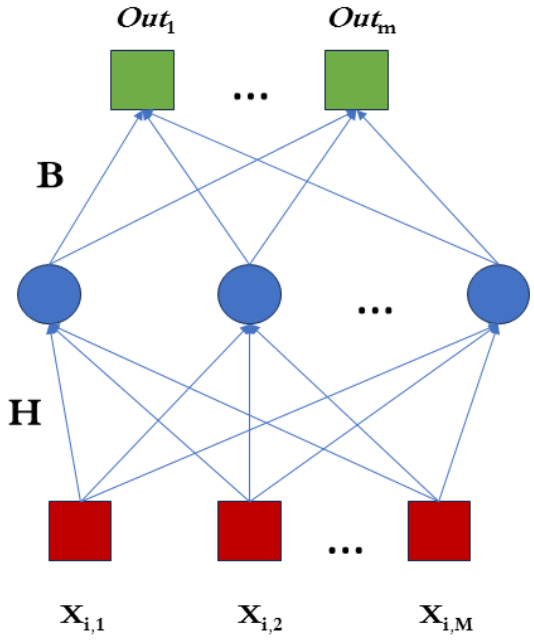


Figure 4.3: Graphical representation of an ELM as a single hidden layer neural network, processing the i-th example. The input matrix, represented by $\mathbf{X}$, is mapped by the random function $\boldsymbol{H}$ to produce a latent vector, highlighted in blue. The $\boldsymbol{Out}$ vector is then predicted by the function B.

4.3.3 Federated Approach

As previously stated, a decentralized and distributed training approach was implemented, enabled by the federated environment that is comprised of two actors: a *server*, which is responsible for managing the entire environment, and a series of *clients* (or nodes), which execute the *server*'s instructions in a federation that is characterized by a Master-Slave(s) approach. In the context of our case study, each node represents a single device used to monitor the individual patient. Consequently, there are 12 *clients*. Each node maintains a local and private dataset that is continuously updated, as it is always measuring new samples of BGL. The intrinsic robustness and security of FL from a privacy perspective is derived from the fact that each node trains a learning model on its own dataset, without sharing it with any other participant in the federation. Subsequently, each *clients* performs its own

training process, which is arbitrarily defined by the *server* but consistent across all *clients*. The parameters, computed by the *clients*, which contain the knowledge extracted from the data, are then sent to the *server*. The method by which the parameters from the various *clients* are aggregated is arbitrary and depends on the specific problem. For instance, simple averaging may be employed. The aggregation produces a set of parameters that is compatible with the learning model chosen by the *server* beforehand and it is used to produce a trained global model, which is then sent back to the *clients* so that they can update it on the new data. This is an iterative process that lasts until the end of the available data or up to a set maximum number of iterations (commonly called rounds).

Initialization

The process of initialising the federated environment involves a number of fundamental steps necessary for proper peer-to-peer efficent communication:

- **Setup**: *clients* wishing to participate in the federation must notify the *server* of their intention.
- **Verification**: In this phase, the *server* verifies that the information sent by the *clients* is correct.
- **Defining the environment**: The *server* informs the participating *clients* of important information: the chosen learning model and its hyperparameters, the maximum number of rounds, the number of iterations per round, the characteristics of the object containing the learned parameters and its size.

Rounds

: Each round follows the same procedure. All the *clients* in the federation perform one or more training iterations (depending on what was specified during initialisation) and store the parameters resulting from the training in memory, to send it to the *server* at the end of the current round. The training phase ends when the *server* receives positive feedback, concerning the execution of the training calculation, from all the *clients* participating in the federation. The *server* informs all the *clients* to send their parameters. There is then a phase of aggregation of these parameters *server*-side. It results in an object that is compatible with the established learning model, so that global parameters can be obtained, which are sent back to all *clients* and determines the end of the current round.

4.4 Experimental setup

In this section we present the choices made to adapt the system to the BGL forecasting task and the aspects investigated through a series of experiments.
Regarding the process of constructing the training examples, the step (SWstep) at which the observation window, that generates the input vector, moves over the BGL series, was considered as a hyperparameter to be investigated. We started with a window moving with a SWstep of 1 sample per example (i.e. the j-th example is found lagged by 1 sample from the (j-1)-th example), up to a SWstep of 3 samples per example (i.e. the j-th example is found lagged by 3 samples from the (j-1)-th example). Larger SWsteps have been neglected in order to avoid updating the model too infrequently, which would have been clearly contrary to the objectives of this study, one of which is to propose a predictive system capable of adapting very quickly to the variability of the data.
Focusing on the hyperparameters of ROS-ELM, due to the relative simplicity of the model and its extremely short training times, we decided to tune everything that could be tuned, in particular we find the size of the training batch (N, interpreted as the number of examples that the network sees simultaneously and on which it solves the linear problem), the number of neurons in the hidden layer ($\hat{N}$), constrained, as shown in [53], by the size of the batch. We also have the value of the L_2 regularisation parameter (C) and, finally, the activation function used in the hidden layer (g).
In the federated environment, only the aggregation method aspect was investigated, in particular the mathematical function required to aggregate the parameters sent from the *clients* to the *server*. These functions were tested in terms of their robustness against the presence of outlayers artificially inserted into the dataset through dedicated experiments.
For the hyperparameters, a grid search was performed and the ranges of all the investigated values are reported in Table 4.2.

Hyperparameter	**Range**	**Step**
SWstep	[1, 3] samples	1 sample
Batch Size	[10, 200]	10
Hidden Neurons	[50, 100]	10
C	[0.001, 0.1]	0.005
g	[ReLU, Sigmoid, Tanh]	//

Table 4.2: Ranges of the tuned hyperparameters.

The federated aggregation methods investigated are:

- **The Geometric Mean**: it is the simplest and computationally least expensive aggregation method selected, with a complexity that can be approximated as $\boldsymbol{O(\hat{N}p)}$ where $\boldsymbol{\hat{N}}$ represents the size of the matrix $\boldsymbol{B}$, which, in a regression task, is reduced to a vector of dimensions $\boldsymbol{(\hat{N}, 1)}$, while $\boldsymbol{p}$ represents the number of vectors over which to compute the mean (i.e. the number of participants in the federation, 12 in our case);

- **The Weighted Geometric Mean**: this type of aggregation is characterized by the multiplication of each element of the vector $\boldsymbol{B}$ by a weight ***score*** that quantifies the the inverse of the error (i.e. Root Mean Square Error: RMSE) and thus the accuracy that the single *clients* presents on a local test set containing samples related to the same subject on which the *clients* itself has been trained. The complexity of this method is still comparable to that of the simple Geometric Mean, which can again be approximated by $\boldsymbol{O(\hat{N}p)}$;

- **The Geometric Median ($\boldsymbol{GM}$)**: it is a particularly robust aggregation method in the presence of outliers (vectors $\boldsymbol{B}$ geometrically very distant from all others), it is extremely effective in reducing the deterioration of the global model's performance (consequent to aggregation) in cases of malfunctions or corruption of one or more *clients*. This method has been implemented through an iterative algorithm, whose convergence is guaranteed in a finite number of iterations, through a tolerance set a priori on the distance in the space $\mathbb{R}^{\hat{N}}$ and equal to 10^{-6}. The complexity of this method grows compared to the previous ones and can be approximated by $\boldsymbol{O(I\hat{N}p)}$ where $\boldsymbol{I}$ represents the number of iterations required for convergence:

$$\arg\min_{\boldsymbol{GM}} \sum_{i=1}^{p} (\|\boldsymbol{B}^{i,2} - \boldsymbol{GM}\|^2) \tag{4.11}$$

  In the preceding expression, $\boldsymbol{B}^i$ denotes the $\boldsymbol{B}$ vector sent to the *server* by the i-th *client*.

The RMSE is utilized not only within the second aggregation method but also as a performance metric to assess the performance of the global model trained on the test set. However, predictive accuracy does not take into account the clinical risk associated with the predictions made by the model. Therefore, relying solely on the RMSE would be a superficial approach. For this purpose, a metric widely accepted in diabetes research has been used: the Clarke Error Grid or CEG. The CEG represents a graphical tool for assessing the clinical applicability of blood glucose predictive or measuring instruments. In particular, the CEG is

presented as a two-dimensional grid that relates predictions (or measurements) to a reference glycemic value, which is considered reliable. The 2D plan is divided into five zones (A, B, C, D, E) characterized by increasing levels of risk associated with the clinical measures that can be taken consequently to the predictions (or measurements) of the instrument being assessed. The risk levels range from no risk to a risk of serious complications, which may result in death.

4.4.1 FLOP estimation

It is not straightforward to ascertain the complexity of a training algorithm when the implementation process is unknown. One might consider training time as a metric of complexity, but this is highly dependent on the machine used to perform the calculations. Furthermore, it is not always possible to trace the hardware specifications of the machines used to train the predictive models in the literature. We were assisted by [106], which presents a method for estimating the number of floating-point operations (FLOPs) required for the training process of the most common deep learning architectures. The number of FLOPs represents the number of mathematical operations carried out within an algorithm and allows us to estimate its computational complexity, as opposed to processing times, which are highly dependent on hardware specifications. In the aforementioned article, it is evident that there is a clear distinction between architectures trained with backpropagation and those trained without. For each architecture under investigation, the number FLOPs associated with the processing of either a single training example, a training batch or an entire epoch can be calculated. The training-related complexity is defined as the total processing complexity, including the updating of the architecture parameters. In particular, the latter can be approximated as three times the FLOPs required for processing alone (i.e. without updating the parameters). This allowed us to estimate the complexity involved in the training of the models selected in the literature and highlighted above as comparison models.

4.4.2 Inference modes

We decided to evaluate performance, and then extract RMSE and CEG, in two different test configurations. The purpose is precisely to show the behavior of the framework in two different application contexts:

- **Test on test set**: In the first case, we simply use the test data provided directly by the publishers of the dataset. In each federated round, after aggregating the parameters from the different *clients*, we have a test phase on the entire test set.

- **Online test**: In the second case, the approach changes. We wanted to simulate a real application context; in fact, in each round, the model should make predictions on data immediately following the data on which it has just learned (a classic online learning condition). To do this, in each round the global model is tested on the training batch immediately following the one it has just seen.

In both scenarios, the performance obtained is the average of the performance obtained by the global model on the data for the 12 different subjects.

4.5 Results and discussion

First, before presenting the results, you can appreciate the outcome of the grid search in Table 4.3. The hyperparameters shown in that table are those used to obtain all the results.

Hyperparameter	**Value**
SWstep	1 samples
Batch Size	100
Hidden Neurons	100
C	0.01
g	ReLU

Table 4.3: Optimal hyperparameters obtained via grid search.

4.5.1 Model Performance: Single vs Federated Learning

The trained global model proved to achieve particularly low prediction error values in the test phase, confirming the initial hypotheses about its excellent generalisation capabilities. This is true both for the online test conditions and for the entire test set, as shown in Figure 4.4. In the same figure, the rapid convergence of learning to a quasi-stationary value can also be observed. In particular, the global model achieves an average minimum RMSE (averaged over the 12 *clients*) of 14.73 for the test on test set and 18.48 for the online test during the FL process. The RMSE value converges very quickly after 20 rounds in both test cases, suggesting a model pre-tuning window of about one week (consistent with the sampling times used in the acquisition of this dataset) before reliable performance is achieved. It can be seen that there is a substantial difference between the mean error committed by the global model in consecutive rounds, evidenced by high variability in the trend of RMSE. The percentage

absolute error committed in relation to the target BGL value is depicted in Figure 4.5, to highlight the difficulties encountered by the model in making predictions under different glycemic conditions. It can be observed that the predictive accuracy is particularly high in the euglycemic zone, but it deteriorates in the hypoglycemic and hyperglycemic zones but this is justified by the imbalance in the dataset between the different glycemic conditions. With regard to the clinical applicability of the method, the analysis carried out by the CEG showed a good degree of clinical reliability. In fact, as can be seen in Figure 4.6 a percentage of predictions greater than 97.88% can be found in zones A and B (characterized by low risk) for all subjects, a percentage of predictions of about 0.02% in zone C and about 2.20% in zone D. As can be seen from the graph of the CEG, the predictions are distributed on or around the diagonal.

Triple Regressor In the case of the triple regressor model, performance improves significantly. A minimum RMSE averaged over the 12 subjects of 13.73 in the case of the on test set and 12.03 in the case of the online test is achieved, as can be appreciated in Figure 4.7. There is a clear advantage in terms of RMSE over the single regressor model in both test cases, also both the variability of the mean RMSE and the mean standard deviation over all rounds are reduced. This is due to the greater accuracy with which predictions are made mainly in the hypoglycemic zone but also in hyperglycemic one, as the triple regressor model provides specialized sub-models for different glycemic conditions. Figure 4.8 is intresting because it indicates that employing three regressors significantly enhances the performance achieved in hypoglycemia compared to the single regressor, while the performance on hyperglycemia remains essentially unaltered. The reason for the fluctuations in this graph (for BGL equals to 80 mg/dL and 100 mg/dL) is due to the method used to select the expert sub-model for the current forecast. As mentioned above, this method is implemented using a hard approach, which inevitably makes the transitions discontinuous. However, it should be noted that these oscillations are limited to a range in which the percentage of error is under 6%. The utilization of the three regressors does not affect the rate of convergence during training. This is reasonable to assume, given that no novel information is introduced into the dataset in comparison to the single regressor case, and that the three regressors do not share any internal information. It is quite obvious that the introduction of specialized models for different glycemic conditions reduces the level of risk associated with predictions. This is evident from the CEG in Figure 4.9. In particular, we have a 98.75% prediction rate for the trained global model in zones A and B, about 0.08% in zone C and about 1.16% in zone D. In general, from Figure 4.9, it is possible to visually observe a decrease in the concentration of points in zone D between 0 and 70 mg/dL of the reference value, indicating

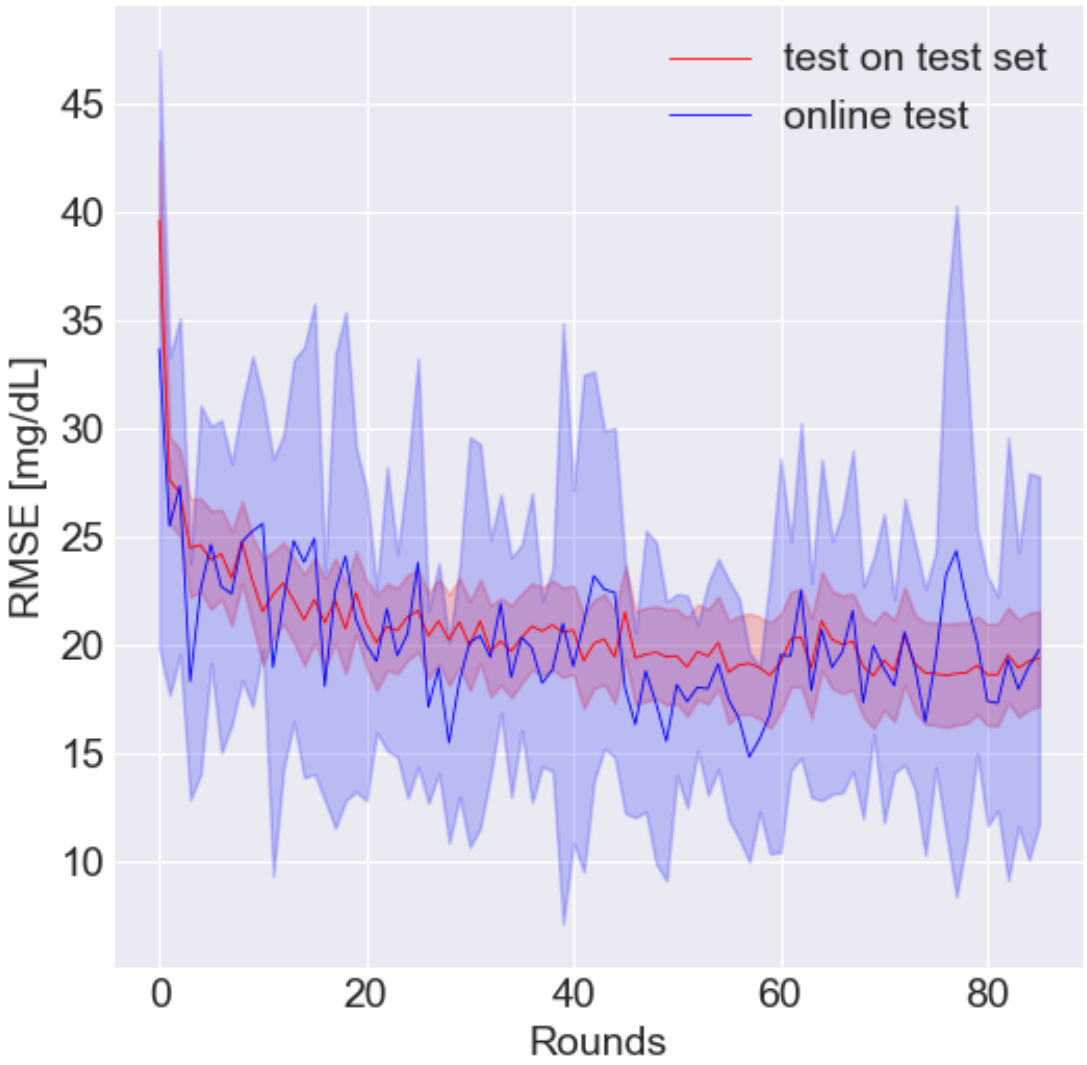


Figure 4.4: Representation of the average error committed by the global model in testing phase for both testing methods. The solid lines indicate the RMSE value averaged over the data of the 12 subjects, while the opaque area indicates the standard deviation obtained under the same conditions.

a reduction in risk compared to the single regressor case; there is also a greater adherence of the points to the plane bisector.

It is important to note that the introduction of the three regressors does not significantly increase the complexity of the algorithm. In fact, the additional FLOPs compared to the single regressor case can be considered negligible. This is true because, during training,

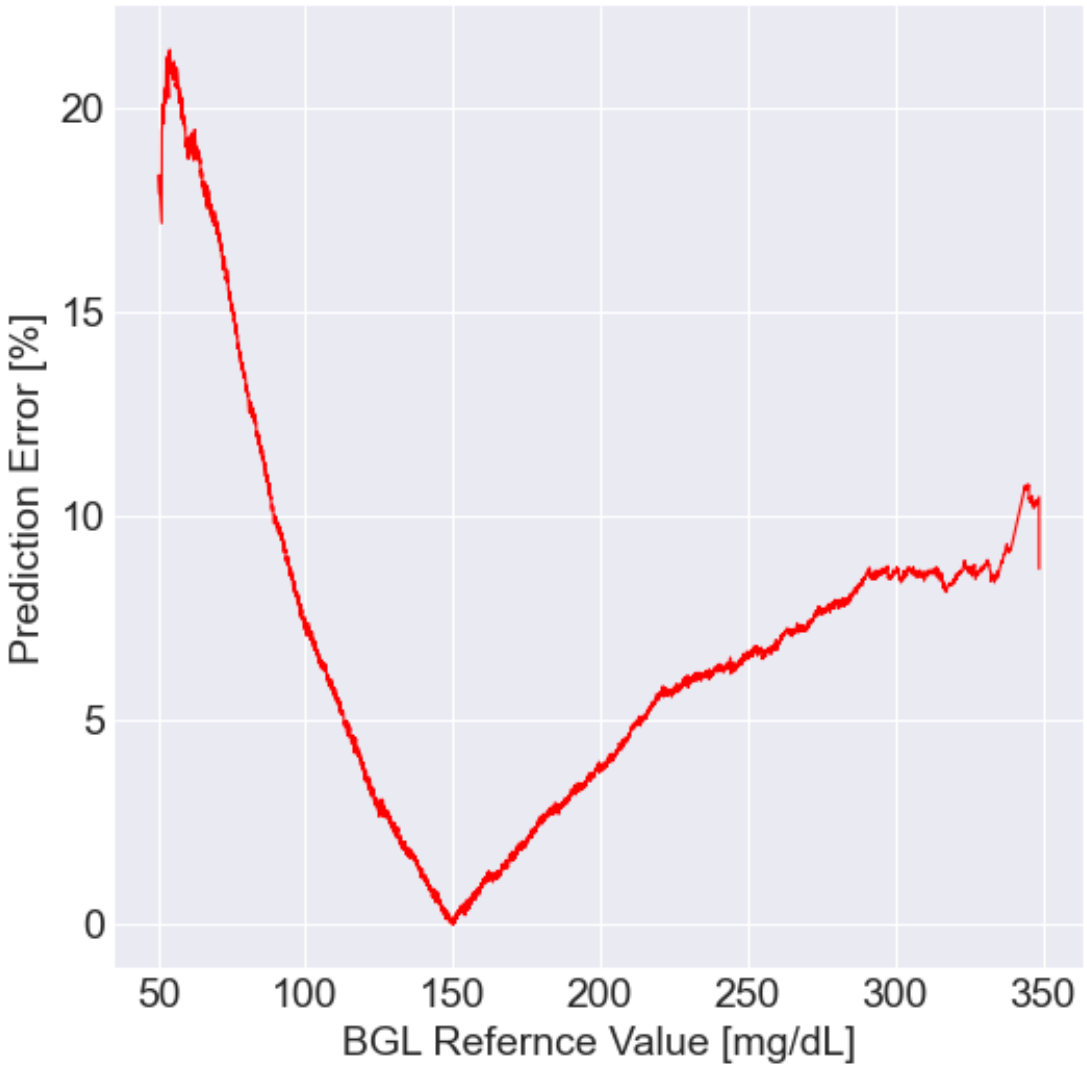


Figure 4.5: Percentage error committed by the global model. The graph shows a high level of predictive accuracy within the euglycemic zone (100–180 mg/dL), with a marked decrease in accuracy for BGL values below 100 mg/dL, as well as in the hyperglycemic zone (BGL > 180 mg/dL).

for each training sample, the competence model is selected from the input vector for the network, as previously explained. This mutually exclusive model selection system presents a computational complexity that can be neglected. The additional complexity that must be considered pertains to the training of the model under conditions that are dubious in terms of glycemic control. These conditions are those that are situated near the thresholds

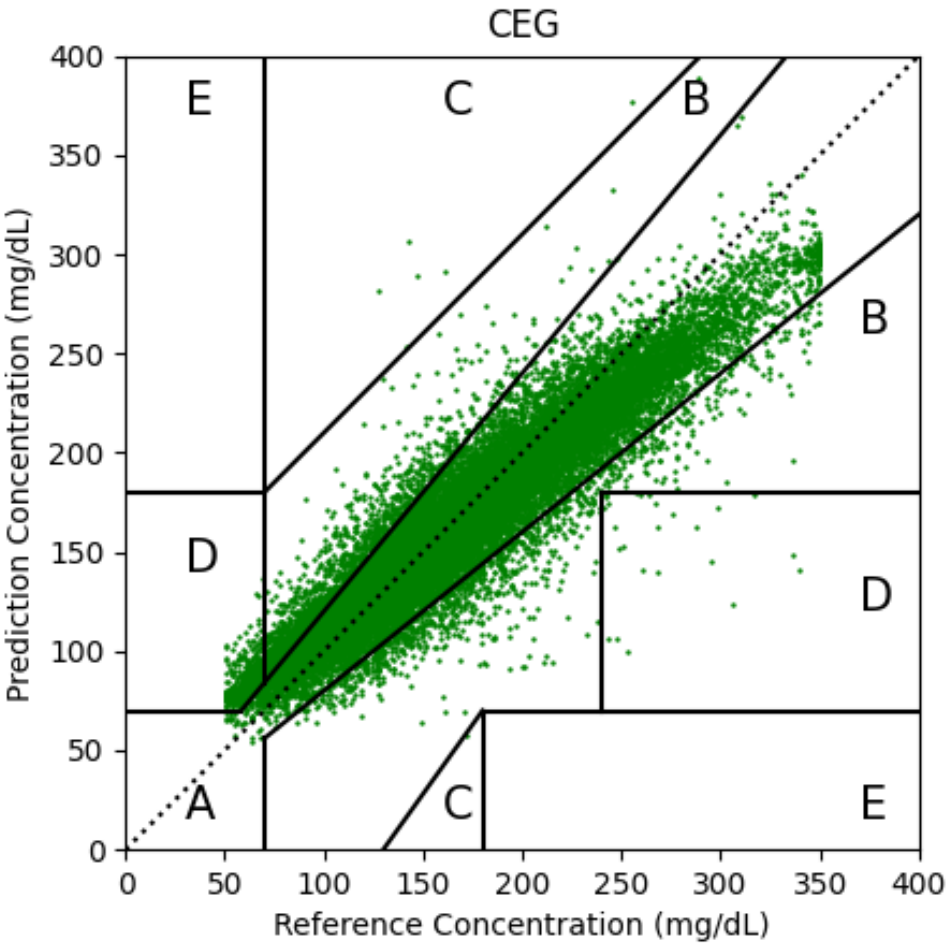


Figure 4.6: Distribution of predictions generated by the global model on the test set of data from 12 subjects, within the five zones of the CEG.

that separate euglycemia from hypoglycemia and euglycemia from hyperglycemia. In such instances, all the sub-models in question are trained simultaneously in order to facilitate a seamless transition between sub-models in terms of RMSE and percentage error.

Federated aggregation The outcome of the robustness experiments demonstrated the previously established properties of the geometric median, which was identified as the optimal aggregation method among those investigated. It is noteworthy that the advantages of employing the geometric median are only discernible under specific circumstances, such as *clients*-side malfunctions or *server-clients* communication issues. In such instances, the geometric median is observed to result in a reduction in the extent of performance deterioration in comparison to the other methods that were investigated.

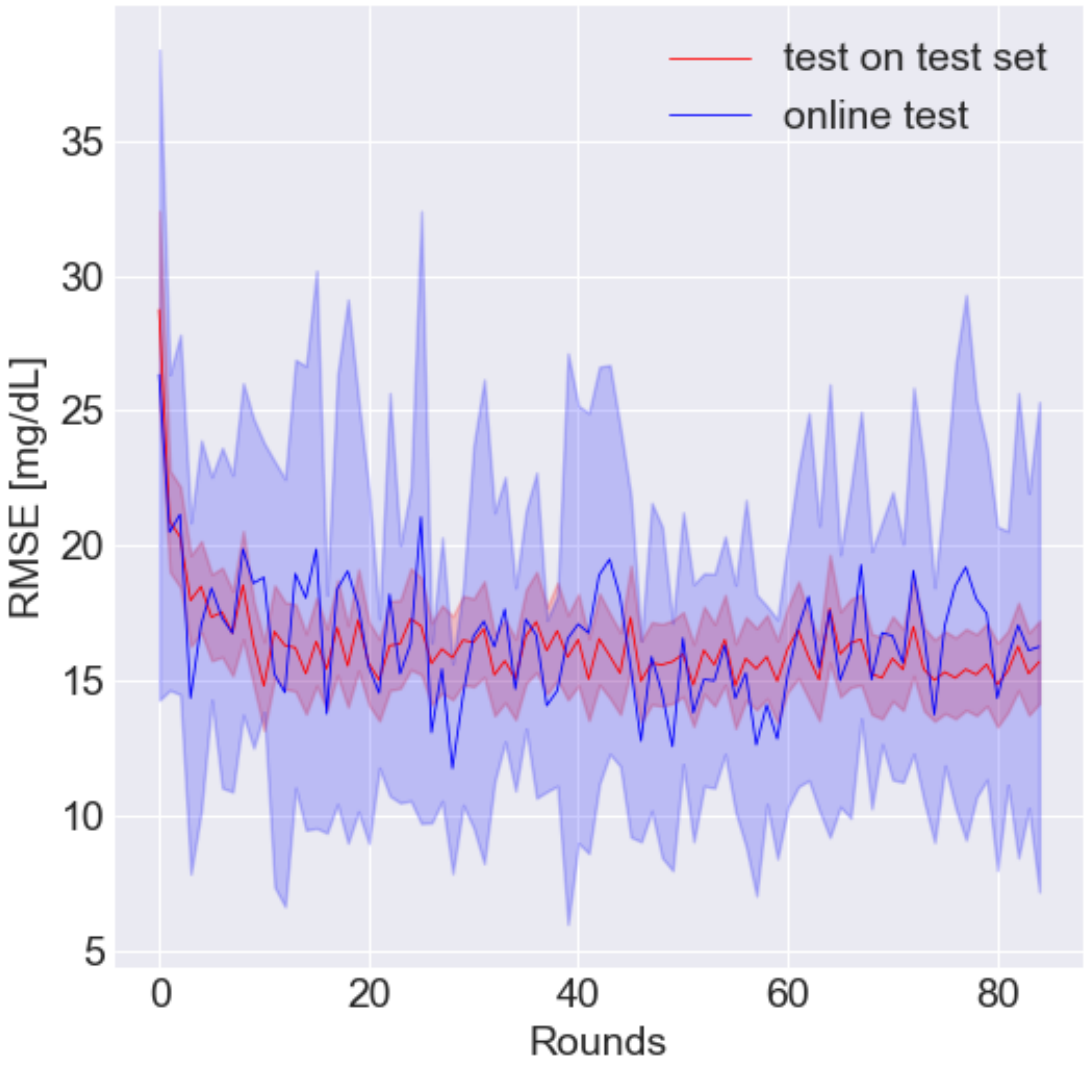


Figure 4.7: Representation of the average error committed by the triple regressor in both tests. The RMSE was plotted in relation to the federated rounds, obtained by performing inference by both the test on test set and online test. The solid lines indicate the RMSE average value, while the opaque areas indicate the standard deviation obtained under the same conditions.

4.5.2 Comparison with Existing Approaches

Throughout the experimental phase, the proposed system has shown remarkable capabilities that demonstrate its validity and superiority over the comparison models. In particular, the triple regressor FedROS-ELM has been shown to obtain overall lower RMSE values than

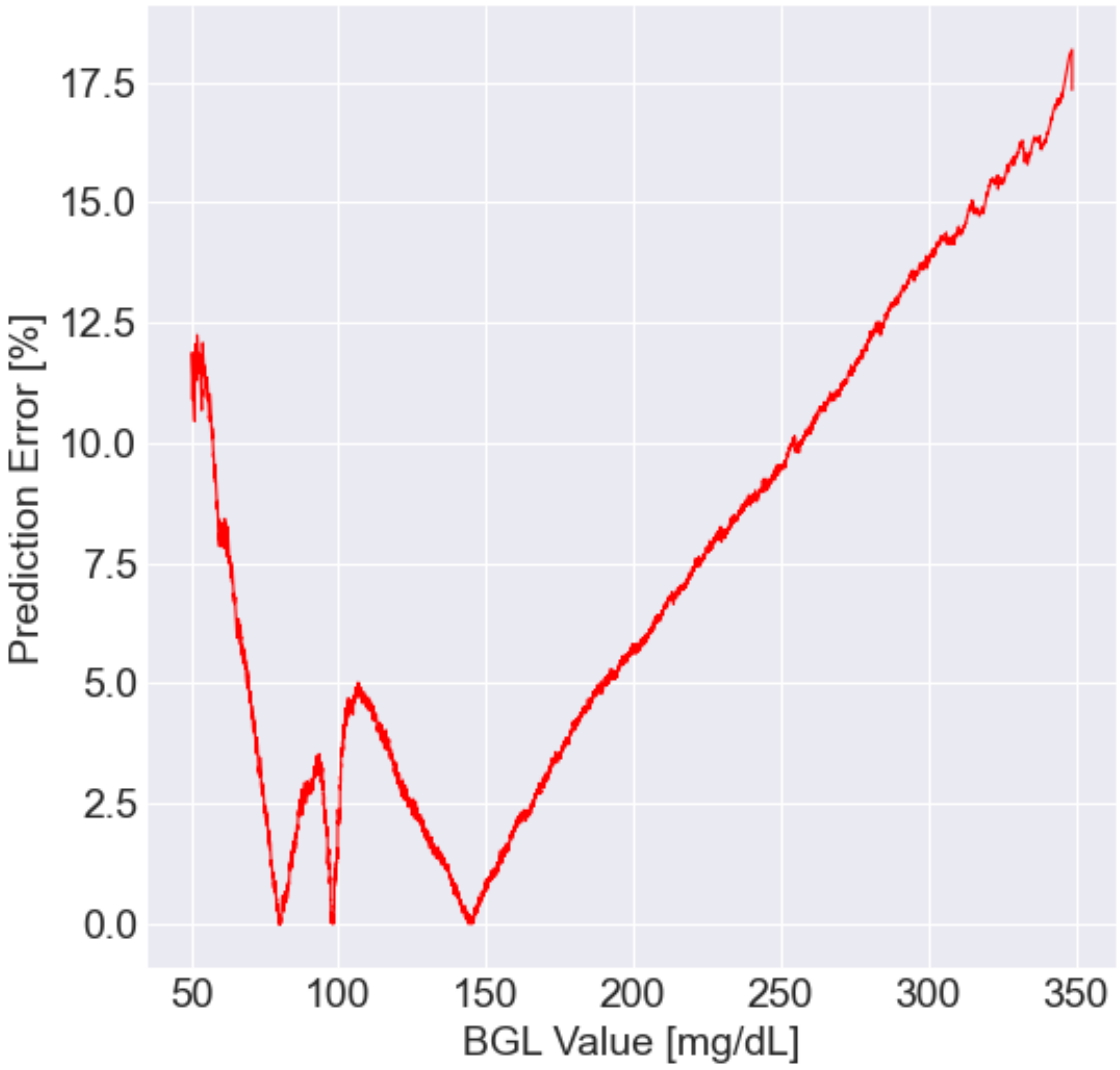


Figure 4.8: Representation of the percentage error committed by the global triple regressor model.

comparison models. However, it is important to note that the performance of [12] is affected by an analysis that includes data from both the Ohio T1DM dataset and a private dataset, a circumstance that could compromise the rigour of the direct comparison.

The trade-off between performance and complexity of the proposed model is particularly favourable. This is demonstrated by an average RMSE of 15.03 mg/dL, achieved with a notably low computational cost of approximately 10^3 FLOPs. This represents a 10.59% reduction in RMSE compared to the best-performing competitor model [12], which, in contrast,

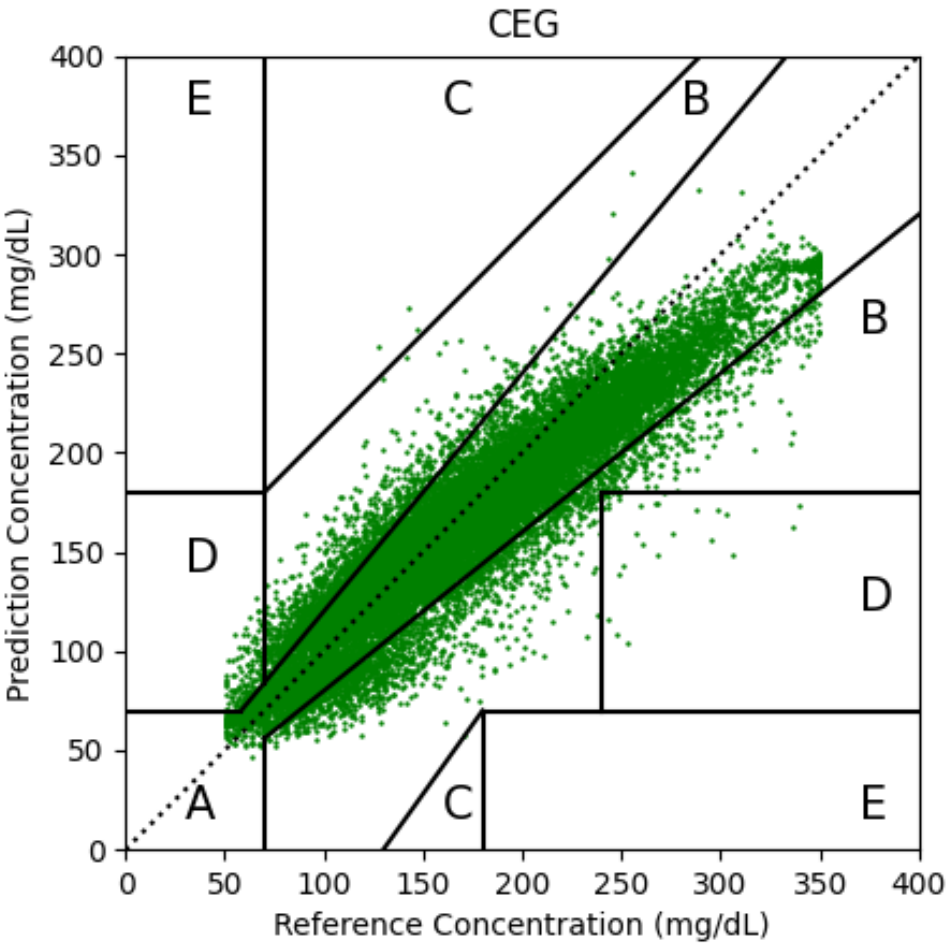


Figure 4.9: Distribution of predictions performed by the global triple regressor model within the five zones of the CEG.

demands three orders of magnitude more FLOPs. Importantly, the use of linear interpolation for missing samples in [12] may have simplified the prediction process, directly affecting the performance comparison. Table 4.4 summarizes the comparison.

Model	**RMSE**	FLOPs
Proposed	**15.03**	$\sim \mathbf{10^3}$
[89](1)	19.63	$\sim 10^7$
[89](2)	19.64	$\sim 10^8$
[89](3)	19.62	$\sim 10^8$
[63]	19.97	$\sim 10^6$
[12]	16.81	$\sim 10^6$
[80]	18.87	$\sim 10^6$

Table 4.4: Comparison between the proposed approach and models selected from the literature in terms of RMSE and number FLOPs.

Chapter 5

Deep Reinforcement Learning for Personalized Insulin Control

Managing blood glucose levels in patients with Type 1 Diabetes Mellitus (T1DM) is crucial for maintaining long-term health and preventing medical complications. In this study, we propose [74] a hybrid closed-loop control system for an artificial pancreas consisting of a personalized insulin delivery system based on a Dual Proximal Policy Optimization (Dual PPO) controller, optimized for each patient through a grid search on pre-trained models to handle both hyperglycemia and hypoglycemia. Personalized insulin bounds characterize the system incorporating a safe-control mechanism to prevent insulin administration if the glucose level falls below a predetermined threshold. We conducted experiments on 10 *in silico* adult patients generated using the UVA/Padova simulator, evaluating the performance of the system in terms of time spent in euglycemic, hyperglycemic, severe hyperglycemic, hypoglycemic, and severe hypoglycemic regions for five days in a randomized meals scenario. Dual PPO achieves significant improvements in time spent in the euglycemic range (Time in Range (TIR) = 69.30% $\pm$ 1.61), compared to single PPO model (TIR = 61.69% $\pm$ 1.54). Concurrently, our hybrid closed-loop control system requires only minimal interaction from patients through meal announcements, distinguishing itself from other open-loop systems such as Basal-bolus controller and Proportional–Integral–Derivative controller that require carbohydrate estimation for each meal. Our findings highlight the potential of Dual PPO controller for personalized insulin delivery in patients with T1DM. The proposed approach holds promise for advancing precision medicine in diabetes management and may pave the way for future clinical applications in real-world settings.

5.1 Reinforcement Learning in Glycemic Control

State of the Art Reinforcement Learning (RL) is an area of machine learning where agents learn optimal policies by interacting with an environment. In recent years, RL algorithms have gained popularity for Type 1 Diabetes Mellitus (T1DM) control [119]. RL involves training an agent to maximize returns from a reward signal [112], with the environment modeled as the patient's body and actions (e.g., insulin dosage) affecting blood glucose levels.

We can distinguish two fundamental RL approaches: model-based and model-free. Whereas model-based RL uses a model of the environment for planning actions [34], model-free RL learns directly from experience, requiring more data but being more flexible. Recent efforts favor model-free policy gradient algorithms like Trust Region Policy Optimization (TRPO) [101] or PPO [102] (for a short review see Appendix A), showing promising results in simulations [3, 124, 125]. PPO, in particular, stands out due to its robust and efficient training performance, characterized by stability and sample efficiency, making it well-suited for complex, dynamic environments, as shown by recent applications in natural language processing, such as fine-tuning language models like ChatGPT, effectively balancing exploration and exploitation to achieve outstanding results [93, 137].

5.2 The Dual PPO Framework

In the present study we introduce a novel strategy using a dual-controller system with two independent PPO agents, each trained in parallel on the same simulated patient but operating within its own state and action spaces, in order to reduce the training time while improving sample efficiency. An insulin cap constraint limits the maximum insulin delivery, and a safe control mechanism prevents insulin administration if blood glucose drops below a threshold, reducing hypoglycemia risk. Smaller insulin doses are administered over extended periods to mitigate risks associated with large bolus injections and there is no requirement for CHO consumption data (hybrid closed-loop).

Architecture of the Dual PPO System The proposed Dual Proximal Policy Optimization (PPO) framework uses two PPO agents, each pre-trained on the whole glycemic range for a given patient, but operating in distinct glycemic regions bounded through personalized insulin caps. Both the region boundaries and the cap values are model hyperparameters optimized through a grid search. The structure and interaction of the agents are detailed.

5.3 Methodology

In this study, we develop a real-life usable control system for optimizing insulin administration in patients with T1DM, based on two PPO agents, which we call Dual PPO (Fig. 5.1). In our model, we employ two distinct agents, each responsible for administering a continuous quantity of insulin. These agents operate under different constraints, i.e., each agent differs in the maximum rate of insulin delivery allowed. The first agent is subject to a higher cap, denoted as Cap_H, whereas the second operates under a lower cap, Cap_L, which is less than Cap_H. The transition between the two agents is governed by the CGM value related to a predetermined patient-specific *transition threshold*. When the CGM value crosses this threshold, the control shifts between the two agents, ensuring that only one agent is active at any given time. Furthermore, a safe-control mechanism is implemented to handle hypoglycemia by interrupting the administration of insulin in dangerous situations, which takes action if the glycemic index drops below 90 mg/dL (*safety threshold*).

In brief, the proposed system acts in three distinct regions based on the current CGM value:

- **High-Cap region**: above the *transition threshold* (a patient-specific value, typically around 160-170 mg/dL), roughly marking the passage to the hyperglycemic condition, where the first agent operates;
- **Low-Cap region**: between the *safety threshold* and the *transition threshold*, in the euglycemic range, where the second agent operates;
- **Safe-Control region**: under the *safety threshold* immediately above the hypoglycemic region, where the insulin injection from the controller is suspended.

To accomplish this goal, we used the UVA/Padova simulator to generate *in silico* data for 10 T1DM adult patients. The simulator was configured with Insulet, a popular insulin pump system, and a CGM sensor that reads blood glucose level every 3 minutes.

A grid search is then performed to find the best hyperparameters (caps and threshold) combination in terms of TIR for the two PPO agents. Finally, the Dual PPO system is evaluated on the patients.

5.3.1 State Space

In our research, we confront the inherently difficult challenge of accurately representing the complete state of a T1DM patient. Given the multitude of factors that can influence a patient's condition, achieving a comprehensive representation is essentially unattainable in both

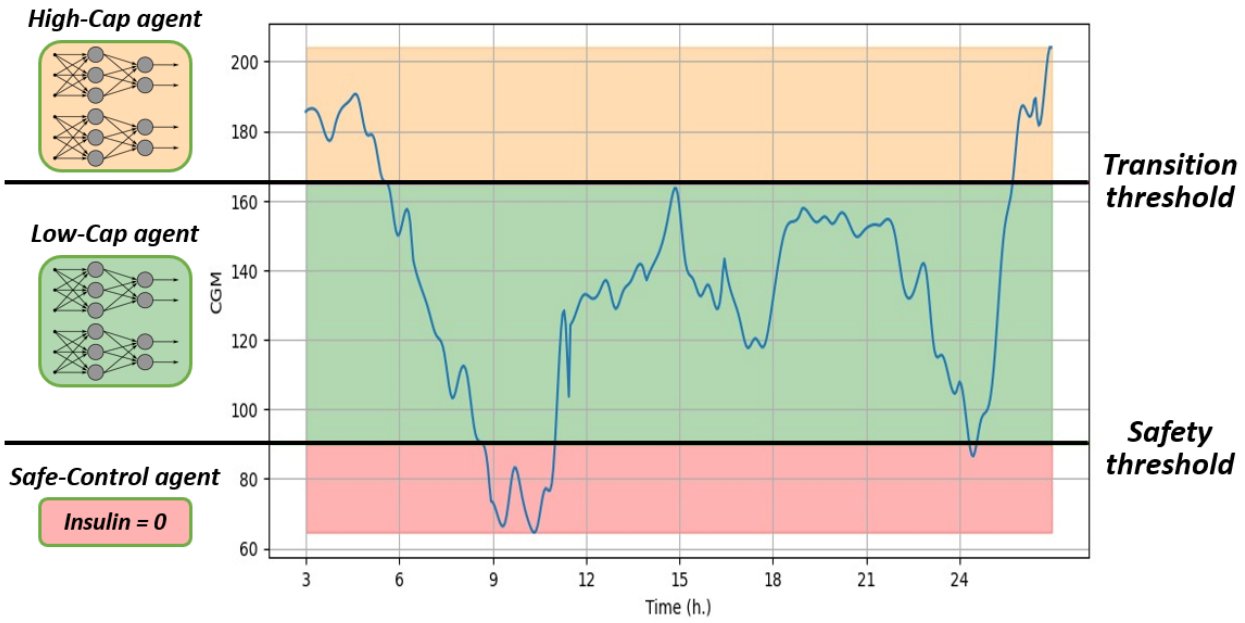


Figure 5.1: Structure of the proposed system based on two PPO agents (Dual PPO). One of the two agents operates in the High-Cap region when the glycemic curve is above the transition threshold, while the other operates in the Low-Cap region (between the transition threshold and the safety threshold). The safe-control mechanism, acting below the safety threshold prevents the administration of insulin in the Safe-Control region.

real-life and simulated scenarios. To face this complexity, we utilize a Partially Observable Markov Decision Process (POMDP) framework. This approach is more suited to our context than a fully observable Markov Decision Process (MDP), as it effectively accommodates the intrinsic uncertainties and limited observability associated with this problem.

In classical RL, the environment is typically modeled as a MDP, described as a set of states (S), actions (A), a transition function (P), and a reward function (R). It assumes full observability, where the agent knows its current state entirely and aims to maximize cumulative reward through a policy (π). However, in real-world scenarios like T1DM management, a POMDP is more appropriate due to the partial observability of the patient's state, where full state observability is not possible, including set of observations (Ω) and an observation function (O) that account for the uncertainties and partial information inherent in blood glucose levels influenced by various factors. This allows for robust insulin dosing decisions in a complex and partially observable environment. For this reason, at least formally, the problem must be formulated as a POMDP, extending MDP to handle partial observability.

For this reason, our selection of patient features was guided by the objective of maximizing informational richness while minimizing reliance on patient input. In the simulation, each timestep corresponds to a three-minute interval in real time. This temporal resolution

was chosen to match the minimum temporal resolution of our CGM sensor. The chosen observable features are:

- CGM(t): Continuous Glucose Monitoring value at time t
- $\frac{d\text{CGM}}{dt}$: The discrete derivative of CGM with respect to time t
- Time Slot: Categorical variable that cycles during the day and identifies a specific 2-hour interval
- IOB(t): Insulin On Board at time t
- Meal Announcement: a boolean variable indicating food intake at a specific timestep

CGM(t) and its discrete derivative $\frac{d\text{CGM}}{dt}$ represent the minimum information necessary to capture the dynamics of the evolution of glucose in the patient's blood - respectively the value and its future trend. The Time Slot variable - twelve 2-hour time intervals into which we have divided the day - represents a rough indicator for our algorithm for learning to distinguish the type of daily meal (breakfast, lunch, dinner or some snacks). The Insulin On Board (IOB) at a given time t is defined as the sum of the insulin action remaining from all previous insulin doses:

$$IOB(t) = \sum_{k=0}^{T_{\max-1}} a(k)I(t-k) \tag{5.1}$$

where $a(k) = \frac{T_{\max}-k}{T_{\max}}$ is the decaying curve of the insulin action, $T_{\max} = 180$ min. (3 hours for the Insulet pump we chose for our experiment), and $I(t)$ represents the amount of insulin infused by the pump at time t, obtained from the insulin delivery data.at time t. It represents the amount of insulin that is still active at time t, taking into account the insulin action decaying curve.

Finally, the meal announcement is a boolean variable that signals the start of a meal, and is the only variable that requires even minimal manual intervention by the user. However, recent advancements in machine learning have led to the development of automated meal detection algorithms (e.g. [95, 87]) and it is feasible that our system could be enhanced through integration with such automated meal detection algorithms.

5.3.2 Action Space

The action space for any given agent capable of administering insulin lies within a continuous interval $[0, Cap]$ expressed in Insulin Unit (IU), where Cap represents the maximum insulin rate, that is the maximum amount deliverable by the insulin pump in the time unit (equal to 1 minute for most insulin pumps), and will be on the order of 10^{-2} to 10^{-1} IU per minute.

As stated in the introduction, we administer small insulin doses over a prolonged period, avoiding larger bolus injections to mitigate the risk of hypoglycemia. Nonetheless, it is worth noting that the set of potential cap values remains greater than the minimum basal insulin increment for commercial pumps, which is on the order of 10^{-3} IU per minute.

5.3.3 Reward Function

In the development of reinforcement learning (RL) models for artificial pancreas systems, the design of reward functions plays a crucial role in shaping agent behavior and ensuring safe and effective glycemic control. For such a reason we have investigated and compared two different reward mechanisms, Parabolic and Magni, comparing the results obtained at the end of this study.

Parabolic This function has been chosen in order to give a positive reward to an agent capable of maintaining the CGM within the euglycemic range (e.g. in Del Giorno et. al. [34]), and at the same time penalizing it with negative rewards when in hypoglycemic or hyperglycemic regimes, while keeping it as simple as possible.

So we have defined the reward function as a quadratic function of CGM values expressed in mg/dL:

$$R_{Parabolic} = -R_0 \cdot (\text{CGM} - 70) \cdot (\text{CGM} - 180) \tag{5.2}$$

where R_0 is an arbitrary positive constant (from which RL algorithms do not depend and that for our convenience we set equal to $R_0 = 0.1$).

Magni For this case the reward function has been derived from the Magni risk function [17] which has been extensively utilized in simulation studies and clinical trials to evaluate glycemic outcomes:

$$\text{Risk(CGM)} = 10 \cdot \Big(1.509 \cdot ((\ln(\text{CGM}))^{1.084} - 5.381)\Big)^2,$$

Then, we have defined a Magni reward function as in [117]:

$$R_{Magni} = 1 - \frac{\text{clip}_{\epsilon_1,\epsilon_2}\left(\text{Risk(CGM)}\right)}{7.75},$$

where:

$$\text{clip}_{\epsilon_1,\epsilon_2}(x) = \min(\epsilon_1, \max(\epsilon_2, x)),$$

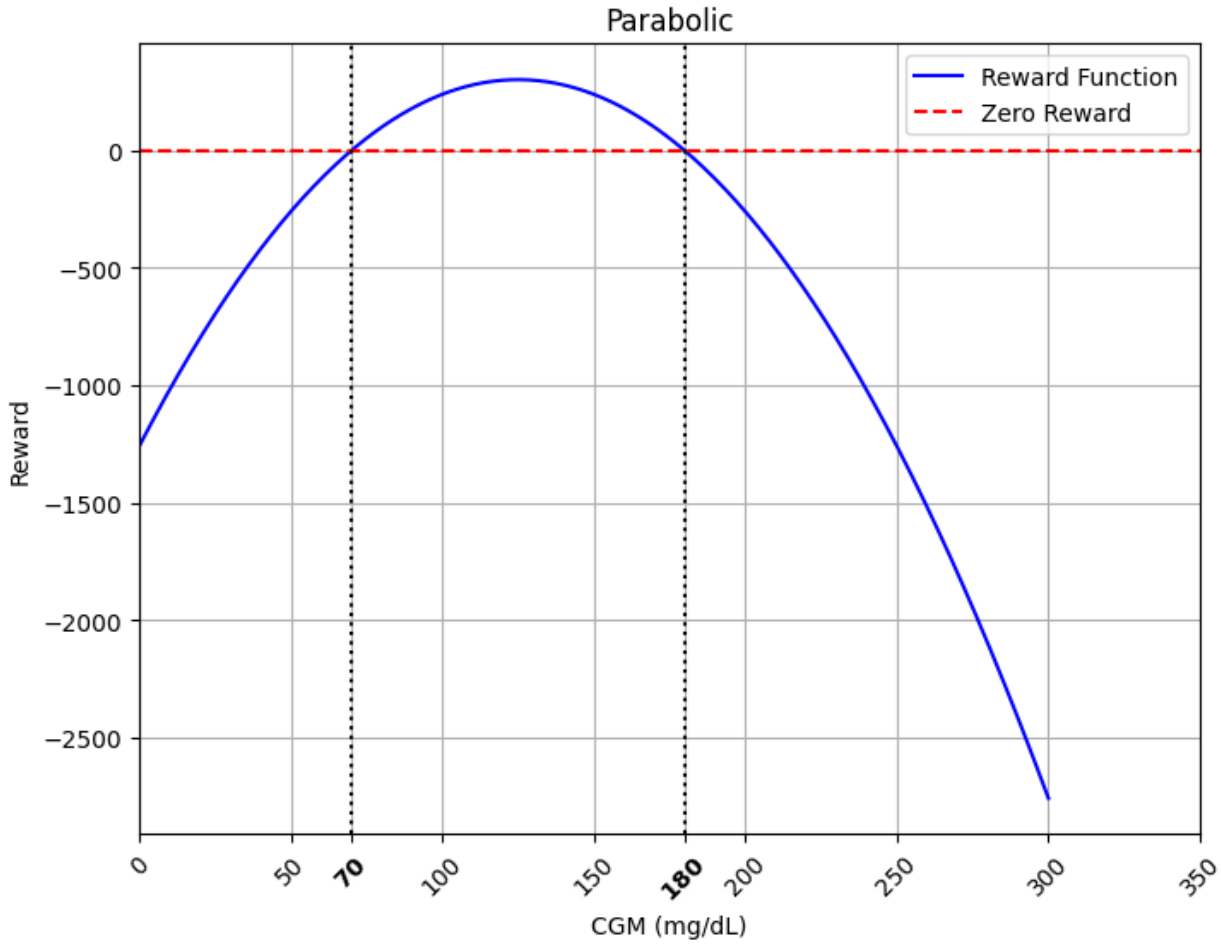


Figure 5.2: Parabolic reward function

$$\epsilon_1 = 15.5, \quad \epsilon_2 = 0.$$

It introduces a more clinically interpretable reward mechanism, penalizing much more hypoglycemia (< 70 mg/dL) than hyperglycemia, with a constant negative reward value of -1 in hypoglycemic regions as shown in 5.3.

Together, these reward functions offer complementary insights into glucose regulation: the parabolic function emphasizes simplicity and smooth penalization across ranges, while the Magni risk function integrates clinical relevance and real-world safety thresholds.

5.3.4 Dataset

The study included 10 *in silico* adult subjects with T1DM. To gather data, various simulations were conducted using the Simglucose library [120], specifically designed for modeling patients with diabetes, and the UVA/Padova software, an useful alternative to animal trials

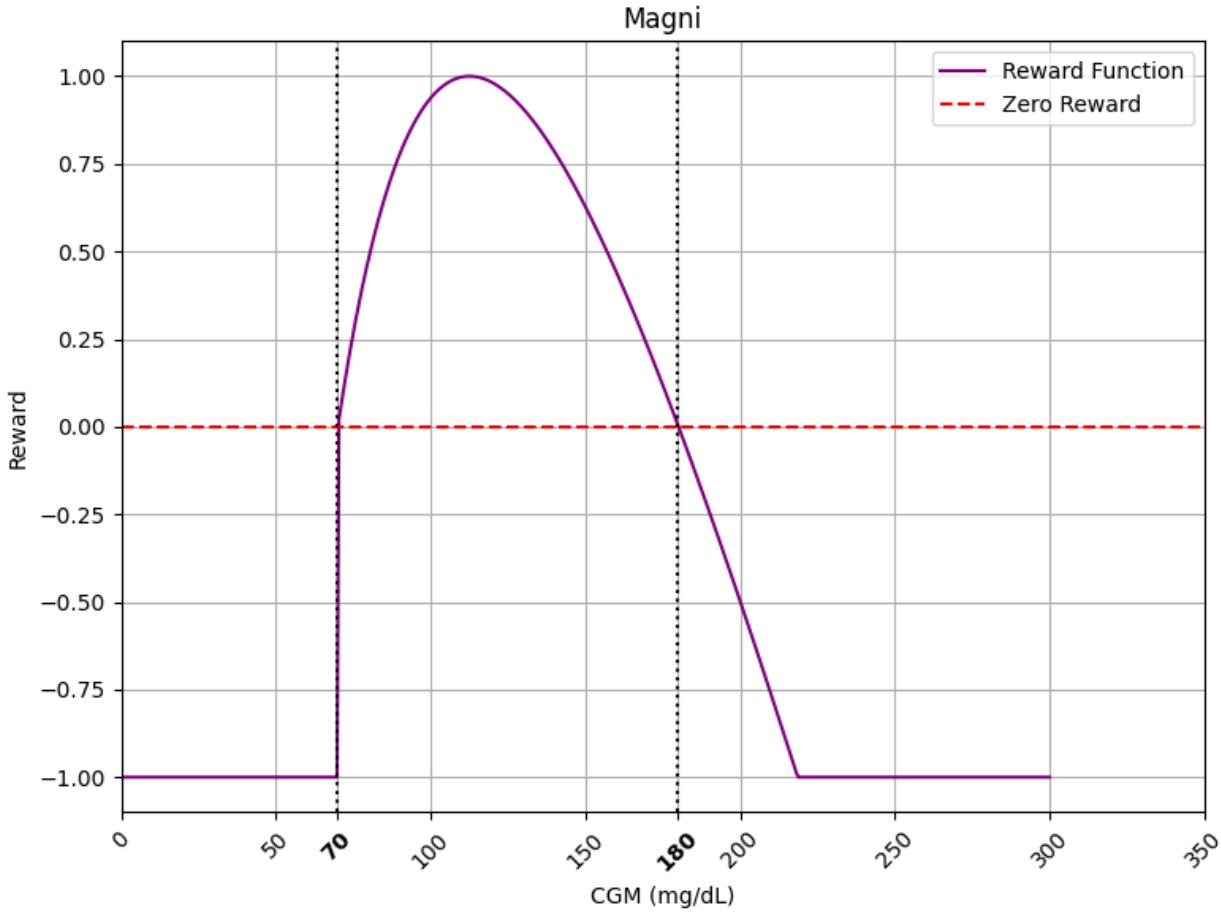


Figure 5.3: Magni reward function

for preclinical testing of insulin treatments which received FDA approval in 2008 [22]. The simulator is capable of accurately representing the glucose-insulin dynamic model of an *in silico* population, encompassing adults, adolescents, and children. We focused on simulating adult individuals due to the availability of reliable joint parameter distributions at the initial release of the simulator, based on a nondiabetic adult population [22].

To simulate real-world food intake scenarios, we developed a function that creates 100 random scenarios of food intake, each spanning a period of five days in order to match the test duration for each run. For each kind of meal (breakfast, lunch, dinner, two snacks) the function generates with a certain probability (Prob. (%), Table 5.1) a food intake that is taken from a normal distribution, with mean equal to a fixed share (CHO_{Meal}) of daily carbohydrates and its own meal-dependent standard deviation (σ_{CHO}) in according to the Dietary Reference Intakes for Carbohydrate [127]. Even the meal time is decided extracting it from a normal distribution, with a meal-dependent mean and standard deviation (respectively Time and σ_{Time} columns in Table 5.1).

Table 5.1: Probability, mean and standard deviation of meal occurrence and mean and standard deviation of CHO intake for random scenario generation.

Meal	**Prob. (%)**	**Time (h.)**	$\sigma_{\mathbf{Time}}$ **(h.)**	$\mathbf{CHO}_{Meal}$ **(g.)**	$\sigma_{\mathbf{CHO}}$ **(g.)**
Breakfast	60	08:00	1.5	34	7.5
Lunch	99	13:00	0.5	104	22.5
Snack 1	30	17:00	1	12	2.5
Dinner	95	21:00	1	80	17.5
Snack 2	3	24:00	1	12	2.5

The function randomly selects whether or not each meal will be included in the scenario based on predefined probabilities. The function returns a list of tuples, where each tuple contains a time stamp representing meal time and the corresponding amount of carbohydrates consumed. An example of meal distribution for a randomly generated scenario is shown in Figure 5.4.

Workflow In our workflow (Figure 5.5) virtual patients are generated through the UVA/Padova simulator. These simulations are utilized to train two sets of twelve distinct PPO agents, differentiated by unique insulin dosage limits, referred to as *Caps*.

The central innovation of this study is the development of a dual-agent control system, specifically engineered for diabetes management, employing a PPO agent optimized for hyperglycemic conditions, another PPO agent optimized for managing euglycemic conditions, and a safety mechanism to halt insulin delivery in hypoglycemic conditions.

System parameters such as insulin dosage caps and transition thresholds between euglycemic and hyperglycemic management have been optimized through a comprehensive grid search across the simulated patients. Then the dual-agent system's effectiveness has been evaluated over a 5-day virtual trial, recording the time each patient spent in various glycemic conditions.

Additionally, the study conducted an ablation test using a single-agent setup to bench-

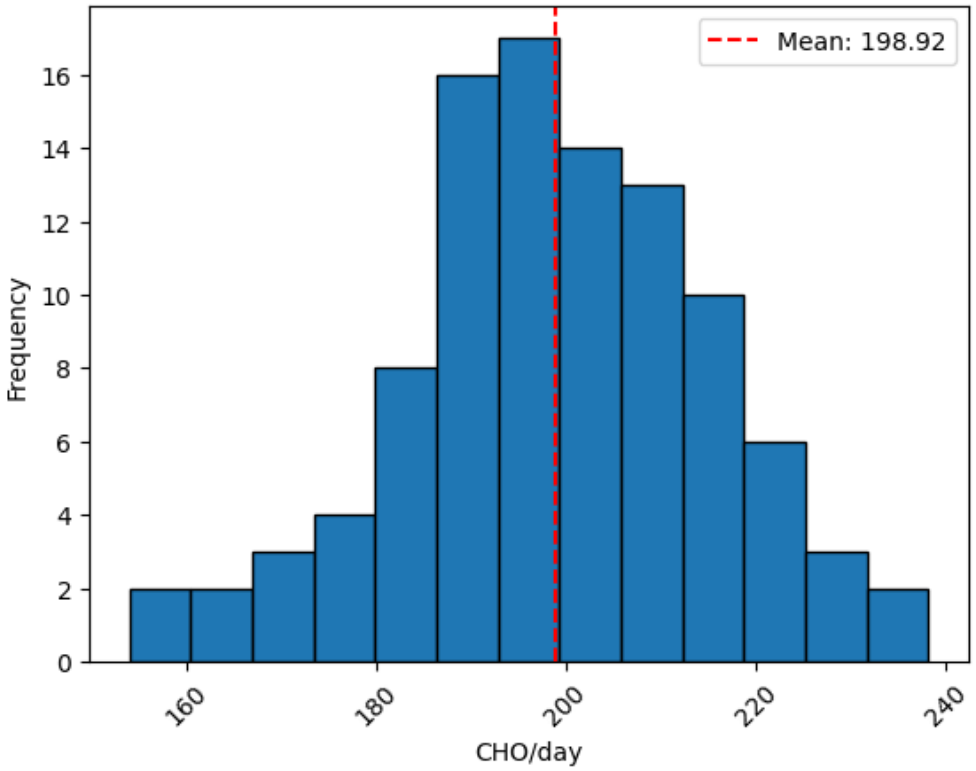


Figure 5.4: Histogram of CHO/day in grams for 100 generated scenarios.

mark its performance against the dual-agent approach and against other classical strategies present in the literature.

Experimental setup The study was conducted using Spyder, an open-source integrated development environment (IDE) usually used for scientific programming in Python language. As introduced in section 5.3.4, a simulation of the patient is generated via the Simglucose library to train a class of PPO agents (*pre-trained models*) through the OpenAI Gym RL library [9]. Then, following grid search optimization of the two caps and the *transition threshold*, we set up the Dual PPO system for validation on the virtual replica of the patient for a 5-days period, providing an analysis of the time spent in the euglycemic condition or out of it.

Pre-trained models For each patient, $N = 12$ different PPO agents were trained, all equal to each other except for a different Cap_i value, so that each was characterized by a different action space (Cap_i ranging from 0.04 to 0.15 IU, where $i = 1, 2, ..., N$). Hence, for each patient we obtained 12 models, with different caps, trained for 1024 timesteps (approximately two days, a timestep is the equivalent of 3 minutes in the real world), to

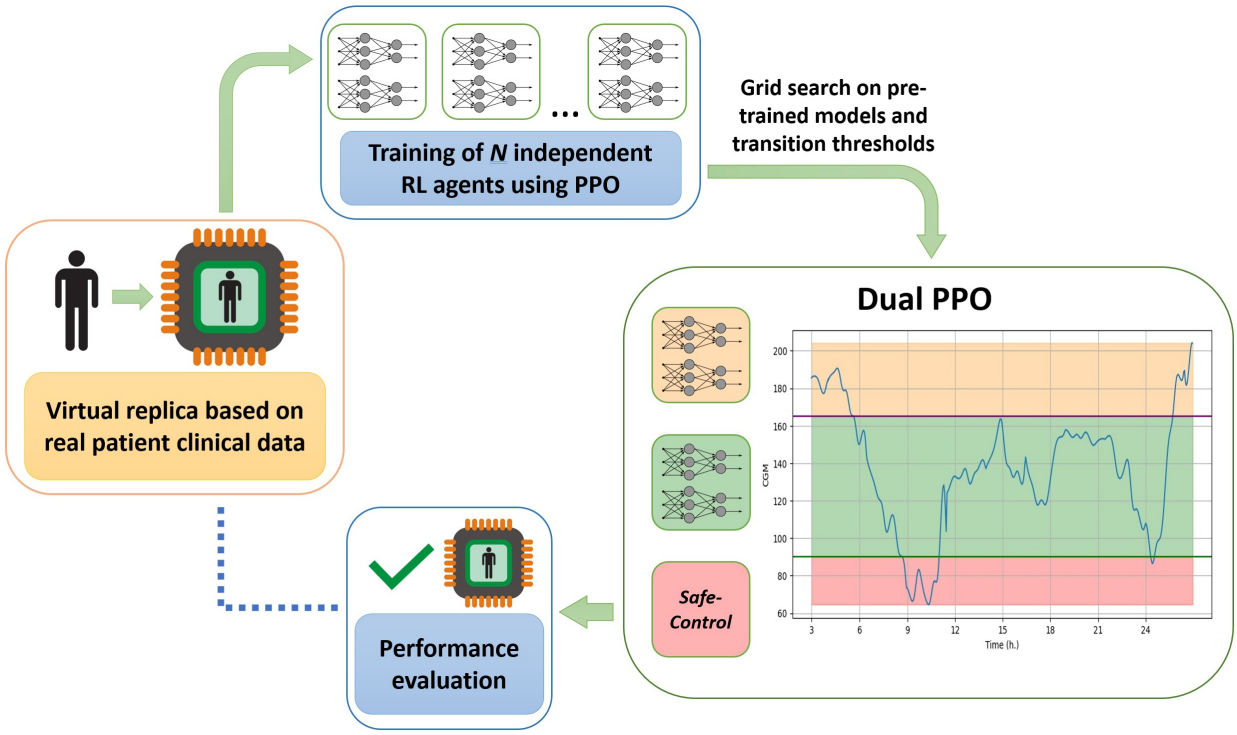


Figure 5.5: A comprehensive workflow depicting the proposed methodology. The process starts from a virtual replica of a patient, and involves a grid search on pre-trained models and transition thresholds. After the implementation of Dual PPO, the evaluation of performance on the virtual replica is conducted.

ensure that the agent had sufficient time to learn the optimal insulin dosage. Each model was trained under different random seeds to ensure the robustness of the learning process and mitigate the impact of randomness in parameter initialization.

The PPO algorithm introduces several hyperparameters that control its behavior besides the number of interactions (timesteps). These hyperparameters include the discount factor γ, the Generalized Advantage Estimation (GAE) factor λ, the number of epochs for the optimization step, the learning rate, the clipping parameter ϵ, and the value function coefficient and entropy coefficient for the loss function.

In our work, we found that the default hyperparameters provided by the Stable Baselines3 library [99] were sufficient for our needs, and we did not need to perform extensive hyperparameter tuning. This is in line with the findings of the original PPO paper [102], which showed that PPO's performance is relatively robust to the choice of hyperparameters. The hyperparameters used for training the PPO agents are presented in Table 5.2.

Regarding the architecture, The PPO agents employ both for the actor and the critic neural networks an architecture with two hidden layers, each containing 64 neurons, as the function approximator. This default configuration was chosen to balance model performance

and computational efficiency.

Table 5.2: List of hyperparameters for PPO training.

Hyperparameters	**Value**
Interactions	1024
Batch size	64
Num. of epochs for surrogate loss optimization	10
Learning rate	3 x 10^{-4}
PPO clip range (ϵ)	0.2
Discount factor (γ)	0.99
GAE (generalized advantage estimator) (λ)	0.95
Value function loss coeff.	0.5
Max gradient clipping value	0.5

5.3.5 System Architecture and Optimization

The idea behind our proposal is to set up a patient-specific dual-agent control system built on:

- A first PPO agent, whose insulin delivery action space is $[0, Cap_H]$, expert in handling hyperglycemic scenarios where the glycemic index exceeds the *transition threshold*
- A second PPO agent, whose insulin delivery action space is $[0, Cap_L]$, expert in handling euglycemic scenarios where the glycemic index falls beetwen a *safety threshold* and the *transition threshold*
- A safety-control mechanism to prevent insulin administration when the glycemic index drops below the *safety threshold*

where the *safety threshold* has been fixed at 90 mg/dL, Cap_H and Cap_L represent the maximum insulin that can be administered by the insulin pump in the time unit respectively by the first and the second PPO agent and where holds the condition $Cap_H > Cap_L$ in order to guarantee that in the High-Cap region - characterized by higher blood glucose - the first PPO agent is able to deliver more insulin than the second PPO agent.

In order to account for interindividual variability among patients, we optimized the architecture of our model through a personalized grid search. Specifically, for each patient, we determined the optimal values of three hyperparameters - Cap_H (maximum insulin rate for the high-cap agent), Cap_L (maximum insulin rate for the low-cap agent), and the transition threshold — averaged over 50 runs. Please note that while the *safety threshold* is the same for all patients and equal to 90 mg/dL, the *transition threshold* is a hyperparameter that can vary from patient to patient.

To evaluate and select the optimal grid search results, we prioritized Time in Range (TIR) as the primary metric for success due to its established clinical relevance. TIR reflects the percentage of time a patient's blood glucose levels remain within the euglycemic range (70–180 mg/dL), which correlates strongly with reduced risks of long-term complications and overall improved quality of life for individuals with Type 1 Diabetes Mellitus. However, we recognize that TIR alone does not provide a complete picture of glycemic control and safety. Therefore, we also evaluated time spent in hypoglycemia (50–70 mg/dL), hyperglycemia (180–250 mg/dL), severe hypoglycemia (<50 mg/dL), and severe hyperglycemia (>250 mg/dL) as defined in previous works [45] [116]. These additional metrics are crucial for capturing the system's ability to minimize both short-term risks, such as severe glycemic excursions, and broader glycemic variability. By combining TIR with these complementary metrics, we aimed to achieve a balanced evaluation of the model's efficacy and safety profile. We defined distinct conditions based on CGM values:

$$\begin{cases} \text{Severe Hypoglycemic:} & \text{if } CGM < 50 \text{ mg/dL} \\ \text{Hypoglycemic:} & \text{if } 50 \text{ mg/dL} \leq CGM < 70 \text{ mg/dL} \\ \text{Euglycemic:} & \text{if } 70 \text{ mg/dL} \leq CGM \leq 180 \text{ mg/dL} \\ \text{Hyperglycemic:} & \text{if } 180 \text{ mg/dL} < CGM \leq 250 \text{ mg/dL} \\ \text{Severe Hyperglycemic:} & \text{if } 250 \text{ mg/dL} < CGM \end{cases}$$

Hyperparameters were varied discretely in our grid search, with Cap values ranging from 0.04 IU to 0.15 IU (in steps of 0.01 IU) while strictly respecting inequality $Cap_H > Cap_L$,

and the *transition threshold* varying discretely from 150 mg/dL to 200 mg/dL in steps of 5 mg/dL, for a total grid dimension equal to $\binom{12}{2} \times 11 = 726$.

Table 5.3 shows the results obtained, with the optimal caps and thresholds for each patient.

We also conduct an ablation study executing a similar procedure with a single PPO approach to ascertain the benefits of employing a dual-agent system. In this case, for each patient one of the $N = 12$ corresponding models were used to build a single PPO control system with the task of handle all glycemic conditions. In addition, the same safe-control mechanism was included. Again, for each patient a grid search was performed for a total of 50 runs to determine which insulin cap was optimal. For single PPO approach, we found that the range for the optimal training caps is between 0.07 and 0.09 (Table 5.3).

For both approaches, after the best-performing models were identified, we evaluated them on each simulated patient for 5 days, repeating each test 100 times, and calculated the mean values with standard deviations of the time spent in euglycemic, hyperglycemic, severe hyperglycemic, hypoglycemic, and severe hypoglycemic conditions.

Statistical Analysis We conducted a Shapiro-Wilk test to assess whether each condition was normally distributed for each patient. Since for some conditions and patients the normality hypothesis was rejected (as shown in Figure 5.6), we subsequently applied a non-parametric ANOVA test (Kruskal-Wallis with Bonferroni correction) to all patients across all conditions to demonstrate statistically significant improvements between Single and Dual PPO approach.

5.4 Results and Discussion

In this section we present and discuss the outcomes achieved through the utilization of the proposed Dual PPO system. Additionally, a comparative analysis is conducted with a conventional single PPO approach as well as with traditional control methods such as BBC and PIDC. Each table reports on the columns tested patients and the percentage of TIR and of time spent in hyperglycemic, severe hyperglycemic, hypoglycemic and severe hypoglycemic conditions.

5.4.1 Dual PPO performance

The results of Dual PPO approach are shown in Table 5.4, with the last row showing the average performance. Our system achieves an average TIR of 69.30%, limiting time spent

Table 5.3: Optimal caps and thresholds for each patient for single PPO and Dual PPO approaches.

Patient	**Single PPO**	**Dual PPO**		
	Cap	**$\mathbf{Cap_H}$**	**$\mathbf{Cap_L}$**	**Transition Threshold**
#001	0.08	0.09	0.06	160
#002	0.08	0.14	0.08	165
#003	0.08	0.11	0.06	160
#004	0.07	0.09	0.05	165
#005	0.08	0.13	0.08	165
#006	0.09	0.15	0.07	170
#007	0.07	0.11	0.07	160
#008	0.07	0.10	0.06	160
#009	0.07	0.14	0.06	190
#010	0.07	0.14	0.07	160

in hypoglycemia ($< 6\%$) and severe hypoglycemia ($\sim 1.91\%$), conditions that are a more immediate concern in the short term.

Hyperglycemia percentage was recorded at 18.71%, while severe hyperglycemia percentage accounted for less than 5% of the time, indicating that the controller effectively prevents severe high blood sugar levels.

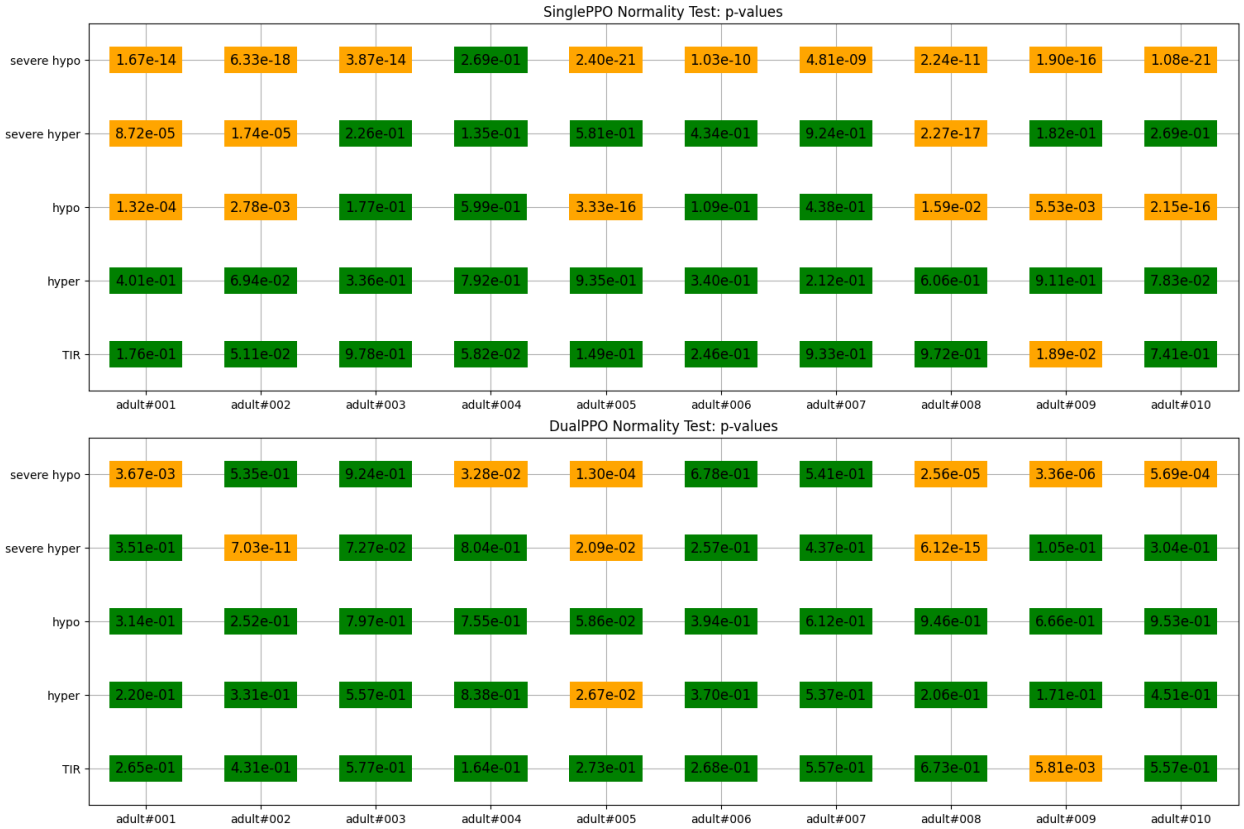


Figure 5.6: Results of normality tests (Shapiro-Wilk) for various conditions in SinglePPO and DualPPO treatments across different patients. Orange background indicates that the null hypothesis that the distribution is normal has to be rejected under the chosen level of significance (p-value < 0.05), while green background indicates that it cannot be rejected (p-value $>= 0.05$).

At the level of individual patients, the results show more variability, with patient #008 showing the best overall outcome, with a TIR percentage of 93.30%, hyperglycemia of 4.22%, and almost negligible severe hyperglycemia, while patient #009 exhibiting the highest percentages of severe hypo- and hyperglycemia (5.21% and 13.99% respectively).

5.4.2 Single PPO performance

Table 5.5 presents the results for the same patients using a single RL agent with a PPO algorithm. The overall TIR percentage was observed to be 61.69%. Hyperglycemia percentage was recorded at 21.30% and hypoglycemia percentage was observed to be 3.21%. Patients experienced severe hyperglycemia an high percentage of time (13.41%), whilst severe hypoglycemia percentage was recorded at 0.40%.

Table 5.4: Performance for each patient and overall mean with one standard deviation achieved on 100 run with Dual PPO approach.

Test	Severe Hypo (%)	Hypo (%)	TIR (%)	Hyper (%)	Severe Hyper (%)
#001	**0.15±0.10**	3.12±0.45	71.11±1.81	24.79±1.52	0.82±0.31
#002	0.25±0.10	5.24±0.54	86.61±1.14	7.85±0.82	0.04±0.05
#003	0.61±0.24	3.50±0.54	71.49±1.67	22.79±1.13	1.62±0.50
#004	2.50±0.43	8.73±0.70	62.85±1.52	17.21±0.73	8.72±0.78
#005	0.22±0.12	2.58±0.46	76.48±1.44	20.30±1.06	0.42±0.18
#006	4.65±0.68	7.63±0.54	**49.88±1.50**	24.97±1.11	12.88±1.40
#007	4.97±0.59	**10.66±0.62**	64.54±2.12	15.11±0.97	4.72±0.82
#008	0.16±0.10	**2.32±0.43**	**93.30±0.85**	**4.22±0.67**	**0.00±0.01**
#009	**5.21±2.95**	5.96±0.76	54.68±2.47	20.16±1.44	**13.99±1.38**
#010	0.36±0.14	4.24±0.49	62.08±1.55	**29.73±1.22**	3.59±0.65
	1.91±0.55	5.40±0.55	69.30±1.61	18.71±1.07	4.68±0.61

Dual PPO vs single PPO: performance comparison For a full comparison between Single and Dual PPO we reported box plots for all patients and different conditions (Figures 5.7, 5.8, 5.9, 5.10 and 5.11). Our proposed system shows a significant improvement in percentage TIR ($\Delta = +7.61\%$) as shown by p-values statistical analysis for a 0.05 significance level (see below), as well as in severe hyperglycemia (Δ = -8.73%), with only a minimum increase in severe hypoglycemia ($\Delta = +1.51\%$).

To prove this we have done a Kruskal-Wallis statistical test and calculated the p-values on 100 run for each condition and patient (on the left in Figure 5.12 is reported the statistic H for the test, on the right the p-values), showing improvements for the Dual PPO controller

Table 5.5: Performance for each patient and overall mean with one standard deviation achieved on 100 run with Single PPO approach.

Test	Severe Hypo (%)	Hypo (%)	TIR (%)	Hyper (%)	Severe Hyper (%)
#001	0.10±0.18	2.32±1.40	67.89±6.48	24.37±4.70	5.32±3.85
#002	0.04±0.10	1.68±0.97	78.93±3.88	16.95±3.26	2.40±1.85
#003	0.21±0.37	3.10±1.24	63.28±4.92	25.37±4.18	8.04±3.82
#004	**2.47±1.17**	**11.37±1.68**	54.41±4.46	15.02±2.52	16.73±3.35
#005	**0.01±0.03**	0.12±0.27	59.04±4.53	**30.47±4.05**	10.36±4.67
#006	0.33±0.43	3.26±1.26	**45.47±5.37**	21.67±4.18	**29.27±6.84**
#007	0.39±0.44	3.93±1.36	57.24±5.55	23.25±4.17	15.19±5.02
#008	0.35±0.47	4.32±1.56	**87.06±4.81**	**8.08±3.94**	**0.19±0.4**
#009	0.07±0.15	1.89±1.24	49.36±4.89	25.64±4.89	23.04±6.51
#010	**0.01±0.04**	**0.09±0.21**	54.21±3.84	22.19±3.38	23.50±5.09
	0.40±0.11	3.21±0.35	61.69±1.54	21.30±1.24	13.41±1.31

compared to the Single PPO controller for the TIR condition across all patients, and for a large number of patients for the remaining conditions.

Specifically, enhancements are observed in TIR for all patients, with p-values indicating high statistical significance (P<.001 for most patients). In addition to TIR, almost all other conditions show significant improvements (see Figure 5.12). For instance, the incidence of hypoglycemia and severe hyperglycemia are markedly reduced with Dual PPO, with p-values consistently below .001, highlighting the robustness of the Dual PPO controller in managing these conditions.

It is noteworthy that for a few specific conditions and patients, the p-values did not

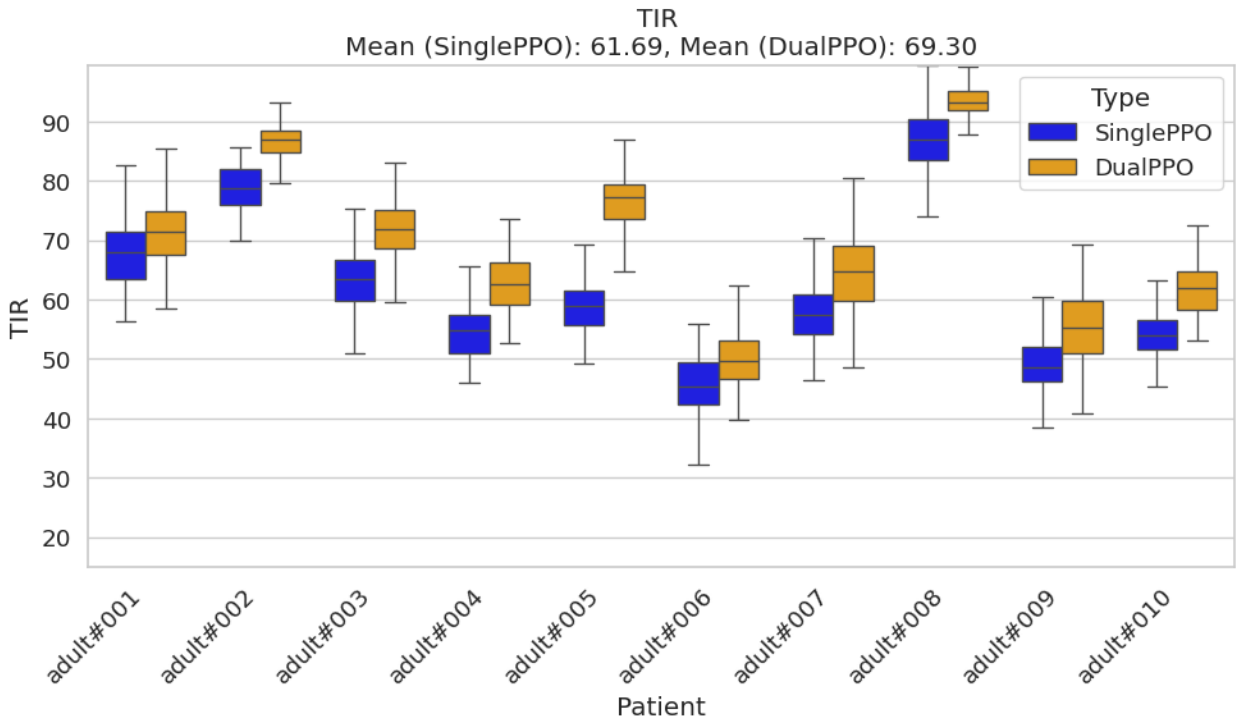


Figure 5.7: TIR condition comparison: Single PPO vs. Dual PPO.

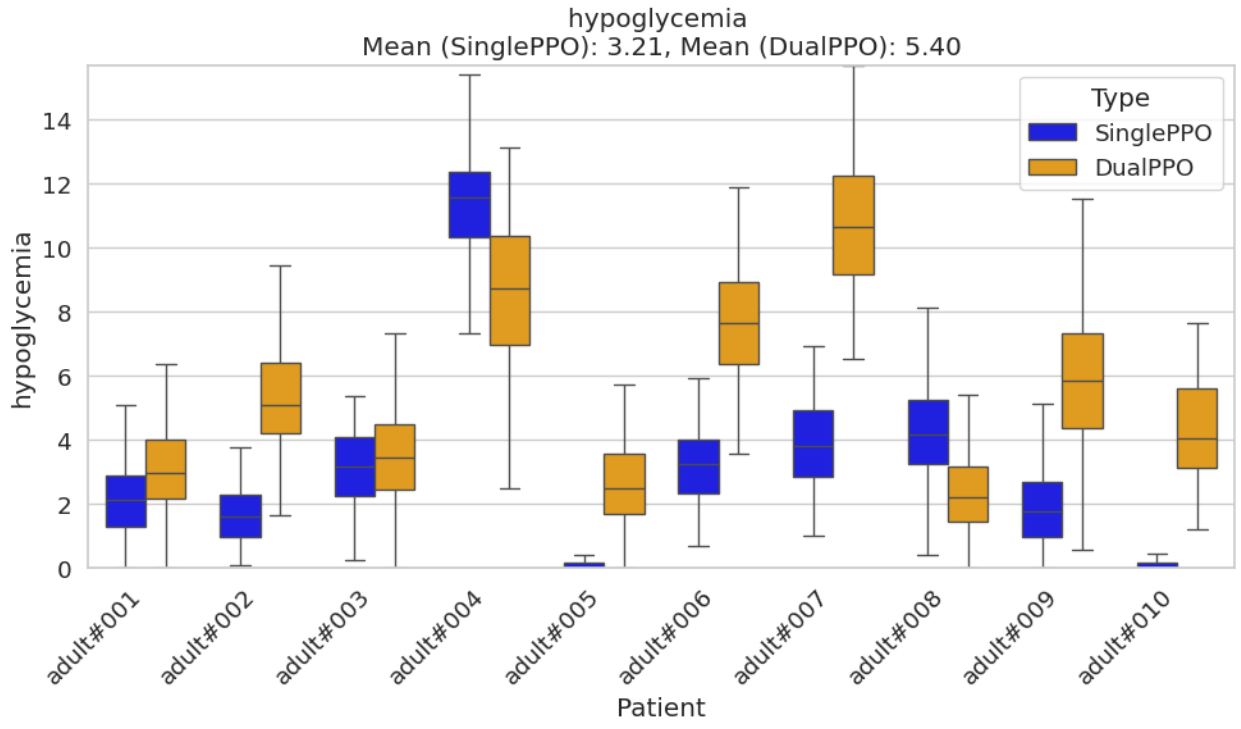


Figure 5.8: Hypoglycemic condition condition comparison: Single PPO vs. Dual PPO.

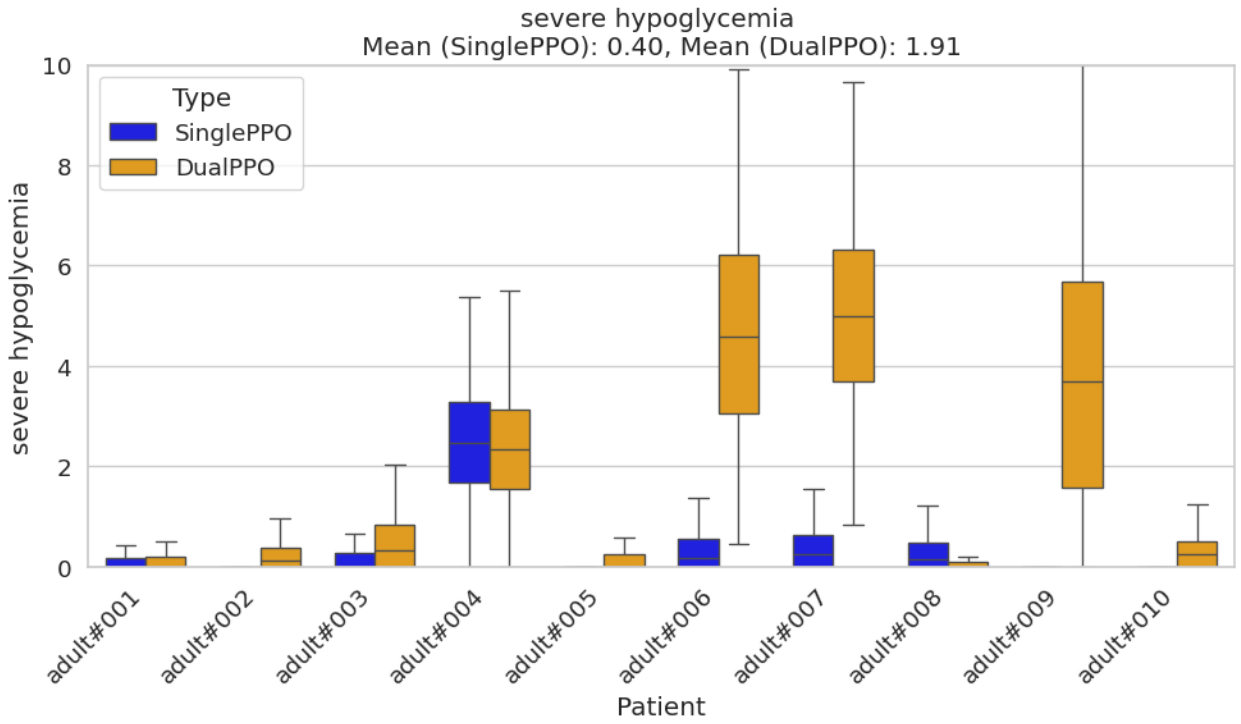


Figure 5.9: Severe hypoglycemic condition comparison: Single PPO vs. Dual PPO.

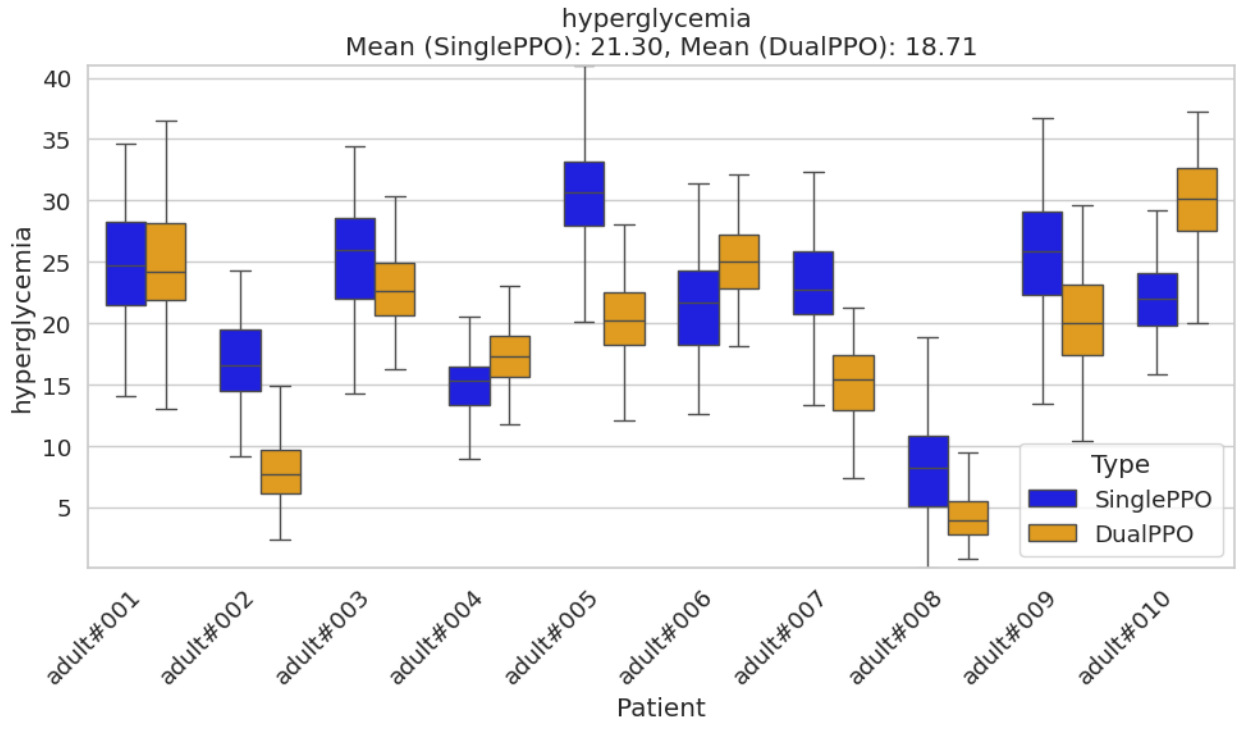


Figure 5.10: Hyperglycemic condition comparison: Single PPO vs. Dual PPO.

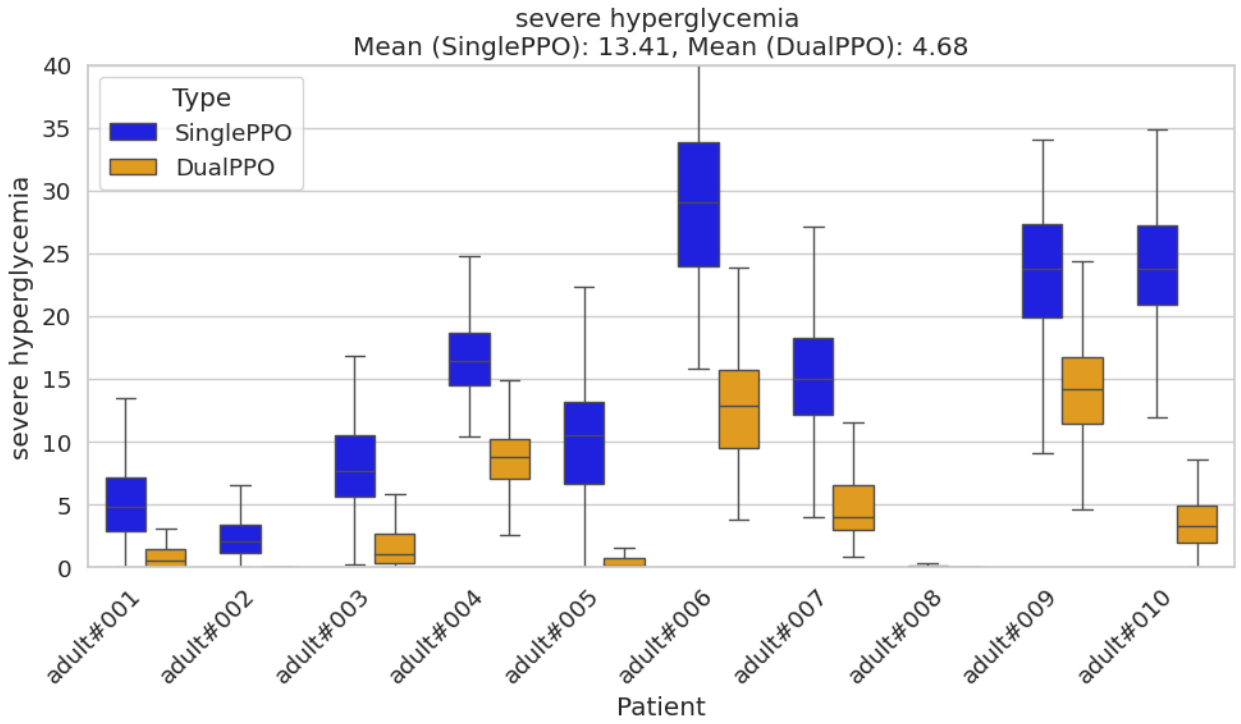


Figure 5.11: Severe hyperglycemic condition comparison: Single PPO vs. Dual PPO.

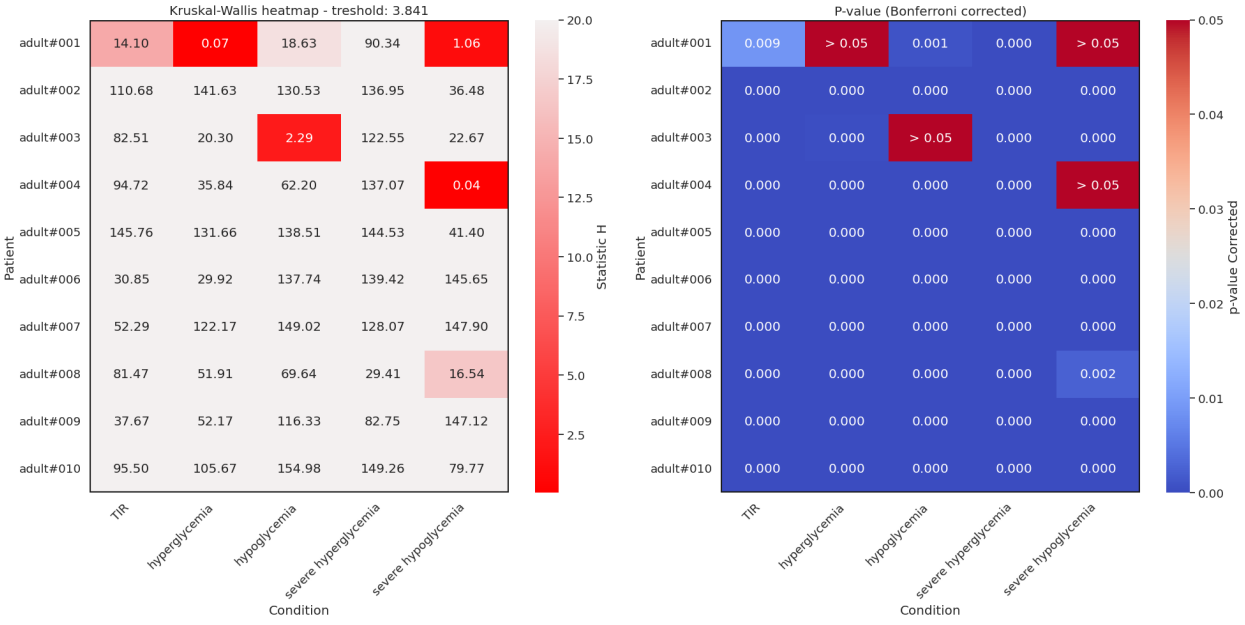


Figure 5.12: Kruskal-Wallis and p-value statistics comparing Single PPO vs. Dual PPO approach.

reach statistical significance. For example, Patient #001 showed a p-value greater than 0.05 for Severe Hypoglycemic and for Hyperglycemic range, Patient #003 for Hypoglycemic.

Similarly, Patient #004 had a p-value beyond the 0.05 treshold for Severe Hypoglycemic.

These deviations, however, can be understood considering the different action spaces for the two PPO agents, the first for the High-Cap region, the second for the Low-Cap region, characterized by two different insulin caps Cap_H and Cap_L, where $Cap_H > Cap_L$. When the CGM shifts from the High-Cap region to the Low-Cap region, control switches from the first PPO controller trained with a broader action space - useful for managing the hyperglycemic zone - to the other, dual PPO controller trained with the dual action space - specialized in managing euglycemic and hypoglycemic conditions. However, this may result in increased IOB and then, for patients with higher insulin sensitivity, may lead to a slightly longer stay in the hypoglycemic condition compared to a single PPO controller.

Similar findings with other controllers have been reported in other studies [122, 34], indicating a distinct patient's responsiveness to insulin administration, leading to increased glycemic variability of some patients compared to the rest of the adult cohort.

It suggests that enhancing control in one region may inadvertently affect the other, indicating the need for a finely-tuned insulin delivery strategy. This delicate balance is an evidence of the intricate nature of T1DM management and the sophistication required in designing effective control algorithms.

5.4.3 Comparison with classical methods

In addition to the ablation study above comparing a single PPO approach with our proposed system, we report in table 6.2 the comparison with classical approaches such as BBC and PIDC, with the performance averaged over all patients.

Table 5.6: Comparison of the proposed system overall results with single PPO, BBC controller and PIDC.

Controller	**System**	**Reward Function**	**Severe Hypo (%)**	**Hypo (%)**	**TIR (%)**	**Hyper (%)**	**Severe Hyper (%)**
Dual PPO	closed-loop	parabolic	1.91±0.55	5.40±0.55	69.30±1.61	18.71±1.07	4.68±0.61
Single PPO	closed-loop	parabolic	0.40±0.11	3.20±0.35	61.69±1.54	21.30±1.24	13.41±1.31
Dual PPO	closed-loop	Magni	1.95±0.42	5.53±0.52	64.23±1.55	18.42±0.98	9.87±1.08
Single PPO	closed-loop	Magni	0.23±0.08	2.42±0.34	62.42±1.65	22.25±1.29	12.68±1.31
BBC	open-loop	N/A	1.62±4.23	3.03±0.80	88.35±3.16	6.68±0.71	0.32±0.31
PID	open-loop	N/A	75.92±9.49	2.01±0.72	15.06±2.73	3.81±1.32	3.20±1.15

The BBC still scores the best results in terms of TIR and other conditions, but with an open-loop system, a conventional approach which lacks automation and relies on the patient's manual entry of CHO intake during meals. Furthermore, it is noteworthy to observe that the percentage of time spent in severe hypoglycemia (1.62%) is only slightly lower than that of Dual PPO (1.91%).

The PIDC, instead, obtain the worst result with a TIR percentage of 15.07% and a severe hypoglycemia percentage of 75.92%.

The inability of the PIDC to effectively manage the glycemic curve of patients and the resulting poor performance could be attributed to the duration of the conducted tests (5 days), thus highlighting how extended management might necessitate a more sophisticated and personalized approach, such as the utilization of adaptive control algorithms.

Chapter 6

Multi-Agent Reinforcement Learning for Cooperative Insulin Delivery

The application of DRL and MARL to the management of Type 1 Diabetes (T1D) represents a growing area of research focused on enhancing glycemic control through advanced artificial intelligence methodologies. Over the past few years, several studies have contributed to this field, showcasing a progression of innovative strategies and techniques.

In 2020, Zhu et al. [136] introduced an insulin bolus advisor leveraging deep reinforcement learning and continuous glucose monitoring to optimize insulin dosing during meals. Their model, based on an actor-critic framework, demonstrated notable improvements in the mean time-in-range (70–180 mg/dL) for both adult and adolescent subjects with T1D. This foundational work underscored the potential of DRL in managing meal-related insulin therapy and set the stage for subsequent advancements in the field.

Building on this foundation, multiple studies emerged in 2023, marking significant strides in the use of reinforcement learning for T1D management. Jaloli and Cescon proposed a closed-loop framework for insulin administration tailored to patients undergoing multiple daily injection (MDI) therapy [56]. By employing a reinforcement learning agent based on the soft actor-critic algorithm, their approach achieved a substantial reduction in glycemic variability and increased the time patients remained within the target glucose range. In another contribution, Jaloli and Cescon [57] explored the potential of MARL in a system designed to provide personalized basal and bolus insulin administration recommendations. This system utilized a metabolic glucose model in conjunction with a soft actor-critic multi-agent framework, demonstrating improved glycemic control through reduced variability and increased time-in-range.

Lv et al. [71] further advanced the field by introducing a hybrid control policy for artificial

pancreas systems. Their approach combined model predictive control (MPC) with ensemble DRL policies to leverage the strengths of both methodologies, resulting in enhanced glycemic regulation for individuals with T1D. This innovative hybrid system illustrated the versatility and effectiveness of integrating different control strategies.

Another notable contribution came from El Fathi and Breton [38], who investigated the use of reinforcement learning to simplify meal-related insulin dosing. Their research focused on an RL agent capable of recommending optimal insulin doses based on qualitative meal strategies, thereby eliminating the need for precise carbohydrate counting. The in-silico experiments highlighted the system's ability to improve time-in-range while reducing hypoglycemic episodes, emphasizing the practicality of reinforcement learning in real-world scenarios.

Together, these studies illustrate the rapid evolution of reinforcement learning applications in T1D management, showcasing a trajectory toward increasingly personalized and effective therapeutic interventions. The integration of MARL and DRL approaches continues to offer promising solutions for addressing the complexities of glycemic control, paving the way for further advancements in the field.

Cooperative MARL has gained significant traction as a framework for solving complex, decentralized decision-making problems. Among the various paradigms, Centralized Training for Decentralized Execution (CTDE) stands out for its ability to harness the strengths of centralized information during training while enabling fully decentralized execution. This section explores the theoretical underpinnings, methodologies, and key contributions in this area, focusing on value function factorization and centralized critic methods applied to the T1D case study.

6.1 Partially Observable Environments in Multi-Agent Settings

Managing Type 1 Diabetes (T1D) requires dynamic and personalized decision-making to maintain optimal blood glucose levels. This challenge is inherently complex due to the uncertainties in physiological responses, external factors such as physical activity, and the delayed effects of insulin administration.

POMDP - Uncertainty of a single agent A POMDP captures the sequential decision-making problem under uncertainty by considering not only the dynamic nature of the environment but also the partial observability of the underlying state. Formally, a POMDP is

a framework for sequential decision-making under uncertainty [59] that extends the concept of MDP [98] and is defined by the tuple:

$$\langle S, A, T, R, \Omega, O, \gamma \rangle$$

where:

- S: A finite set of states with a designated initial state distribution Σ_0.
- A: A finite set of actions available to the agent.
- T: A state transition probability function $T : S \times A \times S \to [0, 1]$ that specifies the probability of transitioning from state $s \in S$ to state $s' \in S$ after taking action $a \in A$, i.e., $T(s, a, s') = \Pr(s'|s, a)$.
- R: A reward function $R : S \times A \to \mathbb{R}$, representing the immediate reward received for being in state $s \in S$ and taking action $a \in A$.
- Ω: A finite set of observations available to the agent.
- O: An observation probability function $O : \Omega \times A \times S \to [0, 1]$, specifying the probability of observing $o \in \Omega$ after taking action $a \in A$ and transitioning to state $s' \in S$, i.e., $O(o, a, s') = \Pr(o|a, s')$.
- $\gamma \in [0, 1]$: A discount factor for future rewards.

Dec-POMDP - Uncertainty of many agents The Decentralized Partially Observable Markov Decision Process (Dec-POMDP) generalizes the POMDP to a multi-agent setting. Formally, a Dec-POMDP is defined by the tuple:

$$\langle I, S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma \rangle$$

where:

- I: A finite set of agents, where $|I| = n$.
- A_i: A finite set of actions for each agent $i \in I$, with $A = \times_{i \in I} A_i$ as the set of joint actions.
- Ω_i: A finite set of observations for each agent $i \in I$, with $\Omega = \times_{i \in I} \Omega_i$ as the set of joint observations.

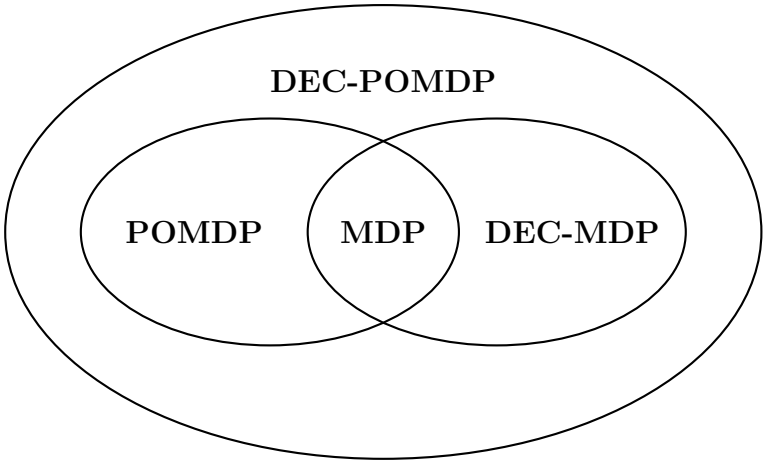


Figure 6.1: The relationships among the frameworks.

whereas S, T, R, O and γ are the same as for a POMDP.

The introduction of Dec-POMDP and their subset, Decentralized Markov Decision Process (Dec-MDP), extends traditional MDP frameworks to encompass scenarios where multiple agents act based on local observations, without access to the complete state of the environment or each other's information. While these models provide a formal structure to address real-world distributed decision-making problems, such as multi-robot coordination or operations in which multiple decision-makers are involved, they introduce a dramatic increase in computational complexity. Specifically, solving finite-horizon Dec-POMDP and Dec-MDP has been proven to be NEXP-complete [5], meaning these problems do not admit polynomial-time solutions and likely require doubly exponential time in the worst case. This complexity fundamentally differentiates decentralized problems from their centralized counterparts, which are solvable in polynomial or exponential time for finite horizons. Such results underscore the intrinsic difficulty of reducing decentralized planning problems to centralized formulations or directly adapting algorithms from POMDP. Nevertheless, recent advancements in reinforcement learning have demonstrated that algorithms such as Proximal Policy Optimization (PPO) can achieve strong performance in cooperative multi-agent settings, effectively addressing some of the computational challenges inherent in decentralized decision-making processes [125].

6.2 GLUMARL: a Multi-Agent RL framework for T1D

In this study, we propose GLUMARL, a novel MARL approach for the automated and adaptive management of insulin administration in Type 1 Diabetes Mellitus (T1DM). Our system relies on the cooperation of multiple agents trained together within a shared environment to

optimize insulin administration for individual patients, within the following framework:

- **Shared exploration**: Agents explore the environment together, learning to coordinate their actions based on the shared reward signal and feedback from the patient model.
- **Cooperative learning**: The agents adjust their policies simultaneously, with the goal of optimizing the collective glucose management. Through repeated episodes of interaction with the environment, the agents learn to balance insulin delivery in both postprandial (after-meal) and fasting states, as well as during hypoglycemic conditions.
- **Policy updates**: After each training episode, the agents' policies are updated based on the observed outcomes and rewards. PPO's clipping mechanism ensures that policy updates are stable and prevent overly large changes in the agents' actions.

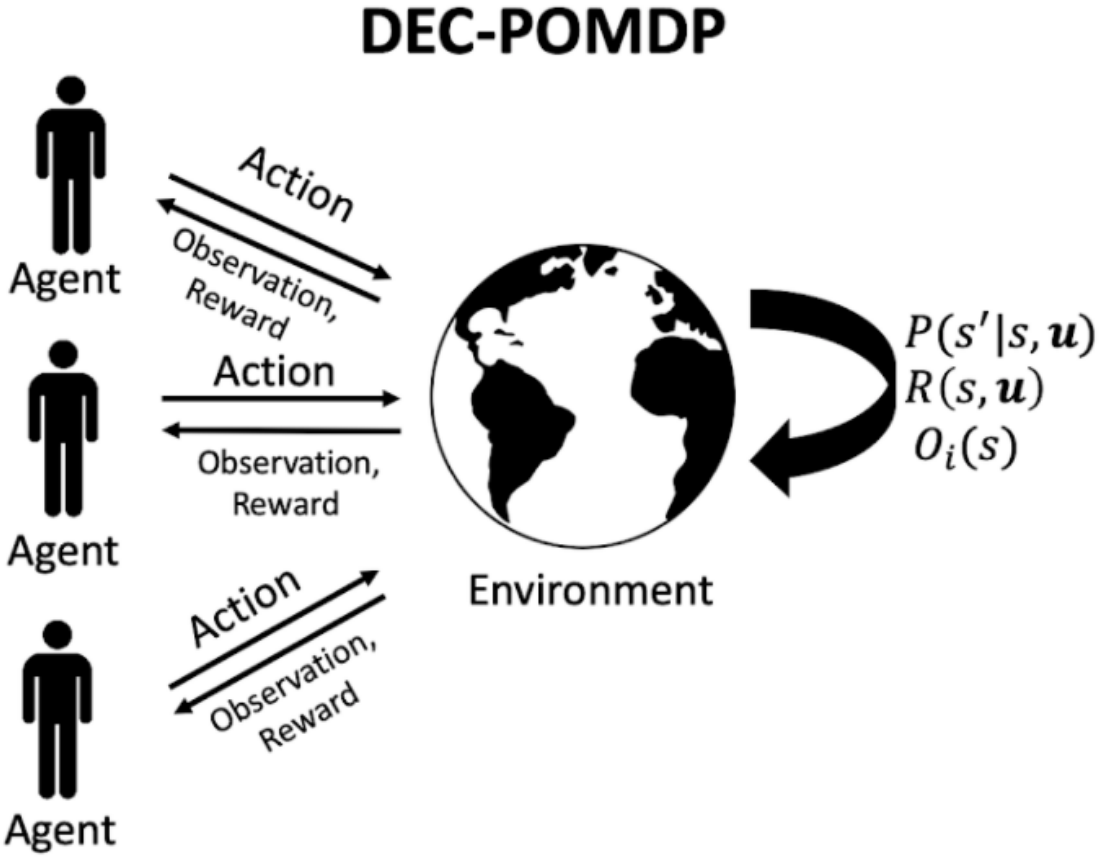


Figure 6.2: Multi-Agent - Environment interaction setting in a decentralized POMDP.

The agents are trained on a series of virtual patient profiles over several episodes, trying to find an optimal cooperative strategy for insulin administration. Each agent is characterized

by a distinct maximum insulin delivery rate (*Cap*) while the three agents are activated in different glycemic ranges as follows:

- **Agent 1**: This agent operates when the CGM detects blood glucose levels above a predetermined *hyperglycemia threshold*. It administers insulin up to Cap_H, the highest cap. The actual insulin dose is dynamically adjusted according to the patient's needs but constrained by this upper limit.
- **Agent 2**: When the CGM is between the *hyperglycemia threshold* and the *hypoglycemia threshold*. It administers insulin with a lower maximum rate, Cap_L, where $Cap_L < Cap_H$. Again, the amount of insulin delivered is adjusted dynamically but is limited by Cap_L.
- **Agent 3**: This agent intervenes when the CGM detects a blood glucose level below the *hypoglycemia threshold*. In this case, the insulin delivery is completely halted by setting it to zero, preventing hypoglycemic episodes.

6.3 Dataset

In order to generate *in silico* data for a cohort of 10 adult patients diagnosed with T1DM we employed the FDA-approved UVA/Padova simulator [120]. The simulator was configured to use the Insulet insulin pump system, a widely adopted technology, in conjunction with a CGM sensor that recorded blood glucose levels at 3-minute intervals. The performance of the MARL system was subsequently evaluated on this virtual patient cohort.

6.4 State, Action and Reward Space

In this study, the State space and Action space for each agent are defined to mirror those employed in the previous study in 5.3.1 and 5.3.2. However, a significant modification has been introduced to adapt the state space to the multi-agent setting: it is constrained within specific bounds, that we can think as *game rules* inspired by the terminology commonly used in multi-agent reinforcement learning within the context of game theory. These ranges are determined by the caps and thresholds described in detail in Section 6.2, effectively limiting the agents' perception and actions to a predefined operational domain tailored to the dynamics of the problem at hand.

The reward function employed in this framework is the parabolic function introduced in Section 5.3.3, designed to incentivize cooperative behavior among agents while optimizing individual contributions toward the shared goal of effective glycemic management. This approach redefines the multi-agent reinforcement learning environment as a cooperative game where all agents are incentivized to maximize the return of a single, shared reward function. This design choice emphasizes collaboration rather than competition, aligning the objectives of individual agents to collectively improve the system's overall performance.

A notable distinction between this study and previous approaches lies in the training methodology. In earlier frameworks, agents were trained independently in isolation, and the optimal agents were subsequently selected through a grid search based on their performance. In contrast, the current study adopts a more integrated approach where all agents are trained simultaneously within a shared environment that is the patient. This simultaneous training process ensures that the policies learned by each agent are inherently compatible with those of others, fostering more effective coordination. The agents operate on the same patient model during training, which enhances their ability to adapt to the shared environment and encourages consistent policy updates aligned with the dynamics of the patient-specific physiology.

6.4.1 Model Optimization

To further refine the model, a grid search is performed, but its purpose is fundamentally different from its use in the previous study. Instead of selecting individual agents, the grid search is applied to identify the most effective *game rules* defined by the hyperparameters (caps and thresholds) described in Section 6.2. This configuration is chosen to maximize the efficacy of the multi-agent system on a specific patient, advancing the paradigm of personalized medicine. The result is a framework where not only are agents trained in a coordinated manner, but the entire system is also optimized to adapt to the unique physiological characteristics and glycemic response patterns of individual patients.

Additionally, the multi-agent environment has been designed to incorporate specific constraints that prevent simultaneous insulin delivery by multiple agents. This is achieved through the implementation of an action-masking mechanism (using the MaskingPPO library [108]), which enforces a turn-based protocol. The transition between turns occurs whenever the patient's glycemic state shifts from one region to another, as defined by the thresholds outlined in Section 6.2. This mechanism ensures that the system operates within a structured decision-making framework, further enhancing its ability to maintain stability and prevent adverse outcomes such as over-delivery of insulin.

6.4.2 Algorithm Choice

In alignment with previous studies, the PPO algorithm has been selected as the RL backbone for this framework. This choice enables a direct comparison of the results obtained in this study with those from earlier work, facilitating a clear evaluation of the benefits introduced by the multi-agent setting and the cooperative game-based approach. Overall, this design reflects a comprehensive effort to integrate advanced reinforcement learning techniques into the domain of precision medicine, providing a robust and adaptive solution for the management of T1DM.

6.4.3 Results

In this section, we present and analyze the performance of our proposed multi-agent reinforcement learning system. To further contextualize its effectiveness, we also compare the results with traditional control techniques, specifically basal-bolus control (BBC) and proportional-integral-derivative control (PIDC).

Each table summarizes the performance across all patients, detailing the percentage of time spent within the target glucose range (TIR) and the percentage of time in hyperglycemic, severe hyperglycemic, hypoglycemic, and severe hypoglycemic states.

The performance results for the MARL-based approach are displayed in Table 6.1, where the last row shows the overall average performance across all patients. The MARL system achieved a mean TIR of 75.09%, with time in hypoglycemia kept under 5% and severe hypoglycemia at approximately 0.63%. These values reflect the system's success in minimizing hypoglycemic episodes, which are critical for immediate patient safety.

Hyperglycemia was observed for an average of 15.44% of the time, while severe hyperglycemia remained below 5%, suggesting the controller's ability to prevent dangerously high blood glucose levels.

On an individual level, patient results varied significantly. Patient #008 demonstrated the highest overall performance, with a TIR of 93.41%, hyperglycemia at just 2.94%, and an almost non-existent occurrence of severe hyperglycemia (0.01%). In contrast, patient #009 displayed the most challenging outcomes, with the highest percentages for severe hypoglycemia (4.04%) and severe hyperglycemia (6.52%), indicating areas where control could be further optimized.

Additionally, patient #006 exhibited the lowest TIR (60.94%) and the highest rates of both hyperglycemia (22.69%) and severe hyperglycemia (12.89%). Conversely, patient #007 recorded the lowest percentages of severe hypoglycemia (0.03%) and hypoglycemia (0.84%), reflecting exceptional control in maintaining safe blood glucose levels.

Table 6.1: Performance for each patient and overall mean with one standard deviation achieved on 1000 runs.

Test	Severe Hypo (%)	Hypo (%)	TIR (%)	Hyper (%)	Severe Hyper (%)
adult#001	0.24±0.34	3.49±1.76	80.93±4.48	13.99±3.17	1.35±1.51
adult#002	0.29±0.32	6.27±1.59	84.54±3.52	8.51±2.70	0.38±0.66
adult#003	0.85±0.61	6.67±1.78	73.08±5.44	16.66±4.04	2.74±2.11
adult#004	0.04±0.09	1.27±0.78	69.03±4.17	18.87±3.39	10.79±3.45
adult#005	0.31±0.33	6.35±1.68	82.36±4.44	10.46±3.47	0.52±0.88
adult#006	0.22±0.31	3.26±1.47	**60.94±5.47**	**22.69±3.95**	**12.89±3.80**
adult#007	**0.03±0.08**	**0.84±0.68**	71.12±5.10	21.70±4.04	6.31±3.20
adult#008	0.25±0.35	3.38±1.60	**93.41±2.86**	**2.94±1.79**	**0.01±0.08**
adult#009	**4.04±1.90**	**7.62±1.91**	63.29±6.31	18.53±3.83	6.52±3.57
adult#010	0.04±0.10	1.50±0.90	72.18±4.17	20.00±3.49	6.28±2.87
Overall Mean	0.63±0.44	4.07±1.42	75.09±4.60	15.44±3.39	4.78±2.21

Table 6.2 presents a performance comparison between our proposed MARL system and traditional controllers, including basal-bolus control (BBC) and proportional-integral-derivative control (PIDC). The results in the table reflect averaged performance across all tested patients.

The BBC controller achieves the highest TIR at 88.36%, along with favourable results across other glycemic conditions. However, it is an open-loop system that lacks full automation, requiring patients to manually log their carbohydrate intake (CHO) during meals. Additionally, the time spent in severe hypoglycemia with BBC (1.62%) is worse than the performance achieved by our system (0.63%), highlighting some limitations of BBC in preventing low blood sugar episodes.

On the other hand, the PIDC approach performs poorly, with a TIR of just 15.07% and

a severe hypoglycemia rate as high as 75.92% . These results underscore PIDC's inability to handle the complex dynamics of glucose control effectively, especially over an extended test duration of five days. This performance shortfall suggests that a more adaptive and personalized control system, such as GLUMARL, is crucial for long-term glycemic management.

In summary, while BBC achieves strong results in a structured, open-loop setting, GLUMARL shows significant promise by delivering comparable safety in hypoglycemic control, along with the flexibility and autonomy of a closed-loop approach.

Table 6.2: Comparison of the proposed system overall results with BBC controller and PIDC.

Controller	**System**	**Severe Hypo (%)**	**Hypo (%)**	**TIR (%)**	**Hyper (%)**	**Severe Hyper (%)**
GLUMARL	closed-loop	0.63±0.44	4.07±1.42	75.09±4.60	15.44±3.39	4.78±2,21
BBC	open-loop	1.62±4.23	3.03±0.80	88.36±3.16	6.68±0.71	0.32±0.31
PID	open-loop	75.92±9.49	2.01±0.72	15.07±2.73	3.81±1.32	3.2±1.15

6.4.4 Comparison between Simulated and Real-World Patients

In order to contextualize the performance achieved by the simulated adult patients discussed previously, Table 6.3 presents glycemic metrics extracted from an anonymized cohort of real patients in the Ohio dataset [78] in the same clinically relevant categories.

Overall, the mean values for these real patients are characterized by a *Severe Hypoglycemic* percentage (0.41%) that is comparable to the average ranges found among most of the simulated subjects (whose means vary between approximately 0.03% and 4.04%, albeit with one simulated patient, adult #009, displaying significantly higher severe hypoglycemic episodes). This alignment suggests that, on average, both cohorts experience low rates of severe hypoglycemia, although outliers may be present in either case.

Interestingly, the real patients display a mean hypoglycemic percentage of approximately 2.89%, which is somewhat lower than the mean value of 4.07% observed in the simulated cohort's overall summary. This difference could be attributed to several factors, including inter-patient variability, differences in insulin dosing regimens or daily behaviours, or the presence of compensatory physiological mechanisms that simulated models may not fully capture.

A more evident discrepancy can be observed in the euglycemic (TIR) percentage. The

Table 6.3: Glycemic metrics extracted from the anonymized Ohio dataset. Percentages of time spent in five categories: Severe Hypoglycemic, Hypoglycemic, Euglycemic, Hyperglycemic, and Severe Hyperglycemic. Rows show data for each anonymized patient (Patient ID) and the final row displays the mean across all patients in this dataset.

Patient ID	**Sev. Hypo (%)**	**Hypo (%)**	**TIR (%)**	**Hyper (%)**	**Sev. Hyper (%)**
540	0.64677	5.376272	70.996429	17.395405	5.585124
544	0.044981	1.259465	64.727491	25.001874	8.966189
552	0.104858	3.171968	73.558196	17.747291	5.417686
559	0.563486	3.050338	56.709241	27.069872	12.607062
563	0.095277	1.898734	71.791207	23.472166	2.742616
567	1.056843	5.616366	64.724088	21.499207	7.103495
570	0.029140	1.464268	40.489546	37.349749	20.667298
575	1.155230	6.447150	68.213890	18.407582	5.776148
584	0.209247	0.688491	50.043874	32.932838	16.125548
588	0.110168	0.686929	61.979133	32.266217	4.957553
591	0.786360	3.145440	64.893070	24.972441	6.202690
596	0.110132	1.879589	74.324523	20.198238	3.487518
Overall Mean	0.409374	2.890417	63.537557	24.859407	8.303244

real-world individuals exhibit a mean of about 63.54%, whereas the simulated cohort attained higher average TIR values (around 75.09%), with several simulated patients even surpassing 80% or 90%. This contrast may be indicative of idealized or optimized conditions embedded in the simulation framework, suggesting that real patients face additional complexities—such

as lifestyle factors, meal variations, or imperfect patient adherence—that reduce their time in the euglycemic range.

An even more pronounced divergence emerges in the hyperglycemic and severe hyperglycemic intervals: the Ohio dataset shows a mean of 24.86% in hyperglycemia and 8.30% in severe hyperglycemia. By contrast, the simulated patients tend to present lower hyper ($\sim 15.44\%$) and severe hyper ($\sim 4.78\%$) conditions, on average. Additionally, certain real patients (e.g., ID 570 and ID 584) demonstrate high Hyperglycemic and Severe Hyperglycemic percentages, whereas the simulated group exhibits milder levels of hyperglycemia, except for specific outliers adult #006 and adult #009).

From a statistical standpoint, these findings highlight the importance of evaluating both groups not only in terms of mean values but also via variability and outlier analysis. Indeed, some real-world patients (e.g., ID 570 and ID 584) have relatively extreme hyperglycemic excursions, while some simulated patients (such as adult#009) exhibit unusual patterns in severe hypoglycemia. Such comparisons underscore the necessity of robust modeling and the potential benefit of tailoring simulation parameters to replicate more faithfully the inter-patient variability seen in clinical practice.

In conclusion, while the simulated data display generally favorable control—particularly for euglycemia—the real-world data underscore more pronounced hyperglycemic excursions and slightly lower risk of extreme hypoglycemia. These differences may derive from the simplified assumptions in simulation models, residual confounding in real patients —related to lifestyle, genetics, comorbidities, and so forth — , or a combination thereof. Future work should integrate more complex real-world factors into the simulation pipeline to better mirror the broader variability observed in clinical populations.

Chapter 7

Conclusions

This thesis explores the design and assessment of novel methodologies for managing T1D, with a primary focus on mitigating glycemic variability through the integration of advanced machine learning architectures. By harnessing real-time data analytics, the developed frameworks enable precise early detection of hypo- and hyperglycemic events while supporting personalized optimization of glycemic regulation.

In Chapter 3 we investigated a layered meta-learning approach, utilizing multi-expert models for a three-class classification task (hypoglycemia, normoglycemia, hyperglycemia). Specifically, we proposed two meta-learning models, ME-LSTM-DT and ME-CNN-DT, for early prediction of glycemic events in T1D management. Both models demonstrated robust performance, achieving recall rates exceeding 81% for hypoglycemia and 83% for hyperglycemia with a 15-minute prediction horizon (PH), while maintaining low false alarm rates (0.45–0.46/day). Notably, the ME-LSTM-DT achieved an average time gain of 22.8 and 24.0 minutes for hypo- and hyperglycemia, respectively, outperforming existing literature benchmarks (15–20 minutes). The ME-CNN-DT exhibited superior precision (87% for hypoglycemia at $\alpha = 1$) but traded marginally lower advance detection for fewer false positives.

Qualitative comparisons with state-of-the-art methods (Table 3.2) highlighted the efficacy of the meta-learning framework, particularly in addressing dataset imbalance and minimizing false alarms. Edge implementation tests confirmed real-time feasibility, with inference times under 0.1 ms per prediction, suitable for deployment on resource-constrained devices (Table 3.8). However, performance degraded on the private UCBM dataset, underscoring challenges in generalizing across heterogeneous patient populations and sensor technologies. While longer PHs (60–120 minutes) reduced predictive accuracy due to increased uncertainty, the 30-minute PH proved optimal, balancing clinical utility and reliability.

In Chapter 4 we evaluated federated learning-based glucose prediction models, comparing a single regressor and a triple regressor architecture. The single regressor achieved an average

RMSE of 14.73 mg/dL (test set) and 18.48 mg/dL (online testing), with rapid convergence but reduced accuracy in hypoglycemia/hyperglycemia due to dataset imbalance. Clinically, 97.88% of predictions fell within low-risk CEG zones A/B. The triple regressor, employing specialized sub-models for hypoglycemia, euglycemia, and hyperglycemia, outperformed its counterpart, reducing RMSE to 13.73 mg/dL (test) and 12.03 mg/dL (online), while minimizing variability. It improved hypoglycemic accuracy by over 6% and enhanced clinical reliability (98.76% in CEG zones A/B). Minor transition fluctuations ($\leq$ 6% error) were clinically negligible. Compared to existing frameworks, the model demonstrated superior efficiency (RMSE: 15.13 mg/dL online, $\sim 10^3$ FLOPs) versus competitors requiring $\sim 10^6$-10^8 FLOPs. Its low computational cost and robust performance position it as a viable tool for real-time glucose forecasting.

Adaptive insulin delivery strategies were another focal point in this research, exemplified by hybrid closed-loop control systems validated using *in silico* patient simulations.

The Dual PPO system proposed in Chapter 5 achieved an average TIR of 69.30%, while effectively limiting hypoglycemic episodes (5.40%) and severe hypoglycemia (1.91%). Statistical analysis confirmed that the Dual PPO controller significantly outperformed the Single PPO approach ($\Delta_{TIR} = +7.61\%$, $p < 0.001$) and substantially reduced severe hyperglycemia (-8.73%). Dual PPO architecture's effectiveness stems from its specialized dual action space design, with distinct controllers for High-Cap and Low-Cap regions. Our comparative analysis revealed also that the Parabolic reward function yielded superior glycemic outcomes compared to the widely used Magni function (TIR: 69.30% vs. 64.23%). Furthermore, our approach demonstrates remarkable computational efficiency compared to state-of-the-art reinforcement learning methods, requiring approximately 1.0×10^3 training iterations versus 8.0×10^5 to 2.9×10^6 iterations reported in comparable studies.

When benchmarked against classical methods, our approach showed competitive performance despite fundamental differences in system design. While the BBC achieved the highest overall TIR (88.35%), it requires manual meal announcements, lacking the automation that characterizes closed-loop solutions like our proposed system.

In this context, the GLUMARL system was developed and extensively tested using *in silico* patients generated by the UVA/Padova simulator. This novel hybrid closed-loop control system presented in Chapter 6 integrates a MARL framework with multiple cooperating RL agents, enabling personalized and adaptive insulin administration without requiring continuous carbohydrate information. GLUMARL achieved significant improvements in glycemic control, with an average TIR of 75.09%, closely matching the BBC performance in managing hypoglycemia and effectively reducing severe hypoglycemia from 1.62% to 0.63%, balancing adaptability and safety, while offering superior autonomy in glycemic regulation.

7.1 Towards an Integrated Framework for T1D Management

A key contribution of this thesis lies in the complementary nature of the developed components, which together form a cohesive ecosystem for comprehensive T1D management.

The predictive models developed in Chapters 3 and 4 provide different approaches to anticipating glycemic events, each with distinct advantages. The meta-learning architecture investigated in Chapter 3 excels at early event detection with minimal false alarms, offering patients crucial time to take preventive action. Meanwhile, the federated learning framework examined in Chapter 4 ensures continuous, privacy-preserving forecasting that can adapt to changing patient conditions while maintaining data security across distributed environments.

These prediction capabilities form the foundation upon which the control strategies in Chapters 5 and 6 build. The Dual PPO architecture in Chapter 5 introduces a zone-based approach to insulin administration, optimizing delivery based on distinct glycemic regions. The GLUMARL system presented in Chapter 6 extends this concept by implementing multi-agent cooperation, further enhancing the robustness and adaptability of insulin control.

In a fully integrated system, the prediction models could provide early warnings and trend forecasts that directly inform the reinforcement learning control mechanisms. For instance, when the meta-learning model predicts an impending hypoglycemic event, this information could be fed into the GLUMARL framework to initiate preemptive action, such as reducing insulin delivery before blood glucose levels fall to critical ranges. Similarly, the continuous forecasting capabilities of the FedROS-ELM model could help the Dual PPO system optimize its transition between control zones based on anticipated glycemic trajectories rather than just current states.

Crucially, the integration of predictive outputs into control decisions enables anticipatory rather than purely reactive management—a paradigm shift exemplified by using hypoglycemia forecasts to trigger preemptive insulin attenuation before glucose levels enter clinically dangerous thresholds.

The technical interoperability of these components is reinforced by their shared computational efficiency, with all subsystems demonstrating real-time operational feasibility on edge devices. This characteristic is pivotal for clinical translation, as it enables deployment across resource-constrained wearable platforms without sacrificing prediction accuracy or control responsiveness. Furthermore, the frameworks' modular design permits incremental enhancement, allowing future integration of ancillary modules for meal detection, physical activity monitoring, or stress response modeling.

7.2 Future Directions

Future research should expand on these findings by addressing physiological variability and noisy data in real-world settings, validating performance across heterogeneous patient populations, and integrating additional functionalities such as automated meal detection. Several specific avenues for advancement include:

- **Enhanced Integration:** Development of communication interfaces between prediction and control modules, ensuring seamless data flow and coordinated decision-making
- **Precision Medicine:** Refinement of adaptation algorithms to enable personalized treatment strategies by accounting for patient-specific factors, including lifestyle patterns, genetic predispositions, concurrent medical conditions, and age-related physiological differences.
- **Advanced Sensing Integration:** Incorporation of additional biomarkers and physiological signals beyond CGM data, such as heart rate variability, physical activity levels, and stress indicators
- **Security Framework:** Development of robust encryption and privacy-preservation techniques specifically tailored for medical IoT environments with constrained computational resources
- **Clinical Validation:** Progression from *in silico* testing to controlled clinical trials, evaluating both the individual components and the integrated system across diverse patient populations

The exploration of these directions will be critical for translating the promising results presented in this thesis into practical, widely-adopted clinical solutions that meaningfully improve outcomes for individuals with T1D.

7.3 Concluding Remarks

In conclusion, this study establishes a foundation for AI-driven, patient-centric solutions in T1D management, offering innovative tools for personalized glycemic control with significant potential for clinical application. By developing complementary approaches to both prediction and control, this research creates a comprehensive framework that addresses the full spectrum of challenges in diabetes management.

The integration of these components represents a significant step toward fully autonomous artificial pancreas systems that can adapt to individual needs while maintaining safety and efficacy. The focus on computational efficiency, privacy preservation, and real-world applicability further enhances the translational potential of these technologies.

Beyond T1D management, the methodologies developed in this research—particularly the combination of edge computing, federated learning, and multi-agent reinforcement learning—provide a template for addressing broader healthcare challenges characterized by continuous data streams, privacy concerns, and the need for personalized interventions. This extends the impact of this work beyond glycemic data to potential applications in other settings requiring continuous monitoring and adaptive management strategies.

Bibliography

[1] 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.

[2] American Diabetes Association. Introduction: Standards of medical care in diabetes—2022, 2022.

[3] Meysam Bastani. Model-free intelligent diabetes management using machine learning. 2014.

[4] 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.

[5] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[6] 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.

[7] 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.

[8] Jeffrey A Bluestone, Kevan Herold, and George Eisenbarth. Genetics, pathogenesis and clinical interventions in type 1 diabetes. *Nature*, 464(7293):1293–1300, 2010.

[9] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[10] 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.

[11] 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.

[12] Hatim Butt, Ikramullah Khosa, and Muhammad Aksam Iftikhar. Feature transformation for efficient blood glucose prediction in type 1 diabetes mellitus patients. *Diagnostics*, 13(3):340, 2023.

[13] 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.

[14] Eda Cengiz and William V. Tamborlane. A tale of two compartments: Interstitial versus blood glucose monitoring. *Diabetes Technology & Therapeutics*, 11(S1):S–11–S–16, 2009. PMID: 19469670.

[15] 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.

[16] 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.

[17] William Clarke and Boris Kovatchev. Statistical tools to analyze continuous glucose monitor data. *Diabetes technology & therapeutics*, 11(S1):S–45, 2009.

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

[19] Claudio Cobelli, Eric Renard, and Boris Kovatchev. Artificial pancreas: past, present, future. *Diabetes*, 60(11):2672–2682, 2011.

[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] Philip E Cryer and Ana María Arbeláez. Hypoglycemia in diabetes. *Textbook of diabetes*, pages 513–533, 2017.

[22] 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.

[23] 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.

[24] Thomas Danne, Revital Nimri, Tadej Battelino, Richard M Bergenstal, Kelly L Close, J Hans DeVries, Satish Garg, Lutz Heinemann, Irl Hirsch, Stephanie A Amiel, et al. International consensus on use of continuous glucose monitoring. *Diabetes care*, 40(12):1631–1640, 2017.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Philip De Chazal, Jonathan Tapson, and André Van Schaik. A comparison of extreme learning machines and back-propagation trained feed-forward networks processing the mnist database. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2165–2168. IEEE, 2015.

[32] Ivanoe De Falco, Antonio Della Cioppa, Tomas Koutny, Martin Ubl, Michal Krcma, Umberto Scafuri, and Ernesto Tarantino. A federated learning-inspired evolutionary algorithm: Application to glucose prediction. *Sensors*, 23(6):2957, 2023.

[33] Benedetta De Paoli, Federico D'Antoni, Mario Merone, Silvia Pieralice, Vincenzo Piemonte, and Paolo Pozzilli. Blood glucose level forecasting on type-1-diabetes subjects during physical activity: A comparative analysis of different learning techniques. *Bioengineering*, 8(6):72, 2021.

[34] 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.

[35] Olesya Didyuk, Nicolas Econom, Angelica Guardia, Kelsey Livingston, and Ulrike Klueh. Continuous glucose monitoring devices: past, present, and future focus on the history and evolution of technological innovation. *Journal of diabetes science and technology*, 15(3):676–683, 2021.

[36] 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.

[37] Dominic Ehrmann, Bernhard Kulzer, Timm Roos, Thomas Haak, Mohammed Al-Khatib, and Norbert Hermanns. Risk factors and prevention strategies for diabetic ketoacidosis in people with established type 1 diabetes. *The Lancet Diabetes & Endocrinology*, 8(5):436–446, 2020.

[38] Anas El Fathi and Marc D Breton. Using reinforcement learning to simplify meal-time insulin dosing for people with type 1 diabetes: In-silico experiments. *IFAC-PapersOnLine*, 56(2):11539–11544, 2023.

[39] 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.

[40] Nelly Elsayed, Zag ElSayed, and Murat Ozer. Early stage diabetes prediction via extreme learning machine. In *SoutheastCon 2022*, pages 374–379. IEEE, 2022.

[41] 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.

[42] Virginie Felizardo, Diogo Machado, Nuno M Garcia, Nuno Pombo, and Pedro Brandão. Hypoglycaemia prediction models with auto explanation. *IEEE Access*, 10:57930–57941, 2021.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] Eleni I Georga, Vasilios C Protopappas, Demosthenes Polyzos, and Dimitrios I Fotiadis. Online prediction of glucose concentration in type 1 diabetes using extreme learning machines. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3262–3265. IEEE, 2015.

[48] 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.

[49] 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.

[50] Lutz Heinemann, Michael Schoemaker, Günther Schmelzeisen-Redecker, Rolf Hinzmann, Adham Kassab, Guido Freckmann, Florian Reiterer, and Luigi Del Re. Benefits and limitations of mard as a performance parameter for continuous glucose monitoring in the interstitial space. *Journal of Diabetes Science and Technology*, 14(1):135–150, 2020.

[51] 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.

[52] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 985–990. Ieee, 2004.

[53] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[54] 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.

[55] 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.

[56] Mehrad Jaloli and Marzia Cescon. Reinforcement learning for multiple daily injection (mdi) therapy in type 1 diabetes (t1d). *BioMedInformatics*, 3(2):422–433, 2023.

[57] Mehrad Jaloli and Marzia Cescon. Basal-bolus advisor for type 1 diabetes (t1d) patients using multi-agent reinforcement learning (rl) methodology. *Control Engineering Practice*, 142:105762, 2024.

[58] 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.

[59] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[60] Lindy Kahanovitz, Patrick M Sluss, and Steven J Russell. Type 1 diabetes—a clinical perspective. *Point of care*, 16(1):37–40, 2017.

[61] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

[62] 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.

[63] Heydar Khadem, Hoda Nemat, Jackie Elliott, and Mohammed Benaissa. Blood glucose level time series forecasting: nested deep ensemble learning lag fusion. *Bioengineering*, 10(4):487, 2023.

[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] 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.

[66] 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.

[67] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, 2021.

[68] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI open*, 3:111–132, 2022.

[69] 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.

[70] 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.

[71] Wenzhou Lv, Tianyu Wu, Luolin Xiong, Liang Wu, Jian Zhou, Yang Tang, and Feng Qian. Hybrid control policy for artificial pancreas via ensemble deep reinforcement learning. *IEEE Transactions on Biomedical Engineering*, 2024.

[72] 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.

[73] Bradley A Malin, Khaled El Emam, and Christine M O'Keefe. Biomedical data privacy: problems, perspectives, and recent advances. *Journal of the American medical informatics association*, 20(1):2–6, 2013.

[74] Alessandro Marchetti, Daniele Sasso, Federico D'Antoni, Francesco Morandin, Maurizio Parton, Margherita Anna Grazia Matarrese, and Mario Merone. Deep reinforcement learning for type 1 diabetes: Dual ppo controller for personalized insulin management. *Computers in Biology and Medicine*, 191:110147, 2025.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] John Martinsson, Alexander Schliep, Björn Eliasson, and Olof Mogren. Blood glucose prediction with variance estimation using recurrent neural networks. *Journal of Healthcare Informatics Research*, 4:1–18, 2020.

[81] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[82] 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.

[83] 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.

[84] Kellee M Miller, Nicole C Foster, Roy W Beck, Richard M Bergenstal, Stephanie N DuBose, Linda A DiMeglio, David M Maahs, and William V Tamborlane. Current

state of type 1 diabetes treatment in the us: updated data from the t1d exchange clinic registry. *Diabetes care*, 38(6):971–978, 2015.

[85] Xue Mo, Youqing Wang, and Xiangwei Wu. Hypoglycemia prediction using extreme learning machine (elm) and regularized elm. In *2013 25th Chinese Control and Decision Conference (CCDC)*, pages 4405–4409. IEEE, 2013.

[86] Mohammad Moshawrab, Mehdi Adda, Abdenour Bouzouane, Hussein Ibrahim, and Ali Raad. Reviewing federated machine learning and its use in diseases prediction. *Sensors*, 23(4):2112, 2023.

[87] Clara Mosquera-Lopez et al. Enabling fully automated insulin delivery through meal detection and size estimation using artificial intelligence. *npj Digital Medicine*, 6(1):39, 2023.

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

[89] Hoda Nemat, Heydar Khadem, Mohammad R Eissa, Jackie Elliott, and Mohammed Benaissa. Blood glucose level prediction: advanced deep-ensemble learning approach. *IEEE Journal of Biomedical and Health Informatics*, 26(6):2758–2769, 2022.

[90] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, 55(3):1–37, 2022.

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

[92] World Health Organization. Diabetes Fact Sheet, 2023. https://www.who.int/news-room/fact-sheets/detail/diabetes.

[93] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[94] 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.

[95] Emilie Palisaitis, Anas El Fathi, Julia E von Oettingen, Ahmad Haidar, and Laurent Legault. A meal detection algorithm for the artificial pancreas: a randomized controlled clinical trial in adolescents with type 1 diabetes. *Diabetes Care*, 44(2):604–606, 2021.

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

[97] 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.

[98] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[99] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, 22(1):12348–12355, 2021.

[100] 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.

[101] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[102] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[103] 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.

[104] 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.

[105] 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.

[106] Jaime Sevilla, Lennart Heim, Marius Hobbhahn, Tamay Besiroglu, Anson Ho, and Pablo Villalobos. Estimating training compute of deep learning models. *Epoch, January*, 20, 2022.

[107] 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.

[108] Stable-Baselines3 Contributors. *PPO-Mask: Masking Support for Proximal Policy Optimization*. Stable-Baselines3 Community, 2025. Accessed: 2025-01-06.

[109] 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.

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

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

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

[113] Stella Tsichlaki, Lefteris Koumakis, Manolis Tsiknakis, et al. Type 1 diabetes hypoglycemia prediction algorithms: Systematic review. *JMIR diabetes*, 7(3):e34699, 2022.

[114] Roger H Unger, Alan D Cherrington, et al. Glucagonocentric restructuring of diabetes: a pathophysiologic and therapeutic makeover. *The Journal of clinical investigation*, 122(1):4–12, 2012.

[115] 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.

[116] Martina Vettoretti, Andrea Facchinetti, Giovanni Sparacino, and Claudio Cobelli. Type-1 diabetes patient decision simulator for in silico testing safety and effectiveness

of insulin treatments. *IEEE Transactions on Biomedical Engineering*, 65(6):1281–1290, 2017.

[117] Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, et al. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29(10):2633–2642, 2023.

[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] 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.

[120] Jinyu Xie. Simglucose v0.2.1. GitHub repository, 2018. https://github.com/jxx123/simglucose.

[121] Qingqing Xu, Liye Wang, and Sujit S Sansgiry. A systematic literature review of predicting diabetic retinopathy, nephropathy and neuropathy in patients with type 1 diabetes using machine learning. *Journal of Medical Artificial Intelligence*, 3, 2020.

[122] 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.

[123] 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.

[124] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[125] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[126] 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.

[127] Gordon A Zello. Dietary reference intakes for the macronutrients and energy: considerations for physical activity. *Applied Physiology, Nutrition, and Metabolism*, 31(1):74–79, 2006.

[128] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[129] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[130] 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.

[131] 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.

[132] Taiyu Zhu, Kezhi Li, and Pantelis Georgiou. A dual-hormone closed-loop delivery system for type 1 diabetes using deep reinforcement learning. *arXiv preprint arXiv:1910.04059*, 2019.

[133] 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.

[134] 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.

[135] 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.

[136] 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.

[137] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendices

Appendix A

Proximal Policy Optimization

Policy gradient methods The goal of an agent is to find a policy π that maximizes $J(\pi)$, defined as in [112] as the expected cumulative discounted reward over time:

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{A.1}$$

To maximize $J(\pi)$, policy gradient methods exploit the Policy Gradient Theorem [112] for a parametric policy π_θ, through gradient ascent of the θ parameters:

$$\theta \to \theta + \alpha \nabla_\theta J(\pi_\theta) \tag{A.2}$$

where we can express the gradient of the expected return as:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a)\right], \tag{A.3}$$

and where $A^{\pi_\theta}(s,a)$ is the advantage function:

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s), \tag{A.4}$$

representing the relative value of taking action a in state s compared to the expected value under the current policy.

Trust Region Policy Optimization By leveraging the Conservative Policy Iteration (CPI) [61] algorithm, TRPO paper introduces a *surrogate* objective function J^{CPI} [101] to ensure monotonic improvement from an old policy $\pi_{\theta_{\text{old}}}$ to a new policy π_θ:

$$J^{CPI}(\pi_\theta) = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right], \tag{A.5}$$

where the ratio $r_t(\theta)$ is defined as:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)},$$

$\hat{A}_t$ is an estimator of the advantage under the policy parameterized by θ_{old} and $\hat{\mathbb{E}}_t[...]$ indicates the empirical average over a finite batch of samples (see also [102]). To ensure stable and conservative policy updates within a *trust region*, a constraint is imposed on the Kullback-Leibler (KL) divergence between the old and new policies [101]:

$$\hat{\mathbb{E}}_t\left[D_{\text{KL}}\left(\pi_{\theta_{\text{old}}}(\cdot|s_t) \,\|\, \pi_\theta(\cdot|s_t)\right)\right] \leq \delta, \tag{A.6}$$

where δ is a step-size parameter that controls the proximity between consecutive policy updates, preventing excessively large deviations that could lead to performance degradation.

Proximal Policy Optimization PPO [102] further simplifies this concept by replacing the hard KL constraint with a clipped surrogate objective function, explicitly limiting policy changes:

$$J^{CLIP}(\pi_\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{A.7}$$

where:

$$\text{clip}(x, k_-, k_+) := \max(k_-, \min(k_+, x)) \quad \forall x, k_-, k_+ \in \mathbb{R} \tag{A.8}$$

and ϵ is a hyperparameter (typically set to 0.2) controlling the degree of policy update clipping.

This clipping mechanism ensures stable updates by restricting the policy ratio $r_t(\theta)$ within a predefined interval $[1-\epsilon, 1+\epsilon]$, preventing excessively large policy deviations. Such controlled updates enhance the robustness and stability of the learning process, making PPO particularly suitable for continuous control tasks [102].

Appendix B

Contributions in Computer Science

Deep Reinforcement Learning

Reference Pasqualini, L., Parton, M., Morandin, F., Amato, G., Gini, R., Metta, C., Fantozzi, M., & Marchetti, A. (2022). Score vs. winrate in score-based games: which reward for reinforcement learning? Proceedings of the 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). https://doi.org/10.1109/icmla55696.2022.00099

Abstract In the last years, the DeepMind algorithm AlphaZero has become the state of the art to efficiently tackle perfect information two-player zero-sum games with a win/lose outcome. However, when the win/lose outcome is decided by a final score difference, AlphaZero may play score-suboptimal moves because all winning final positions are equivalent from the win/lose outcome perspective. This can be an issue, for instance when used for teaching, or when trying to understand whether there is a better move. Moreover, there is the theoretical quest for the perfect game. A naive approach would be training an AlphaZero-like agent to predict score differences instead of win/lose outcomes. Since the game of Go is deterministic, this should as well produce an outcome-optimal play. However, it is a folklore belief that "this does not work". In this paper, we first provide empirical evidence for this belief. We then give a theoretical interpretation of this suboptimality in general perfect information two-player zero-sum game where the complexity of a game like Go is replaced by the randomness of the environment. We show that an outcome-optimal policy has a different preference for uncertainty when it is winning or losing. In particular, when in a losing state, an outcome-optimal agent chooses actions leading to a higher score variance. We then posit that when approximation is involved, a deterministic game behaves like a nondeterministic game, where the score variance is modeled by how uncertain the position is. We validate

this hypothesis in AlphaZero-like software with a human expert.

Deep Learning

Reference Metta, C., Fantozzi, M., Papini, A., Amato, G., Bergamaschi, M., Galfrè, S. G., Marchetti, A., Veglio, M., Parton, M., & Morandin, F. (2024). Increasing biases can be more efficient than increasing weights. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. https://doi.org/10.1109/WACV57701.2024.00279

Abstract We introduce a novel computational unit for neural networks that features multiple biases, challenging the traditional perceptron structure. This unit emphasizes the importance of preserving uncorrupted information as it is passed from one unit to the next, applying activation functions later in the process with specialized biases for each unit. Through both empirical and theoretical analyses, we show that by focusing on increasing biases rather than weights, there is potential for significant enhancement in a neural network model's performance. This approach offers an alternative perspective on optimizing information flow within neural networks.