BACHELOR THESIS

BSc in Computer Engineering

dIAbetes

Analysing diabetes mellitus data using artificial intelligence techniques

Author
Mario García Jiménez

Tutor
Oresti Baños Legrán
Claudia Villalonga Palliser

School of Technology and Telecommunications Engineering of the University of Granada

Granada, September, 2023

dIAbetes: Analysing diabetes mellitus data using artificial intelligence techniques

Mario García Jiménez

Keywords: Type 1 Diabetes Mellitus, Blood Glucose, Forecasting, Prediction, Continuous Glucose Monitoring, Machine Learning, Deep Learning

Abstract

Diabetes is a chronic disease that currently affects to more than 500 million of people. People with type 1 diabetes mellitus need to maintain blood glucose levels within a normal range, which is usually a challenging task. The introduction of continuous glucose monitoring technologies allows the patients to known their blood glucose levels in real time, but also to create a history of blood glucose measurements that can be exploited by using machine learning techniques. The development of an accurate blood glucose forecasting model would have a significant impact, as it could help patients to take early actions to avoid harmful situations. Unfortunately, this is still an open challenge.

This bachelor thesis, conducted in collaboration with the Clinical Unit of Endocrinology and Nutrition of the San Cecilio University Hospital of Granada in Spain, explores general and personalized strategies using Long Short Term Memory (LSTM) neural networks and linear regression models to forecast future blood glucose levels at 30 and 60 minute prediction horizons on the T1DiabetesGranada dataset. The best results were obtained using the general LSTM model (Root Mean Square Error (RMSE) of 17.74 mg/dL for 30 minutes and 32.40 mg/dL for 60 minutes) and the general linear model (RMSE of 18.42 mg/dL for 30 minutes and 33.46 for 60 minutes). These results can compete with the current state of the art publications. In addition, the results provided by the general linear model suggest that the T1DiabetesGranada dataset does not require complex models to accurately forecast future blood glucose levels.

dIAbetes: Análisis de datos sobre diabetes mellitus mediante técnicas de inteligencia artificial

Mario García Jiménez

Palabras clave: Diabetes Mellitus Tipo 1, Glucosa en Sangre, Predicción, Monitorización Continua de Glucosa, Aprendizaje Automático, Aprendizaje Profundo

Resumen

La diabetes es una enfermedad crónica que afecta actualmente a más de 500 millones de personas. Las personas con diabetes mellitus de tipo 1 necesitan mantener los niveles de glucosa en sangre dentro de unos márgenes normales, lo que suele ser una ardua tarea. La introducción de tecnologías de monitorización continua de la glucosa permite a los pacientes conocer sus niveles de glucosa en sangre en tiempo real, pero también crear un historial de mediciones de glucosa en sangre que puede aprovecharse mediante técnicas de aprendizaje automático. El desarrollo de un modelo preciso de predicción de la glucosa en sangre tendría un impacto significativo, ya que podría ayudar a los pacientes a tomar medidas preventivas para evitar situaciones perjudiciales. Desgraciadamente, a día de hoy esto sigue siendo un reto pendiente.

Este trabajo de fin de carrera, realizado en colaboración con la Unidad Clínica de Endocrinología y Nutrición del Hospital Universitario San Cecilio de Granada en España, explora estrategias generales y personalizadas que utilizan redes neuronales de memoria a largo plazo (LSTM) y modelos de regresión lineal para predecir futuros niveles de glucosa en sangre en horizontes de predicción de 30 y 60 minutos en el conjunto de datos T1DiabetesGranada. Los mejores resultados se obtuvieron utilizando el modelo LSTM general (raíz del error cuadrático medio (RMSE) de 17,74 mg/dL para 30 minutos y 32,40 mg/dL para 60 minutos) y el modelo lineal general (RMSE de 18,42 mg/dL para 30 minutos y 33,46 para 60 minutos). Estos resultados pueden competir directamente con los resultados obtenidos en las publicaciones del estado del arte. Además, los resultados proporcionados por el modelo lineal general sugieren que el conjunto de datos T1DiabetesGranada no requiere modelos complejos para predecir con precisión los futuros niveles de glucosa en sangre.

I, **Mario García Jiménez**, student of the degree Computer Engineering of the **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación of the University of Granada**, with DNI 25620713R, authorise the following copy of my Final Degree Project in the centre's library so that it can be consulted by any person.

Fdo: Mario García Jiménez

Granada, September 5, 2023

D. **Oresti Baños Legrán**, professor in the area of **Ingeniería de Sistemas y Automática Dpto. Ingeniería de Sistemas y Automática** at the University of Granada.

D. **Claudia Villalonga Palliser**, professor in the area of **Arquitectura y Tecnología de Computadores Dpto. Ingeniería de Computadores, Automática y Robótica** at the University of Granada.

Inform:

That the present work, entitled **dIAbetes: Analysing diabetes mellitus data using artificial intelligence techniques**, has been carried out under their supervision by Mario García Jiménez, and authorizes the defense of such work before the appropriate court.

Issues and signs this report in Granada on September 07, 2023.

Tutors:

Oresti Baños Legrán **Claudia Villalonga Palliser**

Acknowledgements

To Ciro Rodriguez, Claudia and Oresti for guiding and supporting me throughout this adventure. For making an effort to understand me and for being available whenever I needed them.

To my dear grandma, who has lived most of her life with type 1 diabetes, but always with a smile on her face. This work is mine, but also yours.

Contents

Chapter 1

Introduction

1.1 Context and motivation

Diabetes is a global and lifelong disease with an estimated prevalence of more than 400 million people worldwide in 2019, rising to 578 million by 2030 [33]. According to the World Health Organization (WHO), 1.5 million deaths are directly attributed to diabetes each year [43]. In type 1 Diabetes Mellitus (T1DM), the pancreas stops producing insulin, which affects to the blood glucose (BG) regulations in the body. As a consequence, T1DM patients need to keep their BG levels (BGLs) within a safe range, as levels below or above this safe range can cause hypoglycaemia or hyperglycaemia, respectively. These conditions can be fatal, causing loss of consciousness or even death.

For patients, keeping BGLs within a safe range is not an easy task. BGLs are the result of complex interactions between various physiological, hormonal, and metabolic processes that are not yet fully understood. They are influenced by several factors such as age, weight, gender, diet, physical activity, illness, menstruation, stress, and medication. The BG behaviours can vary widely between different patients, and even within the same patient over time, imperceptible changes in the routine can lead to significantly different outcomes [34]. For all these reasons, it is very difficult to know whether a patient's BGLs will remain stable over time or whether they will change rapidly and unexpectedly.

However, with the advancement of technology, continuous glucose monitoring (CGM) sensors have been introduced to ordinary patients to measure their BGLs every few minutes. CGM facilities the BG concentration control and guides the medical treatments, but also promotes research into forecasting future BGLs. The use of CGM allows the collection of a history of BG measurements, which conforms the appropriate input for machine learning prediction models. Given a large set of BG measurements, machine learning models, and in particular deep learning models, are the most powerful tool to capture the complex relationships between BGLs and to predict future BG values. The results obtained in other BGLs prediction publications using machine learning models encourage further research in this direction. Achieving the goal of predict accurately the BGLs would undoubtedly mark a before and after in the lives of people with type 1 diabetes.

1.2 Proposed solution

The intricate and non-linear underlying relationship between the BG values and the variability between patients makes BGLs prediction a task that demands sophisticated modeling techniques. In this bachelor thesis, linear models and neural network models are implemented and evaluated on the T1DiabetesGranada dataset. Linear models are convenient because they are easy to implement and relatively quick to train. The motivation for using neural networks is given by the universal approximation theorem, which demonstrates that given enough hidden neurons, a neural network can approximate any continuous function. This ensures that no matter how difficult a patient's BG behaviour is, there is a neural network that can model it with a certain degree of accuracy. In addition, the use of both models on the same data allows for more equal comparisons than could be made with either of state of the art papers.

1.3 Thesis goal

The main objective of this thesis is to develop a Machine Learning model that allows the prediction of blood glucose levels in patients with type 1 diabetes using data from real patients. In order to achieve this objective, the following sub-objectives must first be achieved:

- Investigate type 1 diabetes and the most innovative techniques for predicting BGLs.
- Analyse the longitudinal sensor-based T1DiabetesGranada dataset in terms of its overall and individual characteristics.
- Preprocess the T1DiabetesGranada dataset to facilitate the learning and application of forecasting models.
- Generate various forecast models using different machine learning approaches.
- Evaluate the developed forecasting models.

1.4 Project planning

A general idea of the plan was conceived at the beginning of the thesis. It was modified during the course of the project according to the needs and problems found. These changes were mainly due to the data pre-processing and the experiments took longer than expected. The figure shows the final version of the schedule. Figure 1.1 shows the planned schedule for all the tasks required to complete this thesis.

1.5 Project budget

The basic salary of a data science engineer in Spain is around €23,000 per year. The company that would have this worker on its payroll would have to cover additional costs such as social security for its employees, office rent, electricity, water, computer servers, etc. It would also have to make a profit. Therefore, the company's fees can be assumed to be €50 per hour, so over 7 months, working 40 hours per week, the total cost would be around €56,000.

1.6 Thesis structure

The rest of the thesis is structured in the following chapters:

- **State of the Art:** This chapter provides a general review of the techniques used to address the problem of BGLs prediction, with a particular focus on publications within the last five years.
- **Data Exploration:** This chapter describes the used dataset (T1DiabetesGranada), with particular emphasis on its temporal dimension and the relationship between the BG measurements.
- **Data Preprocessing:** This chapter describes the preprocessing techniques applied on the T1DiabetesGranada dataset.
- **Prediction Models:** This chapter deals with the development, evaluation, and comparison of BGLs prediction models.
- **Conclusions:** This chapter presents the final discussions on the proposed experiments.

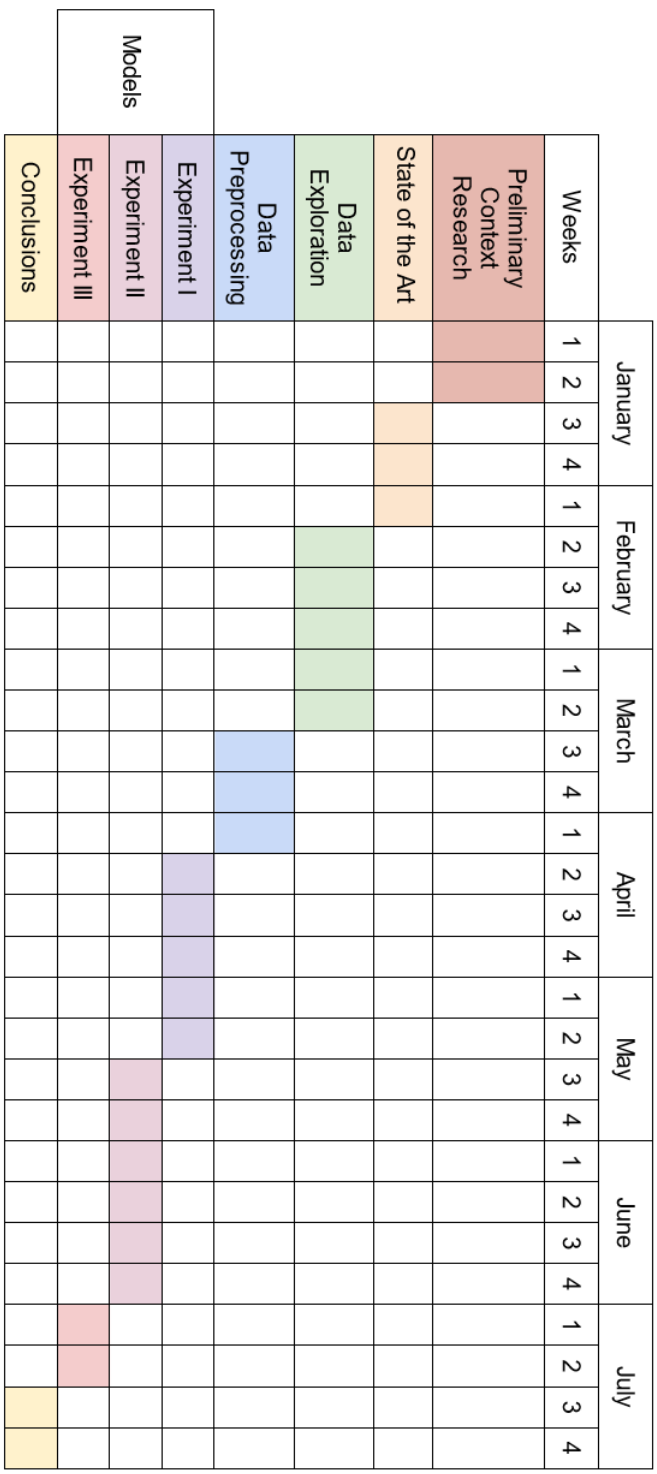


Figure 1.1: Project planification.

Chapter 2

State of the art

2.1 Search methods

To analyse the current state of the art three different approaches were taken: several reviews were consulted, the Web of Science database was searched, and citation search was applied. A total of 22 publications that were found to be relevant and consistent with the present work have been summarised in the Tables 2.1, 2.2, 2.3, 2.4, 2.5, hereafter referred to as state of the art summary tables. The main findings of the state of the art analysis are presented in Section 2.2.

2.1.1 Consulted reviews

Three reviews were considered [3, 40, 47]. They compare relevant scientific publications based on the prediction models and techniques used, input datasets, objectives, results, and limitations. The review by Zhu et al. [47] was published in 2021 and the reviews by Tsichlaki et al. [40] and Alhaddad et al. [3] were published in 2022, so they cover publications up to 2021 and the first part of 2022.

The review [47] includes only publications that use deep learning techniques in the field of diabetes. It shows that the most popular deep learning models to predict futrue BGLs are currently based on convolutional neural networks and recurrent neural networks. It also collects some papers that create ensemble models combining both of them with good results. In addition, this review shows the typical preprocessing techniques applied to the dataset and the main error metrics to compare the results. Finally, it includes a summary of the limitations and challenges identified by the selected articles.

The review [40] includes work on diabetes hypoglycemia prediction, which is a problem within BGLs prediction. Most of the cited papers are treated as classification problems, but there are also some interesting regression models. The machine learning models used in these papers are the same as those used in the papers cited in the review above.

The last one [3] is the most general one. The interest of this review lies in the fact that it goes beyond the field of machine learning, including publications that tackle the problem using statistical prediction methods. This provides information on how this problem is being tackled using completely different models and the results obtained. Taken together, these reviews provide a broad and up-to-date perspective on the progress made in predicting BGLs in patients with type 1 diabetes.

2.1.2 Database searching

In addition to the reviews considered, a search was carried out using a custom query on Web of Science with the aim of finding other publications that might be of interest in understanding the most up to date techniques used to predict BGLs. The query finds the articles (DT) with the expression *blood glucose* in the title (TI) and with the expressions *predict* or *forecast*, and the expressions *neural network* or *deep* or *machine learning* or *nonlinear* in the topic (TS) published in 2022 or 2023 (PY). Those with the terms *type 2 diabetes* or *T2D* or *ECG* (electrocardiogram) in the topic were excluded. Only papers published in 2022 or 2023 were included in the search because the previous years were considered in the consulted reviews.

((TI= ((blood glucose))) AND (TS=((predict OR forecast) AND (neural network OR deep OR machine learning OR nonlinear) NOT (type 2 diabetes OR T2D OR ECG))) AND (PY==("2023" OR "2022"))) AND (DT==("ARTICLE"))

On 6th March, Web of Science returned 12 results. Of these, [9, 42, 45, 22, 41] were rejected because they are clearly not relevant to the topic. [38, 15] were also rejected after reading the abstract and the introduction for the same reason. [18] focuses on BGLs prediction, but the dataset includes only patients with type 2 diabetes, so this paper was also not considered. The 4 remaining results [5, 49, 30, 46] were reviewed and summarized in the state of the art summary tables.

2.1.3 Citation searching

The citation searching method is a research approach that consists of examining the references, citations, or connections of an initial set of publications, in this case consisting of the publications considered in the reviews and the query search, in order to identify additional sources that contribute to the understanding of the research topic. This category includes works considered relevant by the author, previous publications, complementary studies, contributed code, etc. This process helped to ensure the rigour and validity of the articles, as well as to understand the motivation behind specific choices made by the authors. No works found by this method were included in the state of the art summary tables.

2.2 Blood glucose prediction models

The prediction of BGLs plays a critical role in the management and treatment of diabetes by enabling individuals and healthcare providers to anticipate and respond to fluctuations in BGLs. By accurately predicting future BGLs, people with diabetes can proactively adjust their medication, diet and lifestyle to maintain optimal blood glucose control. The following sections describe the progress that has been made in the field of prediction models, the current status of the datasets and the performance achieved.

2.2.1 Prediction models

Over the years, several techniques have been tried to predict BGLs in individuals with varying degrees of success. In [11], Bunescu proposed a physiological model that uses mathematical equations to attempt to model the trend of BGLs. Equation-based solutions can give acceptable results, but their formulation is cognitively demanding and not scalable, as the model has to be redesigned when new features become available. This mathematical model was able to outperformed the physician's predictions. It can also be easily combined with others, as in [13], where it is used to model the effects of carbohydrates and insulin on BGLs in combination with a heuristic search algorithm.

Statistical approaches have also been widely employed. Traditionally, ARIMA has been the most commonly used technique in time series forecasting. In recent years, variants of ARIMA have been used in BGLs prediction, such as [44] which uses ARIMA with adaptive orders to try to capture the non-stationary changes of the BGLs over time. In [31], an ARIMA model was tested on the same dataset as thirty different linear and nonlinear predictive algorithms, including Long Short Term Memory (LSTM) and feed-forward neural networks, outperformed them all with results of 22.15 (mg/dl) RMSE error for 30 minutes prediction. [5, 46, 49] are examples of papers that use other statistical methods. Remarkably, Aljamaan [5] experimented with a long-term prediction horizon of 6 days, while most of the works use 30-60 minute forecast horizons, which are considered rigid and challenging to improve.

In the last few years, the most popular approaches have been based on neural networks. In [39], Idrissi suggests a 1-dimensional filter convolutional neural network (CNN) and tests it with 5 different forecasting strategies. The model outperforms an LSTM model also implemented by the author, obtaining an RMSE of 8.68 (mg/dl) for a 30-minute prediction horizon. In [21], Li proposed a dilated CNN model. Dilated CNNs can increase the receptive field of the network without increasing the number of parameters, allowing them to capture temporal patterns over a wider range of time steps. In addition, Artificial Neural Networks and Autoregressive Neural Networks are examined in [47, 4].

However, among the various machine learning models employed in blood glucose prediction, recurrent neural networks (RNN) have gained significant popularity and are considered the leading approach. RNNs are a powerful technique that implements the concept of "memory" to neural networks. The main problem with classical recurrent neural networks (RNN) is the vanishing or exploding gradient problem [6]. For this reason, most of the publications implement variations of classical RNN units such as LSTM [26, 4, 27, 2, 25, 48, 28, 29] or dilated RNN [48]. The work of Mirshekarian [26] creates a personalized LSTM model per patient, improving the results of the previous work [11] using the same dataset. On the other hand, Aliberti [4] utilizes the LSTM model to study a large and heterogeneous cohort of patients and then applies it to completely new patients. The main objective of this work is to learn a generalizable prediction model that can be easily extrapolated to real-life applications. Several works have taken the theoretical base of LSTM and modified it to try to get stronger models that give better results. Examples include [27] with a Memory-Augmented LSTM that allows searching for similar BGLs trends further back in time, [2] whose LSTM takes into account both past information and the suggested future information as inputs or [29] that combines RNN with Restricted Boltzmann Machines, a type of artificial neural network that is used for unsupervised learning.

Another approach worth mentioning is ensemble models. Ensemble models try to enhance the weaknesses of each algorithm by combining multiple models to improve the overall performance and accuracy of the predictions. In the literature, there are proposals of ensemble models, such as LSTM and bidirectional LSTM models (bi-LSTM) [37, 30], CNN and LSTM [20] or personalized models as the autoregressive multi-output model with polynomial forecasting system [14]. The combination strategy applied to the results can vary widely between ensemble models, [30] explains three different approaches and compares them.

2.2.2 Datasets

The dataset is a key element in the success of a BGLs prediction model. A rich dataset should reflects the BGLs of the patients in the real world and therefore a good result in the dataset should translate into a good result in the real world. It is possible to divide the BG datasets into two broad groups:

1. **In-silico data:** Consists of computer-generated data. The most popular generator is the UVA/Padova T1D simulator [23]. Several papers exploit this generator to complement the real patients' data [37, 20, 26, 21, 2, 48], but others use it to generate the complete input dataset [13, 5].

2. **Real patients data:** Data obtained from real patients under professional supervision using continuous glucose monitoring (CGM) technology. Although the most realistic conclusions can be drawn from this type of data, not all the cited publications have open-source datasets [14, 44, 2], making it difficult to contrast results.

It is important to note that publications that reporting results using simulated and real patient data show that the prediction error is almost certainly significantly lower for simulated data than for real data. This could explain the exceptionally good results in models that use only simulated data as in [5], which presents less than 1 mg/dl RMSE for 15 and 30 minutes estimation windows. Furthermore, in some publications that use real patient data, the dataset is extremely short: 3 days from 12 patients in [46] or 4 days from 9 patients in [28]. This casts doubt on whether the good results could be maintained in larger and more diverse datasets.

Input information may also include complementary information such as carbohydrate measurements (meals), injected insulin, physiological statistics, medical tests, etc [27]. The use of complementary BGLs data can help to understand the factors that contribute to abrupt changes in BGLs levels. It can also improve machine learning models by learning the relationship between these variables and subsequent BGLs, allowing the models to detect patterns and uncover hidden correlations. Nevertheless, one of the challenges of using complementary data is the difficulty of obtaining frequent, aligned and accurate measurements. Regular monitoring of carbohydrate intake, precise recording of insulin administration, and continuous monitoring of physiological statistics require significant effort from patients and healthcare providers. Moreover, medical tests may not be always available or accessible. These practical limitations can impact the quantity and quality of complementary data that can be incorporated into BGLs studies. State of the art publications that add complementary measurements

typically use insulin pumps and sensor bands [47].

2.2.3 Models perfomance

In order to measure model performance, it is important to use common evaluation metrics. As can be seen from the state of the art summary tables, the most popular loss function for BGLs prediction is the Root Mean Square Error (RMSE). The RMSE is a useful loss function when large errors are more costly and it is important to minimize them. In addition, the RMSE is expressed in the same units as the measurements, making it a useful metric for direct interpretation of the error. Another widely used evaluation tool in the BG field is the Clarke Error Grid Analysis (EGA) [12]. The EGA is specifically designed to assess the accuracy of BGLs estimates. The most important aspect of the EGA is its ability to determine the clinical significance of the difference between the estimated BG value and the actual BG value.

Once the error metrics have been established, the prediction horizons over which the model performance is measured must be defined. The use of prediction horizons of 30 and 60 minutes is generally accepted. According to the current state of the art, predicting blood glucose values 15 minutes into the future is not challenging, and for more than 60 minutes the results are usually too poor to be considered. Every few months, new publications improve the results for 30 and 60-minute horizons, but very few are venturing to increase these prediction horizons. Taking into account the papers from the state of the art that use data from real patients and have at least two weeks of data per patient, the best BGLs prediction models achieves an RMSE between 15 and 20 mg/dL for a 30-minute prediction horizon, and between 24 and 32 mg/dL for 60 minutes [4, 27, 20, 25, 29, 30].

Year	Cite	Models	Dataset	Device Brand	Measures	Kind of Model	Objectives	Error Metric Results
2017	[26]	Long short term memory (LSTM) and a linear layer	400 days of 5 real patients data	-	BG, insulin and meals	Personalized model	Creates a personalized LSTM model that improves author's model proposed in [11]	RMSE using BG: 30 min: 21.4 mg/dL, 60 min: 38.0 mg/dL. RMSE using BG, insulin and meals: 60 min: 37.4 $\pm$ 0.5 mg/dL
2018	[37]	LSTM, bi-LSTM, 4 dense layers	1) GoCarb dataset (20 adults) 2) UVa/Padova simulation of 38-day data for 11 virtual patients	-	BG	General model	Predicts blood glucose levels based on CGM measurements.	RMSE: 15 min: 11.63 mg/dL, 30 min: 21.75 mg/dL, 45 min: 30.22 mg/dL, 60 min: 36.92 mg/dL
2018	[14]	Deep sequential polynomial multi-output (ensemble model)	3 years for 40 patients (nearly 550,000 measurements)	-	BG	General model	Compare multi-output and single-output prediction prediction strategies	Absolute percentage error: 30-min: 4.87%
2020	[20]	CNN, LSTM, 2 fully connected (FC) layers	1) 10 Uva/Padova patients 2) 6-month data of 10 real patients	Dexcom G4 Platinum CGM sensors	BG, insulin and meals	-	Predicts blood glucose levels based on CGM measurements	RMSE: 30 min: 21.07 mg/dL, 60 min: 33.27 mg/dL
2019	[4]	Non-linear autoregressive neural network, LSTM and FC layer	RT_CGM dataset (451 patients)	Abbott Diabetes, DexCom and Medtronic Enlite	BG	General model	Learns a generalizable blood glucose level prediction model from a multi-patient training	RMSE: 30 min: 19.47 mg/dL, 45 min: 26.47 mg/dL, 60 min: 32.38 mg/dL

Table 2.1: Summary of publications on BGL prediction models (Part 1).

Year	Cite	Models	Dataset	Device Brand	Measures	Kind of Model	Objectives	Error Metric Results
2019	[27]	Memory-augmented LSTM	1) OhioT1DM dataset: 8 weeks of 6 real patients data 2) 24 hours of 40 simulated patient data with AIDA 3) 90 days of 10 simulated patients with UVa/Padova	Medtronic Enlite® sensors	BG, meals, insulin, heart rate, skin temperature, skin conductance, time of day	Personalized model	Demonstrates that LSTM models are robust to noise and can easily incorporate additional features without any change in the architecture	RMSE using BG, meals, insulin: 30 min: 18.07 mg/dL, 60 min: 28.28 mg/dL RMSE using all features: 30 min: 17.99 mg/dL, 60 min: 28.20 mg/dL
2020	[21]	Dilated convolution neural network	1) UVA/Padova 180 days of 10 patients 2) ABC4D: 6 weeks of 10 real patients 3) OhioT1DM dataset: 8 weeks of 6 real patients data	Medtronic Enlite® sensors	BG, meals, insulin and sleep	Personalized model	Predicts probabilistic distribution of short-term glucose values.	RMSE in ABC4D: 30 min: 19.19 $\pm$ 2.74 mg/dL, 60 min: 31.78 $\pm$ 3.85 mg/dL RMSE in OhioT1DM: 30 min: 19.29 $\pm$ 2.76 mg/dL, 60 min: 31.83 $\pm$ 3.49 mg/dL
2020	[2]	2 branches of LSTM cells (past and future information)	1) UVa/Padova T1D simulator (100 adults) 2) 1 real patient dataset	G4 Platinum CGM system, Dexcom Inc.	BG, insulin, meals	General model	Forecast glucose levels using past features (BG, insulin, meal) and future features (suggested insulin and meal future information).	RMSE for real patient: 30 min: 21.09 mg/dL
2020	[25]	LSTM and 2 FC layers	OhioT1DM dataset	Abbott FreeStyle	BG	Personalized model	Predicts blood glucose levels based on CGM measurements.	RMSE: 30 min: 18.867 mg/dL, 60-min: 31.403 mg/dL

Table 2.2: Summary of publications on BGL prediction models (Part 2).

Year	Cite	Models	Dataset	Device Brand	Measures	Kind of Model	Objectives	Error Metric Results
2013	[11]	Support vector regression (SVR)	5 patients dataset	CGM device	BG, insulin, meals	Personalized model	Uses a generic physiological model of blood glucose dynamics to generate informative features for a SVR model that is trained on patient specific data	RMSE: 30 min: 19.5 mg/dL, 60 min: 35.7 mg/dL
2017	[13]	Hybrid model: physiological models and grammatical evolution metaheuristic	UVa/Padova 14 days of 100 virtual patients	-	BG, insulin, meals	Personalized model	Designs a hybrid approach comprising physiological models for insulin and grammatical evolution with predictions for 120 minutes	RMSE in nocturnal segment using only simulated measurements: 120 min: 11.80 mg/dL
2020	[47]	Artificial Neural Network LSTM	12 patients from OhioT1DM dataset	Medtronic Enlite® sensors	BG	Personalized model	Introduces a deep learning framework for edge inference on a microcontroller unit to predict BG levels	RMSE using ANN: 30 min: 19.81 ± 2.13 mg/dL, 60 min: 33.58 ± 3.61 mg/dL LSTM: 30 min: 19.10 ± 2.04 mg/dL, 60 min: 24.25 ± 2.84 mg/dL
2020	[48]	Dilated recurrent neural network	1) UVA/Padova 360 days of 10 patients 2) 8 weeks of 6 real patients from OhioT1DM dataset	Medtronic Enlite® sensors	BG, insulin, meals	Personalized model	Predicts blood glucose levels based on CGM measurements	RMSE: 30 min: 18.9 ± 2.6 mg/dL
2020	[28]	LSTM	1) AIDA 40 simulated patients 2) DINAMO dataset 4 days of 9 real patients	Zephyr BioHarness 3	BG, insulin, meals	-	Predicts blood glucose levels based on CGM measurements	RMSE: 30 min: 6.42 mg/dL, 60 min: 11.35 mg/dL

Table 2.3: Summary of publications on BGL prediction models (Part 3).

Year	Ref	Models	Dataset	Device Brand	Measures	Kind of Model	Objectives	Error Metric Results
2019	[44]	ARIMA with adaptive orders	From 1.3 to 7 days of 100 patients with T1D (49 patients) and T2D (51 patients)	Glutalor CGM DS-02	BG	Personalized model	Prevent hyperglycemia or hypoglycemia	Relative absolute deviation for T1D patients: 30m: 6.27%
2020	[39]	1) CNN, a flatten layer, 2 dense layers 2) LSTM	10 real patients from DirecNetInpatientAccuracyStudy dataset	-	BG	General model	Determining the best configuration of the proposed CNN Determining the best strategy of multi-steps forecasting (MSF) using the obtained CNN for a prediction horizon of 30 min Comparing the CNN and LSTM models for one-step and multi-steps prediction	Best model (CNN) RMSE: 30 min: 8.68 mg/dL
2020	[29]	RNN with restricted boltzmann machines	10 randomly selected patients from a 110 real patients dataset	IoT CGM device	BG	Personalized model	Test a new proposed wearable CGM system to predict the blood glucose level from glucose level history using a method run in the Cloud	RMSE: 30 min: 15.59 mg/dL
2021	[31]	Autoregressive (AR), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), feed-forward neural network, LSTM...	141 T1D patients data	Dexcom G6 CGM sensor	BG	Both	Perform a head-to-head comparison of thirty different linear and nonlinear predictive algorithms using the same dataset	Best model (ARIMA) RMSE: 30 min: 22.15 mg/dL

Table 2.4: Summary of publications on BGL prediction models (Part 4).

Year	Ref	Models	Dataset	Device Brand	Measures	Kind of Model	Objectives	Error Metric Results
2022	[5]	Box-Jenkins model (Statistical method)	UVA/Padova 1 week of 30 patients	-	BG, insulin and meals	General model	Develop an accurate long-term prediction method for blood glucose levels for T1D patients to avoid hypo/hyper-glycemia incidents	RMSE with 1 day of training using simulated measurements: 144 hours: 7.13 mg/dL. RMSE with 3 days of training using simulated measurements: 96 hours: 0.80 mg/dL.
2022	[49]	FEEMD-SSA-KELM model (statistical method)	Open Source: Dataset of the Children's Network Diabetes Research Center 3 randomly selected patients with more than 3 days of measurements each one	Abbott Diabetes, DexCom and Medtronic Enlite	BG	Personalized model	Predicts blood glucose levels based on CGM measurements	RMSE: 30 min: 5.46 mg/dL
2022	[30]	Ensemble model that uses LSTM, bi-LSTM and linear models	OhioT1DM dataset. 8 weeks of 12 patients	Medtronic Enlite® sensors	BG	Personalized model	Predicts blood glucose levels based on CGM measurements	RMSE: 30 min: 19.62 mg/dL, 60 min: 33.45 mg/dL
2022	[46]	CEEMDAN-IBFOGRU model (statistical method)	3 days of 12 patients	MKB0805, YUNKEAR Ltd and YUWELL Ltd	BG	Personalized model	Predicts blood glucose levels based on CGM measurements	RMSE: 15 min: 0.38 mg/dL, 30 min: 0.47 mg/dL

Table 2.5: Summary of publications on BGL prediction models (Part 5).

Chapter 3

Data exploration

3.1 General information

The dataset used is the T1DiabetesGranada dataset [32]. It is a longitudinal multi-modal dataset of type 1 diabetes mellitus dataset that includes over four years of data from 736 T1D patients from the province of Granada in Spain. It provides not only continuous BG levels, but also patient demographic and clinical information. The dataset includes four comma-separated values (CSV) files. The patient_info file contains information about the patients, such as demographic data, start and end dates of BG level measurements and biochemical parameters, number of biochemical parameters or number of diagnostics. The glucose_measurements file contains the continuous BGLs measurements of the patients recorded with the Freestyle Libre 2 sensor [1]. This sensor performs automatic readings of the BGLs every 15 minutes and manual readings of BGLs at any time. The biochemical_parameters file contains data of the biochemical tests performed on patients to measure their biochemical parameters. The diagnostics file contains diagnoses of diabetes mellitus complications or other diseases that patients have in addition to type 1 diabetes mellitus.

In this work, only the glucose_measurements file is used. The file size is 770.000 KB and contains more than 22.5 million BG measurements from 736 type-I diabetes patients. Each measurement is associated with a patient ID, measurement date (YYYY-MM-DD format), and time (24-hour format with minute precision). The glucose levels are measured as natural numbers between 40 and 500 mg/dL, both included. Table 3.1 shows an example of ten rows of the dataset:

Patient_ID	**Measurement_date**	**Measurement_time**	**Measurement**
LIB193263	2020-06-09	19:08:00	99
LIB193263	2020-06-09	19:23:00	92
LIB193263	2020-06-09	19:38:00	86
LIB193263	2020-06-09	19:53:00	85
LIB193263	2020-06-09	20:08:00	85
LIB193263	2020-06-09	20:23:00	87
LIB193263	2020-06-09	20:38:00	88
LIB193263	2020-06-09	20:53:00	93
LIB193263	2020-06-09	21:08:00	106
LIB193263	2020-06-09	21:23:00	134

Table 3.1: Glucose measurements example.

3.2 Time series data characterisation

When working with time series data, it is crucial to analyze and understand the time dimension. Time plays a fundamental role in time series analysis as it provides valuable insights into the temporal patterns, trends and dependencies within the data. This section explores the temporal particularities of the T1DiabetesGranada dataset and their impact on the forecasting process.

3.2.1 Data amount

The data were collected between January 2018 and March 2022. During the experiment, the number of patients was not fixed: new patients joined, some patients left, and others temporarily stopped participating. This has important implications for the T1DiabetesGranada dataset.

1. As the number of patients was not always the same over the months of the study, the number of BG measurements also varied. Figure 3.1 shows a general increase in the number of patients, and therefore the number of measurements, as the study progressed. This variation results in an unbalanced dataset. Figure 3.2 shows the result of counting the total number of measurements per each month over the duration of the experiment. There is a difference of more than one million measurements between the month with more representation (January) and the month with less representation (April).

2. Patients are not aligned in time, i.e. the measurements for different patients do not start or end at the same time. Figure 3.3 shows the average BG value per month for six randomly selected patients. Each point indicates that there is at least one measurement for that patient during that month. Looking at this graph, it is easy to see when the data starts and ends for each patient and if there are any long gaps of more than a month. For example, patient *LIB193278* (orange) has three months of measurements (August 2019, January 2020 and February 2020), while patient *LIB193277* (blue) was part of the experiment from June 2018 to March 2022. In addition, patients *LIB193278* (orange), *LIB193268* (light purple) and *LIB193274* (green) have gaps in their measurements.

3. The total number of BG automatic readings is different for each patient and depends on the time the person was part of the experiment. Figure 3.4 shows that exist a wide difference between the patients with the highest amount of data and the patients with the lowest amount of data.

3.2.2 Measurements time interval

The sensor used to measure the BGLs in the T1DiabetesGranada dataset allows two types of measurements: automatic and manual. Ideally, the sensor should take an automatic measure every 15 minutes and the patient can take manual readings at any time, interspersed with the automatic readings. This would create a dataset with continuous information on BGLs every 15 minutes provided by the automatic readings. However, in the real world the timing of the automatic measurements is subject to variations that break this continuity of the samples separated by 15 minutes. This break in the continuity has some presence in the data, for example, the number of samples with a time difference of 16, 17 or 18 minutes from the previous measurement represents almost the 5% of the total measurements.

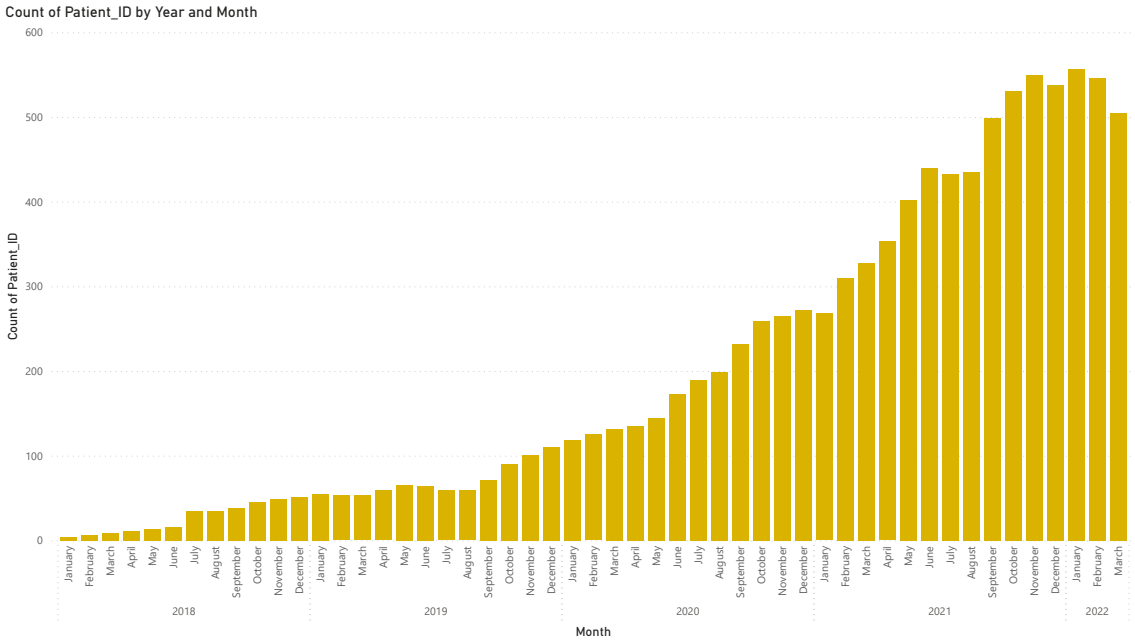
In addition, the T1DiabetesGranada dataset contains missing values which, when extended over time, create gaps. Gaps are temporal discontinuities in the data where BGLs were not recorded. During these gaps, there is a complete absence of information regarding the BGLs. As a result, valuable information is lost, and when data collection resumes, there is a risk that the newly collected data may not align with the previous data timestamp.

Tables 3.2, 3.3 and 3.4 show examples from the real data set where measurement continuity is broken for one reason or another. The columns *t* and *t-1* represent the time at which the measurement was taken and the time at which the previous measurement was taken. The column *Minutes Difference* shows the time difference with the previous sample in minutes. The *Measurement* column shows the BG value in mg/dL.

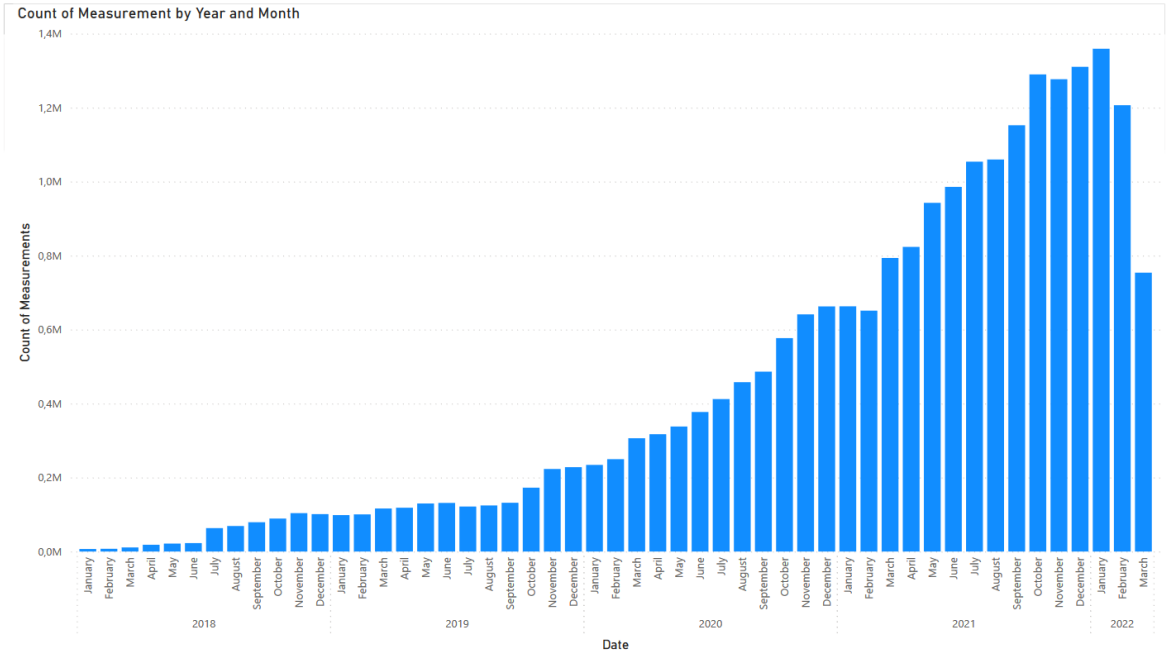
Table 3.2, there are three measurements. They are all automatic measurements as they are separated by 15 minutes or more. The second measurement has a time difference of 17 minutes from the first one, which can be translated into a delay of 2 minutes. This measurement breaks the continuity of the automatic measurements and shifts the rest of the following measurements two minutes into the future.

Table 3.3, there are two automatic readings and one manual reading. The second sample is the manual measurement because the time difference between the third sample and the first sample is 15 minutes. If this measurement were ignored, the time differences within this example dataset would be the expected ones.

Table 3.4 shows an example of a patient with a gap of one and a half hours. This means that the BG value of 5 expected measurements is unknown.



(a) Amount of patients per month and year.



(b) Amount of measurements per month and year.

Figure 3.1: Amount of patients and measurements read over the duration of the T1DiabetesGranada dataset collection.

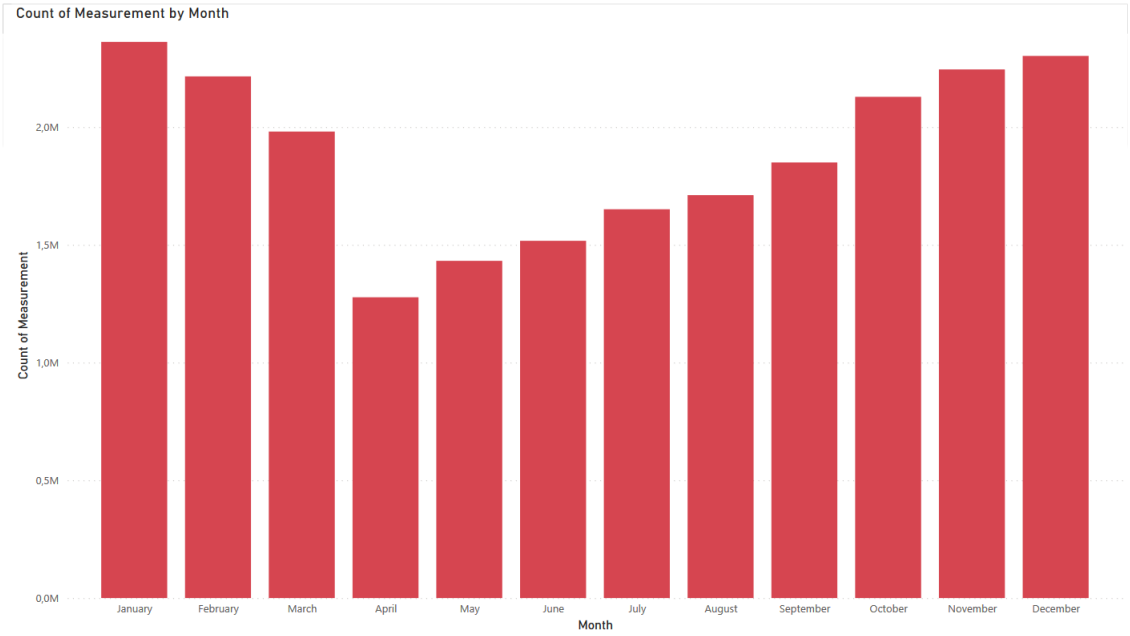


Figure 3.2: Total number of measurements per month obtained by summing the number of measurements for each month across all years of the T1DiabetesGranada dataset.

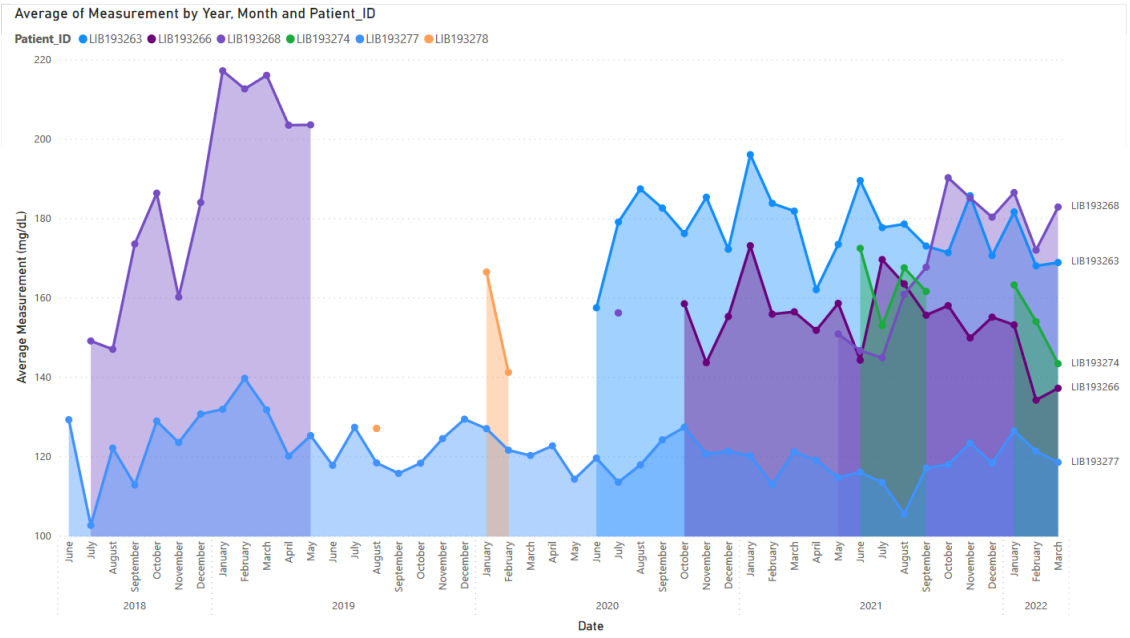
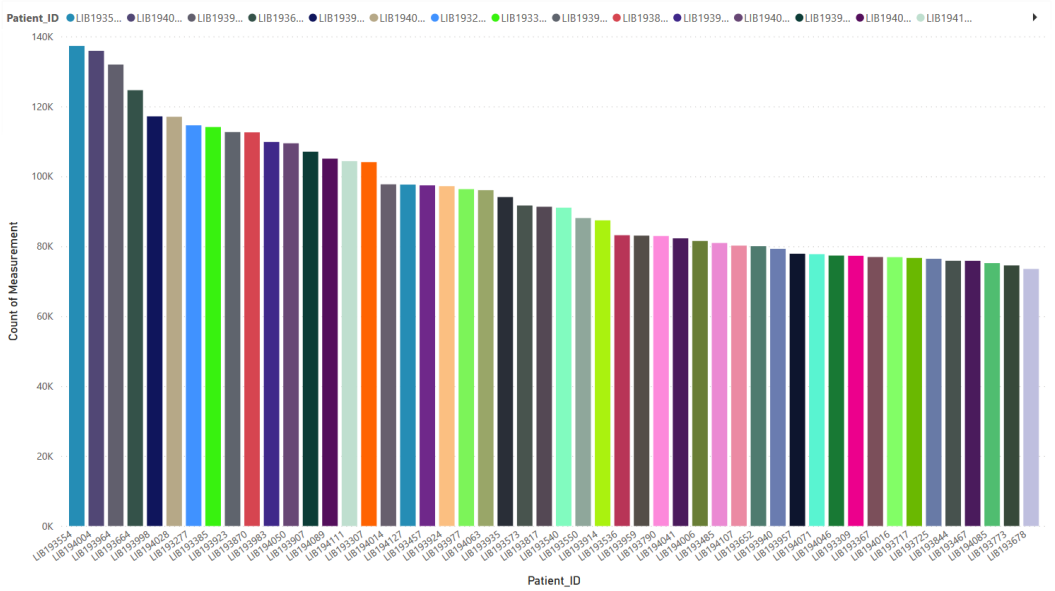
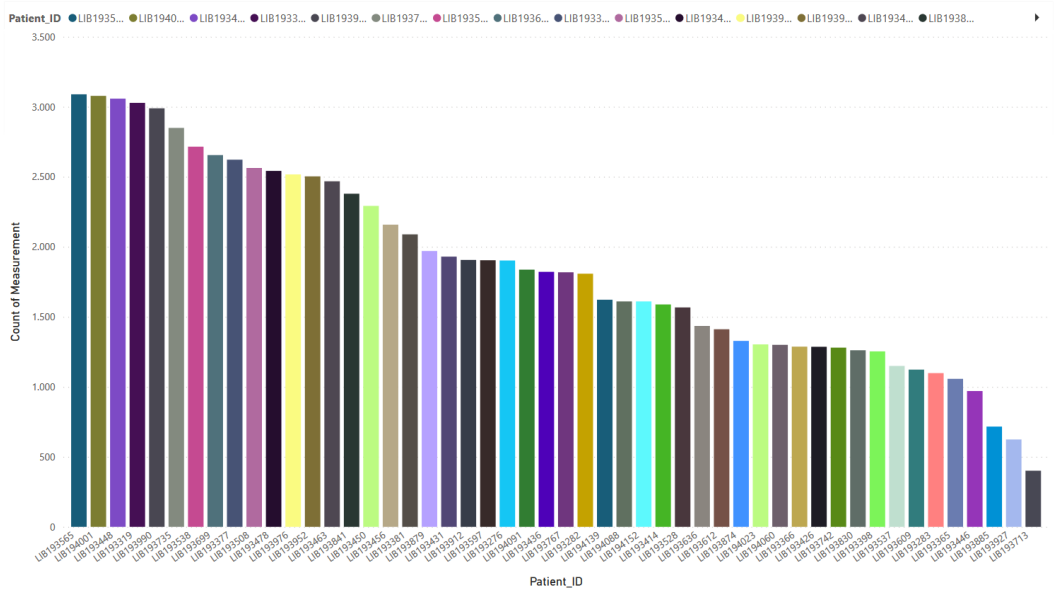


Figure 3.3: Average BGLs per month and year for seven patients.



(a) The 50 patients with the most amount of measurements.



(b) The 50 patients with the lowest amount of measurements.

Figure 3.4: Patients with most and lowest amount of data.

Patient_ID	t	t-1	Minutes Difference	Measurement
LIB193266	02:40:00	02:25:00	15	201
LIB193266	02:57:00	02:40:00	17	200
LIB193266	03:12:00	02:57:00	15	192

Table 3.2: T1DiabetesGranada example of a delay in an automatic sensor reading.

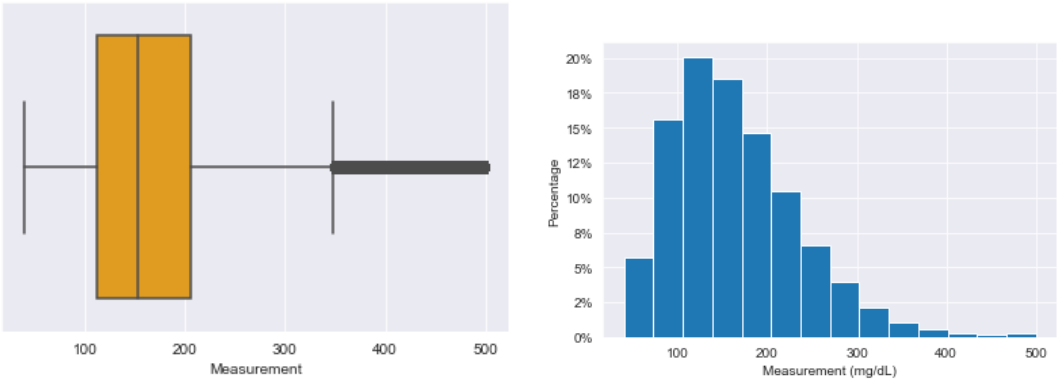
Patient_ID	t	t-1	Minutes Difference	Measurement
LIB193266	07:54:00	07:39:00	15	144
LIB193266	08:00:00	07:54:00	6	144
LIB193266	08:09:00	08:00:00	9	136

Table 3.3: T1DiabetesGranada example of a manual sensor reading.

These irregularities result in a dataset where the time interval between measurements is constantly changing, making it challenging to establish a consistent temporal context for the data. Without a reliable and consistent temporal context, it becomes arduous to identify patterns, trends, or correlations that may exist within the glucose data. It is clear from what has been seen in this section that a resampling is necessary during the pre-processing phase of the data.

3.3 Blood glucose measurements distribution

Figure 3.5 shows the distribution of the BG measurements. Most measurements are close to an ideal glucose range: 70 - 180 mg/dL. As the values deviate further from this range, the percentage of occurrences decreases progressively. This pattern is logical and can be attributed to patients' efforts to maintain their glucose levels within the safe range.



(a) Boxplot.

(b) Percentage histogram.

Figure 3.5: Blood glucose measurements distribution for the T1DiabetesGranada dataset.

However, when the BG measurements distribution is analyzed patient by patient, a different picture emerges. The big problem with the BGLs trend is that it is highly patient-dependent. Each individual's body reacts differently to various factors such as diet, exercise, medication, and general health conditions. So what is normal for one person may not be normal for another. In order to highlight these differences, two analyses have been carried out.

The first anlysis consists in comparing boxplots for patients with a a similar number of measurements without taking into account whether the dates of the measurements are aligned or not. Figure 3.6 shows the boxplots for the selected patients. Figure 3.6a compares patients *LIB193340* and *LIB193315*, both of whom recorded approximately 30.000 samples. Patient *LIB193340* exhibits a significantly longer interquartile range (IQR) compared to patient *LIB193315*, indicating a greater dispersion of BG values. This disparity can be primarily attributed to the fact that the measurements for patient *LIB193340* consistently remain below 300 mg/dL throughout the entire duration of the experiment. Furthermore, the median BG value of the measurements of patient *LIB193315* would be considered an outlier

Patient_ID	t	t-1	Minutes Difference	Measurement
LIB193313	21:25:00	21:10:00	15	224
LIB193313	22:55:00	21:25:00	90	199
LIB193313	23:10:00	22:55:00	15	184

Table 3.4: T1DiabetesGranada example of a gap.

in patient *LIB193340* dataset. This implies that the typical blood glucose level for patient *LIB193315* falls outside the range of normal values observed for patient *LIB193340*. Figure 3.6b compares patients *LIB193340* and *LIB193315*, both of whom recorded approximately about 16.000 measurements. It shows that the IQR for patient *LIB193315* is very close to the ideal range of BGLs. In contrast, the IQR for patient *LIB193340* is completely above this range, with values that could indicate hyperglycaemic states. Patient *LIB193340* may be considered abnormal or atypical compared to the expected values. The existence of such patients poses a challenge to the prediction of BGLs, as their data significantly differs from that of the general patient population. The special characteristics of patient LIB193340 may require specific modeling approaches adapted to their distinct glucose patterns. Figure 3.6c compares the 8 patients with the most measurements, all of them with more than 100.000 samples. This demonstrates that the differences between patients are not due to an insufficient number of measurements, and remains also for patients who stayed for a long time in the data collection.

The second analysis is performed using time-aligned measurements from randomly selected patients. Figure 3.7 illustrates that the differences between patients persist. The results of the two analysis suggest that the observed variations in BGLs distribution are inherent to the individual patients themselves, rather than being a result of mismatched data sizes or measurement dates, although these factors may potentially amplify the differences.

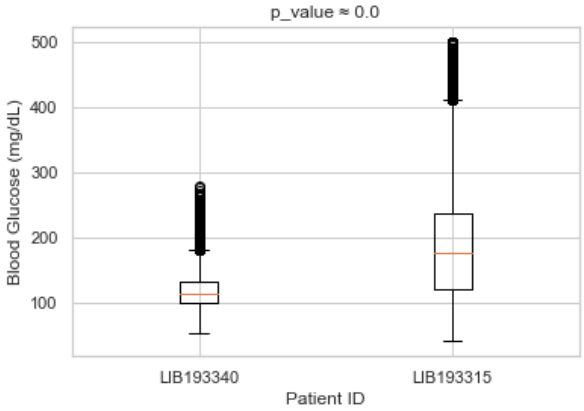
3.4 Lagged blood glucose measurements

In time series prediction problems, the new values are predicted based on the previous values. This implies the existence of a correlation between values along the time dimension. The goal of the prediction algorithms is to model this relationship, which allows the prediction of future values based on the known information. For this reason, it is essential to analyze the autocorrelation of the dataset. In other words, the success of the BGLs prediction depends directly on the correct understanding of how the values of previous BGLs measurements influence the current ones. As stated by Jason Brownlee in *Introduction to Time Series Forecasting with Python*, "Time series modeling assumes a relationship between an observation and the previous observation. Previous observations in a time series are called lags, with the observation at the previous time step called lag1, the observation at two time steps ago lag2, and so on" (Page 48, Section 6.7 Lag Scatter Plots) [10].

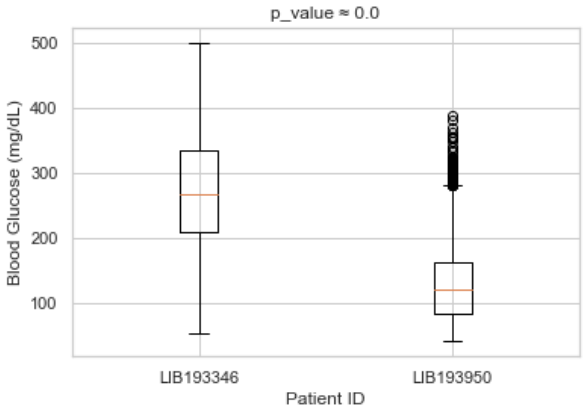
The autocorrelation function (ACF) measures the correlation between a time series and its own lagged values. It quantifies the relationship between measurements at different time lags, indicating the strength and direction of the dependence. The partial autocorrelation function (PACF) also measures the correlation between a time series and its lagged values, but the PACF captures the unique correlation between the two measurements, removing the indirect effects of all other intermediate lags. These functions cannot be plotted for the full range of lagged measurements, as the number of lagged values would be too large to be visualised. Therefore, a threshold of 50 lagged samples were selected to explore the correlation of the BG values.

As it would be expected, Figure 3.8 shows that the autocorrelation is different for each of the patients. There are patients like *LIB193264* who lost the correlation of BG measurements prematurely. This indicates a lack of long-term dependence in the values. On the other hand, patients like *LIB193361* that have a high correlation during all the selected lagged samples. This may be one of the indicators of how difficult is the prediction of future values for a patient, since the ability to accurately predict future values relies on the presence of persistent patterns and dependencies in the data.

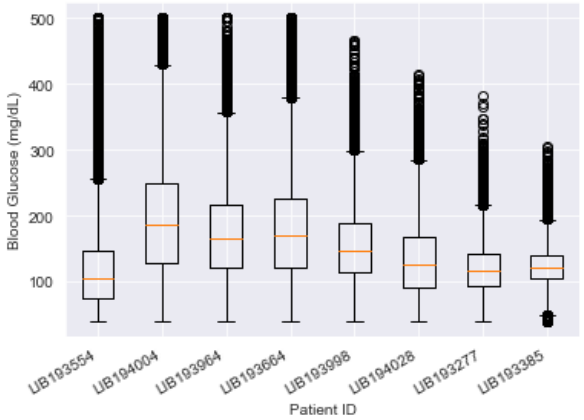
Figure 3.9 shows that the PACF shows that for almost all patients the partial autorrelation persists between 3 and 10 lagged samples. To ensure the best performance of the prediction model, the selection of the correct number of lagged values is essential. The use



(a) Boxplots of patients *LIB193340* and *LIB193315*.



(b) Boxplots of patients *LIB193346* and *LIB193950*.



(c) Boxplots of the 8 patients with the most measurements.

Figure 3.6: Blood Glucose measurements distrubution for patients with similar number of measurements.

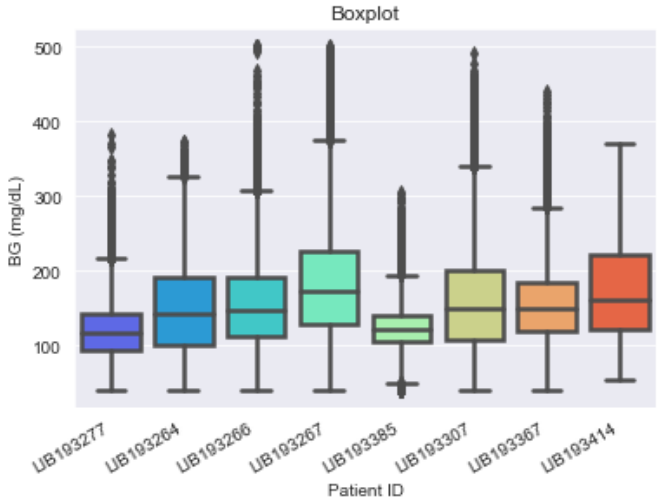


Figure 3.7: Boxplots for 8 randomly selected patients for March 2020.

of an insufficient number of lagged values as model input will limit its ability to capture the relevant patterns and dynamics of BGLs. A model that fails to capture the complexity of the data can easily lead to underfitting. On the other hand, using an excessive number of lagged values can introduce noise and redundant information into the model. This may lead to overfitting, where the model is too closely tailored to the training data and performs poorly on new, unseen data. Therefore, finding the right balance is crucial, and the analysis of the results of these plots is a good starting point.

In addition, some of the patients exhibit significant long-term correlations as shown in Figure 3.10. Patients *LIB194063* and *LIB193659* show relationships with values of 8-9 hours in the past. Patient *LIB193279* with values from 12 hours before. The existence of these correlations for some of the patients could be exploited using models that can capture long-term dependencies.

However, the most important information provided by the graphs is undoubtedly the presence of autocorrelations of lagged values outside the confidence interval cone (the light blue zone). This observation demonstrates that the BGLs time series studied in this problem is not a white noise series. A white noise series is a sequence of unconnected values that cannot be predicted. The fact that the autocorrelations extend beyond the lagged values indicates the existence of dependencies between the BG measurements. These dependencies suggest that the changes in BGLs are governed by underlying patterns and ensure that, if these patterns are captured, prediction of BGLs is possible for this T1DiabetesGranada dataset.

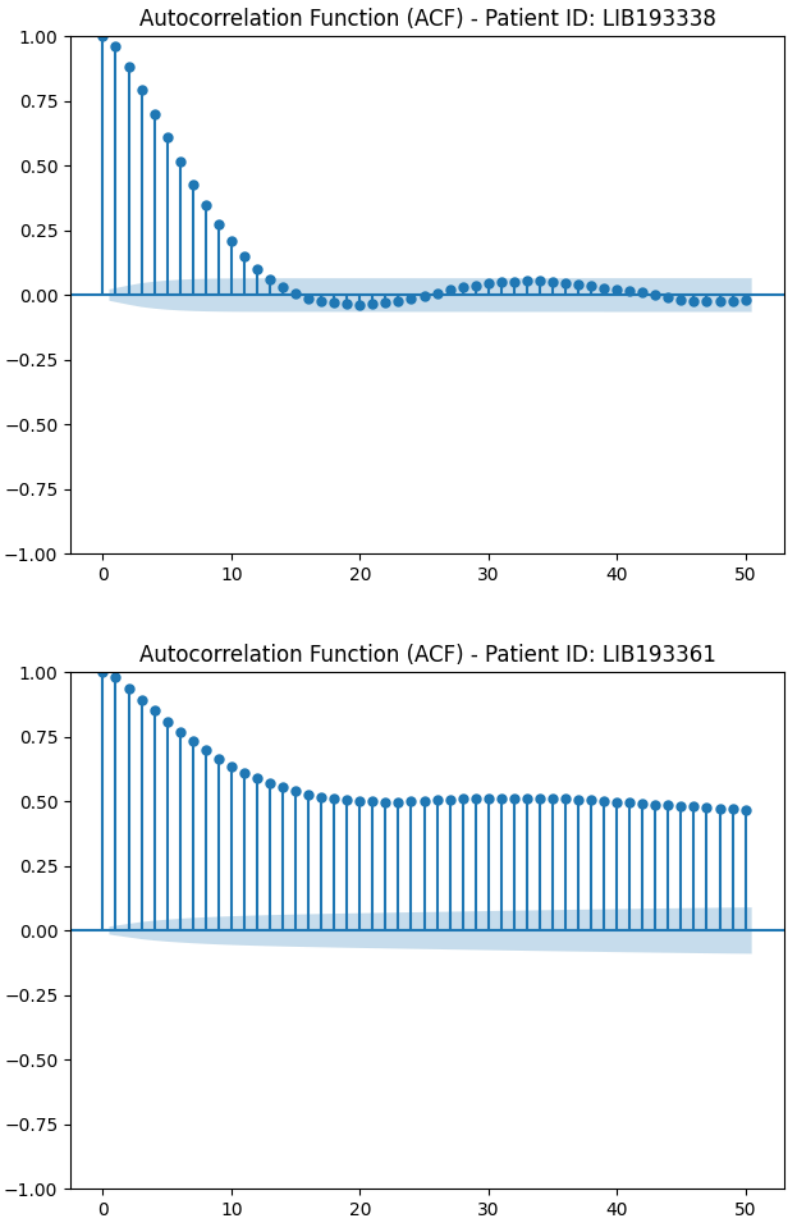


Figure 3.8: Autocorrelation Functions (ACF) for patients *LIB193338* and *LIB193361*.

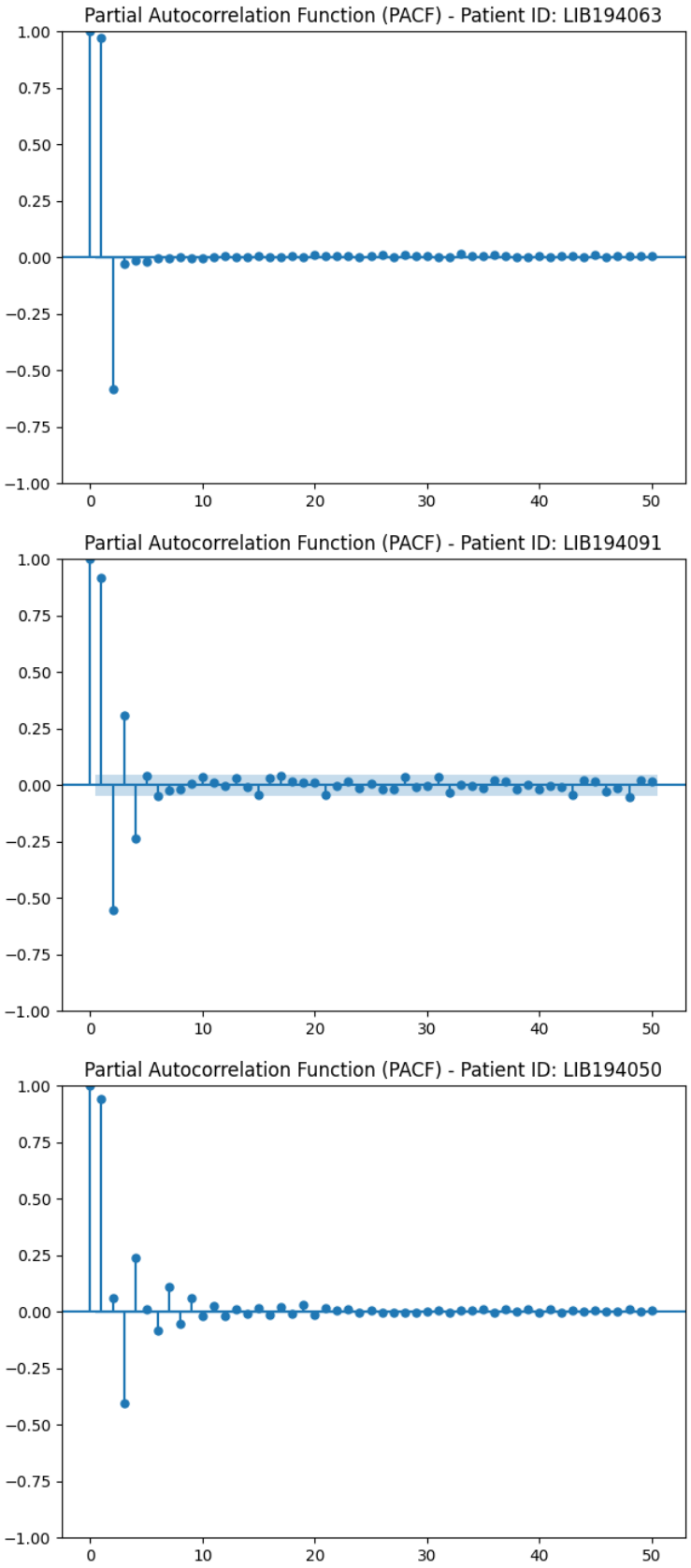


Figure 3.9: Partial Autocorrelation Functions (PACF) of patients *LIB194063*, *LIB194091* and *LIB194050*.

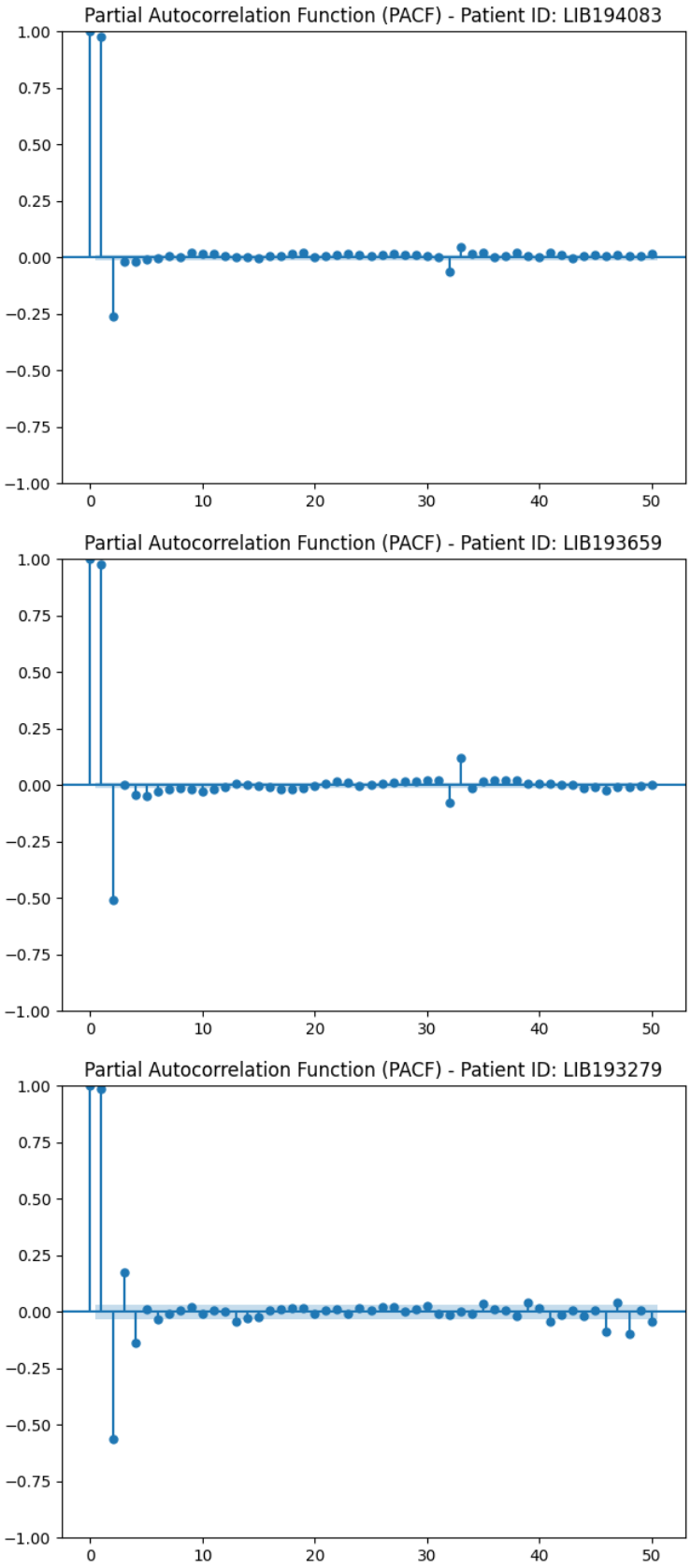


Figure 3.10: Partial Autocorrelation Functions (PACF) of patients *LIB194083*, *LIB193659* and *LIB193279* with long-term correlations.

Chapter 4

Data preprocessing

The data preprocessing summary tables (Table 4.2 and Table 4.3) show the preprocessing techniques used and the models employed for some interesting papers selected from papers that were considered in the state of the art. Those preprocessing techniques are discussed below.

4.1 Removing duplicate samples

A duplicate sample refers to a reading with the same measurement time and BG value as another sample for a patient. No duplicate samples were found in T1DiabetesGranada dataset.

4.2 Removing outliers

The sensor measures BGLs between 40 and 500 mg/dL. If the BGLs of a patient were lower than 40 mg/dL or higher than 500 mg/dL, the device would register 40 and 500 mg/dL respectively. These two values are extreme and highly unusual, but possible. Therefore, it is difficult to draw conclusion considering BG values in isolation. For this reason, the only way to find outliers is to look at the relationships between measurements. In order to find outliers, two different strategies have been discussed: rate of change of blood glucose values and extreme blood glucose values.

4.2.1 Rate of change of blood glucose values

By looking at the relative differences between measurements, it is possible to understand how BGLs change over a known period of time. It is expected that with a fixed period of time between measurements, the rises and falls in BGLs will follow some rules, such as being below a maximum threshold of change. For example, it would not be feasible to consider a blood glucose reading of 40 mg/dl 15 minutes after a reading of 500 mg/dl for the same patient. Unfortunately, it is incredibly challenging to set a global threshold that separates realistic variations from unrealistic ones and applies to all patients.

Figure 4.1 represents how much the BGLs change with respect to the previous BG measurement, using a maximum time between measurements of 18 minutes (15 minutes plus 3 minutes of affordable delay). It shows that almost all differences between measurements are small, below 60 mg/dL. This gives an overall picture of BGLs behaviour, but by the moment no conclusions can be obtained about the outliers. According to experts at the Clinical Unit of Endocrinology and Nutrition of the San Cecilio University Hospital of Granada (Spain), the changes in an average patient can reach 100 mg/dL over a period of 15 minutes, and even more abrupt changes in extreme cases such as an overdose of the medical prescription to combat hypoglycemia or hyperglycemia.

Table 4.1 shows the percentiles of the differences between the BG measurements and their previous measurement. $\mathbf{P}_{99.00}$ reaches 50 mg/dL. In order to reach the value assumed by doctors (100 mg/dL), $\mathbf{P}_{99.98}$ must be calculated. There are 5,461 consecutive samples with differences from the previous sample greater than 100 mg/dL, which is less than the 0.01% of the total dataset. These results suggest that the BGLs tendency of the dataset is consistent

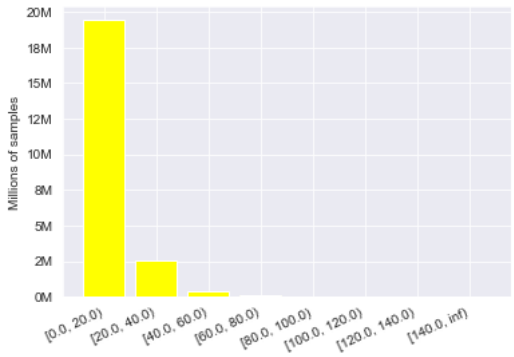


Figure 4.1: Number of samples per range of relative variation (mg/dL) between the measurements and the previous measurements.

Percentile	BG difference (mg/dL)
P50.00	7
P75.00	13
P99.00	52
P99.98	108

Table 4.1: Percentiles of the relative variation between BG measurements and the previous measurements.

with the medical assumptions, which means that this strategy has not found any clear outliers.

4.2.2 Extreme blood glucose values

In this section, the distribution of the BG values is examined in order to find outliers. This can be analyzed using the boxplots discussed in the data exploration section (see Figure 3.6). They show values that could be the considered outliers for each of the patients, representing them as a circles. It can be clearly seen that all these suspected outliers are inside or at least really close to hypoglycemia (less than 70 mg/dL) or hyperglycemia (more than 250 mg/dL) scenarios.

The problem with this approach is that the prediction of BGLs is particularly important when the patient is in any of these two scenarios because their health is at risk. If these suspected outliers were removed, much of the information from these scenarios would be lost. It is likely that removing values near the extremes of the sensor readings would contribute to better performance for the prediction model, but this better performance is less important than the information that these extreme values can provide. The use of additional information such as meals, exercise, or insulin could help to judge whether these measurements should be considered as outliers, but in this case this information is not available so no outliers were removed.

4.3 Resampling

Resampling is required because as seen in data exploration the time between consecutive samples is not always the same. As mentioned before, the sensor takes an automatic reading every 15 minutes, but patients can take manual readings at any time by scanning the sensor, which adds measurements between automatic readings. In addition, the dataset contains time variability in these automatic measurements and gaps due to complications during data collection, such as sensor failures, delays, sensor running out of battery, or patients leaving the study and then returning. Whatever the reason, the used prediction models need equal time differences between measurements to give consistent predictions.

After the preprocessing, the dataset should have a sample every 15 minutes with no samples in between. This can be understood as a downsampling process to convert 15-minute intervals with more than one sample to 15-minute intervals with only one sample and an

upsampling process to create empty measurements for missing values. The code of the resampling function can be seen in Listing 7.1.

4.3.1 Downsampling multiple measurements in the same interval

The downsampling process removes any manual readings taken in between the 15 minute automatic readings and realigns the readings that had small time variations. The sensor has a precision of minutes, so for these readings that were taken at the same minute but different seconds, the last one would be selected to represent the BG measurement. Please note that this case is not a duplicate measurement because the time of the measuremts is equal, but the BG value is not.

Figure 4.2 shows two examples of the resampling process. In example I, the pink circles represent four different measurements taken sequentially. A timestamp of 15 minutes is created from the first measurement. The first, the second and the fourth measurements match with the timestamp, so they are considered to be automatic measurements. The third measurement is a manual reading between the second and third marks of the timestamp. In the downsampling process this third measurement would be removed and the other three measurements would be included in the dataset. The Example II represents three automatic measurements, but the last two of them have small time variations and they fall into the same interval. In this case, the strategy would be to find the closest measurement to the timestamp. Both would be found and the three values would be realigned.

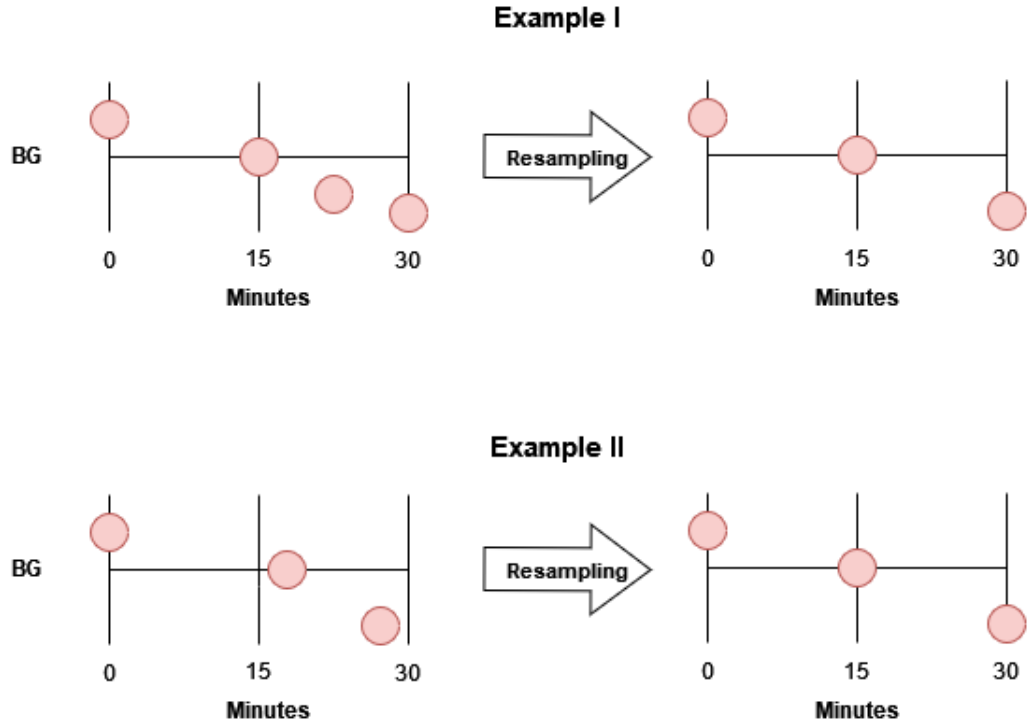


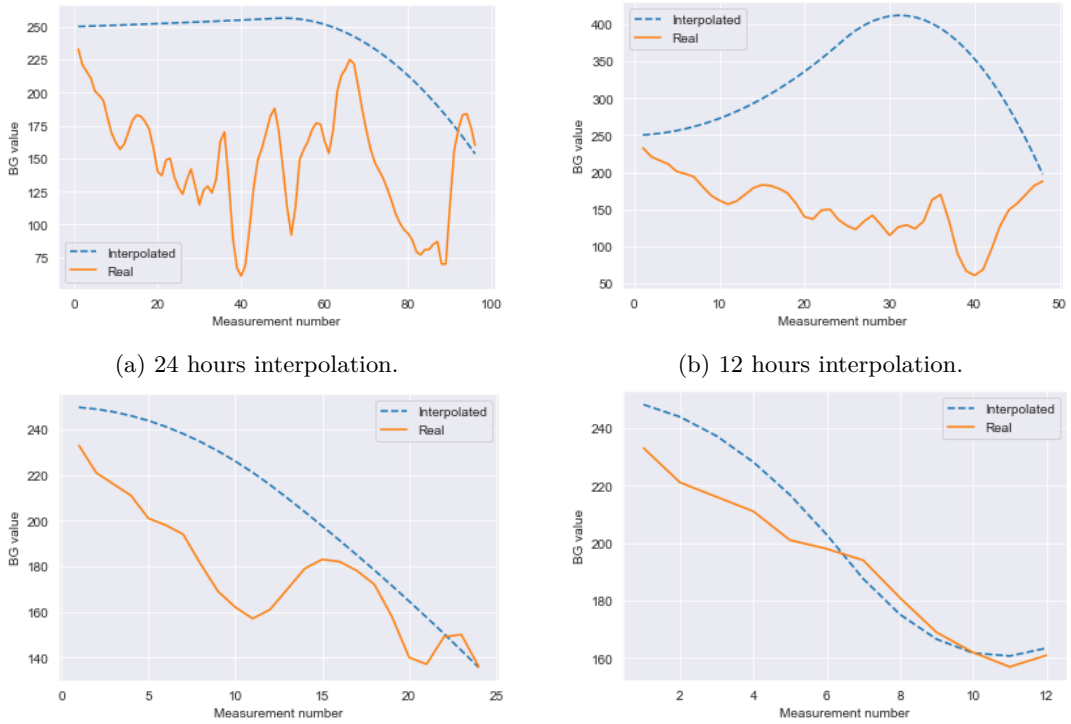
Figure 4.2: Examples of the resampling process. In example I, a set of measurements with automatic and manual measurements is resampled. In example II, a set of measurements with automatic measurements recorded with time variability is resampled.

4.3.2 Upsampling missing values and gaps treatment

The upsampling process is more complex to manage because it involves dealing with a loss of information. In the literature, the two main strategies followed for dealing with gaps are to simulate the missing values or to split the dataset into sub-datasets. Sometimes both of them are combined. These techniques are not ideal and their use should be carefully considered, as they can strongly condition the results.

The simulation of BG measurements implies the use of algorithms to assign values to the missing measurements. The problem with simulating missing data is that the techniques

are not precise enough and the BGLs can change rapidly and non-linearly. Therefore, the simulated data is likely to have little to do with the missing real data. In addition, using values generated by algorithms can lead to overfitting. If the simulation of missing values is carried out, it should be as short as possible. Long simulations produce trends that are completely different from the real ones because the error is being added to each simulated measurement. Figure 4.3 shows an example of interpolation using the 2nd-order polynomial method for a random patient of the dataset. The results for simulations of 24 and 12 hours are poor. The results improve as the interpolation time decreases, obtaining moderate results for simulations of 6 hours (24 measurements) and good results for 3 hours (12 measurements).



(a) 24 hours interpolation.

(b) 12 hours interpolation.

(c) 6 hours interpolation.

(d) 3 hours interpolation.

Figure 4.3: BGLs interpolation for patient *LIB193367*.

The other common strategy is to split the dataset when the number of correlative missing values is above a threshold. Spliting a dataset creates two subdatasets, each consisting of a part of the original dataset. The problem with splitting the dataset is that it breaks the continuity of the samples and, the use of multiple datasets for each of the patients makes the BGLs prediction problem more difficult to handle. In [20], the missing values strategy is to interpolate gaps shorter than 1 hour (equivalent to 4 samples in this dataset) and to split gaps longer than 1 hour into two segments. When the same process is applied to this dataset, longer gaps are so common that several subsamples per patient are obtained.

These experiments on interpolation of missing values and splitting the data show that these strategies are not a good approach for this dataset. As a first approach, neither of these strategies was followed. Instead, all missing data are kept as unknown samples, i.e. the upsampling procedure consists only of generating samples with empty measurements corresponding to the missing values. This involves that any sequence with one or more missing data points will be discarded during model training. This strategy worked well so the use of missing data simulation was not necessary.

4.4 Normalisation and standardisation

The data has not been normalized or standarized. The only information from the dataset used to predict the new BGLs is the history of past BG values. As only one feature is used, it was decided to keep the original values of the BG measurements. This idea is supported by most of the BG publications that predict the values using only the BG information [4, 30, 31, 37, 39, 49].

4.5 Removal of patients with insufficient data

Patients with less than 1 month of data, excluding empty measurements generated by the upsampling process, were excluded. This threshold was chosen because this is the minimum amount of data required to perform the proposed experiments that would allow the model to be extrapolated to a hypothetical real-world application. This is explained in more detail in the Section 5. As a result of this process, the data of 45 patients were removed, leaving 691 patients..

Reference	Models	Preprocessing Techniques
[26]	LSTM with a linear layer	• Interpolation for missing values. • Transform absolute measurement values into the difference between a measurement and its lagged measurement.
[20]	CNN + LSTM + 2 FC layers	• Data alignment. • Data Normalization. • 1D Gaussian kernel filter for missing and outlier data. • Interpolation / Extrapolation for missing values.
[37]	LSTM + bi-LSTM + 4 dense layers	• Outliers. • Interpolation for missing values. • Division into sub-datasets when a gap is longer than 5 days.
[4]	Deep sequential polynomial multi-output (ensemble model)	• Remove sequences with several gaps. • Tikhonov regularization. • Remove short sequences with less than 30 samples.
[21]	Dilated convolution neural network (DCNN)	• Data alignment. • Remove outliers. • Interpolation and extrapolation for gaps shorter than 1 hour.

Table 4.2: Summary of preprocessing techniques used in state of the art publications (Part 1).

Reference	Models	Preprocessing Techniques
[25]	LSTM + 2 FC layers + Output layer	• Scale measurements by multiplying 0.01.
[48]	Dilated Recurrent Neural Network	• First-order interpolation for missing values in training set. • Median filter to remove spikes and outliers in training set. • Extrapolation for missing values in testing set. • Data recombination to increase dataset size.
[39]	CNN + Flatten Layer (FL) + 2 Dense layers, LSTM	• Remove outliers and redundant records.
[31]	AR, ARMA, ARIMA, Support Vector Machine, Random Forest, Feed-forward neural network, LSTM	• Data alignment. • Third-order interpolation for gaps shorter than 15 minutes. • Split into subdatasets for gaps longer than 15 minutes.
[30]	Ensemble model using LSTM, bi-LSTM, and linear models	• Linear interpolation for missing data.

Table 4.3: Summary of preprocessing techniques used in state of the art publications (Part 2).

Chapter 5

Blood glucose level forecasting

The forecasting models used to predict the future BGLs in the T1DiabetesGranada dataset but were inspired by the state of the art forecasting models used in other datasets. There are numerous papers that address the same forecasting problem as this work, using a similar dataset of real patients, and some of them have also published the code associated with their forecasting models. This search for the best forecasting models that can be extrapolated to the T1DiabetesGranada dataset led directly to the Blood Glucose Level Prediction Challenge 2020 (BGLP) [35]

The BGLP is a research competition focused on forecasting BGLs in diabetic patients using machine learning models on a subset of the OhioT1DM dataset [24]. Publications from the BGLP Challenge were reviewed to identify models that could fit with the lines of this work. The OhioT1DM dataset includes relevant patient information beyond the BG values: Insulin information, biometric statistics, meals and exercise, and most of the BGLP2020 models attempt to take advantage of one or more of these additional features. As a result, the models are highly coupled to this additional information. However, only BG measurements are used from the T1DiabetesGranada dataset, so most of these models are difficult to use.

Fortunately, the publication with the paper ID 1 in the BGLP ranking [8] studies the prediction of future glucose levels from historical glucose levels only. Furthermore, Figure 5.2 shows that this model has the second best performance in 30 minute RMSE and the best performance in 60 minute RMSE. It is also the only non-personalized model that is listed in the final ranking. This brings the advantage that once trained it can be used directly with new patients. These characteristics make this model the perfect starting point for the prediction of BGLs on our dataset. From now on, this paper [8] and its prediction model will be referred to as *reference paper* and *reference model*, respectively.

Official ranking (April 26, 2020).

	30 min		60 min				
Paper ID	RMSE	MAE	RMSE	MAE	**Overall**	Online	Personalized
13	18.22	12.83	31.66	23.60	86.31	No	Yes
6	19.21	13.08	31.77	23.09	87.15	No	Yes
16	18.34	13.37	32.21	24.20	88.12	No	Yes
15	19.05	13.50	32.03	23.83	88.41	No	Yes
1	18.23	14.37	31.10	25.75	89.45	No	No
14	19.37	13.76	32.59	24.64	90.36	Yes	Yes
7	19.60	14.25	34.12	25.99	93.96	No	Yes
9	20.03	14.52	34.89	26.41	95.85	Yes	Yes

Figure 5.1: BGLP 2020 general ranking. Source [35].

The reference paper proposes three types of models: A linear model, a feed-forward neural network, and recurrent neural network. The results considered in the ranking challenge are obtained using the recurrent neural model with LSTM type cells. The code of the LSTM recurrent neural network and the linear model, avaliable at [7], have been adapted to T1DiabetesGranada dataset. This will make it possible to see if the good results obtained

RMSE at 30 minutes ranking.

	30 min		60 min				
Paper ID	RMSE	MAE	RMSE	MAE	**RMSE at 30**	Online	Personalized
13	18.22	12.83	31.66	23.60	18.22	No	Yes
1	18.23	14.37	31.10	25.75	18.23	No	No
16	18.34	13.37	32.21	24.20	18.34	No	Yes
15	19.05	13.50	32.03	23.83	19.05	No	Yes
6	19.21	13.08	31.77	23.09	19.21	No	Yes
14	19.37	13.76	32.59	24.64	19.37	Yes	Yes
7	19.60	14.25	34.12	25.99	19.60	No	Yes
9	20.03	14.52	34.89	26.41	20.03	Yes	Yes

RMSE at 60 minutes ranking.

	30 min		60 min				
Paper ID	RMSE	MAE	RMSE	MAE	**RMSE at 60**	Online	Personalized
1	18.23	14.37	31.10	25.75	31.10	No	No
13	18.22	12.83	31.66	23.60	31.66	No	Yes
6	19.21	13.08	31.77	23.09	31.77	No	Yes
15	19.05	13.50	32.03	23.83	32.03	No	Yes
16	18.34	13.37	32.21	24.20	32.21	No	Yes
14	19.37	13.76	32.59	24.64	32.59	Yes	Yes
7	19.60	14.25	34.12	25.99	34.12	No	Yes
9	20.03	14.52	34.89	26.41	34.89	Yes	Yes

Figure 5.2: BGLP RMSE rankings. Source [35].

by this model in the challenge will hold on a different and larger dataset.

5.1 Theoretical framework

5.1.1 LSTM model

Long Short Term Memory (LSTM) neural networks models are the most popular models for predicting BGLs in the current state of the art. The LSTM neural network is an evolution of the classic Recurrent Neural Network (RNN). An RNN is a type of neural network suitable for sequential data such as a time series. RNN is very powerful in theory, but it has some issues that make harder the learning over long sequences, such as the vanishing gradient problem [19, 6]. The LSTM network solves the problems of the RNN by ignoring useless information in the network.

Figure 5.3 represents an scheme of a LSTM cell. The LSTM cell is defined by the following equations (5.1) to (5.6), where σ is the sigmoid activation function that allows the gates to have a value between 0 and 1, the W_z is the corresponding weight matrix for the respective gate (z) neurons, x_t is the input vector,and b_z is the bias for the respective gate (z). In the notation used, the subindexes can refer to the gate, i for the input gate, f for the forget gate and C for the control gate, but also to the timestep, t for the current timestep and $t-1$ for the previous one. The hidden state (h_{t-1}) referred for all the equations is calculated in the previous timestep using equation (5.6).

The input gate (i_t) decides which information will be transferred to the cell. It is defined as:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.1}$$

The forget gate (f_t) decides which information from the input should be forgotten. It is defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.2}$$

The control gate (C_t) controls the update of the cell state from the last timestep C_{t-1} to the current one C_t, based on the following equations:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5.3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{5.4}$$

The output gate (o_t) is responsible for generating the output and updating the hidden state (h_t) that will to all the gates in the next timestep. This process is defined as:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5.5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{5.6}$$

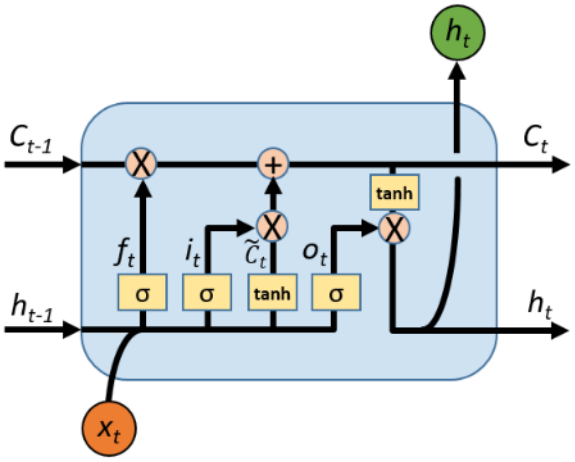


Figure 5.3: Structure of the LSTM cell. Source [37].

5.1.2 Error metric

The models that will be proposed in this section were all trained to minimize the root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5.7}$$

where $\hat{y}_i$ is the predicted value, y_i is the real value and n is the total number of measurements in the used set. This error metric was chosen because it is used in most state of the art papers, including the reference paper. This metric provides a non-clinical measure of how well a model fits a set of data, which is directly interpretable as its units are mg/dL.

5.1.3 Clinical evaluation

The Clarke Error Grid (CEG) [12] is the most widely accepted method for evaluating the accuracy of BGLs prediction models. Unlike the RMSE metric, which is a context-blind measure of error, this metric is an assessment of how harmful an error could be to a patient. It is typically represented by a scatter plot that compares the real measurements with the predicted measurements (see Figure 5.4). The plot is divided into five regions (A, B, C, D, and E) based on the clinical impact of the prediction error:

- **Zone A**: This zone represents clinically accurate predictions. Includes points with no expected impact on treatment decisions. Contains predicted values that are within the 20% of the real measurement.

- **Zone B**: This zone indicates benign errors that may lead to altered treatment decisions but are unlikely to result in severe consequences. Includes points that are outside the 20% of the real measurement.
- **Zone C**: This zone indicates errors that have the potential to lead to unnecessary treatment or failure to treat.
- **Zone D**: This zone indicates dangerous errors which may lead to inappropriate treatment. Include points that fail to detect potentially hypoglycaemic or hyperglycaemic events.
- **Zone E**: This zone represents critical errors that endanger patient health. Contain points that confuse hypoglycaemia and hyperglycaemia treatments.

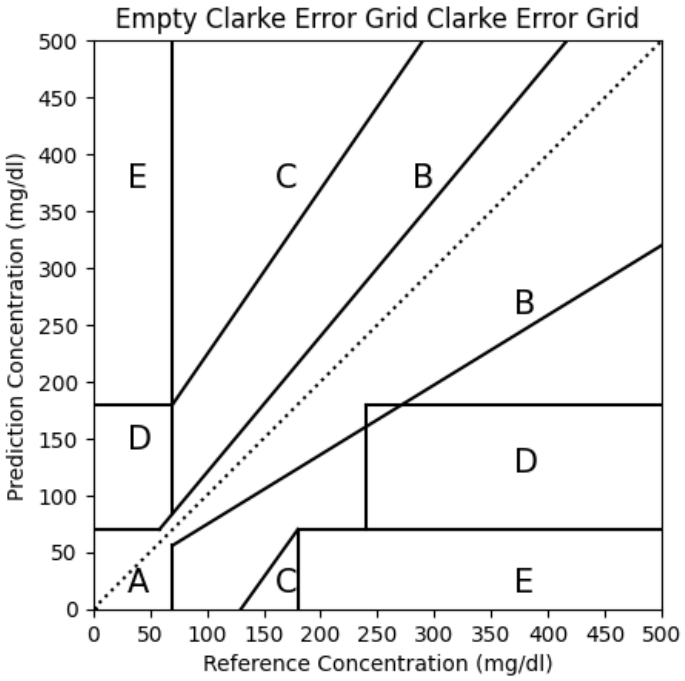


Figure 5.4: Clarke Error Grid. Source [36].

5.2 Models parameterization

5.2.1 Sliding window

The dataset must be restructured as a supervised learning problem in order to use a model to predict future BGLs. This means that the data must be reorganized as input sequences of measurements and their corresponding outputs or labels. The simplest strategy to do this is to use all the measurements that are before the value to be predicted as input data and the value to be predicted as the output or label. Figure 5.5 shows an example of this technique. Each of the cells in the figure represents a measurement of the dataset. Each of the windows is an input-output pair formed by the selected measurements, where the orange cells are the input sequence and the purple cell is the desired output. It is important to note that the input values (also called the history length) increase with each new prediction.

The problem with this approach is that the length of the input sequence will be so long that the prediction cannot be calculated at some point. To solve this problem, it is necessary to set a common size for all the windows. The size of the window is determined by the history length plus the prediction horizon. The prediction horizons are established by the state of the art publications: 30 and 60 minutes, corresponding to samples 2 and 4. The selection of the history length was made by a tuning process that is explained in section 5.2.4. The length of the input data will also determine the number of measurements required before the first prediction. This loss of information is not a problem in this case because the usual window sizes are much smaller than the amount of data. The total number of windows is also reduced by

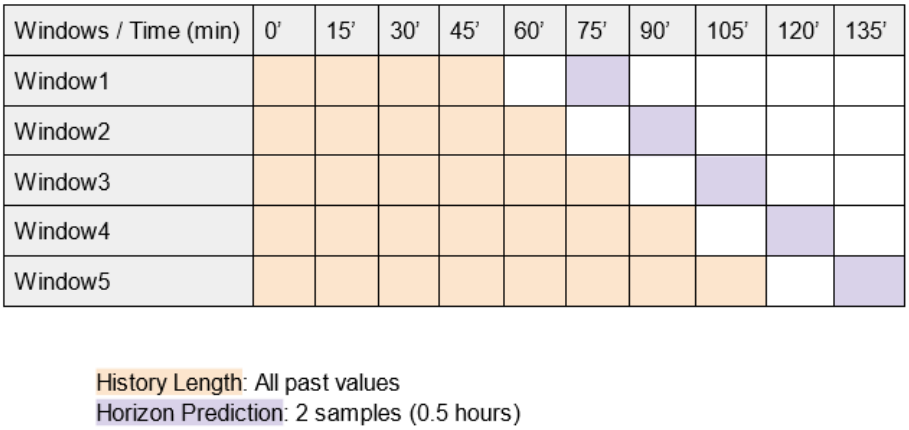


Figure 5.5: Sliding window scheme for a cumulative history length without maximum window size.

the number of empty measurements, as the windows containing empty measurements, either in the history length or in the prediction horizon, are discarded.

Another important decision is how the window slides along the dataset, i.e. how many samples are between one window and the next. Using a small scroll allows many windows to be created, giving our model more opportunities to learn. On the other hand, overlapping measurements in different windows could cause overfitting problems, but the rapid changes in BGLs and the short duration of the correlation between the measurements suggest that this will not be very problematic in this case. Therefore, it was decided to consider one sample shift for each new window. This is equivalent to a time interval between the windows of 15 minutes. Figure 5.6 shows an example for a window size of six samples, four as the history length and the other two as the prediction horizon, with a window shift of one sample. The code used to create the windows can be seen in Listing 7.2

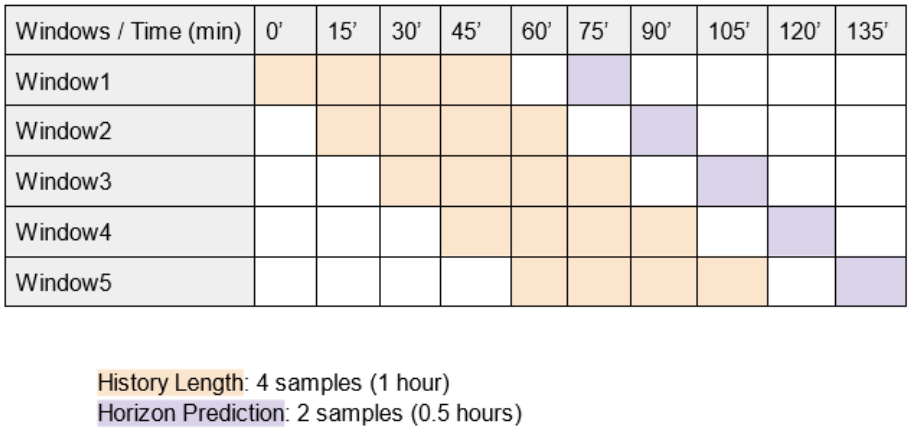


Figure 5.6: Sliding window scheme for fixed window size.

5.2.2 Single-output vs multi-output forecasting

The number of blood glucose values that the model can predict at each step determines the output strategy. The single-output technique attempts to directly predict the value corresponding to the desired horizon. In contrast, the multi-output technique can provide the prediction of multiple blood glucose values. For example, Figure 5.6 shows windows for

a prediction horizon of 2 samples. In a single-step approach, a model would only predict the value of 75 minutes in the window number one, but in a multi-step approach, the output could also include the prediction of the value of 60 minutes. It is important to take into account that the implications of these approaches go further than the number of output values. The use of a single-output strategy means that a different model must be trained for each prediction horizon. On the other hand, a multi-output model is capable of predicting the values of more than one horizon in the same prediction, but this prediction is inherently more difficult because it requires modelling the joint probability of these future values. This topic is addressed in paper [14], which supports the multi-step strategy. However, in this thesis the single-step strategy was chosen to follow the same logic of the reference paper. This means that in the following experiments, two different models must be trained for the 30 and 60 minute prediction horizons.

5.2.3 Models architecture

The reference paper [8] proposed three models: an LSTM model, a feed-forward Neural Network model and a linear model. In this thesis, an LSTM model and a linear model are proposed using these LSTM and linear models as reference. For the LSTM model, the selected hyperparameters are the same as those selected by selected in the reference paper due to the good results offered. The chosen hyperparameters are listed below:

- Hidden units of the LSTM model: 128.
- Recurrent Cells Type: LSTM.
- Number of LSTM layers: 1.
- Output dropout: 0.

Figure 5.7 shows the architecture of the LSTM prediction model. The linear model is a classical prediction model, its architecture can be seen in Figure 5.8.

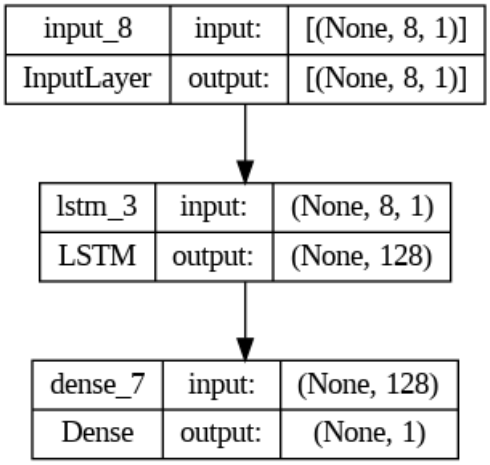


Figure 5.7: LSTM architecture scheme.

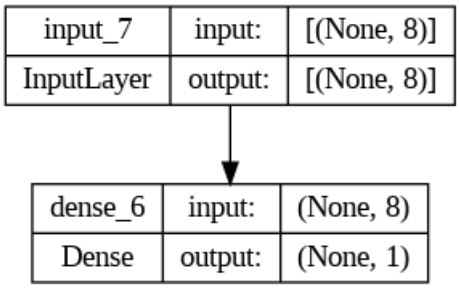


Figure 5.8: Linear model architecture scheme.

5.2.4 History length tuning

The reference paper investigated the effect of 30 minute, 60 minute, 2 hour, and 24 hour window history lengths on the forecasting of future BGLs. Figure 5.9 shows the results obtained by the reference paper with each of these history lengths for the proposed models. The best results were obtained with a history length of 30 minutes. It is important to note that the reference paper sensor takes a reading every 5 minutes, so these 30 minutes are equivalent to six samples. For our sensor, these 30 minutes would be equivalent to 2 samples. It is not expected that good results will be obtained using history lengths with fewer samples than the number of samples in the prediction horizon. Therefore, a history length tuning was performed.

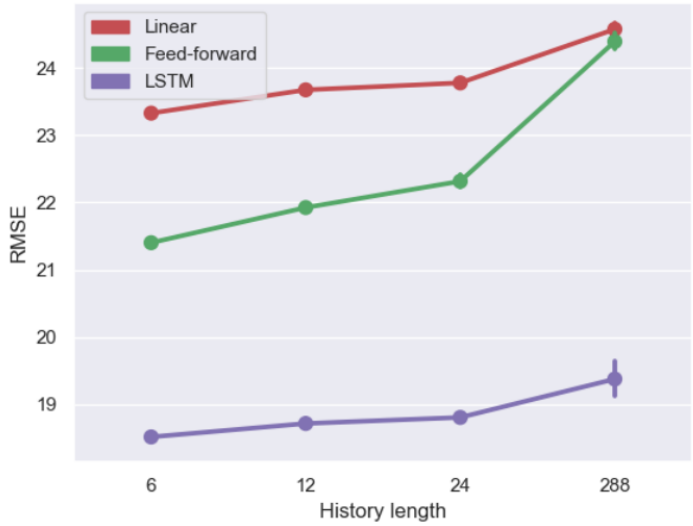


Figure 5.9: Reference paper results of using different history lengths in each of the proposed models. Source [8].

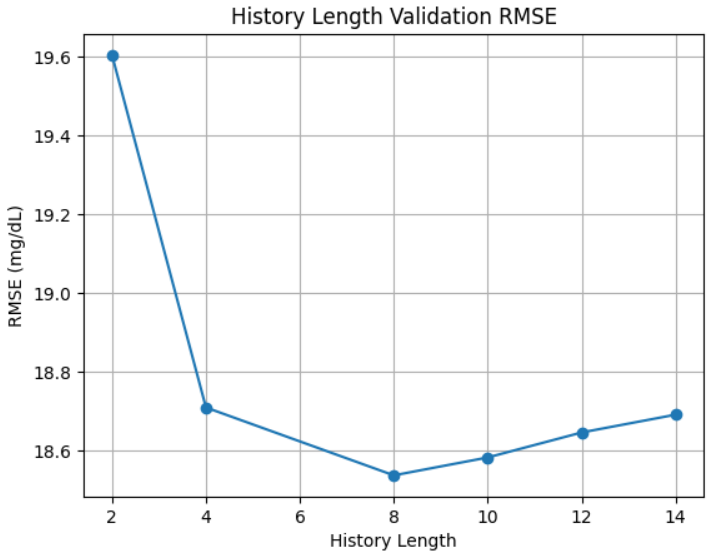
Data exploration showed that the persistence of the correlation between the BG measurements varied between 3 and 10 past values depending on the patient, as show in Section 3.4. Considering this and the history lengths chosen in the reference paper, the following values were selected to tune the history length: 30 minutes (2 samples), 1 hour (4 samples), 2 hours (8 samples), 2.5 hours (10 samples), 3 hours (12 samples), 3.5 hours (14 samples), and 24 hours (96 samples). During the tuning process, the 24-hour history length caused problems due to excessive memory consumption and was eventually discarded from the tuning set.

The tuning process consists of creating windows in the train set conformed by each of these history lengths and a prediction horizon of 30 minutes. Then, for each set of windows, a LSTM model is trained over ten epochs and validated on the validation set of patients. Figure 5.10 shows the results of the history lengths tuning. The best result is obtained for a history length of 2 hours (8 samples). The results for values 10 and 12 are close, but the use of a history length of 8 values is a more interesting choice because the total number of windows generated in the complete dataset will be higher. A history length of 8 measurements means that the data will be prepared in such a way that the model is given an input sequence of 8 past measurements to forecast the measurement that is in the desired horizon. This has conditioned the number of input units in the LSTM architecture (see Figure 5.7). This also means that the order of the linear model is 8. The full window size will be 10 samples for the 30 minute prediction horizon and 12 samples for the 60 minute prediction horizon. At this point, all the patient data are restructured as windows of these two sizes for all the following experiments.

5.3 Experiments motivation

In the following sections, three different experiments are performed with the aim of finding the best forecasting model for the T1DiabetesGranada dataset. The first experiment (see

Figure 5.10: History Length Tunning. Validation results per history length.



Section 5.4) explores the use of general forecasting models. The motivation for using a general model is that it allows future BGLs of new patients to be forecasted directly, without any time for training or adaptation. The second experiment (see Section 5.5) explores the use of personalized models. Personalized models are a good solution when a model need to be adapted to a concrete patient or a similar group of patients. Finally, the third experiment (see section 5.6) adapts personalized models to a real world case where there are new patients with a very limited history of BGLs measurements. The code of the experiments can be reviewed in [16]. Keras library has been used for the development of the prediction models [17].

5.4 Experiment I. General models

The experiment I consists of creating a general model that allows the prediction of BGLs for any patient. A general model is a prediction model that is trained with a large amount of BG data from several patients and then, it is used to predict new data on completely new patients. The tendency of the new patients' data may or may not follow the rules of the patients used for training. Therefore, the basis of a good general model is to use the maximum number of data to try to include the maximum number of BG tendencies. The code of the LSTM and linear models can be reviewed in Listings 7.3 and 7.4, respectively.

5.4.1 Data splitting

Figure 5.11 shows the schema of the data division process. The training set consists of 484 patients, representing 70% of the total. The validation set consists of 69 patients, which is 10% of the total. The test set consists of the remaining 20% of the patients, 138 patients. The selection of the set chosen for each patient was random, but it was ensured that the three sets contain patients with a similar number of measurements. To do this, the patients were ordered by their amount of measurements and grouped into sets of patients with similar amounts of data.

This separation implies that each patient is only used to train, validate or test the model. This approach is in contrast to the one used in the reference paper and in the following experiments, which takes the train and test sets from the same patient data. It is likely that this approach followed in the general model creates a test set that is more distant to the train set than the one created by the reference paper or by the following experiments, as it is expected to have more variability between different patients than between different moments of the same patient. This is important to note because the metrics calculated on these test datasets will represent an estimation of the performance of the model and are used for comparisons.

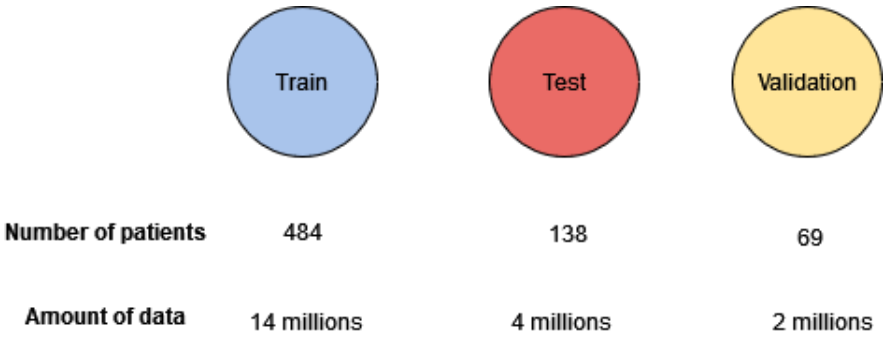


Figure 5.11: General model data division scheme.

5.4.2 Model training

Two general LSTM models and two general linear models were created, one of each for a 30 minute horizon and the other for a 60 minute horizon. All models were trained for a maximum of 1000 epochs with early stopping using a patience of 50 epochs and a batch size of 256. The use of a relatively big batch size is justified because the train set is huge, and smaller batch sizes caused problems with memory efficiency. The Adam optimizer was also chosen because it gives good results in the reference paper.

The use of two distinct forecasting horizons creates two different but closely related problems. These similarities between forecasting the BGLs at 30 minutes and at 60 minutes can be exploited using transfer learning. This means that the resulting training weights of the 30 minute forecas horizon models were used to initialize the training of the 60 minute forecasting models, for both the LSTM and linear models. The motivation for using this technique is that the weights computed in for the 30 minute forecasting models are expected to be closer to an optimal set of weights for the 60 minutes prediction horizon than a set of random weights, thus also reducing training time.

5.4.3 Results discussion

Table 5.1 shows the RMSE results of the LSTM model and the linear model on our dataset. These results have been obtained by evaluating the models with all the test patients considered as a single test set for 30 and 60 minute prediction horizons (PH). The table shows that the LSTM model performs slightly better than the linear model for both prediction horizons, but the improvement is not significant.

	RMSE	
Model	PH=30 min	PH=60 min
LSTM	17.74	32.40
Linear	18.42	33.46

Table 5.1: LSTM and linear general models RMSE comparison.

Figure 5.12 and Figure 5.13 show the Clarke Error Grids (CEG) for the 30 and 60 minute horizons for the complete test set. The ideal CEG would draw a perfect diagonal pattern. The grids for the 30 minute horizons show a narrower pattern than the 60 minute horizon because the prediction is easier. It is expected that the further away the prediction horizon is, the greater the variability that is shown by the points in the grid. At first glance, it appears that the LSTM model performs better than the linear model for both horizons. The most notable difference is that at the bottom of the grid, the linear model throws a large number of points into the C and E areas, while the LSTM model keeps these zones almost empty. At the top of the diagonal, the dispersion of points for the linear model is also much higher than the generated by the LSTM model. Not all the points are visible in the grids because both prediction models predict values outside the sensor ranges. The LSTM model predictions are above zero and reach values of 519 mg/dL for 30 minute prediction horizon. The linear model predictions reach values of -632 and 1039 mg/dL. Negative BG values and values as high as

1000 mg/dL are clearly nonsense.

Table 5.2 shows the percentage of values within each zone in the grids. It shows significant differences between the 30 minute and 60 minute prediction horizons. For both the LSTM model and the linear model, the percentage of values in the zone A decreases by 20% between the 30 and 60 minute horizons. Most of the values that are lost from zone A fall into zone B, but there is also an increase in the values in the undesirable zones C, D and E. This is further evidence of the difficulty of extending the prediction horizon for BGLs prediction problems. A comparison between the LSTM and the linear results shows that the performance of the models is very similar, as there are no differences of more than 1% of points within the same zones for 30 minute prediction horizons and 3% of points within the same zones for 60 minute prediction horizons.

Zones	LSTM		Linear	
	PH=30 min	PH=60 min	PH=30 min	PH=60 min
A	88.790%	67.902%	88.518%	68.309%
B	9.345%	25.859%	10.294%	27.692%
C	0.016%	0.339%	0.036%	0.587%
D	1.846%	5.859%	1.144%	3.255%
E	0.002%	0.041%	0.008%	0.157%

Table 5.2: Clarke error grid percentages LSTM general model and linear general model.

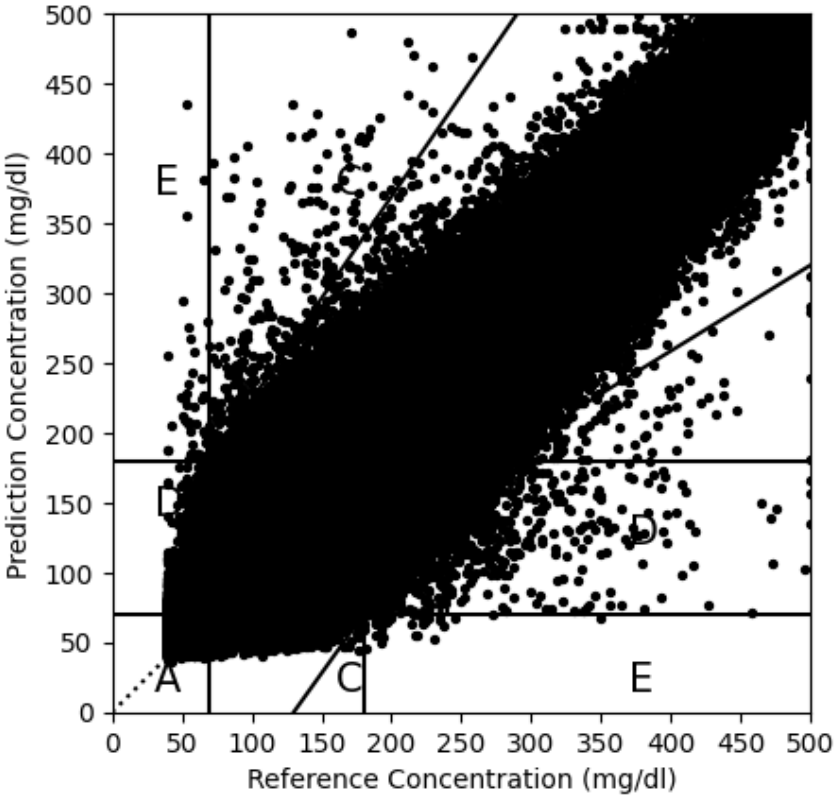
It was expected that the LSTM would clearly outperform the linear model and that the linear models would give poor results as happens in the reference paper. These results of the error metric and the clinical evaluation have not been in line with the expectations. The linear model gives more than acceptable results, its RMSE results are within the state of the art for 30 minute prediction horizons and very close for 60 minute prediction horizons (see Section 2.2.3). The improvements provided by the LSTM layer are very small, especially considering the large increase in computational complexity that this layer adds. It is possible that the use of a general training set that mixes the BG long-term dependencies of many patients does not allow the LSTM model to capture clear trends that could be generalised to all test patients. For this reason, the LSTM general model was also evaluated on each test patient individually to provide information on how good it is per each patient. Figure 5.14 shows a boxplot of the results of this evaluation. It can be seen that the performance of the model is highly patient dependent, especially at a prediction horizon of 60 minutes. There is a wide range of results, from those that would be considered exceptionally good by the state of the art, to those that are very poor. Table 5.3 shows statistics on these results. The worst results for both prediction horizons were obtained with patient *LIB193319*. The best results for the 30 minute prediction horizon were obtained with patient *LIB193353* and for 60 minute prediction horizon with patient *LIB193939*. The standard deviation results show that the dispersion of the results is higher for the 60 minute prediction horizon than for the 30 minute prediction horizon.

	RMSE	
	PH=30 min	PH=60 min
Best Results	10.95	19.99
Worst Results	31.30	59.61
Mean	18.05	33.05
Standard Deviation	3.03	6.02

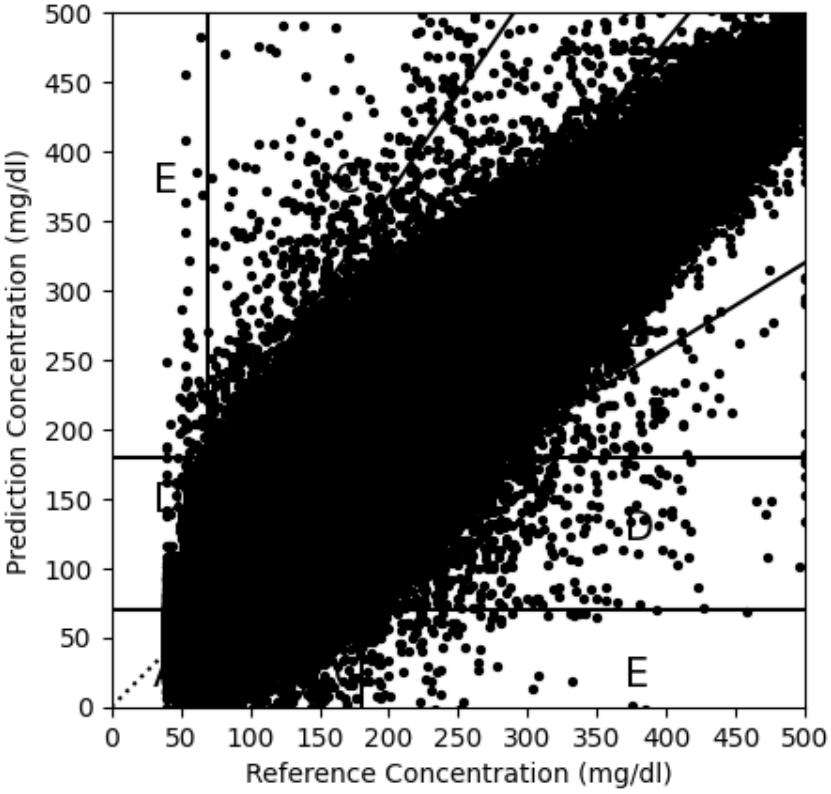
Table 5.3: Best, worst and average RMSE of the LSTM general model tested for each patient.

Large differences between patients are also found in the CEG analysis. Table 5.4 shows the minimum and the maximum percentages of points found in each of the zones when the general model is tested individually on patients Note that the percentages on different zones do not necessarily belong to the same patient. The most pronounced differences are found in zones A, B and D for both prediction horizons.

This individual patient analysis demonstrates that there are patients whose BGLs patterns are much more easier or harder to predict by the LSTM model than others. It is possible

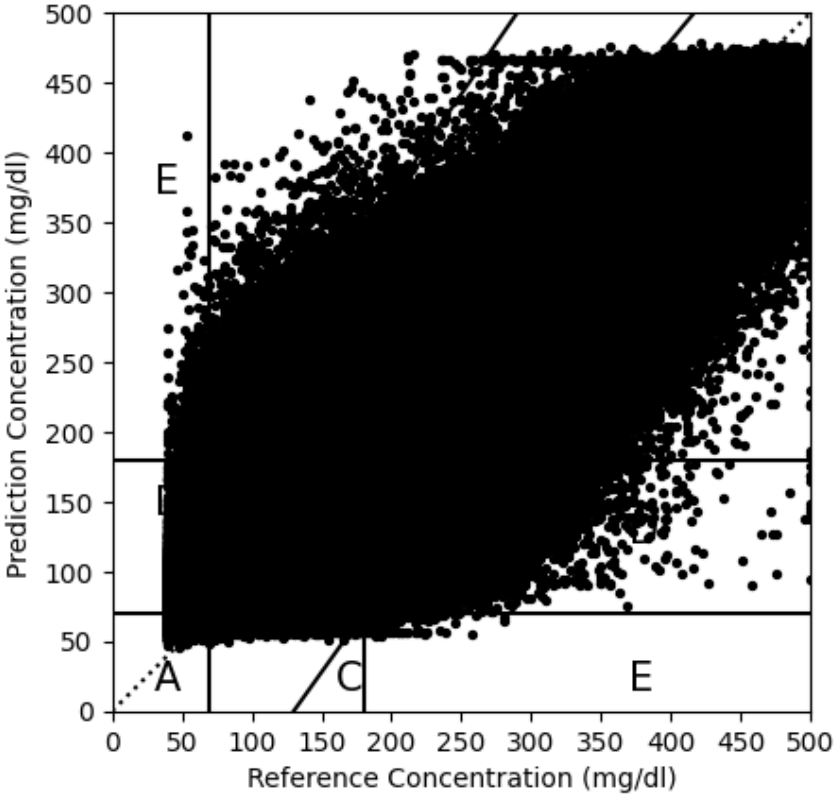


(a) LSTM model.

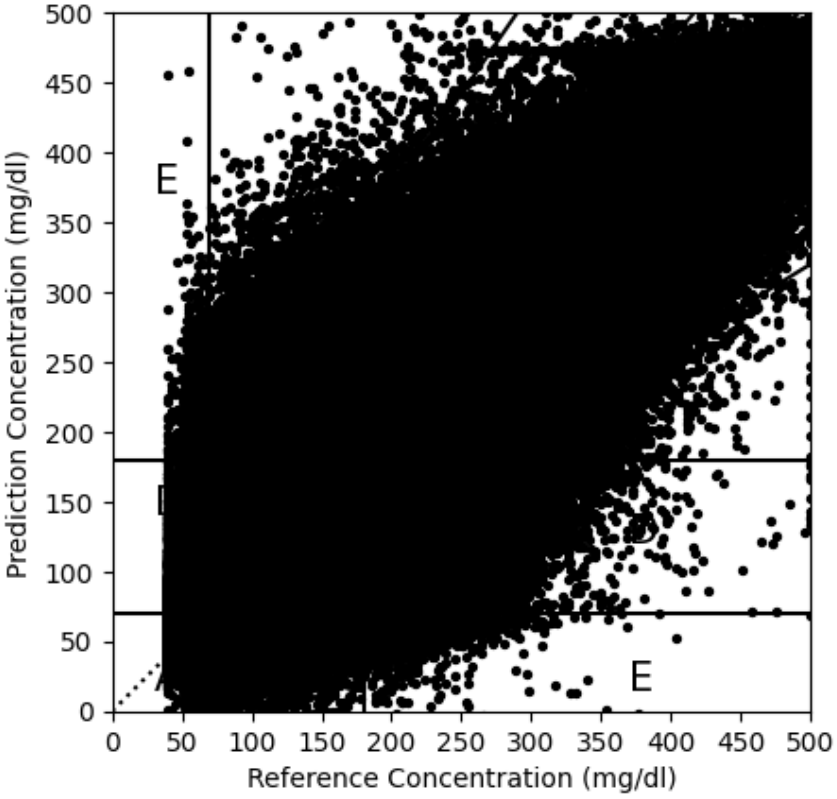


(b) Linear model.

Figure 5.12: Clarke error grid prediction horizon of 30 minutes.



(a) LSTM model.



(b) Linear model.

Figure 5.13: Clarke error grid prediction horizon of 60 minutes.

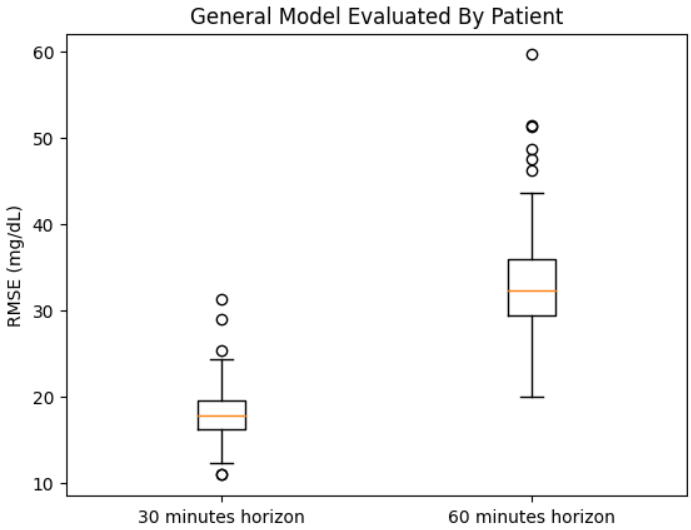


Figure 5.14: Boxplot of the RMSE error when LSTM general model is evaluated for each patient.

Zones	PH=30 min		PH=60 min	
	Minimum	Maximum	Minimum	Maximum
A	75.46%	96.12%	49.53%	83.16%
B	3.68%	18.55%	15.18%	35.70%
C	0%	0.28%	0%	1.76%
D	0.11%	6.05%	0.42%	17.87%
E	0%	0.08%	0%	0.44%

Table 5.4: Clarke error grid minimum and maximum percentage of points in each of the zones for the general LSTM model evaluated for each patient.

that the patients who give very poor results with the general model have trends that are out of the norm. The training set contains a large number of patients, but it is reasonable to assume that there are patients with BG behaviours that are outside the behaviours used to train the models. A possible solution to achieve good results in these types of patients would be to use personalized models per patient instead of a global general model. This strategy is widely used in the state of the art, as it allows the model to be directly adapted to the specific tendencies of the patient.

5.5 Experiment II. Personalized models

The experiment II consists of creating a personalized model that can be adapted to a specific patient. A personalized model uses the information of a patient to predict the future BGLs of the same patient. Each personalized model is directly dependent on one patient and it is not expected to give good results for other patients. In contrast to the general model presented in Section 5.4, personalized models require previous data from the patient for whom the future values are to be predicted in order to be trained.

5.5.1 Data splitting

Creating a personalized model for each patient in the T1DiabetesGranada dataset would be a very demanding task. In its place, four personalized models were trained. It is important to remark that the process followed to create these personalized models could be extrapolated to any of the other patients. Personalized models are a good idea especially for patients who are out of the norm. Therefore, the patient who gave the worst results (*LIB193319*) in the RMSE evaluation of the general model can be considered as the perfect candidate. In addition, it is interesting to see if the use of personalized models could improve the results of patients fully controlled by the general model, so the patients who gave the best results, *LIB193353* for 30 minutes and *LIB193939* for 60 minutes prediction horizons, were also selected. For each of these patients, the dataset is divided into 80% of the data for training and 20% for testing.

This division is done respecting the temporal order. Figure 5.15 shows the schema for the division of the data.

Figure 5.15: Personalized model data division scheme.

5.5.2 Model training

Four personalized LSTM models and four personalized linear models were trained using a similar logic to the general model. This means a maximum of 1000 epochs, with early stopping implemented after 50 epochs of patience, and the Adam optimizer. As the number of samples is much smaller in this case, a batch size of 32 was used as this gave good performance in the reference paper. In addition, the transfer learning technique was not used in this experiment in order to create as specialised models as possible.

5.5.3 Results discussion

In order to fairly compare the general models presented in Section 5.4 with each of the personalized models proposed in this experiment, the general models and the personalized models were both evaluated in the last 20% of the data for each of the chosen patients. Table 5.5 shows these RMSE results for the general models and for the new proposed personalized models for the patients who were selected in the *Data Division* Section 5.5.1. This table shows that the results obtained with the personalized models are very close to the results obtained with the general models. The personalized models only improve the results of the general models for the patient *LIB193939*, which is the patient with the best results for the 60 minute prediction horizon. For the other patients, the personalised models predict slightly worse than at least one of the general models. Comparing the LSTM and the linear personalized model shows that the improvements of the personalized LSTM model are quite small, the same as for the general models. Again, in this experiment the results of the linear models are much better than expected, to the point that for patient *LIB193319* both the general and the personalized linear models give better results than their respective LSTM versions. An important conclusion is that none of the models gives sufficiently good results to claim to be incodicionaly better than the other models.

Figure 5.16, Figure 5.17, Figure 5.18 and Figure 5.19 compare the CEG of the general and the personalized models that gave the best results in Table 5.5 for each of the selected patients. The grid for the patient *LIB193319* in Figure 5.16 is interesting because, although it corresponds to the patient who gave the worst results in the general LSTM model, the points do not look particularly misplaced. This is because the RMSE error metric and the

	PH=30 min		PH=60 min	
	LIB193319 †	LIB193353 ‡	LIB193319 †	LIB193939 ‡
General LSTM	33.30	11.94	55.92	20.93
General Linear	33.52	12.20	55.21	21.90
Personalized LSTM	33.34	12.17	58.26	18.28
Personalized Linear	34.46	12.41	58.02	18.40

Table 5.5: Selected patients RMSE comparison between the LSTM and Linear general model tested for each patient and the personalized models. Patients with † and ‡ are those for whom the general LSTM model gave the worst and the best results respectively in each prediction horizon.

CEG clinical evaluation are two different ways of evaluating the prediction of a model. They are complementary to each other, and a good or bad result in one does not mean the same thing in the other.

It is difficult to draw conclusions from the grids because the points are distributed very similarly in all the comparisons. However, these clinical results can be analyzed in Table 5.6 and Table 5.7, which show the percentage of values within each zone of the grids for the LSTM general model, the LSTM personalized model and the linear personalized model for the same set of patients. The tables show that all the models give satisfactory results for all the patients, keeping more than 97% of the points in no risk zones (A and B) for the 30 minute horizon and more than 90% of the points for the 60 minute horizon. When comparing the models with each other, it can be seen that the results for the patients with a 30 minute prediction horizon (*LIB193319* and *LIB193353*) are too similar to draw any conclusions. For patient *LIB193319* on a 60 minute prediction horizon, the LSTM personalized model may appear to be the worst performance model. However, this is difficult to confirm because, despite the decrease in points in zone A and the increase in zones B, C and E, it seriously reduces the percentage of points in zone D. Finally, the only patient who shows significant differences between personalized and general models models is patient *LIB193939* on a 60 minute prediction horizon. The LSTM and the linear personalized models give almost the same results, but both of them offer an important increase of points of 15% in zone A, and a decrease of 10% in zone B and 5% in zone D. These results only allow to ensure that the use of personalized models performs better than the general LSTM model for the data of patient *LIB193939*.

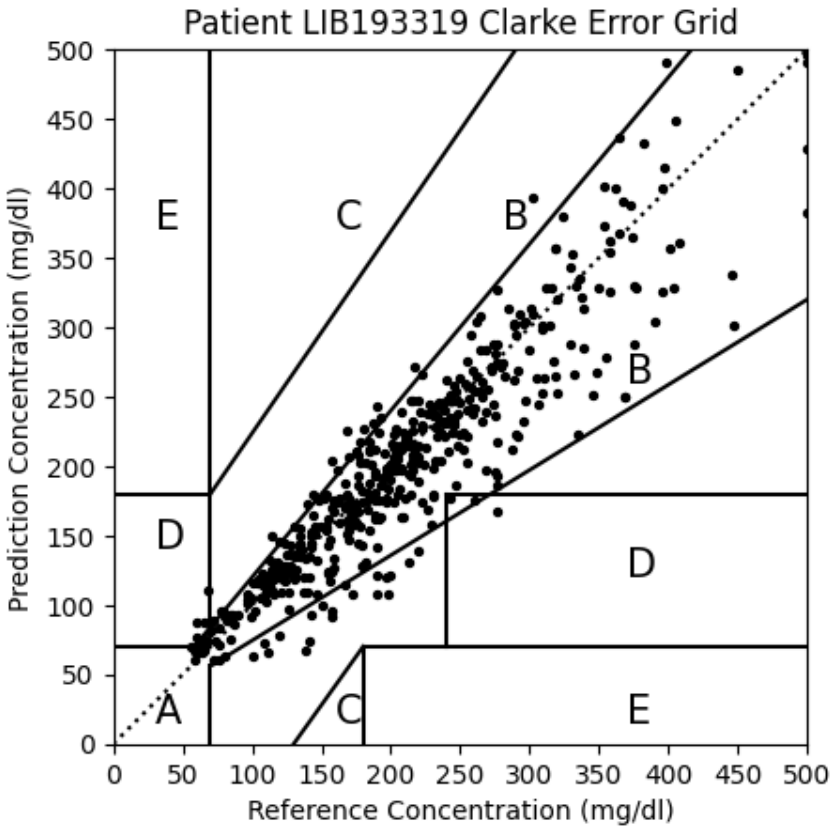
Zones	LSTM General Model		LSTM Personalized Model		Linear Personalized Model	
	PH=30 min	PH=60 min	PH=30 min	PH=60 min	PH=30 min	PH=60 min
A	80.99%	58.56%	80.37%	53.36%	77.27%	59.87%
B	17.36%	32.32%	18.39%	39.91%	20.45%	32.32%
C	0.00%	0.22%	0.00%	1.52%	0.00%	0.00%
D	1.65%	8.89%	1.24%	4.77%	2.27%	7.59%
E	0.00%	0.00%	0.00%	0.43%	0.00%	0.22%

Table 5.6: Clarke error grid percentages of LSTM general, LSTM personalized and linear personalized models for patient *LIB193319* for both prediction horizons.

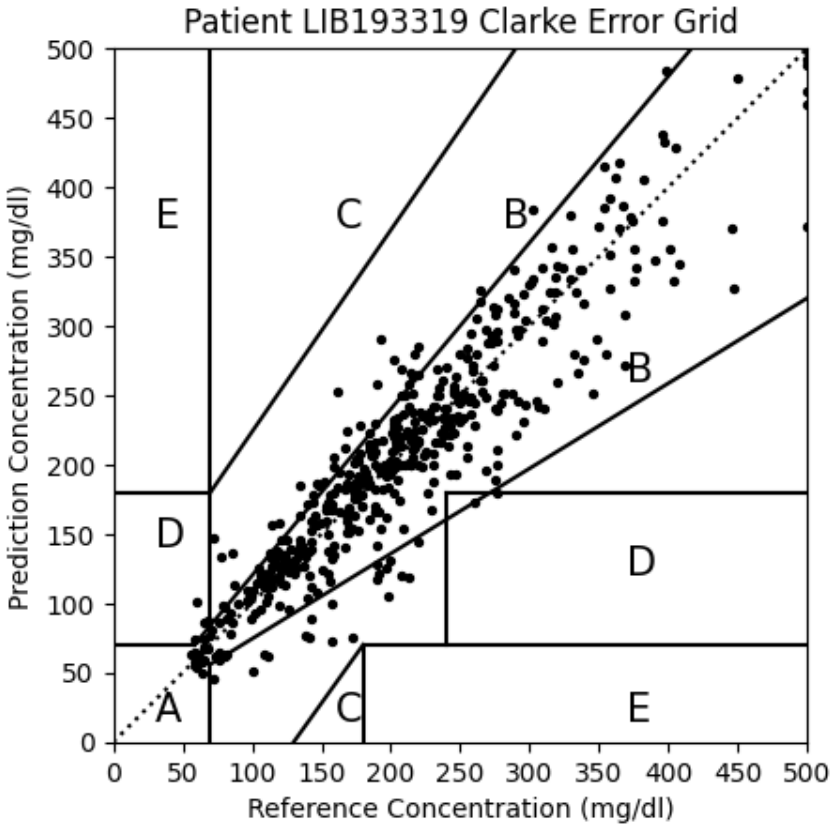
Zones	LSTM General Model		LSTM Personalized Model		Linear Personalized Model	
	PH=30 min	PH=60 min	PH=30 min	PH=60 min	PH=30 min	PH=60 min
A	94.52%	61.42%	94.20%	76.57%	93.96%	76.38%
B	4.71%	31.05%	5.31%	20.55%	5.68%	20.95%
C	0.00%	0.02%	0.00%	0.00%	0.00%	0.00%
D	0.77%	7.52%	0.49%	2.88%	0.37%	2.67%
E	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 5.7: Clarke error grid percentages LSTM general, LSTM personalized and linear personalized models for patient *LIB193353* for 30 minute horizon and patient *LIB193939* for 60 minute horizon.

The analysis of all the results raises many doubts about whether personalized models could be the solution to fix the problems of the general prediction models. Although the use of personalized models has provided a clear improvement for one of the patients (*LIB193939*),

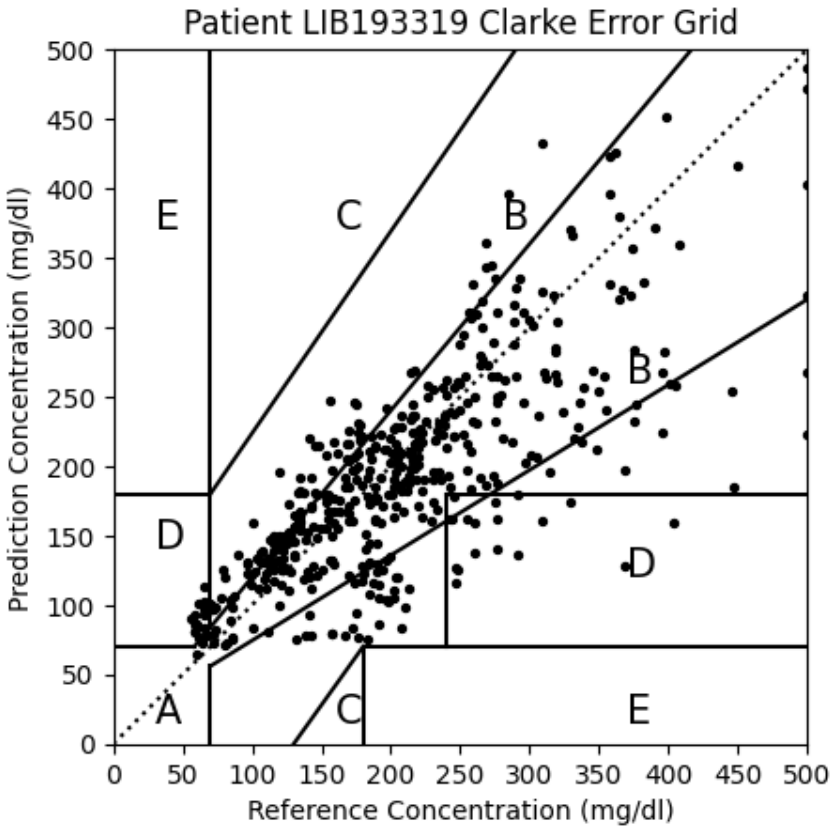


(a) General LSTM model.

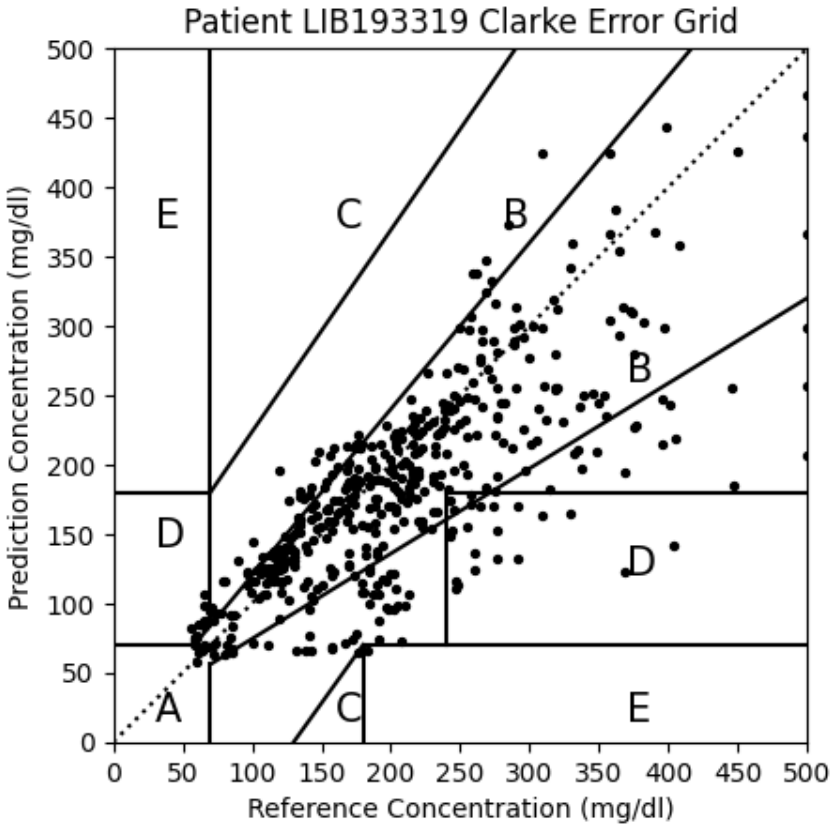


(b) Personalized LSTM model.

Figure 5.16: CEG comparative for patient *LIB193319* using a 30 minute prediction horizon.

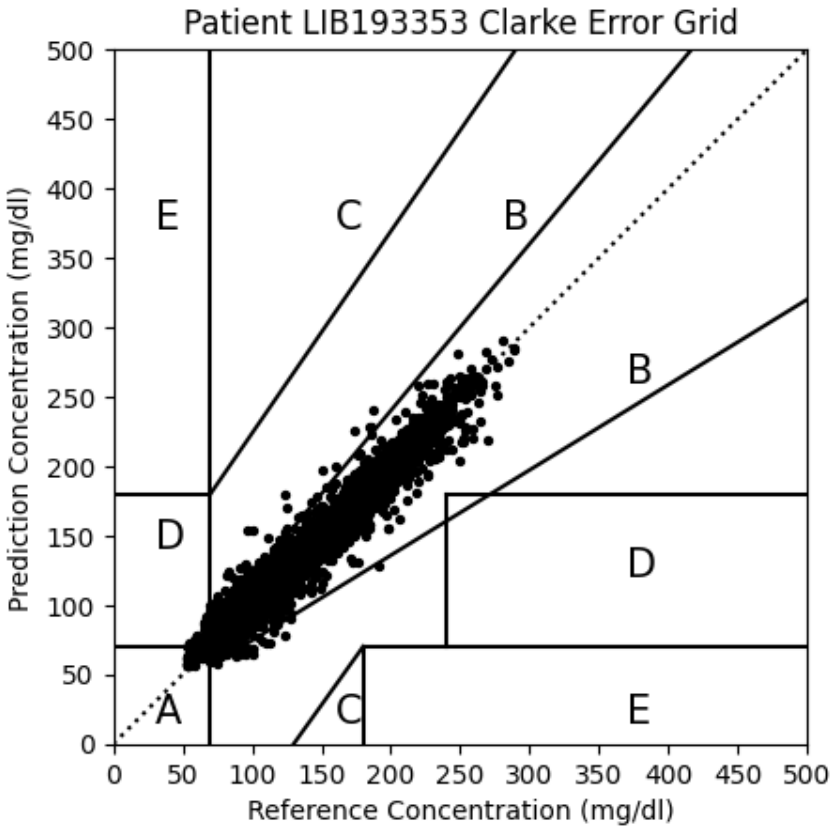


(a) General LSTM model.

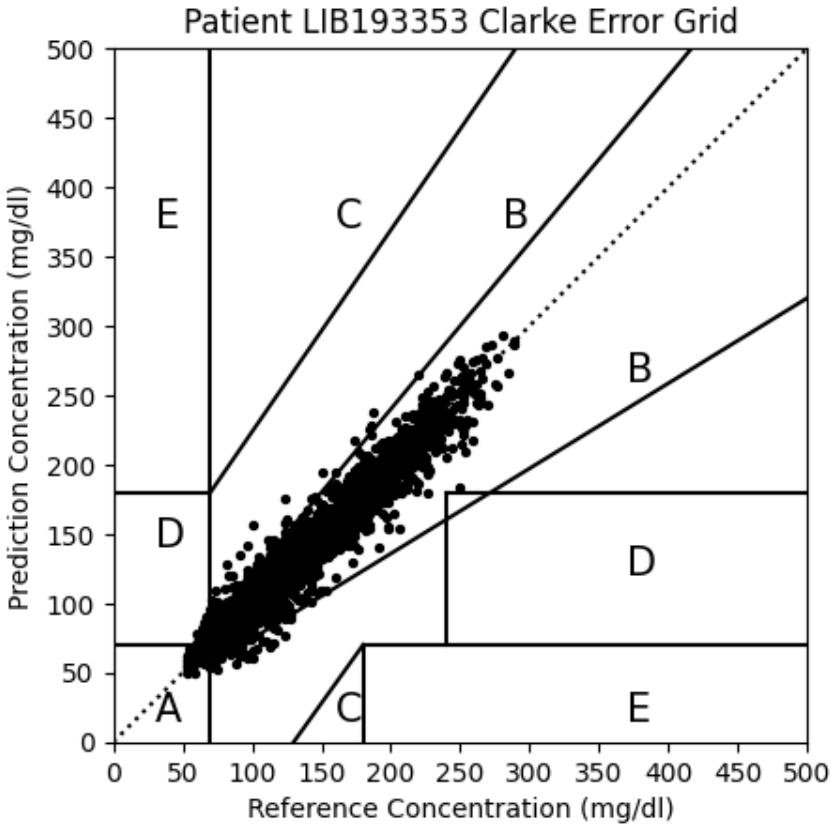


(b) Personalized LSTM model.

Figure 5.17: CEG comparative for patient *LIB193319* using a 60 minutes prediction horizon.

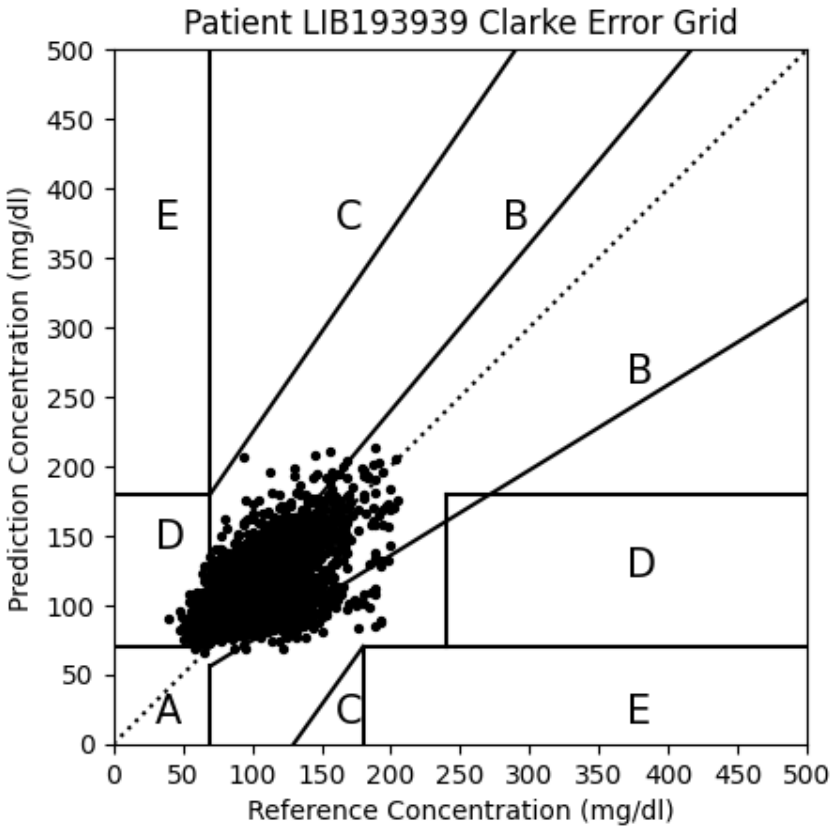


(a) General linear model.

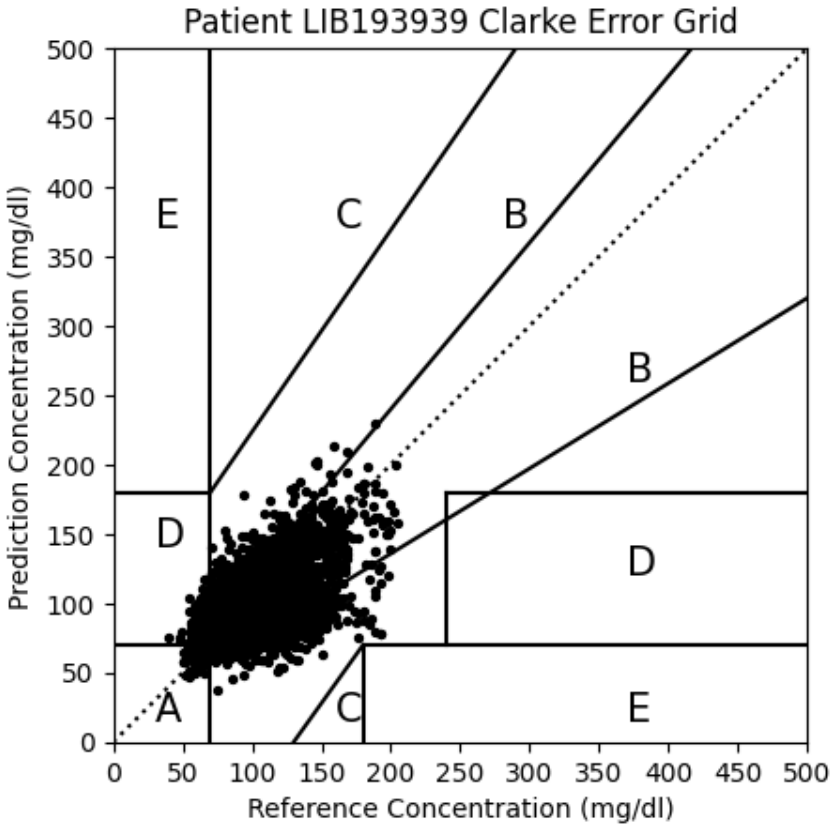


(b) Personalized linear model.

Figure 5.18: CEG comparative for patient *LIB193353* using a 30 minutes prediction horizon.



(a) General LSTM model.



(b) Personalized LSTM model.

Figure 5.19: CEG comparative for patient *LIB193939* using a 60 minutes prediction horizon.

the results of the other patients are the same or worse. On the basis of the patients studied, it cannot be said that personalized models perform better than general models for our dataset. However, the main problem with personalized models goes beyond this, as the building of a personalized model is difficult to replicate in the real world. This is because creating a personalized model with a good performance requires a huge amount of data from the patient. It would be unacceptable to require large amount of information for a new patient in a real-world application, as it would mean that the patient would have to wait an enormous amount of time for the model to calibrate before receiving a future BG prediction. This problem does not exist in the nature of a general model, but it could also be mitigated by using sets of measurements to train the personalized model that are small enough to be extrapolated to the real world.

5.6 Experiment III. Personalized model for the real world

The experiment consists of creating a personalized model that could be extrapolated to a real-world application. This model follows all the principles of the personalized models, but uses an amount of information to build the model that would be affordable for a new patient with no measurement history available. This experiment can also be seen as a study of how the size of the training set affects to the final performance of the personalized model.

5.6.1 Data splitting

The aim of this experiment is to obtain a good performance model for each of the patients considered in Experiment II. In this case, the process of building these models could also be extrapolated to any of the other patients. For each of these patients, a tuning process was performed to determine the training set size, or in other words, the number of BG measurements needed to build the model. This tuning process evaluates the use of 1, 2, or 3 weeks of data to train the personalized model. The training set is chosen immediately prior to the time of the test data in order to maintain equal conditions with the classical personalized models, proposed in Section 5.5. The last 20% of the patient data is used to test the model. The tuning is done on a validation set, which corresponds to the amount of available patient data that is outside of the three weeks of data tuned for training and the test set. Figure 5.20 shows the schema for the division of the data.

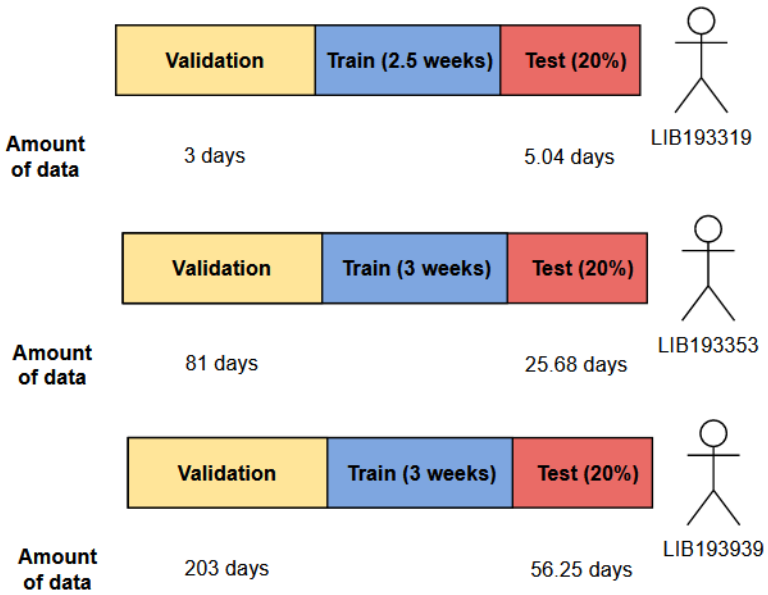


Figure 5.20: Real world personalized model data division scheme.

5.6.2 Model training

The training of the real world personalized models tried to follow the same points as the previous experiment presented in Section 5.5. The models were trained using the Adam optimizer

and a batch size of 32 samples. However, the peculiarity of this case is that the amount of data used to train the models is small enough to be concerned about an overfitting problem. For this reason, the maximum number of epochs was also selected in the tuning process from the following values: 500, 750 and 1000 epochs. In the tuning process, a new model was trained for each of the possible combinations of the number of epochs and the number of weeks. This means 9 LSTM models and 9 linear models for each patient, and a total of 72 different models. Each of these models was evaluated on the corresponding validation patient data, as shown in Figure 5.21 and Figure 5.22. It is important to note that this tuning process was performed to select a fixed number of epochs and weeks that would be maintained for every new patient in the real world. It would be ideal to be able to perform a tuning procedure to select the optimal number of epochs and weeks for each new patient model, but this would obviously go against the principle of using a small amount of patient data.

Figure 5.21 and Figure 5.22 show that the tuning results are diverse. For the LSTM model, patient *LIB193319* achieves the best results for 30 and 60 minute horizons with 1000 epochs and 3 weeks of training, *LIB193353* with 500 epochs and 1 week, and *LIB19393* with 500 epochs and 2 weeks. For the linear model, the best results for patients *LIB193319* and *LIB193353* for 30 minute horizons are obtained with 1000 epochs and 3 weeks of training, and for patients *LIB193319* and *LIB193939* with 1000 epochs and 1 week. From these results it can be seen that none of these combinations is clearly better, and while some work well for a particular patient and a particular horizon, there may be others for whom they do not. The use of 1000 epochs and 3 weeks of training was chosen on the basis that it gave the best results for half of the patients in the LSTM and linear models.

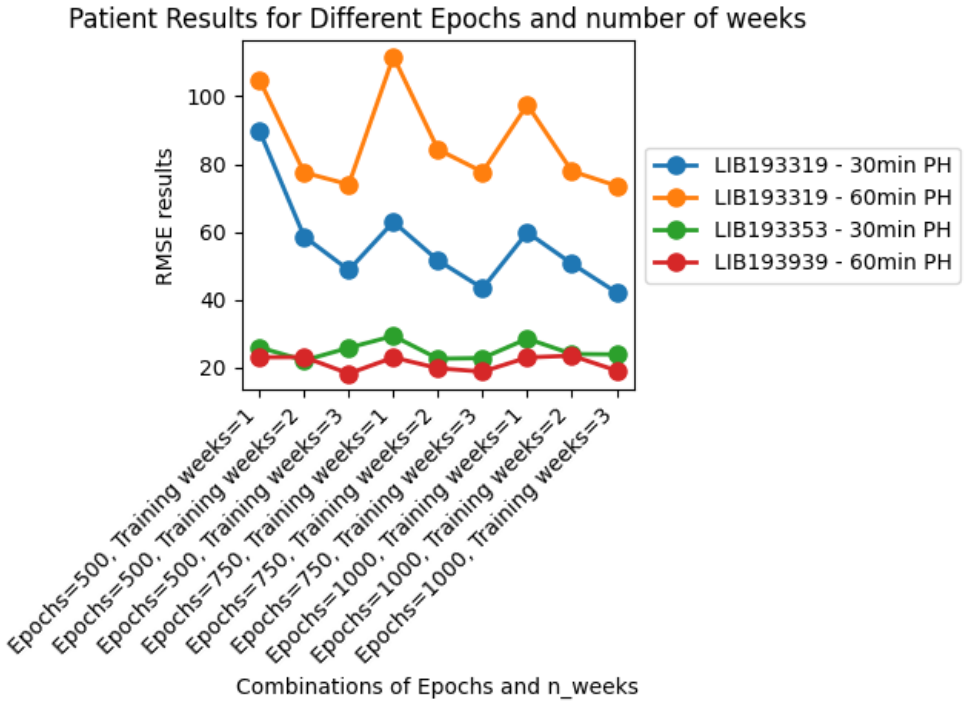


Figure 5.21: Tuning results of the number of epochs and weeks for the LSTM model.

5.6.3 Results discussion

Table 5.8 shows the final RMSE results of the selected real world models in the selected patients and the results of experiments proposed in Section 5.4 and Section 5.5. Once again, the results of the LSTM and the linear models are quite similar. Comparing the new results with those of the previous experiments, the new models are in a bad position. As happened with the personalized models in the previous experiment, the real world personalized models only manage to improve the results of the general model for patient *LIB193939*. The use of a smaller amount of training data is also reflected in the results, as the personalized models achieve better results than the real world models in all the cases. The percentage of improvement of the personalized models over their real world versions depends on the patient. For

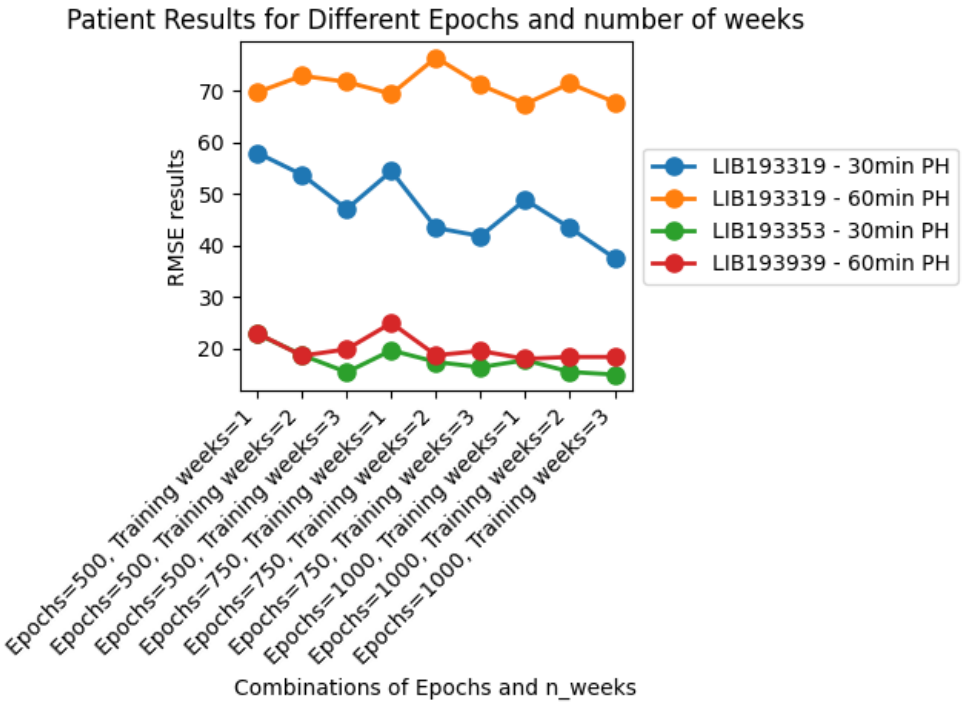


Figure 5.22: Tuning results of the number of epochs and weeks for the linear model.

patient *LIB193319* and patint *LIB193353* and 30 minute prediction horizon the differences are large, the real world results are about 5 mg/dL worse than the personalized results, but for patient *LIB193319* and patient *LIB193939* and 60 minute prediction horizon the improvement of the personalized models is not as significant.

	PH=30 min		PH=60 min	
	LIB193319 †	LIB193353 ‡	LIB193319 †	LIB193939 ‡
General LSTM	33.30	11.94	55.92	20.93
General Linear	33.52	12.20	55.21	21.90
Personalized LSTM	33.34	12.17	58.26	18.28
Personalized Linear	34.46	12.41	58.02	18.40
Real World LSTM	39.88	17.86	61.99	19.77
Real World Linear	39.38	17.39	60.66	21.13

Table 5.8: RMSE comparison between the LSTM and linear general models tested for the selected patients and the personalized models. Patients with † and ‡ are those for whom the general LSTM model gave the worst and the best results respectively in each prediction horizon.

The clinical evaluation using the CGE, which grid results are summarized in Table 5.9 and Table 5.10, shows also show a clear poor performance of the real world models. The real world LSTM and linear models decrease the percentage of values in zone A, and increase the values in zone B, and the undesirable zones C, D or E for the four patients. These results show that the predictive ability of the real world personalized models is below the capacity of the models proposed in Section 5.4 and Section 5.5.

Zones	**LSTM Personalized Model**		**LSTM Real World Model**		**Linear Real World Model**	
	PH=30 min	**PH=60 min**	**PH=30 min**	**PH=60 min**	**PH=30 min**	**PH=60 min**
A	80.37%	53.36%	72.11%	44.47%	74.38%	47.72%
B	18.39%	39.91%	24.17%	47.07%	23.53%	43.82%
C	0.00%	1.51%	1.03%	1.73%	0.00%	1.30%
D	1.24%	4.77%	2.66%	6.51%	2.07%	6.72%
E	0.00%	0.43%	0.00%	0.22%	0.00%	0.43%

Table 5.9: Clarke error grid percentages LSTM personalized, LSTM real world personalized and linear real world personalized models for patient *LIB193319* for 30 minute and 60 minute prediction horizons.

Zones	**LSTM Personalized Model**		**LSTM Real World Model**		**Linear Real World Model**	
	PH=30 min	**PH=60 min**	**PH=30 min**	**PH=60 min**	**PH=30 min**	**PH=60 min**
A	94.20%	76.57%	80.85%	74.67%	87.06%	71.97%
B	5.31%	20.55%	13.18%	22.40%	12.17%	25.63%
C	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
D	0.49%	2.88%	5.96%	2.92%	0.77%	2.40%
E	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 5.10: Clarke error grid percentages for LSTM personalized, LSTM real world, and linear real world models for patient *LIB193353* for a 30-minute horizon and *LIB193939* for a 60-minute horizon.

Chapter 6

Conclusions

6.1 Achieved goals

The main objective of this thesis was to develop a Machine Learning model that allows the prediction of BGLs in patients with type 1 diabetes using data from real patients. This objective has been achieved, different BGLs forecasting models have been developed using the T1DiabetesGranada dataset, with results that are close to the best state of the art results. In order to achieve this objective, it was essential that all the secondary objectives were completed:

- Investigate type 1 diabetes and the most innovative techniques for predicting BGLs: To this end, a comprehensive and in-depth study of the type 1 diabetes mellitus and the BGLs prediction models of the state of the art was carried out.
- Analyse the longitudinal sensor-based T1DiabetesGranada dataset in terms of its overall and individual characteristics: The time series data characterisation, the distribution of the BG measurements and the lagged BG measurements were studied. This had an important influence on the selected preprocessing and forecasting techniques for the T1DiabetesGranada dataset.
- Preprocess the T1DiabetesGranada dataset to facilitate the learning and application of forecasting models: The most important preprocessing was the resampling process, which gave consistency to the dataset and allow the measurements to be evenly spaced in time.
- Generate various forecast models using different machine learning approaches. Different LSTM neural networks and linear regression models were built based on general and personalized strategies.
- Evaluate the developed forecasting models. All the proposed models were evaluated using RMSE and CEG metrics and the results were discussed. The RMSE allows the performance of the models to be measured objectively, while the CEG allows it to be measured clinically.

6.2 Results interpretation

The results of all the proposed experiments can be considered as generally successful. The evaluation of the LSTM and linear general models on a test set of 138 patients achieves a percentage of more than 98% of points for the 30 minute prediction horizon and 93% for the 60 minute prediction horizon in zones A and B, which are considered safe zones for patients. For the four chosen patients, the general and personalized models achieve at least a 94% of points for the 30 minute horizons and a 91% of points for the 60 minute horizons in zones A and B. Therefore, it would hardly be justified to choose one of these models over the others on the basis of such close prediction results for only four patients. However, even though the results are so close, there is still a big difference between the applicability of these models in a real-world context. General models could be used directly to predict the future BG levels of any patient without any prior training. On the other hand, personalized models require a large number of measurements to be built, making it impossible to use these models for new patients without a long history of BGLs measurements. The real world version of the personalized models would require an amount of information that would make it affordable,

but the results are worse than the results of the general models. Nowadays, several state of the art publications create models that give exceptional performance in forecasting BGLs, but would hardly be exported to a new patient application because they require a huge amount of measurements. While these papers can provide very attractive results, it is important to remember that the most important scientific advances occur when they can be used to improve the lives of real people. For all these reasons, it can be concluded that the general models built for the T1DiabetesGranada dataset are a more useful and attractive solution than the personalized models.

The most important finding of the experiments is that it is not necessary to have a complex architecture in order to accurately forecast the future BGLs of the patients in the T1DiabetesGranada dataset. Most of the state of the art publications greatly improve the results of classical machine learning models using any type of deep learning model such as LSTM neural networks, but in this case they do not provide a significant improvement for most of the evaluations. For the general models, the improvement offered by the LSTM model over the linear model is around 1 mg/dL for both prediction horizons. This enhancement provided by the LSTM layer is much less remarkable when considering the huge difference in complexity, time and required computational capacity between the models. In a real world application, these differences are not only manifested during the training phase, but also during the continuous maintenance and improvement of the model. These differences also can be translated into a considerably lower cost and environmental impact of the linear model with respect to the LSTM model. When all these considerations are taken into account, it is clear that the linear general model is the model that should be selected to predict the future BGLs of the patients in T1DiabetesGranada dataset. The evaluation in the test set shows that the linear general model achieves a RMSE result of 18.42 mg/dL for a 30 minute horizon and 33.46 mg/dL for a 60 minute horizon. These results are within the state of the art results for the 30 minute prediction horizon and very close to the state of the art results for the 60 minute prediction horizon. In the clinical evaluation, it achieves a 98.81% of points in zones A and B for a 30 minute prediction horizon and a 96% of points for a 60 minute prediction horizon.

As a final remark, this bachelor thesis, like many other state of the art publications, demonstrates that an accurate prediction of the future BGLs is possible for short prediction horizons. But in addition, this work demonstrates that it not necessary to use complex and deep models in order to guarantee these good predictions for the T1DiabetesGranada dataset. It is relevant to note the importance of this result, as it breaks with the tendency of some publications of using as much as possible complex models to forecast future BGLs. If this linear model gives good results for T1DiabetesGranada dataset, it could be expected that there are other dataset for which linear models can be used as a starting point.

6.3 Limitations and future work

The main limitation faced in the development of this work was the insufficient computing capacity. Training an LSTM neural network model on a large dataset such as the T1DiabetesDataset was too slow on a mid-range computer. The use of Google Collab mitigated this problem, but the wait times for each of the training epochs were still high. It would have been ideal to use a high performance server from the start as this would have freed up a lot of time for other tasks.

For future work it would be interesting to re-train the models extending the prediction horizon to 90 and 120 minutes. There are not many papers that provide results with such distant prediction horizons, but the good results obtained with 30 and 60 minute horizons are very encouraging. It would also be useful to perform a clustering of patients. In this work, this problem was not addressed due to lack of time, but the T1DiabetesGranada dataset is perfect to study it. The knowledge of which patients are most similar would allow the creation of small general models that are adapted to this particular set of patients, i.e. a new kind of models that are in between a general model and a personalized model. Finally, it should be mentioned that the T1DiabetesGranada dataset offers more possibilities than those addressed in this thesis. In future work, the information from the biochemical tests and the diagnostics provided by the dataset could be added to the knowledge of the forecasting models and the results compared with the current ones. This line of research is particularly

interesting because if this additional information could be used properly, the results of the forecasting models could probably become even more competitive.

Chapter 7

Appendix

7.1 Code

The complete code of the thesis can be reviewed in the github repository available a [16]. In the following, snapshots some interesting functions are presented.

```
import pandas as pd
import numpy as np
import datetime

def resample(df, patients = None):
  """
    :param df: Dataframe of BG measurements.
    :param patients: Subset of patients. If None, all
       T1DiabetesGranada patients are considered.
    :return: Array of resampled measurements per patient.
  """

  def get_closest(all_dates_series, expected_date):
    """
    :return: The closest measurement to the expected date if
       exist within the range of 7 minutes.
    """
    allowed_delay = 7
    for delay in range(1, allowed_delay + 1):

      past_value   = expected_date-pd.DateOffset(minutes=delay)
      if not np.isnan(all_dates_series[past_value]):
        return all_dates_series[past_value]

      future_value = expected_date+pd.DateOffset(minutes=delay)
      if not np.isnan(all_dates_series[future_value]):
        return all_dates_series[future_value]

    return None

  resampled_data_per_patient = []

  if not patients:
    patients = get_patient_ids(df)
  for patient in patients:

    df_glucose_patient = df[df['Patient_ID'] == patient]

    start_date = df_glucose_patient['Timestamp'].iloc[0]
    end_date   = df_glucose_patient['Timestamp'].iloc[-1]

    # Series containing all possible dates from the first
```

```
        measurement to the last one separated by one minute.
41      all_dates = pd.date_range(start=start_date, end=end_date,
            freq='T')
42      all_dates_series = pd.Series(dtype='float64', index=
            all_dates)
43
44      # Series containing the dates for which it is expected to
            have a recorded measurements, i.e. each 15 minutes from
            the initial measurement.
45      all_dates_series[df_glucose_patient['Timestamp']] =
            df_glucose_patient['Measurement'].values
46
47      expected_dates = pd.date_range(start=start_date, end=
            end_date, freq='15T')
48
49      # When an expected date does not have a measurement, it is
            searched in the close dates using the all dates series.
50      for expected_date in expected_dates:
51        if np.isnan(all_dates_series[expected_date]):
52          all_dates_series[expected_date] = get_closest(
                all_dates_series, expected_date)
53
54      resampled_data_per_patient.append(all_dates_series[
            expected_dates])
55
56    return resampled_data_per_patient
```

Listing 7.1: Resampling function.

```
1
2  import numpy as np
3  def get_windows_one_step_walk_forward(data, lookback_samples,
       pred_samples):
4  """
5      :param lookback_samples: Number of samples used to predict (
           history length).
6      :param pred_samples: Prediction window (normally 30 or 60
           minutes).
7      :return: Transform data into one step sliding windows
           without missing values.
8      """
9    x_per_patient = []
10   y_per_patient = []
11
12   for data_patient in data:
13
14     # Creating the one step sliding window
15     x = np.lib.stride_tricks.sliding_window_view(data_patient[:-
           pred_samples], lookback_samples)
16     y = np.lib.stride_tricks.sliding_window_view(data_patient[
           lookback_samples:], pred_samples)
17
18     # Removing rows with missing values
19     nan_rows_x = np.isnan(x).any(axis=1)
20     nan_rows_y = np.isnan(y).any(axis=1)
21     x = x[~(nan_rows_x | nan_rows_y)]
22     y = y[~(nan_rows_x | nan_rows_y)]
23
24     x_per_patient.append(x)
25     y_per_patient.append(y)
26
27   x_result = np.concatenate(x_per_patient)
28   y_result = np.concatenate(y_per_patient)
29
```

```
  return x_result, y_result
```

Listing 7.2: Windows generation function.

```

class LSTMModel:
    def __init__(self, input_shape, nb_output_units,
        nb_hidden_units=128, nb_layers=1, dropout=0.0,
        recurrent_dropout=0.0):
        self.input_shape = input_shape
        self.nb_output_units = nb_output_units
        self.nb_hidden_units = nb_hidden_units
        self.nb_layers = nb_layers
        self.dropout = dropout
        self.recurrent_dropout = recurrent_dropout

    def __repr__(self):
        return 'LSTM_{0}_units_{1}_layers_dropout={2}_{3}'.
            format(self.nb_hidden_units, self.nb_layers, self.
            dropout, self.recurrent_dropout)

    def build(self):
        # input
        i = Input(shape=self.input_shape)

        # add first LSTM layer
        x = LSTM(self.nb_hidden_units)(i)

        if self.nb_layers > 1:
            for _ in range(self.nb_layers - 2):
                x = LSTM(self.nb_hidden_units, dropout=self.
                    dropout, recurrent_dropout=self.
                    recurrent_dropout, return_sequences=True)(x)

            # add final LSTM layer
            x = LSTM(self.nb_hidden_units, dropout=self.dropout,
                recurrent_dropout=self.recurrent_dropout,
                return_sequences=False)(x)

        x = Dense(self.nb_output_units, activation=None)(x)

        return Model(inputs=[i], outputs=[x])


# Training
from keras.callbacks import ModelCheckpoint, EarlyStopping
import keras.backend as K

def prepare_model_LSTM(history_length, summary=False, plot=False
    , weights='', nb_hidden_units=128, nb_layers=1, dropout=0.0)
    :

    model = LSTMModel(input_shape=(history_length, 1),
        nb_output_units=1,
        nb_hidden_units=nb_hidden_units, nb_layers=nb_layers,
            dropout=dropout, recurrent_dropout=0.0)
    # build & compile model
    m = model.build()

    m.compile(loss=RMSE,
              optimizer='adam',
              metrics=[RMSE])
    if weights:
      print(f"Weights: {weights}")
```

```
      m.load_weights(weights)
    if summary:
      print(m.summary())
    if plot:
      plot_model(m, to_file='model/LSTM.png', show_shapes=True,
          show_layer_names=True)
    return m

def train(x_train, y_train, history_length, horizon, x_val =
    None, y_val = None,
          batch_size = 32, max_epochs = 500,
              early_stopping_patience = 30, weights = '',
          tunning = False, extra_info_name = '', hidden_units
              =128, dropout=0.0):

  #  Creating and configuring general model

  m = prepare_model_LSTM(history_length, weights = weights,
      nb_hidden_units=hidden_units, dropout = dropout)

  callbacks = []
  if not tunning:
    callbacks.append(ModelCheckpoint(filepath='output/best
        -{3}{0}-hl{1}-hor{2}.pkl'.format(str("LSTM"),
                                    history_length, horizon,
                                    extra_info_name),
                            monitor='RMSE',
                            save_best_only=True,
                            save_weights_only=True))
    callbacks.append(EarlyStopping(monitor='RMSE', patience=
        early_stopping_patience))

  # Adding validation data
  if x_val is not None and y_val is not None:
    print("INFO: Training with validation")
    x_val = np.reshape(x_val, (x_val.shape[0], history_length,
        1))
    validation_data = (x_val, y_val)

  else:
    print("INFO: Training without validation")
    validation_data = None

  # Reshaping the data (needed for LSTM model)
  x_train = np.reshape(x_train, (x_train.shape[0],
      history_length, 1))

  # Train
  hist = m.fit(x_train, y_train,
               batch_size=batch_size,
               epochs=max_epochs,
               shuffle=True,
               validation_data=validation_data,
               callbacks=callbacks
               )

  if tunning:
    m.save_weights('/content/gdrive/MyDrive/TFG/
        GeneralisticModel/LSTM/tunning-LSTM-hl{0}-hor{1}.pkl'.
        format(
         history_length, horizon))

  return hist
```

Listing 7.3: LSTM model functions.

```
1
2 from keras.models import Model
3 from keras.layers import Dense, LSTM, GRU, Lambda, dot,
      concatenate, Activation, Input
4 from keras.utils.vis_utils import plot_model
5
6 class LinearModel:
7     def __init__(self, input_shape=(6,), nb_output_units=1):
8         self.input_shape = input_shape
9         self.nb_output_units = nb_output_units
10
11    def __repr__(self):
12        return 'Linear'
13
14    def build(self):
15        i = Input(shape=self.input_shape)
16        x = Dense(self.nb_output_units, activation=None)(i)
17
18        return Model(inputs=[i], outputs=[x])
19
20 def prepare_model_linear(history_length, summary=False, plot=
      False, weights=''):
21     model = LinearModel(input_shape=(history_length,),
          nb_output_units=1)
22     # build & compile model
23     m = model.build()
24     m.compile(loss=RMSE,
25               optimizer='adam',
26               metrics=[RMSE])
27
28     if weights:
29       print(f"Weights: {weights}")
30       m.load_weights(weights)
31     if summary:
32         print(m.summary())
33     if plot:
34       plot_model(m, to_file='model/LinearBaseline.png',
           show_shapes=True, show_layer_names=True)
35
36     return m
37
38 def train_linear(x_train, y_train, history_length, horizon,
39           batch_size = 32, max_epochs = 500,
                 early_stopping_patience = 50, weights = '',
40           extra_info_name = ''):
41
42   #  Creating and configuring general model
43
44   m = prepare_model_linear(history_length, weights=weights)
45
46   callbacks = []
47
48
49
50   callbacks.append(ModelCheckpoint(filepath='output/best-{3}{0}-
       hl{1}-hor{2}.pkl'.format( str("Linear"),
51                             history_length,
52                             horizon,
53                             extra_info_name),
54                       monitor='RMSE',
```

```
                        save_best_only=True,
                        save_weights_only=True))
    callbacks.append(EarlyStopping(monitor='RMSE', patience=
        early_stopping_patience))

    # Train
    hist = m.fit(x_train, y_train,
                 batch_size=batch_size,
                 epochs=max_epochs,
                 shuffle=True,
                 callbacks=callbacks
                 )

    return hist
```

Listing 7.4: Linear model functions.

Bibliography

[1] Abbott. Freestyle libre 2. `https://www.freestyle.abbott/es-es/productos/freestyle-libre-2.html`. Accessed on 2023-09-02.

[2] Eleonora Maria Aiello, Giuseppe Lisanti, Lalo Magni, Mirto Musci, and Chiara Toffanin. Therapy-driven deep glucose forecasting. *Engineering applications of artificial intelligence*, 87, 1 2020.

[3] Ahmad Yaser Alhaddad, Hussein Aly, Hoda Gad, Abdulaziz Al-Ali, Kishor Kumar Sadasivuni, John-John Cabibihan, and Rayaz A Malik. Sense and learn: Recent advances in wearable sensing and machine learning for blood glucose monitoring and trend-detection. *Frontiers in Bioengineering and Biotechnology*, 10, 2022.

[4] Alessandro Aliberti, Irene Pupillo, Stefano Terna, Enrico Macii, Santa Di Cataldo, Edoardo Patti, and Andrea Acquaviva. A multi-patient data-driven approach to blood of glucose prediction. *IEEE Access*, 7:69311–69325, 2019.

[5] Ibrahim Aljamaan and Ibraheem Al-Naib. Prediction of blood glucose level using non-linear system identification approach. *IEEE Access*, 10:1936–1945, 2022.

[6] Y BENGIO, P SIMARD, and P FRASCONI. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5:157–166, 3 1994.

[7] Robert Bevan. BGLP Repository. `https://github.com/robert-bevan/bglp`, 2020.

[8] Robert Bevan and Frans Coenen. Experiments in non-personalized future blood glucose level prediction. 2020.

[9] Brian Bogue-Jimenez, Xiaolei Huang, Douglas Powell, and Ana Doblas. Selection of non-invasive features in wrist-based wearable sensors to predict blood glucose concentrations using machine learning algorithms. *Sensors*, 22, 5 2022.

[10] Jason Brownlee. *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future*. Jason Brownlee, 2017.

[11] Razvan Bunescu, Nigel Struble, Cindy Marling, Jay Shubrook, and Frank Schwartz. Blood glucose level prediction using physiological models and support vector regression. *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, 1:135–140, 2013.

[12] William L Clarke, Daniel Cox, Linda A Gonder-Frederick, William Carter, and Stephen L Pohl. Evaluating clinical accuracy of systems for self-monitoring of blood glucose. *Diabetes Care*, 10:622–628, 9 1987.

[13] Iván Contreras, Silvia Oviedo, Martina Vettoretti, Roberto Visentin, and Josep Vehí. Personalized blood glucose prediction: A hybrid approach using grammatical evolution and physiological models. *PLOS ONE*, 12:e0187754, 2 2017.

[14] Ian Fox, Lynn Ang, Mamta Jaiswal, Rodica Pop-Busui, and Jenna Wiens. Deep multi-output forecasting learning to accurately predict blood glucose trajectories. pages 1387–1395, 2018. 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), London, ENGLAND, AUG 19-23, 2018.

[15] J j. LI, Z p. Qu, Y w. Wang, and J Guo. Research on multi-parameter fusion non-invasive blood glucose detection method based on machine learning. *European Review for Medical and Pharmacological Sciences*, 26:6040–6049, 2022.

[16] Mario García Jiménez. T1diabetesgranadaforecastingmodels. `https://github.com/rentton/T1DiabetesGranadaForecastingModels`, 2023. Last accessed: 2023-09-02.

[17] Keras. Keras documentation. Accessed on: 2023-09-07.

[18] Dae-Yeon Kim, Dong-Sik Choi, Ah Reum Kang, Jiyoung Woo, Yechan Han, Sung Wan Chun, and Jaeyun Kim. Intelligent ensemble deep learning system for blood glucose prediction using genetic algorithms. *COMPLEXITY*, 2022, 10 2022.

[19] John F Kolen and Stefan C Kremer. Gradient flow in recurrent nets: The difficulty of learning longterm dependencies, 2001.

[20] Kezhi Li, John Daniels, Chengyuan Liu, Pau Herrero, and Pantelis Georgiou. Convolutional recurrent neural networks for glucose prediction. *IEEE Journal of Biomedical and Health Informatics*, 24:603–613, 2 2020.

[21] 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*, 24:414–423, 2 2020.

[22] Ming Liu, Ge Xu, Yuejin Zhao, Lingqin Kong, Liquan Dong, Fen Li, and Mei Hui. Diffuse imaging approach for universal noninvasive blood glucose measurements. *Frontiers in Physics*, 10, 3 2022.

[23] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator. *Journal of Diabetes Science and Technology*, 8:26–34, 1 2014.

[24] Cindy Marling and Razvan Bunescu. The ohiot1dm dataset for blood glucose level prediction: Update 2020. *CEUR Workshop Proceedings*, 2675:71–74, Sep 2020.

[25] 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.

[26] Sadegh Mirshekarian, Razvan Bunescu, Cindy Marling, and Frank Schwartz. Using lstms to learn physiological models of blood glucose behavior. pages 2887–2891. IEEE, 7 2017.

[27] Sadegh Mirshekarian, Hui Shen, Razvan Bunescu, and Cindy Marling. Lstms and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data. pages 706–712. IEEE, 2019.

[28] Mario Munoz-Organero. Deep physiological model for blood glucose prediction in t1dm patients. *Sensors*, 20, 2020.

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

[30] 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*, 26:2758–2769, 6 2022.

[31] 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, 2021.

[32] C. Rodriguez-León, M.D. Avilés-Pérez, O. Baños, M. Quesada-Charneco, P.J. Lopez-Ibarra, C. Villalonga, and M. Muñoz-Torres. T1diabetesgranada: a longitudinal multi-modal dataset of type 1 diabetes mellitus. `https://osf.io/vd45b/?view_only=9949c27596c14f8198f4452f60c98ba4`, 2023.

[33] Pouya Saeedi, Inga Petersohn, Paraskevi Salpea, Belma Malanda, Suvi Karuranga, Nigel Unwin, Stephen Colagiuri, Leonor Guariguata, Ayesha A. Motala, Katherine Ogurtsova, Jonathan E. Shaw, Dominic Bright, and Rhys Williams. Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas, 9th edition. *Diabetes Research and Clinical Practice*, 157:107843, 11 2019.

[34] Signe Schmidt, Daniel A Finan, Anne Katrine Duun-Henriksen, John Bagterp Jorgensen, Henrik Madsen, Henrik Bengtsson, Jens Juul Holst, Sten Madsbad, and Kirsten Norgaard. Effects of everyday life events on glucose, insulin, and glucagon dynamics in continuous subcutaneous insulin infusion-treated type 1 diabetes: Collection of clinical data for glucose modeling. *Diabetes Technology and Therapeutics*, 14:210–217, 3 2012.

[35] Smart Health Research Group. BGLP Results Page. `http://smarthealth.cs.ohio.edu/bglp/bglp-results.html`, 2020.

[36] suetAndTie. Clarkeerrorgrid. `https://github.com/suetAndTie/ClarkeErrorGrid/blob/master/ClarkeErrorGrid.py`. Accessed on 2023-06-15.

[37] Qingnan Sun, Marko V Jankovic, Lia Bally, and Stavroula G Mougiakakou. Predicting blood glucose with an lstm and bi-lstm based deep neural network. 2018.

[38] Baoyu Tang, Yuyu Yuan, Jincui Yang, Lirong Qiu, Shasha Zhang, and Jinsheng Shi. Predicting blood glucose concentration after short-acting insulin injection using discontinuous injection records. *Sensors*, 22, 11 2022.

[39] Ali El Idrissi Touria and Idri. Deep learning for blood glucose prediction: Cnn vs lstm. pages 379–393. Springer International Publishing, 2020.

[40] Stella Tsichlaki, Lefteris Koumakis, and Manolis Tsiknakis. Type 1 diabetes hypoglycemia prediction algorithms: Systematic review. *JMIR Diabetes*, 7:e34699, 7 2022.

[41] Ching-Shih Tsou, Christine Liou, Longsheng Cheng, and Hanting Zhou. Quality prediction through machine learning for the inspection and manufacturing process of blood glucose test strips. *Cogent Engineering*, 9, 12 2022.

[42] Qiaoyun Wang, Feifei Pian, Mingxuan Wang, Shuai Song, Zhigang Li, Peng Shan, and Zhenhe Ma. Quantitative analysis of raman spectra for glucose concentration in human blood using gramian angular field and convolutional neural network. *Spectrochimica Acta - Part A: Molecular and Biomolecular Spectroscopy*, 275, 7 2022.

[43] World Health Organization. Diabetes. `https://www.who.int/health-topics/diabetes#tab=tab_1`, 2019.

[44] Jun Yang, Lei Li, Yimeng Shi, and Xiaolei Xie. An arima model with adaptive orders for predicting blood glucose concentrations and hypoglycemia. *IEEE Journal of Biomedical and Health Informatics*, 23:1251–1260, 5 2019.

[45] Zhenyi Ye, Jie Wang, Hao Hua, Xiangdong Zhou, and Qiliang Li. Precise detection and quantitative prediction of blood glucose level with an electronic nose system. *IEEE Sensors Journal*, 22:12452–12459, 7 2022.

[46] Yongjun Zhang and Guangheng Gao. Optimization and evaluation of an intelligent short-term blood glucose prediction model based on noninvasive monitoring and deep learning techniques. *Journal of Healthcare Engineering*, 2022, 2022.

[47] Taiyu Zhu, Lei Kuang, Kezhi Li, Junming Zeng, Pau Herrero, and Pantelis Georgiou. Blood glucose prediction in type 1 diabetes using deep learning on the edge. IEEE, 2021. IEEE International Symposium on Circuits and Systems (IEEE ISCAS), Daegu, SOUTH KOREA, MAY 22-28, 2021.

[48] Taiyu Zhu, Kezhi Li, Jianwei Chen, Pau Herrero, and Pantelis Georgiou. Dilated recurrent neural networks for glucose forecasting in type 1 diabetes. *Journal of Healthcare Informatics Research*, 4, 9 2020.

[49] Ting Zhu, Wenbo Wang, and Min Yu. A novel blood glucose time series prediction framework based on a novel signal decomposition method. *Chaos Solitons and Fractals*, 164, 11 2022.