FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Hypoglycaemia Prediction on Type 1 Diabetes Patients using Continuous Glucose Monitoring and Health Record Data

Guilherme Lucas Peralta

Mestrado em Bioengenharia

Supervisor: Prof. Vitor Santos Costa

Second Supervisor: Prof. Pedro Brandão

October 19, 2022

Hypoglycaemia Prediction on Type 1 Diabetes Patients using Continuous Glucose Monitoring and Health Record Data

Guilherme Lucas Peralta

Mestrado em Bioengenharia

October 19, 2022

Abstract

Diabetes is a chronic disease characterized by elevated glycaemic values which affects more than 10% of the world's population. These higher than normal blood sugar values are known as hyperglycaemia. A normal treatment in type 1 diabetes patients is the use of insulin to lower the high blood glucose levels. However, if not properly dosed, boluses can give rise to a even worse problem than hyperglycaemias, hypoglycaemias. These consist in low blood glucose levels and can lead to serious consequences in our body and are reported to be one of the biggest concerns of the diabetic population.

Consequently, we felt motivated to develop a system which may help prevent hypoglycaemic events. Given the rising use of artificial intelligence in medicine, we used a combination of physiological models and LSTM based deep neural networks to predict the existence of a hypoglycaemia during the next hour.

Both personalized and generalized models were tested, and we reached the conclusion that, in our experiments, the generalized model worked the best. Several input preprocessors were also tested and the best performing one was a combination of a filtered CGM signal, with the available insulin and carbohydrates values and patient specific information (insulin type, gender and age range). To the best of our knowledge, this model outperformed those described in the literature for the same type of problem, showing a balanced accuracy of 0.876 ± 0.017, a sensitivity of 0.810 ± 0.050, a specificity of 0.942 ± 0.018, a precision of 0.568 ± 0.023 and a MCC of 0.843 ± 0.019 when considering hypoglycaemias point to point.

Looking closely at the model's probability plots, we observe a virtually perfect recall of hypoglycaemic episodes. However, the model still outputs too many false positives, which can make the patient loose trust in the system, suggesting a need for future work.

To sum up, we were able to create a deep learning pipeline able to aid type 1 diabetic patients in their disease management.

Keywords: classification, deep learning, diabetes, glucose, hypoglycaemia, LSTM, prediction

Resumo

A diabetes é uma doença crónica caracterizada por valores glicémicos elevados e que afeta mais de 10% da população mundial. Estes valores de glucose no sangue acima do normal são conhecidos como hiperglicemias. O tratamento mais comum em diabéticos do tipo 1 é o uso de insulina para que os valores de glucose no sangue se reduzam. Porém, se a dose de insulina administrada for incorreta, este tratamento poderá levar a um problema ainda maior do que as hiperglicaemias para o paciente, as hipoglicemias. Estas últimas correspondem a valores de glucose no sangue demasiado baixos e podem resultar em consequências bastante sérias no corpo humano, sendo relatadas como um dos maiores medos na população diabética.

Graças a isto, sentimo-nos motivados a desenvolver um sistema capaz de prevenir a existência de eventos hipoglicémicos. Dado o crescimento do uso de inteligência artificial na medicina, usámos uma combinação de modelos fisiológicos e redes neuronais profundas baseadas em *LSTM* para prever a existência de uma hipoglicemia durante a próxima hora.

Tanto modelos personalizados como modelos genéricos foram testados, tendo-se chegado à conclusão que, nas nossas experiências, os modelos genéricos funcionavam melhor. Testaram-se também vários pré-processadores de *inputs*, sendo o melhor uma combinação de sinal *CGM* filtrado, com os valores de insulina e carbohidratos disponíveis e informações individuais de cada paciente (tipo de insulina, género e intervalo de idades). Tanto quando pudemos verificar, o modelo criado ultrapassa em performance aqueles descritos na literatura para o mesmo tipo de problema, exibindo uma exatidão equilibrada de 0.876 ± 0.017, uma sensibilidade de 0.810 ± 0.050, uma especificidade de 0.942 ± 0.018, uma precisão de 0.568 ± 0.023 e um *MCC* de 0.843 ± 0.019 quando considerando as hipoglicemias ponto a ponto.

Examinando os gráficos de probabilidade à saída do modelo, observamos uma sensibilidade virtualmente perfeito dos episódios de hipoglicemia. No entanto, o modelo continua a emitir demasiados falsos positivos, o que poderá levar o paciente a perder alguma confiança no mesmo, sugerindo a necessidade de trabalho futuro.

Em suma, fomos capazes de criar uma *pipeline* de *deep learning* capaz de ajudar ao controlo da doença em diabéticos do tipo 1.

Keywords: classificação, deep learning, diabetes, glucose, hipoglicemia, LSTM, previsão

Agradecimentos

Aos professores Pedro Brandão e Vitor Santos Costa, pelo apoio, ajuda e conselhos durante este ano letivo.

Aos meus pais e família, por serem o meu porto seguro durante todos estes anos de crescimento.

Ao Gama, Fábio, Jorge, Maria, Matos, Tiago, Patrícia e Francisco, por me proporcionarem ótimos momentos de partilha nas duas residências onde tive prazer de viver e por suportarem os meus *breakdowns*.

Aos meus restantes amigos, estejam em Ovar ou no Porto, por se manterem a meu lado durante todos estes anos, por momentos inesquecíveis e por tornarem estes anos de faculdade bastante melhores.

Uma nova etapa começará, mas sei que se manterão ao meu lado!

A todos, um obrigado do fundo do coração,

Guilherme Peralta

"I am always ready to learn,
although I do not always like being taught."

Richard P. Feynman

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
AUC	Area Under the Receiver Operating Characteristic Curve
Bi-LSTM	Bi-directional Long Short-Term Memory
CGM	Continuous Glucose Monitoring
CPHS	Continuous Glucose Monitoring-Based Prevention of Hypoglycaemia System
GRU	Gated Recurrent Unit
HbA1c	Glycated Haemoglobin
LSTM	Long Short-Term Memory
MCC	Matthews Correlation Coefficient
MSE	Mean Squared Error
NN	Neural Network
PH	Prediction Horizon
PRED-EGA	Prediction Error-Grid Analysis
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
tanh	Hyperbolic Tangent

Chapter 1

Introduction

Diabetes Mellitus is a chronic disease characterized by high blood glucose levels, which affects over 10% of the world's population. In order to avoid fatal consequences, this disease must be coped with in different manners according to the type of diabetes. Diabetic people, especially those with type 1 diabetes, should monitor the amount of blood glucose at regular intervals and use insulin, though there might exist rare cases where they do not use it [68]. Furthermore, the current disease monitoring strategies require a lot of self-care behaviors, being burdensome for the patient, and they are quite costly to the healthcare systems [85][22][33][10].

The use of insulin makes type 1 diabetics quite prone to having hypoglycaemic events [61], which is rather undesirable given the severity of the sequels that can arise from low blood sugar in the human body [35][40]. Some of these include recurrent or persistent psycho-social morbidity, recurrent physical morbidity, or both, long-term impacts in cognition, dementia, and, in some cases, even death [35]. Consequently, hypoglycaemias are a huge concern among diabetics [61].

Hypoglycaemias are almost in their totality explained by iatrogenic causes and are most common in type 1 diabetes [49] [16] [41]. Be that as it may, gender, age, duration of the diabetes condition, HbA1c levels and the pre-existence of other conditions are also factors that can potentiate the existence of hypoglycaemic events. Given that their symptoms and severity are quite specific to each patient, hypoglycaemias may sometimes be neglected, a phenomena known as hypoglycaemia unawareness. This problem increases the patient's risk for a severe low blood glucose reaction, thus they need to be extra careful in glucose monitoring [16].

Nowadays, we are witnessing a rise in the use of technology to aid the patient manage the disease. Such technologies include closed-loop systems, blood glucose event alarms and personalized decision systems [92]. This, allied with the high impact that hypoglycaemic events can have in a diabetic person's life, particularly in type 1 diabetes patients, and the motivated us to develop a classification system that was able to predict in a short to medium term the existence of a hypoglycaemia. With this type of system we also wish to help avoid severe hypoglycaemias in patients suffering from hypoglycaemia unawareness.

With this project we aimed to create a model which surpasses those currently described in the available literature. However, most of the currently available models are rather focused on the prediction of blood glucose values instead of hypoglycaemic events, thus there may be some inaccuracy in the comparison of the results. Out of those that are indeed focused on the prediction of hypoglycaemic events, we believe that they lack a long enough PH for the patient to prepare for the hypoglycaemia, with most of the PHs going only up to 30 minutes [70][75][75][63][32]. So, we would like to enhance this aspect, using a minimum PH of 1 hour.

However, there are still many limitations to the use of these models in real-life situations. The fact that there are a lot of aspects correlated to hypoglycaemic events, it makes it virtually impossible to develop an algorithm that can take into account every aspect of the patient's daily life. This, associated with factors like sensor malfunction, insufficient sensor accuracy and slow subcutaneous absorption of insulin, make it hard to have a model which behaves perfectly.

Provided that, we decided that we would probably need to compromise in the model performance. Additionally, Clarke Error Grid shows us that the clinical severity of failing to predict a hypoglycaemia is quite high, while, most of the time, it is not clinically unsafe to have a false positve [60]. Thus, based on these facts, we prioritized the maximizing the number of true positives over the minimization of the number of false positives.

To develop this work, we used the OhioT1DM Dataset, which is comprised of 12 people with type 1 diabetes using an insulin pump with CGM and a fitness band. This dataset keeps track of 8 weeks' worth of data, with the patients also including self-reported life events, such as carbohydrates intake, sleep quality, illness, among others.

The pipeline developed during this work was successfully integrated in the context of GATEKEEPER AI4DM. GATEKEEPER is a European Multi Centric Large-Scale Pilot on Smart Living Environments with the main objective of creating a platform that connects businesses, entrepreneurs, healthcare providersand elderly citizens and the communities they live in. The project intends to originate an open, trust-based arena for matching ideas, technologies, user needs and processes, aimed at ensuring healthier independent lives for the ageing populations [12]. Figure 1.1 shows the system's architecture, where the patient can interact with the system to populate the database with its data, once there is enough data the system will use it to train the prediction model and, at that point, the user is free to ask the system for hypoglycaemia predictions for the next hour.

This document is organized in five chapters. Chapter 2 includes the fundamental concepts to understand the scope of the developed work. Chapter 3 focuses on the available products and scientific work that tackle the same problem as we did. Chapter 4 refers to analysis of the used dataset, the used methodologies in order to achieve the final pipeline, the results from this work and their analysis and discussion. Finally, the conclusions and future work from this dissertation are exposed in chapter 5.

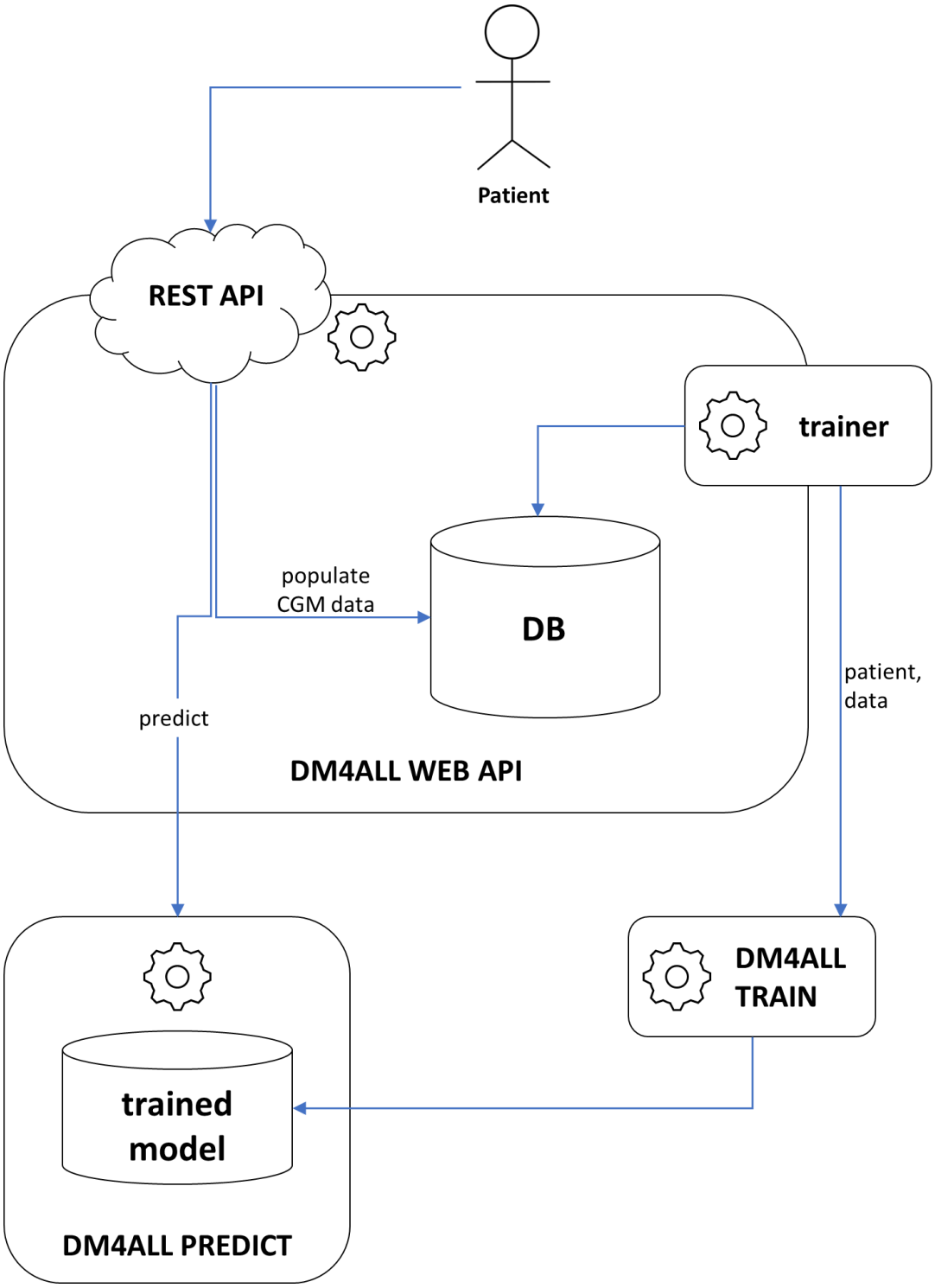

Figure 1.1: Diagram of the system's architecture

Chapter 2

Background Knowledge

Throughout this section we will introduce the basic concepts for the understanding of the work developed. We will introduce diabetes and the costs associated with the disease, we will explain what a time series is, provide a brief history of AI, dive into the world that is ML with a special focus on DL and RNNs, explain the transfer learning process and present the evaluation metrics that are used in this work.

2.1 Diabetes

Diabetes is a chronic disease characterized by elevated glycaemic values, which, in 2021, affected 536.6 million people worldwide, representing more than 10% of the world's population [11][57]. Europe is reported to have 61 million people suffering from the disease, with Portugal alone having a prevalence of 13.6% between the ages of 20 and 79 years old, which corresponds to over 1 million Portuguese [43][9]. The number of cases of diabetes has quadrupled in the last four decades, hence why a global target was set to halt by 2025 both the rise in diabetes and obesity [11][2].

This metabolic disease, occurs either when the body cannot effectively use the produced insulin, or when the pancreas does not produce enough insulin. Since this hormone is key in blood sugar regulation, uncontrolled diabetes has the common effect of higher than normal blood sugar values, known as hyperglycaemia. Over time, hyperglycaemia can result in serious damage of the body's systems, with emphasis on blood vessels and nerves. Even more serious consequences of diabetes are limb amputation and early death [11]. Symptoms include thirst (polydipsia), an excessive excretion of urine (polyuria), weight loss, constant hunger, fatigue and vision changes. As these symptoms might be less pronounced on some types of diabetes, it is important to be on the lookout, or else late diagnosis will arise, coming with further complications [2].

There are three main types of diabetes:

- **Type 1 diabetes** (previously known as juvenile or insulin-dependent diabetes) is characterized by a T-cell-mediated autoimmune response to beta-cells in the pancreas, which results in an absolute lack of insulin production. Out of all cases of diabetes, the prevalence of type 1 diabetes ranges from 5% to 10% [36][78].

- **Type 2 diabetes** (previously known as adult or non-insulin-dependent diabetes) is due to an atypical increased body resistance to insulin, which cannot be overcomed due to an inability to produce enough insulin. However, the way how this insulin resistance provokes beta-cell failure is still unclear. Type 2 diabetes prevails in 90% to 95% of all diabetes cases [36][78].

- **Gestational diabetes** is a type of glucose intolerance reported in some women during gestation [36].

Current strategies to manage diabetes rely on medication, a healthy lifestyle and glucose values' monitoring, requiring a lot of self-care behaviors which are often a big burden for patients [85][22][33]. During the entirety of the day, glycaemic levels will oscillate up and down. Up until a certain range, this blood sugar variation is normal. However, two abnormal situations can occur: the glycaemic levels can be **too high** (>240mg/dL), which is called a **hyperglycaemia**; or blood sugar levels may be **too low** (<70mg/dL), known as **hypoglycaemia** [15][16].

The human brain uses glucose as an energy source and is extremely vulnerable to its deprivation. Under normal conditions, the brain is unable to synthesise and store glucose. Consequently, when in a situation of hypoglycaemia, the normal body's response is the suppression of insulin, followed by the secretion of glucagon and epinephrine, counter regulatory hormones [93].

Hypoglycemic symptoms are age specific and individual [93]. However, reported symptoms are: shakiness; anxiety; sweating, chills and clamminess; irritability or impatience; confusion; accelerated heartbeat; dizziness; hunger; nausea; pallor; sleepiness; weakness; blurred or impaired vision; tingling or numbness in the lips, tongue or cheeks; headaches; coordination problems; sleeping problems; seizures [16]. In the case of iatrogenic hypoglycaemia, i.e., hypoglycaemia induced by medical intervention, it has been reported to cause recurrent or persistent psychosocial morbidity, recurrent physical morbidity, or both, and, in some cases, even death [35]. It can also lead to long-term impacts in cognition and is potentially linked with dementia [40]. This problem causes great corcern among the diabetic population, with the fear of hypoglycaemia being rated the same degree as that of end-stage renal disease or sight-threatening retinopathy [61].

In average, a type 1 diabetic has roughly 2 hypoglycaemic episodes per week, and severe hypoglycaemia has a annual prevalence of 30 to 40%, with an incidence in the same period of 1.0 to 1.7 episodes per patient. Though normally type 1 diabetics are more prone to hypoglycaemias, due to their eventual dependence on insulin therapy, a study found that roughly the same proportion of severe hypoglycaemic episodes requiring emergency medical assistance occurred between type 1 diabetics and type 2 diabetics on an insulin treatment [61].

Potential risk factors for hypoglycaemia in type 1 diabetes depend on the following aspects: gender, age, duration of the diabetes condition, HbA1c levels and the pre-existence of other conditions, such as diabetic neuropathy and micro, macro or neuropathic complications [49]. Furthermore, often severe hypoglycaemia (<50mg/dL) occurs during sleep and, when occurring while the patient is awake, is sometimes not accompanied by warning symptoms [4] [5].

Evidence shows that the onset of moderate hypoglycemia is preceded by release of counter-regulatory hormones, such as glucagon, growth hormone, epinephrine, and cortisol, causing, for instance, variations in heart-rate [94].

One of the most auspicious solutions for glycaemic control is the use of a closed-loop system, also called an artificial pancreas. This system consist of a glucose sensor for continuous glucose monitoring (CGM), that allows an interstitial glucose measurement every 1-5 minutes, an artificial pump and an algorithm which continuously evaluates the data from the glucose sensor and changes the insulin infusion rate. It may need to be finger-stick calibrated, but can also be factory calibrated. Other hormones, like glucagon, can also be pump delivered [29] [38].

Though there have been upgrades to this technology, there are still some limitations to a fully automated closed-loop system. Some of these include sensor malfunction, insufficient sensor accuracy, the 10-15 minutes delay inherent to interstitial measurements when compared to actual blood glucose values, slow subcutaneous absorption of insulin and the fact that no algorithm can take into account every single aspect of the patient's everyday life. There is also a need to address problems like cost-effectiveness and patient and health care professionals training to use the system [29][38][88].

A Clarke error grid divides the blood glucose prediction error in several zones. The theory is that if the prediction error is within zone A, then the treatment is correct, if it is within zone B, then the treatment is deemed as "not inappropriate", however, if the error is within zones C, D or E, the risks associated with such prediction errors will probably lead to clinical issues and are increasingly worse from C to E [60]. When analysing a Clarke Error Grid, as the one depicted in figure 2.1, one can tell that, in the most extreme case, up to true blood glucose value of 130mg/dL, a prediction of hypoglycaemia would either fall into a zone A or B error, either way being clinically safe. However, failing to predict a hypoglycaemia falls into zone D, being considered a quite severe clinical fail.

2.1.1 Cost of Diabetes

In 2021, the estimated world expenses related to diabetes were USD 966 billion, with tendencies to rise up to USD 1054 billion by 2045 [57].

In Portugal, diabetes had, in 2018, an estimated direct cost of €740.7 million to the Portuguese healthcare system. However, if total costs are considered, it is estimated that, in 2014, between €1300 million and €1500 million were spent on diabetes [10].

In 2017, the American Diabetes Association performed an economical study of the costs of diabetes in the U.S.. This study revealed that, in that year, the estimated total costs of diagnosed diabetes were USD 327 billion, of which USD 237 billion are direct medical costs [17]. During

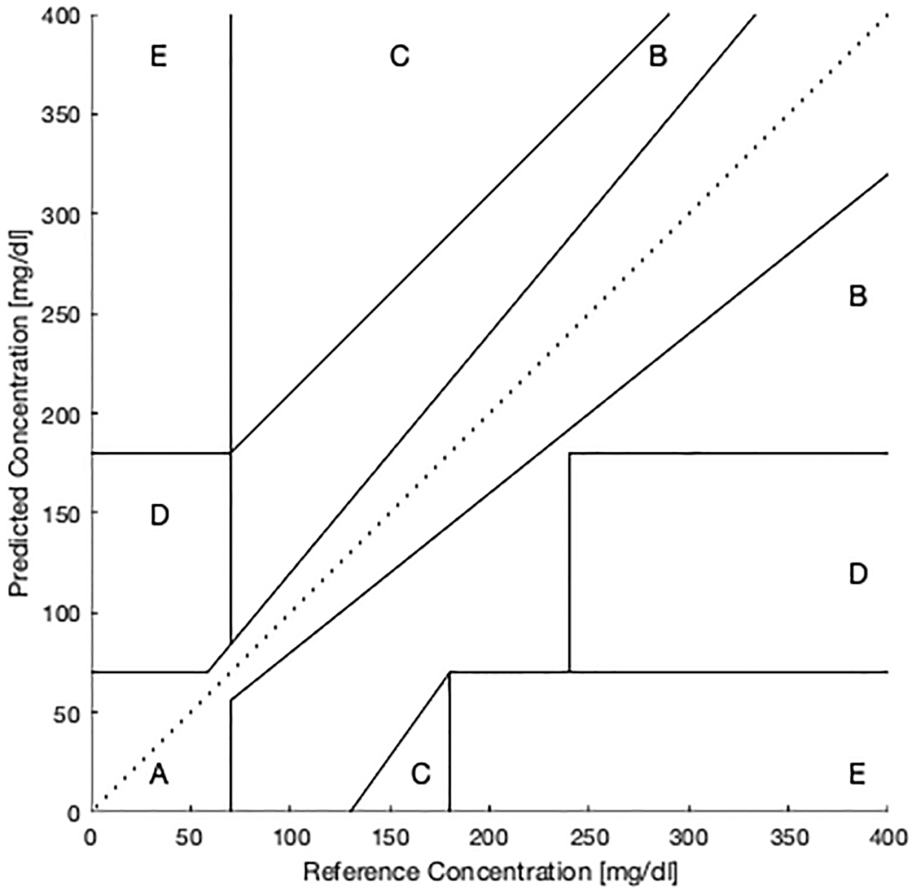

Figure 2.1: Clarke Error Gird. Source: [60]

the same year, the U.S. reported USD 3.5 trillion in total in health care expenses, which means that diabetes covers about 10% of all health care expenses in the U.S. [59]. After adjusting for inflation, this value represented an increase of 26% when compared to 2012 numbers [17]. The total costs in 2017 are especially shocking when you know that, in 2003, the predicted value of total costs in 2020 was USD 192 billion [6].

Segmenting the medical expenditures, its largest components are [17]:

- 30% on hospital inpatient care;
- 30% on diabetes medication prescription;
- 15% on anti-diabetic agents and diabetes supplies;
- 13% on physician office visits.

In the case of the UK, in 2010/2011, diabetes have cost approximately £23.7 billion, which represents 10% of the total health resource expenditure, of which £21.8 billion are related to type 2 diabetes expenses, and is estimated to increase to a value of £39.8 billion in 2035/2036, of which £35.6 billion account for type 2 diabetes expenses. When looking at the cost estimations for moderate hypoglycaemia alone, in 2010/2011 type 1 accounted for £19.2 million and type

2 for £22.6 million, tending to go up to values of £31.7 million and £38.0 million, respectively, by 2035/2036. In the case of severe hypoglycaemia, in 2010/2011 type 1 accounted for £13.9 million and type 2 for £16.4 million, probably reaching values of £19.2 million and £21.5 million, respectively, by 2035/2036 [51].

Some economical studies suggest that improving glycaemic control would have a significant long-term impact on the risks and consequently costs of diabetes [36][18].

2.2 Time Series

A time series consists of a collection of observations of the same variable sequentially measured through time. Time series can be divided into continuous, if the measurements are performed continuously through time, or discrete time series, when the measurements are taken at discrete time points. Nonetheless, the measured variable can be discrete or continuous in either case, i.e., the classification of a time series as continuous or discrete refers exclusively to the time axis. An usual characteristic of time series data is a dependence between current and past observations, thus, in its analysis, the order in which the time points were collected must be taken into account [31].

The main goals when performing a time series analysis are [31]:

- **Description**: represent the data using graphical methods and/or summary statistics.
- **Modelling**: finding an appropriate statistical model which can describe the data-generating process.
- **Forecasting** (or prediction): estimating the values of the series in the future.
- **Control**: if the forecast is good, the analyst may act accordingly and control a given process.

Here are some important aspects to take into consideration when modelling a time series [30]:

- **Trend**: it represents the change in the mean of the data in the long term, which can affect the model's performance, in which case it should be removed.
- **Seasonality**: corresponds to the regular and predictable variations in a given period. For example, sparkling wine sales are expected to be bigger in the new year's period. Just like the trend, it may also affect model performance, in which case it should also be removed.
- **Cyclic Patterns**: these correspond to when there is no fixed period for predictable oscillations. A common example is the economic cycle, where there are periods of recession and periods of growth.
- **Residuals** (or irregular component): corresponds to the remaining part after removing trend, seasonality and cyclic patterns and can also impact the time series' dynamics.

- **Auto-correlation**: it evaluates the degree of dependency of the time series on its historical data. As previously mentioned, the current point in a time series usually is related to what happened in the past. The auto-correlation evaluates the point-to-point dependency of the current time-point to the past ones. If, on all lags, the time series exhibits low auto-correlation values, it is considered white noise.

Thus, at any given point in time, we can decompose the time series (y) as follows [30]:

$$y = Trend + Seasonal + Cyclic + Residuals \tag{2.1}$$

2.3 Artificial Inteligence

2.3.1 Brief History of AI

As Stuart J. Russell and Peter Norvig mention in their book *Artificial Intelligence - A Modern Approach*, the human kind relies on their intelligence to comprehend how the man thinks and to understand, perceive, predict and manipulate the environment by which he is surrounded. However, "the field of artificial intelligence goes further still: it attempts not just to understand but also to build intelligent entities" [79].

But, after all, what is AI? Definitions of this field fluctuate on two main dimensions: *thought processes/reasoning* and *behavior*; and tend to evaluate their success by comparing with human performance or with an ideal concept of intelligence (rationality, i.e. the system is rational if it does the right thing) [79]. According to these divisions, four distinct definitions of AI are summarized in table 2.1.

The field of AI had its first work published in 1943 and several researchers tried to explore these uncharted waters in the following decades, during a time where computers were pictured as something that could only do arithmetic and nothing more. Consequently, AI researchers were constantly breaking barriers in a period known as the "Look, Ma, no hands!" era (1952-1969), during which Rosenblatt developed his famous perceptron convergence theorem [79].

However, the years succeeding this era did not do it justice, as most AI research projects, though working in simple examples, would miserably fail when attempted in more complex ones. Furthermore, the use of representations of basic facts about a given problem and using a series of steps to solve it was a common method in early AI programs. This led to intractability in many of the problems that were attempted to be solved with AI, as the community tried to scale up in complexity. "The fact that a program can find a solution in principle does not mean that the program contains any of the mechanisms needed to find it in practice" [79]. Additionally, AI algorithms still had fundamental limitations, which ultimately conditioned the research in the area [79].

Between 1969 and 1979, some authors successfully tried to create knowledge-based systems [79]. The DENDRAL program and Roger Schank's work on natural language serve as a good example of the advances that domain knowledge allowed in AI [45] [81] [82]. And it was not

Table 2.1: Definitions of AI organized in four different categories. Source: [79]

Systems that think like humans:	**Systems that think rationally:**
"The exciting new effort to make computers think (...) machines with minds, in the full and literal sense" (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning..." (Bellman, 1978)	"The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act" (Wiston, 1992)
Systems that act like humans:	**Systems that act rationally:**
"The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)	"A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990) "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

until the 1980s that AI became a relevant industry, worth around $2 billion in sales in 1988 [79]. In the mid-1980s, there was a reinvention of the back-propagation learning algorithm, which was vastly applied to learning problems, namely in computer science. Since then, a new approach was made to research in AI: rather than proposing new theories, the work is more commonly built upon existing ones, thus basing claims either on rigorous theorems, or on robust experimental evidence instead of on intuition. On top of that, research stopped focusing so much on toy examples to start to further address real-world applications [79].

2.3.2 Machine Learning

Due to its growing importance in the last decade, ML has almost been used as a synonym of the broader field that is AI, with most current advances in AI involving ML. However, it is indeed a subfield of AI which aims to develop computer programs with the ability of automatically improving with experience [24][64].

Using a quite simple analogy, Mikey Shulman explains the gap that ML fills. He compares traditional programming to a precise recipe where the precise amount of each ingredient is explicit, as well as the exact amount of mixing and baking time that the computer can follow. This evidences the difficulty that one would have in writing code, for instance, to recognize pictures of different people. However, this is a fairly simple task for a human to perform. Thus, ML tries to mimic the human approach and lets the computer learn the task themselves through experience [24].

ML currently powers many of the current society's developments, ranging from recommendation algorithms, to speech to text translation, or even biomedical solutions [56]. However, these

algorithms do not learn tasks out of nowhere; the new oil, as Forbes describes it, is necessary for the machinery to work [20]. This mentioned new oil is data, from the most variate sources, to perform the most variate tasks. And the good news for this field is that, as most daily tasks and services can currently be performed using mobile gadgets connected to the internet, data is constantly being gathered. Just to quantify how massive this emerging market is becoming, internet users generate around 2.5 quintillion bytes of data every day, with the big data analytic's market expected to be worth $103 billion by 2023 [73].

ML is usually divided into two phases: training and testing. This allows to develop a robust model by assessing in the test data the degree to which the model has overfitted the training data. If the model is overfitted, it has overly adjusted its parameters to the training set, thus not having a good performance in the test set, and lacking a capacity of generalization. On the contrary, if it is underfitted, it will under-perform in both the train and test sets. But, then, how can one tell how well the model is performing without facing a problem of data leakage, i.e., keeping the test data unseen? The solution lies in a division of the training data in two portions, an actual training set and a validation set, where an anticipation of the test set's performance is expected [27][86].

ML divides, mainly, into three subcategories:

- **Supervised ML**: this is the most usual form of ML. It works by using labelled datasets and training the algorithm with this data. Given an input, the machine will produce an output in the form of a vector of scores, one for each category. The model's performance is optimized by means of an objective function, which evaluates the error between the output score and the expected one, that has its error reduced by an update on the internal adjustable parameters of the algorithm [56].

- **Unsupervised ML**: this form of ML tries to categorize patterns in unlabelled data, which people were not explicitly looking for. A good example would be that of recommendation algorithms [24].

- **Reinforcement ML**: it works using a trial and error mechanism which aims to find the best action by establishing a reward system. A quite common application of this type of algorithm is autonomous driving [24].

ML algorithms can be trained to perform several types of tasks: classification analysis, regression analysis, data clustering, association rule learning, feature engineering for dimensionality reduction, among others. Here are some examples of ML algorithms: Naive Bayes, logistic regression, K-nearest neighbors, support vector machine, random forest and K-means.

2.3.2.1 Deep Learning

Though relevant, conventional ML approaches posed limitations when processing natural data in their raw form. This meant that developing a ML or pattern-recognition system involved transforming the raw data through feature extractors, which meant a lot of careful engineering and

considerable domain expertise, so that a representation which the learning subsystem could interpret would be created [56].

The solution for this problem resides in DL, a subset of ML algorithms which uses representation learning (a series of methods which make possible feeding the system raw data and automatically learn new representations for classification or detection processes). In DL, multiple levels of representation are used successively using a composition of simple but non-linear modules which have the ability to successively represent their input, starting at the raw data, in progressively higher and more abstract levels of representation. When scaled, this approach allows for very complex functions to be learned without human designed features, just by a general-purpose learning procedure [56].

The basic unit of a neural network is the perceptron, which was proposed in 1958 by Rosenblatt. The perceptron consists of a single unit that applies a set of weights to the inputs, adding the bias and applying an activation function to the result of the previous operations, as shown by equation 2.2 [77]. To the nodes through which data and computations flow we call neurons. In deep learning, it is good practice to scale the data in the 0 to 1 range, prior to training.

$$y = h(\sum_{i=0}^{l} x_i \times w_i + b) \tag{2.2}$$

A representation of the perceptron is shown in figure 2.2.

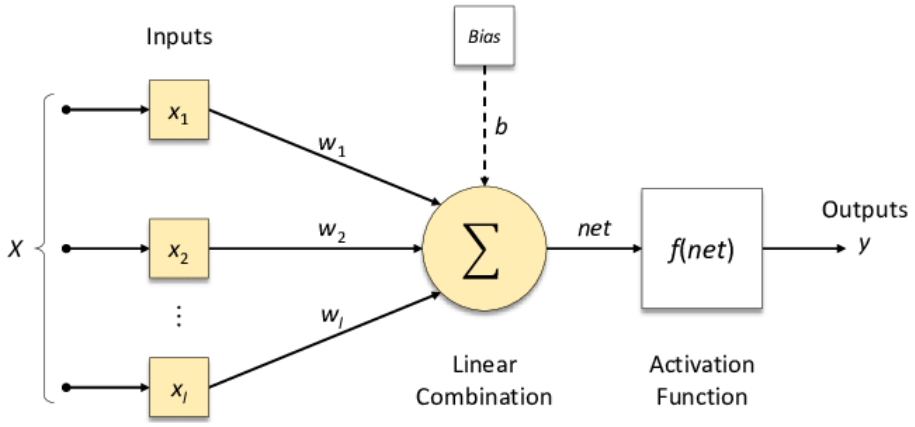

Figure 2.2: Depiction of a Perceptron. Source: [71]

The activation function conditions the output of the unit, bringing non-linearity to the unit's output. There are several distinct activation functions. Some of the most common ones will now be listed [55][28][39]:

- **Sigmoid**: this activation function's output ranges from 0 to 1, having the larger inputs closer to 1 and the smaller inputs closer to 0. This value can be thought of as a probability value. Given that it requires the computation of an exponential, it is highly costly to compute. Additionally, this type of function suffers from a saturation problem, as when it reaches the minimum or the maximum values, the derivative equals zero, which means that the weights

are not updated. Consequently, a process known as vanishing gradients occurs, as the loss function's gradient with respect to the weights fades towards zero. The sigmoid activation function is described by equation 2.3 and is plotted in figure 2.3.

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

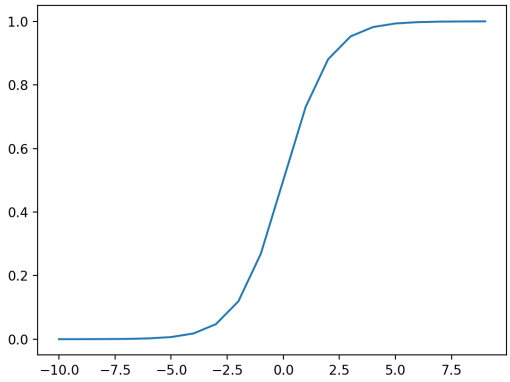

Figure 2.3: Plot of the sigmoid activation function

- **Rectified Linear Unit** (ReLU): this one is the most used activation function. Due to its easy computation and simplicity in backpropagation, the network has a very quick convergence. It also does not saturate in positive values, as the derivative equals 1 for all positive values. However, saturation and vanishing grandient do occur for negative values. The ReLU activation function is described by equation 2.4 and is plotted in figure 2.4.

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2.4}$$

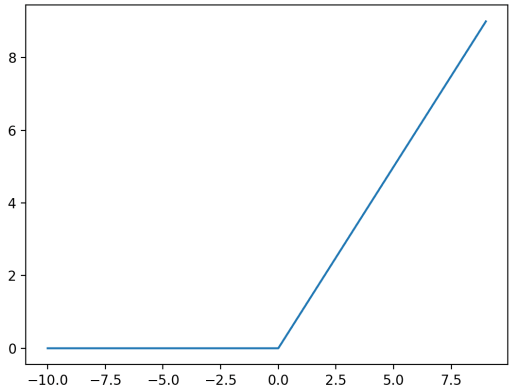

Figure 2.4: Plot of the ReLU activation function

- **Hyperbolic Tangent** (tanh): the output of the neuron will vary between -1 to 1. It is a zero centered function which yields stronger gradients that range between 0 and 1. Just like

the sigmoid, it suffers from vanishing gradient, as once the neuron reaches either -1 or 1 (respectively the minimum and maximum of the tanh function), the derivative equals 0. The hyperbolic tangent activation function is described by equation 2.5 and is plotted in figure 2.5.

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \tag{2.5}$$

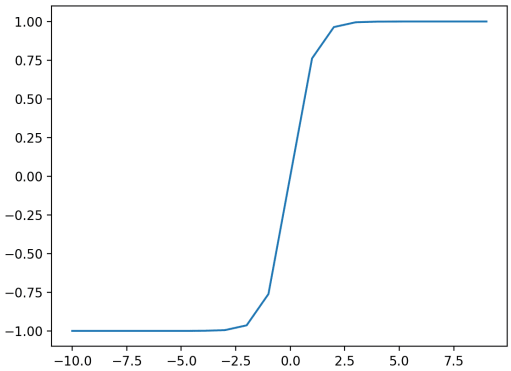

Figure 2.5: Plot of the hyperbolic tangent activation function

Unless we are looking to solve a binary classification problem, which only has two classes to assign the data instances, we will not use a single perceptron. Instead, several perceptrons will be organised in a single layer where all the units are connected to all the network's inputs. This new disposition allows for multi-class classification of problems that are linearly separable, where each unit will be able to classify a distinct class. Each of the units that does not belong to either the input or the output layer is called a hidden unit. Thus, the layers where these units are inserted are called hidden layers. If one intends to solve non-linearly separable problems, it is necessary to stack multiple hidden layers. This works because the hidden layers have the ability to distort the non-linear input in such a way that, by the last layer, it becomes linearly separable. This phenomena is depicted in figure 2.6. The use of deep artificial neural networks, which characteristically have a large number of hidden layer, defines deep learning. Therefore, the depth of a multi-layer network is defined by the amount of hidden layers [56].

Then, just like the human brain, artificial neural networks are comprised of numerous interconnected neurons, which aim to spread the information across the network by receiving sets of stimuli from the surrounding neurons and mapping them to outputs which are fed to the next neuronal layer. The training phase of an artificial neural network requires an intensive search for the weights that can best model the available data, using both back-propagation and gradient descent algorithms. Every time that a forward pass is performed in a network, using a loss function, the network's error can be computed as the difference between the network's output and the ground truth. The back-propagation algorithm computes the gradient of the error to changes in the weights and is comprised of two phases: a forward phase, where the signal is propagated through the network from the input to the output layers; a backward phase, where gradients are propagated to update the weights of the units in the direction of output to input layers [34]. A complete pass

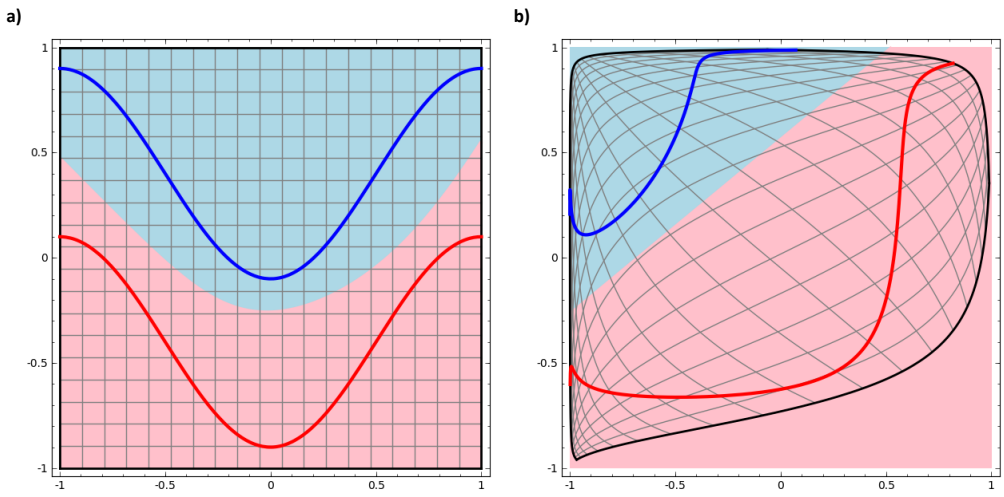

Figure 2.6: Distortion of the input space (**a**)) to generate one that is linearly separable (**b**)). This illustrative example used two input units, two hidden units (sigmoid) and one output unit (sigmoid). Source: [66]

over all the training data is called an epoch, and, normally, a training phase will be comprised of several epochs [19]. The loss function will serve as a measure of the network's performance on a specific task. For each training sample, the loss will be computed and minimized through gradient descent. It will, then, in the backward phase, be propagated to the previous units to calculate the gradient of the various units, serving to update the respective weights [34]. The back-propagation algorithm is shown in figure 2.7.

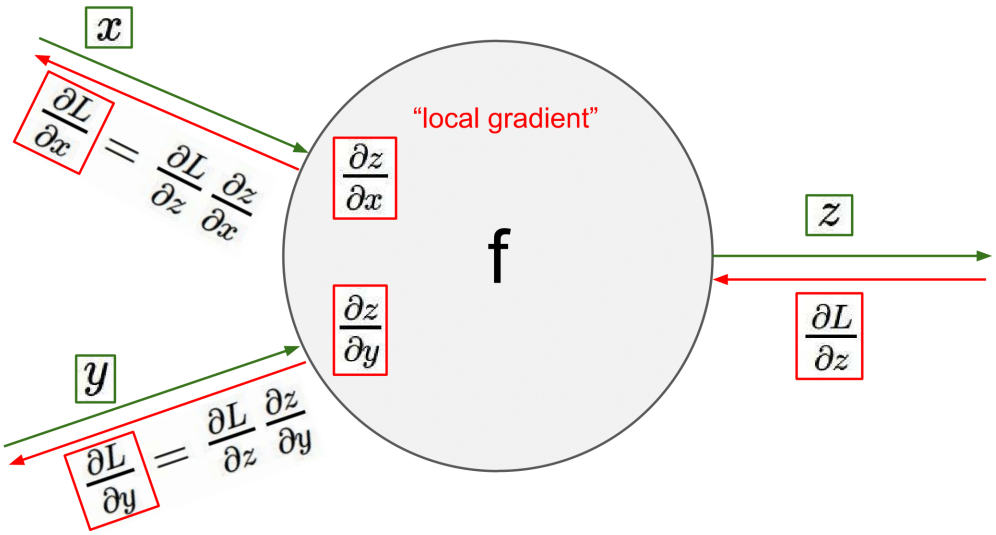

Figure 2.7: Representation of the back-propagation algorithm. Source: [44]

Hence, it becomes obvious the challenge that it can be to choose an appropriate loss function,

as it will be a reflection of what we aim for with a given model and, provided that it is the function that we aim to minimize during training, the performance of our model will be directly related to the quality of such function. Here are some loss functions, where y denotes the true label and $\hat{y}$ the predicted label [26][47]:

- **Zero-one loss**: it is the simplest loss function, which directly compares the training class to the output class. However, it is not a very good function, as it is a step function and we would need to compute the gradient of the loss of a non-differentiable function.
- **Cross-entropy loss**: a widely used loss function, it is differentiable and decreases as the probability of correctly classifying the correct class in the training data increases. It's one common go to option when it comes to classification problems. There are distinct variations of this function for binary (equation 2.6) and multi-class (equation 2.7) classification problems. Though it is so widely used and presents some outstanding results, even this function may under-perform in some contexts, for example when there is label noise in the training data.

$$BinaryCross-Entropy = -\frac{1}{m}\sum_{i=1}^{m}(y_i \cdot log(\hat{y}_i) + (1-y_i) \cdot log(1-\hat{y}_i)) \tag{2.6}$$

$$Multi-ClassCross-Entropy = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot log(\hat{y}_i) \tag{2.7}$$

- **Mean Squared Error** (MSE): this function is the default loss for regression problems. It is calculated as the average of the squared differences between the predicted and actual values. Due to squaring, this functions punishes larger mistakes made by the model. The mathematical formula for the MSE function is presented in equation 2.8.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{2.8}$$

As previously stated in chapter 2.3.2, overfitting is a problem in ML and, just alike, it is also a problem in DL. It is quite normal for a DL model to achieve low error rates on the training set of the data, but being incapable of generalizing this performance to the validation set (and, if it is the case, also to the test set), showing in the latter high error rates. This should, however, be avoided and, to deal with this problem, some regularization techniques have been introduced, such as [69][54]:

- **Dropout**: this type of regularization shuts down some controllable amount of neurons during training, which allows the network to work around those neurons and create a more robust and generalizable model. In figure 2.8 a visual explanation of the dropout mechanism can be found.
- **L1 and L2 Regularization**: consists of adding a regularization term to the loss function, so that the model's weights are penalized, i.e., independently of the loss function's gradient, the weights are a bit smaller in a given update.

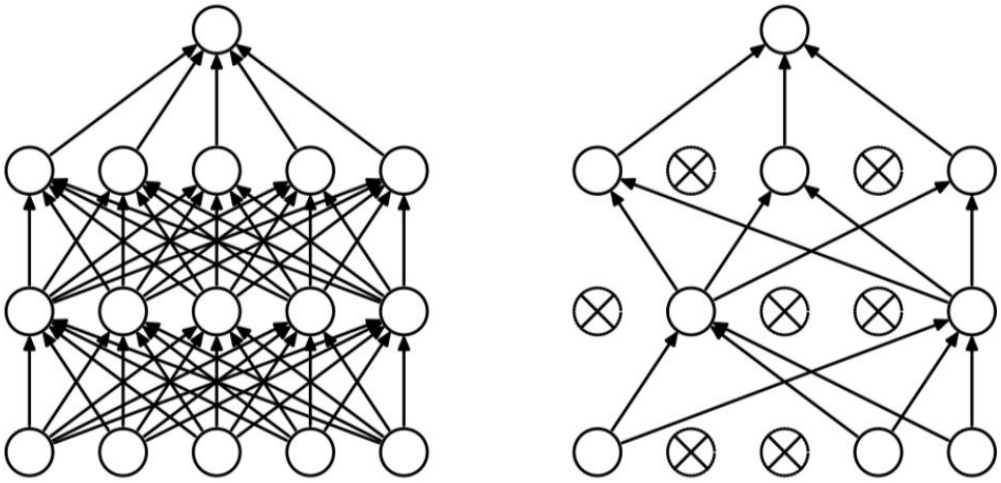

Figure 2.8: (a) Deep NN without dropout and (b) after applying dropout. Source: [14]

- **Early Stopping**: this technique evaluates the performance of the model on the validation set and stops it in the optimal point (minimum or maximum of the validation set). Figure 2.9 depicts this process quite well.

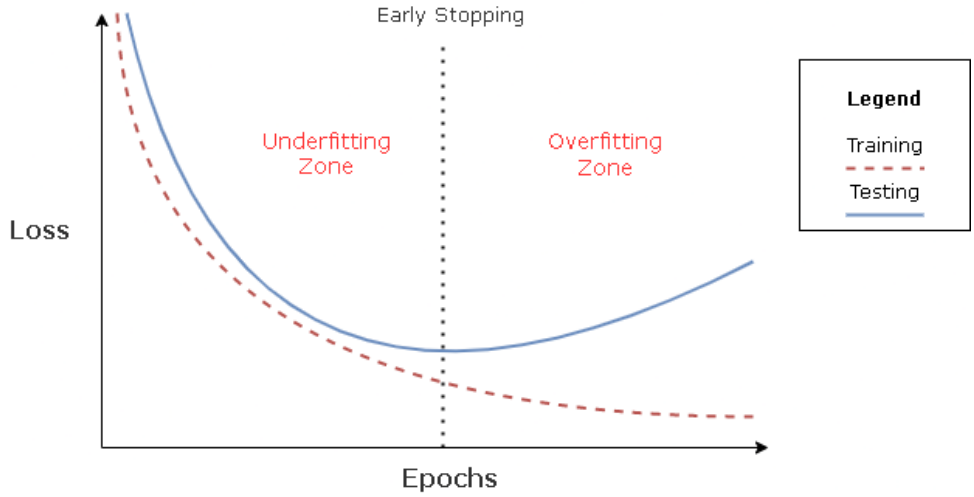

Figure 2.9: Optimal stopping point. Before that point the model would be underfitted and after that point the model would be overfitted. Source: [54]

Artificial neural networks can be divided in two distinct types [50]:

- **Feed-forward neural networks**: in this type of networks the there is an unidirectional information flow in the input to output layer direction, where each layer propagates the signal to the subsequent layer.

- **Recurrent neural networks** (RNNs): on the contrary, RNNs extend feed-forward networks to include feedback connections, allowing units to be connected to previous layer's units.

2.3.2.2 Recurrent Neural Networks

When using machine learning for tasks involving sequential inputs, such as time series, usually the best approach is the use of RNNs. RNN-based models are emerging in diabetes management's literature and have shown superior performance in glucose prediction [96][95][98]. This type of neural networks will perform an element-wise processing of the input sequence, while maintaining in their hidden-units a history of the totality of the past elements of the series saved in a "state vector". The understanding of the training of RNNs becomes simpler if each of the outputs of the hidden units at distinct discrete time steps is thought of as an output of a neuron in a deep multi-layer network (figure 2.10). The neurons receive inputs from other artificial neurons at previous time steps. In that manner, the RNN is able to map the input sequence to the output sequence, with every o_t depending on each previous x'_t (for $t' \leq t$). Thus, one can apply the back-propagation algorithm directly to the flattened sequence on the right of figure 2.10, to compute the total error's derivative with respect to each of the states s_t and each of the parameters. However, this type of neural networks have the issue of typically exploding or vanishing, due to the back-propagated gradients growth or shrinkage, respectively, during training [56].

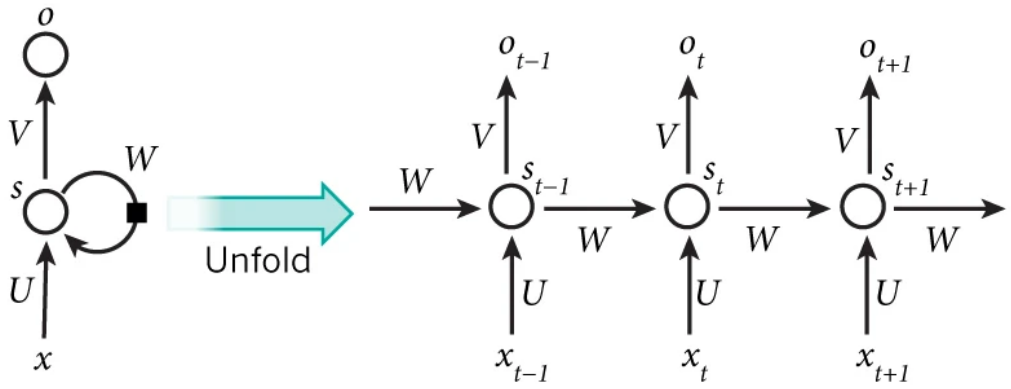

Figure 2.10: RNN and its unfolding through time of the computation involved in its forward computation. The letter x represents the network's input, t represents time at a given point, s represents a hidden unit and o represents the output sequence. *U, V, W* are the parameters matrices, which remain the same at each time step. The black square represents the delay of one time step. Source: [56]

Though their end goal is learning long-term dependencies, it has been shown, both theoretically and empirically, that this type of networks have difficulties learning to store information for very long. One way to fix this problem is to provide an explicit memory to the network, which is exactly what long short-term memory (LSTM) networks do [56]. This type of networks was proposed by Hochreiter & Schmidhuber in 1997, having, since then, been further refined and becoming widely used up to this date [52][67]. Though comprised by the same chained repeating

modules, unlike RNN, each of the LSTM cells are made up of four interacting layers specially intertwined, as represented in figure 2.11. Cell state is represented by the top horizontal part in figure 2.11, as figure 2.12 a) further emphasises. Though it suffers solely some minor linear alterations, it is the key to LSTM networks. These alterations are tweaked using gates, that carefully regulate which information is considered useful. The first gate in the LSTM cell is called the "forget gate", and it decides which of the information is going to be thrown away from the cell state (2.12 b)). This is done by recurring to a sigmoid layer to output a value between 0 (keep nothing) and 1 (keep everything), that is generated based on x_t and h_{t-1}, for every value in the cell state C_{t-1}. This path is summarized by equation 2.9, where W and b are the weights and biases matrices, respectively.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.9}$$

Nextly, a processing of the new information will be made as to decide which of it will be stored in the cell state 2.12 c). This is made recurring to two layers: the "input gate layer", which is a sigmoid function that chooses the values to update and how much to update them (equation 2.10); and a tanh layer that produces a new candidate values vector, $\tilde{C}_t$ (equation 2.11).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.10}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.11}$$

All described steps will now be combined to create a new cell state, by multiplying the old cell state (C_{t-1}) by f_t, and adding $i_t * \tilde{C}_t$. This is depicted in figure 2.12 d). Lastly, a sigmoid layer will decide the parts of the cell state that will be output. The cell state will then be passed through a tanh layer and multiplied by the sigmoid gate's output, as demonstrated in figure 2.12 e) [67].

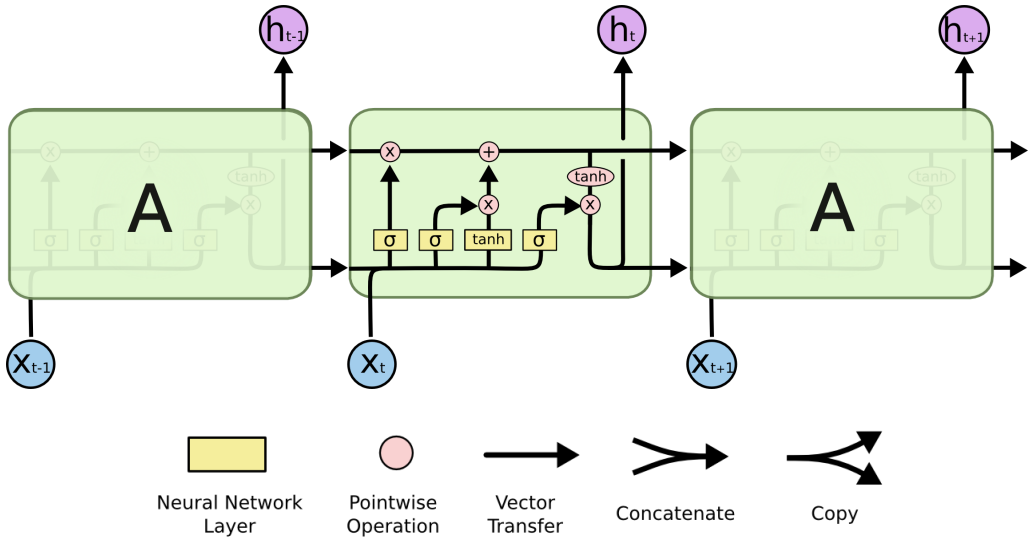

Figure 2.11: Depiction of a LSTM cell and its recurrent connections. Source: [67]

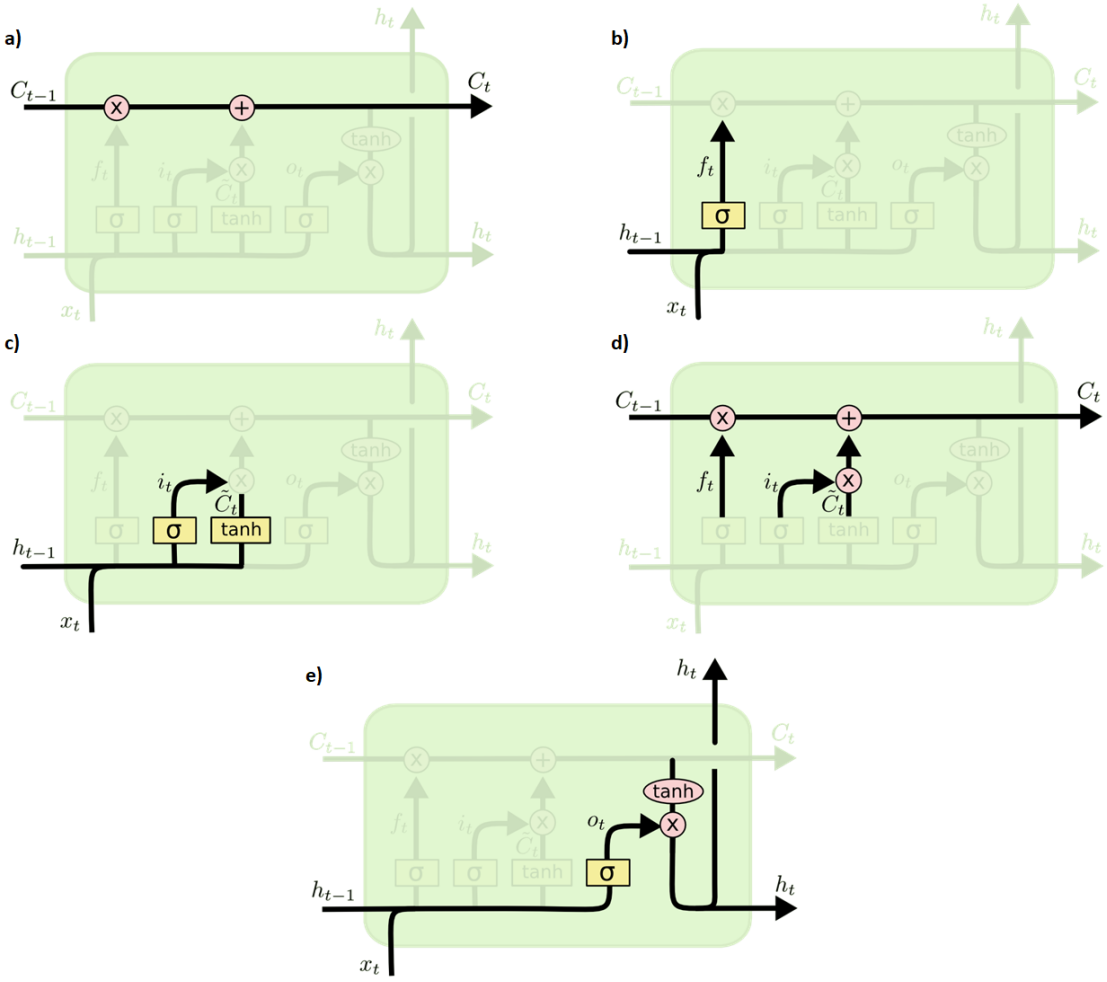

Figure 2.12: Segmentation of LSTM cell and its recurrent connections. Source: [67]

Several variants have been proposed to LSTMs, however a specially relevant one in the ambit of this work is the addition of bi-directionality to the layer, which provides an input flow in both directions, allowing both the future and past information to be preserved, as shown in figure 2.13. The utility of bi-directional LSTMs (BI-LSTM) can be easily explained through a next word prediction example. If we consider the phrase "We went to the [blank space]", there is no way for a network to predict what blank space can be. However, if the network has access to a future sentence "We came from the cafe", it can then more easily predict which word should be in the blank space [90].

2.3.2.3 Transfer Learning

Transfer Learning is a method in ML where a model devoloped and already trained for a given learning task is reused as a starting point in another, usually similar, learning task. This type of learning allows for a less time consuming training phase or improved performance when modelling the second task.

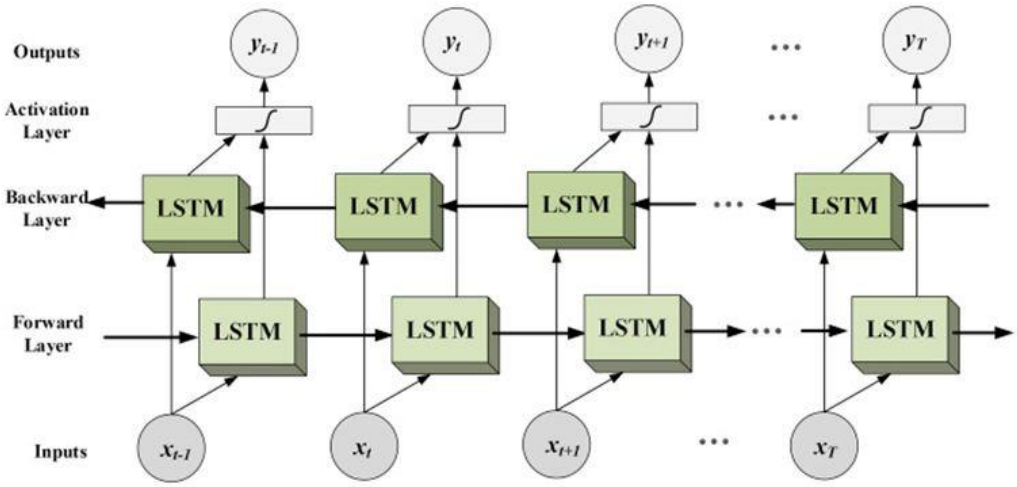

Figure 2.13: Representation of BI-LSTM network. Source: [8]

Given the big amount of resources required to train deep neural networks or the large and challenging datasets usually used, transfer learning has become very popular in this area of artificial intelligence.

Transfer learning requires that the features learned in the first task (the source task) are broader than those aimed for in the second task (the target task) [25].

2.3.2.4 Evaluation Metrics

The following evaluation metrics will be used throughout this work [89][68][74][42][23]:

- **Sensitivity (or Recall)**: This metric evaluates how many of the relevant items were selected.

$$Sensitivity = \frac{TP}{TP+FN} \tag{2.12}$$

- **Specificity**: This metric evaluates how many of the negative items are actually negative.

$$Specificity = \frac{TN}{TN+FP} \tag{2.13}$$

- **Balanced Accuracy**: Corresponds to the arithmetic mean of sensitivity and specificity, being a relevant metric for imbalanced data.

$$BalancedAccuracy = \frac{Sensitivity + Specificity}{2} \tag{2.14}$$

- **Precision**: This metric assesses how many of the retrived items were relevant.

$$Precision = \frac{TP}{TP+FP} \tag{2.15}$$

- **Matthews correlation coefficient (MCC)**: This metric measures the difference between the predicted values and the actual values.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.16}$$

- **Receiver Operating Characteristic (ROC) Curve**: Plots the TPR against the FPR at various threshold settings at which the decision is taken.
- **Area under the ROC curve (AUC)**: It measures the area underneath the ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds.
- **Precision-Recall Curve**: Similarly to the ROC curve, it plots the Precision against the Recall values at various threshold settings at which the decision is taken.

Chapter 3

State Of The Art

During this chapter, we will introduce the existing products in the market to fight the effects of adverse glycaemic events and summarize the main findings in the glycaemia prediction scientific literature.

3.1 Existing Devices

Currently there are a couple of software technologies capable of avoiding hypoglycaemias. In specific, Dexcom has the G6 device (figure 3.1) and Medtronic has the Guardian Connect (figure 3.2). These applications run on a smartphone and connect with the CGM device to collect the data. They then use the collected glucose values to predict the high or low blood glucose levels.

Dexcom proposes a product where glucose is consistently monitored to alert 20 minutes in advance for high and low blood glucose values, independent of the diabetes type [1]. Their patent mentions their technology is dependent on time derivatives of the glucose levels [37]. Medtronic, on the other hand, uses the so called CGM-Based Prevention of Hypoglycaemia System (CPHS),

Figure 3.1: Dexcom G6 Device. Source: [53]

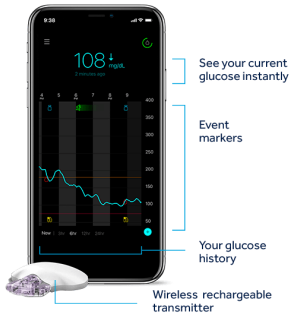

Figure 3.2: Medtronic Guardian Connect Device. Source: [62]

which depends on a mathematical formula they defined for the risk factor based on previous glucose values, to monitor CGM and alerts from 10 to up to 60 minutes in advance both for high and low glucose values [62][3].

3.2 Glycaemia Prediction

The concentration of blood glucose is affected by various factors, such as the amount of administered insulin, the ingestion of carbohydrates, stress, physical activity and parallel pathologies that the patient may have. Consequently, though some models have been defined using solely the CGM signal as input, there is interest in understanding the limitations of such models and which improvements to make in order to increase model performance [70].

Prediction of blood glucose levels is challenging due to the number of physiological factors involved, such as delays associated with absorption of food and insulin, and the lag associated to measurements in the interstitial tissue. Errors in the CGM signal also increase the difficulty of predicting blood glucose values (approximately 9% of the mean absolute relative difference for the best sensors) [33].

Though there has been a modelling of glucose-insulin dynamics since the 1960s, including also models of meal absorption dynamics and of exercise effects, these progressively were replaced by machine learning alternatives, as the field also became increasingly popular.

Large variations in blood glucose, specially after physical activity or a meal, or stress related, have been described to be inherent to each patient. Furthermore, there are several factors that can also affect the blood glucose, which can be related to lifestyle choices, such as fatty food or caffeine intake, alcohol ingestion and travelling, or simply illnesses one can have, medication, growth, weight gain, aging, menstruation and menopause, intense periods of concentration, climate conditions, among others. Specifically type 1 diabetes patients' blood glucose can largely fluctuate during sleep and during heart rate or hormonal variations. All these factors contribute to explain the importance of having an individual patient model, as well as personalized prediction strategies that reflect the behavior of the patient during the day [70].

When looking at the bigger picture, three main types of models have been described to be used for blood glucose modelling [70]:

- Physiological models: they resort to a previous understanding of physiological processes involved in glucose regulation, such as the dynamics of subcutaneous insulin absorption, or carbohydrate digestion and absorption. This type of model usually does not provide satisfying results, as they normally include several parameters and variables that are difficult to adjust.
- Non-physiological (or data-based/driven) models: they rely on CGM data and, sometimes, additional signals to model a patient's physiological response, without the use of physiological variables. These are commonly modelled using either autorregressive or NN models.
- Hybrid models: they use physiological models in the pre-processing phase and non-physiological models in the prediction stage.

Some redundancy when using administered insulin and ingested carbohydrates information has been described, thus recently CGM has been used as a sole input for the model. Additionally, the formalization of these extra inputs in mathematical terms and the extraction of a meaningful signal from them is complicated.

In the analysed literature, *Oviedo et al.* show a prediction horizon (PH) ranging from 15 to 120 minutes, with the most common value being 30 minutes. Some studies did include physical activity and heart rate data, but these additional signals were covered by a low number of studies and there are still lacking some possible extra inputs, such as emotional state and diseases [70].

The same review study reports that the most popular metrics when it comes to classification approaches are accuracy, specificity, sensitivity, precision, ROC and Kappa statistics [70].

By checking the official ranking of the BGLP Challenge, which used the Ohio T1DM dataset, one can tell that all but one of the created models are personalized, meaning that for each of the existing patients, a distinct model was fitted to their training data [7]. Nonetheless, there are disadvantages both in personalized and generalized models. In the case of personalized models, performance can be compromised by intra-subject variability if the dataset in which the model was trained is limited. When it comes to generalized models, if there is a lack of representation of a group of characteristics in the dataset, these models may not be able to generalize well to specific groups of patients [46]. Fiorini et al. describes a variation among devices and individuals of signal properties, such as signal to noise ratio, to be well known, and argues that, for that reason, any prediction model must be personalized [48].

According to Mujahid et al., there are two types of diabetes datasets available, either clinical trials-based datasets, or diabetes simulator-based datasets, such as the UVA/PADOVA simulator. The latter are platforms used to emulate certain physiological characteristics of a diabetic patient, allowing different experiments by controlling distinct parameters related to insulin dosing strategies [65].

Quan et al. used the 30 previous glucose data points to predict the glucose values in a 30 min PH; if the predicted values fell lower than 70mg/dL, the system would consider an hypoglycaemia. The learning phase is performed by alternating through learning and inference using the same dataset and comparing the predicted glucose values with its true values. To do this, they

used a network composed of 1 unit input and output layers and a 30 units LSTM hidden layer with 50% dropout. All glucose data was normalized prior to the input according to the formula: $Normalized\_data = (y - minimum)/(maximum - mininum)$ [75].

The model was trained using a dataset that contained 2000 data points, which corresponded roughly to a week's worth of measurements. They defined the "hypoglycaemia state" as a sole episode from the start point of the hypoglycaemic value to the end point of the hypoglycaemic value, where all the points in between were values on the hypoglycaemic level. The dataset contained 22 of these states. To evaluate model performance, they used precision, sensitivity and a measure they defined as the safety-rate, which consisted on the rate at which the system was able to issue an alert 30 min before the hypoglycemic state. Using this method, they reported values for precision, sensitivity and safety rate of, respectively, 71.8%, 73.9% and 77.3% [75].

However, the authors go on to compare this method with a batch learning one on the same dataset. The latter had a significantly poorer performance, reporting a precision which was divided by zero, and zero sensitivity and safety rate. They justify these results by analysing the autocorrelation plots, where there were a lot more lagged values exceeding the 95% confidence interval within a 30 lag range, than in a 500 lag range. Thus, they concluded that it was more effective to keep learning continuously using data that was closer to the current time than to learn for a longer period of time, even though this represents a higher computational cost [75].

Mhaskar et al. used the DirecNet Central Laboratory dataset, which contained time series for 25 patients under 18 years old. Each time series contains around 160 blood glucose measurements spaced by 5 minutes intervals. In this study, their approach began by dividing the dataset into 30% and 50% distinct training sets and forming clusters based on a 5 minutes prediction of the blood glucose value, using the linear prediction method. Based on this value, they would separate the patient data into 3 distinct clusters, hypoglycaemic, euglycaemic and hyperglycaemic [63].

After the clustering step, they compute three predictors and a judge predictor, with the intent of choosing which one of the predictions is best for each datum. This is judged by checking which predictor assures a best placement in the Prediction Error-Grid Analysis (PRED-EGA).

For the implementation, they employed sampling and prediction horizons of 30 minutes and used a total of 100 trials [63].

They decided to compare the efficacy of this deep network method with that of a shallow feed-forward network with 20 neurons and of a Tikhonov regression using a Gaussian kernel with σ = 100 and regularization constant γ = 0.0001. They also compared with the results obtained when using a flat low-pass Butterworth filter with cutoff frequency 0.8 to smooth the signal. The paper shows better results when the amount of training data increases to 50% and when the data is filtered. The deep neural network outperformed every other predictor, with a percentage of accurate and benign consequence predictions of 96.43% in the hypoglycaemic range, 97.96% in the euglycaemic range, and 85.29% in the hyperglycaemic range [63].

However, it is important to highlight that, according to Boland et al., children usually have prolonged hypoglycemic periods as well as profound postprandial hyperglycaemia, which Mhaskar

et al. highlight as a possible reason for the outperformance of their method when compared to the competitors [21][63].

Rodriguez-Rodriguez et al. retrieved data from 25 type 1 diabetes patients (14 men and 11 women) all under medical treatment and professional supervision, between a age range of 18 to 56 years of age (average 24.51). Blood glucose data was recorded in various sampling frequencies, every 5, 10, or 15 min, and, in total, 5400 h of data were collected [76]. They then proceeded to develop individual patient models for blood glucose predictions. For that, they used sliding windows which contained the blood glucose data from either the previous 3h, 6h, 12h, 26h or 36h. The time window was shifted 1 position up each time, according to the specific sampling frequency being tested at the time. They forecasted values in PHs of 15, 30, 45 and 60 minutes. The following methods were tested during this study: autoregressive integrated moving average (ARIMA), random forests (RF) and support vector machine (SVM). The root mean squared error (RMSE) was used to evaluate model performance [76].

Analysing the results, they came to the conclusion that, as expected, the prediction error increases as the PH increases. Furthermore, the results show that, for all of the developed patient models, the best historical data volume is 6h, with 12h of historical data volume performing worse than the 3h, and higher amounts of data deteriorating even further the achieved accuracy [76]. According to some previous work, there is speculation that these results may be related to the circadian cycle's division of the day in slots of morning, afternoon, and night [87]. They also came to the conclusion that, regardless of the PH, the higher the sampling frequency, the higher the RMSE. Out of all the different used methods, the best performing in all situations was the random forests [76].

Zhu et al. created a system of glucose monitoring via CGM and wristband vital signs that performed real-time glucose prediction and hypoglycaemia and hyperglycaemia warnings. They collected data from 12 T1D patients, where they collected CGM data, vital data and self-reported data, including carbohydrates, protein, fat, insulin boluses, exercise, alcohol, stress, and illness. Insulin and carbohydrates daily entries were converted to insulin on board and carbohydrate on board via physiological models [97].

For the hypoglycaemia prediction task, they have found out that electrodermal activity, interbeat intervals, acceleration and skin temperature were significant predictive factors for hypoglycaemia. Thus they combined these non-invasive measurements with CGM and daily entries and used them in feature selection for the deep learning-based prediction model [97].

The model's architecture consisted on a bidirectional gated recurrent unit (GRU), followed by a normal GRU layer, and an attention mechanism was applied in hope of extracting temporal dependencies regardless of distance. They also used an evidential deep learning approach to train the models and map model uncertainty, which allowed for prediction of rare events (hypoglycaemia and hyperglycaemia), even when the model was not completely confident. They produced a generalized model to be fine tuned specifically to a new subject once he entered the system [97].

As would be expected, the bigger the PH, the worse the model's performance. Considering a PH of 60 minutes, the proposed model presented an accuracy of (88.58 ± 6.53) %, a sensitivity

of (70.30 ± 12.84) %, a specificity of (90.09 ± 8.21) %, a precision of (56.20 ± 10.43) % and a MCC score of 0.56 ± 0.07. They also performed baseline tests with several different methods, among which was a Bi-LSTM that, for the same PH, presented an accuracy of (87.51 ± 6.65) %, a sensitivity of (19.78 ± 12.38) %, a specificity of (97.52 ± 1.92) %, a precision of (47.79 ± 20.69) % and a MCC score of 0.25 ± 0.15 [97].

Cichosz et al. used clinically obtained data from ten male adults who were induced hypoglycaemias using insulin. They decided to use CGM data as well as heart rate variability measures to create a model which predicted a binary output, either normal glucose level or hypoglycaemia. Feature selection was performed and a binary linear logistic regression classifier was used for the proposed task. The data was divided into training and validation using leave-one-out cross-validation [32].

Their algorithm was allowed to predict a hypoglycaemic event up to 10 minutes prior to the blood reference reaching the hypoglycaemia. They also performed an event-based approach to hypoglycaemia prediction, as well as a lead-time detection, where they would measure the time between the detection and the first hypoglycaemic blood glucose value of that specific event [32].

The model using as input data the CGM and the heart rate variability data was compared to one using solely CGM data as input. The results were as follow, respectively: 79% and 33% of sensitivity and 99% and 98% of specificity on the sample based approach; on the event based approach, 16/16 and 12/16 of true positives, zero false positives for both models and a lead time of (22 ± 12) minutes and (0 ± 11) minutes. These significative differences in results allowed them to conclude that the inclusion of heart rate variability data is valuable in hypoglycaemia prediction models. Using CGM and heart rate variability data in the model input, a ROC AUC of 0.98 was obtained [32].

The fact that their data was retrieved under a clinically controlled environment may play a big role on the performance they report, as, for instance, in real world data a simple sudden change in body position may affect the data retrieval process. Furthermore, the fact that the hypoglycaemias were clinically induced by use of insulin, also meant that the percentage of hypoglycaemias within all the retrieved data was much higher than in real world surveillance data. Using data only from males may also influence their results, given that, as they mention, there exist variations in the heart rate variability dependent on gender, age, among other factors. Though they validly argument that a system with high sensitivity is important, otherwise the system could give the patient a false sense of safety, one has already proved by an analysis of the Clarke error grid that, in most cases, a false positive has a much lower clinical risk than a false negative [32].

To sum up, there are already some available technologies which are able to have good performances in the prediction of hypoglycaemias. However, we believe that there may be further potential in the model performances that can be achieved if inputting further data to the model. Some models also use data where hypoglycaemias are medically induced and where the data collection is done in a controlled environment. This type of models may have trouble generalizing to real life data. Additionally, we believe that patients would benefit from a larger PH, so that they can better prepare for the near future. Thus, we will aim to create a model using real life data,

with a bigger PH and ideally an increase in performance, with the intent of filling the gaps that still exist in the literature.

Chapter 4

Experimental Work

This section contains an analysis of the OhioT1DM Dataset, a description of the model architecture used, the training and testing splits considered and a detailed description of the logic behind every step we took when developing this work, as well as a critical analysis of the obtained results and how they compare to the ones described in the literature.

4.1 Dataset Analysis

The Ohio Type 1 Diabetes Mellitus Dataset is comprised of 12 people with type 1 diabetes using an insulin pump with CGM, more specifically either Medtronic 530G or 630G insulin pumps and Medtronic Enlite CGM sensors. This dataset keeps track of 8 weeks' worth of data and patients also provided physiological data via a fitness band (Basis Peak in the case of the 2018 cohort and Empatica Embrace in the case of the 2020 cohort) and reported life-events' data using a custom smartphone app [58].

The dataset contains two XML files for each patient, one containing training data and the other containing test data. This data separation is summed up in table 4.1. Each of the XML files contains the following data fields [58]:

- **<patient>**: holds the patient's ID number and insulin type.
- **<glucose level>**: holds 5 minutes spaced CGM data recordings.
- **<finger stick>**: holds self-monitored blood glucose values.
- **<basal>**: holds the rate of basal insulin infusion. The basal rate begins at a given timestamp ts, and continues until another basal rate is set.
- **<temp basal>**: holds a temporary basal insulin rate that supersedes the patient's normal basal rate. A value set to 0 indicates that the basal insulin flow has been suspended. At the end of a temp basal, the basal rate goes back to the normal basal rate.

- **<bolus>**: holds the amount of insulin delivered to the patient, typically before a meal or when the patient is hyperglycaemic. The most common type of bolus, normal, delivers all insulin at once. Other bolus types can stretch out the insulin dose over the period between ts begin and ts end.
- **<meal>**: holds the self-reported time, type and carbohydrates estimate for a given meal.
- **<sleep>**: holds self-reported sleep and its quality (1 if poor, 2 if fair and 3 if good).
- **<work>**: holds self-reported times of going to and from work. Intensity is evaluated by the patient on a scale of 1 to 10, with 10 the most physically active.
- **<stressors>**: holds self-reported stress.
- **<hypo event>**: holds the time for a self-reported hypoglycaemic episode.
- **<illness>**: holds the time for a self-reported illness.
- **<exercise>**: holds the time and duration, in minutes, of self-reported exercise. Intensity is evaluated by the patient on a scale of 1 to 10, with 10 the most physically active.
- **<basis heart rate>**: holds 5 minutes spaced heart rate data recordings. Only patients wearing the Basis Peak sensor band, have this data available.
- **<basis gsr>**: holds the galvanic skin response.
- **<basis skin temperature>**: holds the skin temperature, in degrees Fahrenheit.
- **<basis air temperature>**: holds the air temperature, in degrees Fahrenheit. Only patients wearing the Basis Peak sensor band have this data available.
- **<basis steps>**: holds a step count, aggregated every 5 minutes. Only patients wearing the Basis Peak sensor band have this data available.
- **<basis sleep>**: holds the times when the sensor band reported the patient to be asleep. Patients wearing Basis Peak, also have available a numeric estimate of sleep quality. However, not all data contributors wore their sensor bands overnight.
- **<acceleration>**: holds the magnitute of acceleration. Only patients wearing the Empatica Embrace sensor band have this data available.

Information about patients' gender, age range, pump model, sensor band and which cohort they belong to is summarized in table 4.2.

There has been a time shift of the months, when de-identifying the dataset. Thus, effects of holidays or seasonality cannot be considered [58].

As, most of the time, a patient's glucose range is that of an euglycaemia, it is obvious that the dataset will be quite imbalanced in its distribution of hypoglycaemias, euglycaemias and hyperglycaemias. This phenomena is illustrated by means of the glucose distribution histogram of

Table 4.1: Number of training and testing CGM data points for each patient

ID	Training	Testing
540	11947	2896
544	10623	2716
552	9080	2364
567	10858	2389
584	12150	2665
596	10877	2743
559	10796	2514
563	12124	2570
570	10982	2745
575	11866	2590
588	12640	2791
591	10847	2760

Table 4.2: Patient gender, age range, pump model, sensor band and cohort

ID	Gender	Age	Pump Model	Sensor Band	Cohort
540	male	20-40	630G	Empatica	2020
544	male	40-60	530G	Empatica	2020
552	male	20-40	630G	Empatica	2020
567	female	20-40	630G	Empatica	2020
584	male	40-60	530G	Empatica	2020
596	male	60-80	530G	Empatica	2020
559	female	40-60	530G	Basis	2018
563	male	40-60	530G	Basis	2018
570	male	40-60	530G	Basis	2018
575	female	40-60	530G	Basis	2018
588	female	40-60	530G	Basis	2018
591	female	40-60	530G	Basis	2018

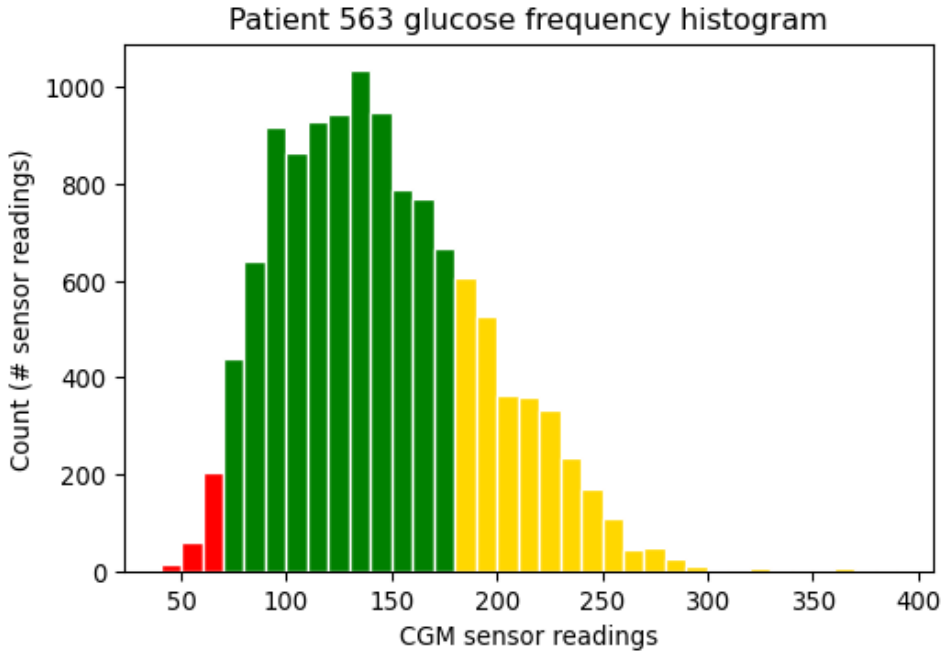

Figure 4.1: Frequency histogram showing counts of CGM sensor readings for patient 563's training data. The red color denotes hypoglycaemic values, the green color denotes euglycaemic values and the yellow color denotes hyperglycaemic values.

patient 563, represented in figure 4.1. Out of all blood glucose values, this patient presents 2.57% of its data within the hypoglycaemic range, 73.89% of its data within the euglycaemic range and 23.54% of its data within the hyperglycaemic range. Consequently, there is a bias towards a prediction of euglycaemia, which needs to be dealt with. This bias is specially enhanced in the case of a hypoglycaemia classification problem.

To begin the analysis of the CGM time series, we used data from patient 563 and plotted it through time, obtaining figure 4.2. Figure 4.3 shows the points filtered using the Savitzky-Golay filter (savgol_filter from the SciPy API [91]) with a time-window of 71 points and polynomial order of 5. This type of filter was described as performing well on simulated CGM data [80]. This step was followed by an autocorrelation plot of the RAW CGM signal, from which we inferred that there was probably some sort of daily periodicity in the CGM signal, however, autocorrelation values were quite low and most of them were bellow the 95% confidence interval. We then analysed up to what point the previous values may relate to the current value by a reduction on the lag-window size, as portrayed in figure 4.5. Though the number of lags above the 95% confidence interval is around 50, we figured that there may be some interest in the information contained in the previous 72 data points (6h of data), as it coincides with the turning point of the autocorrelation function from positive to negative, and is also described by Rodriguez-Rodriguez et al. as the best amount of data to consider, probably due to motives related with the circadian cycle [76][87].

Nonetheless, we used this daily periodicity to extract the trend, seasonal and residual components of the CGM signal (figure 4.6). The model was considered additive and the components

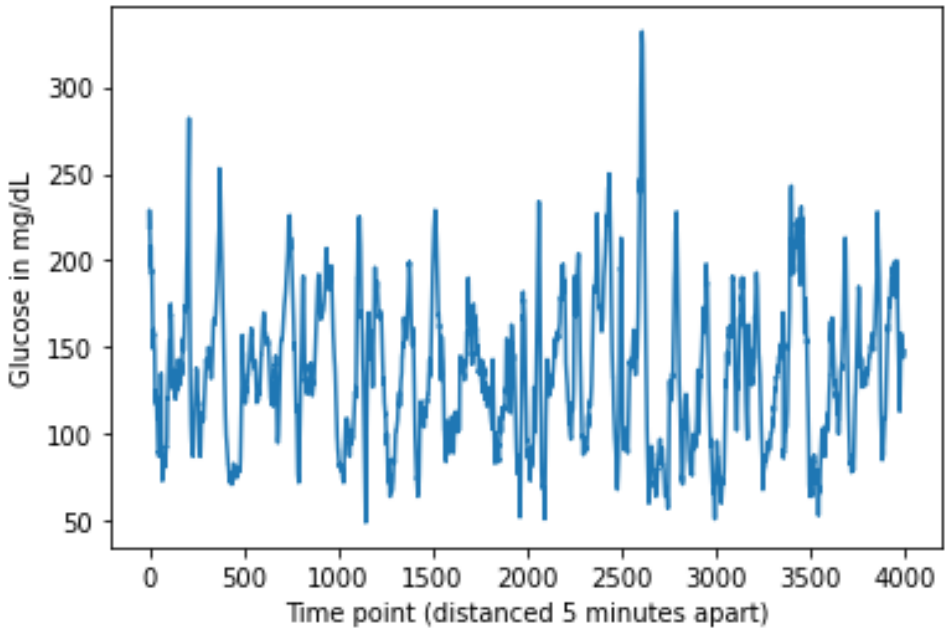

Figure 4.2: Plot of 4000 temporally aligned points from patient 563's CGM signal

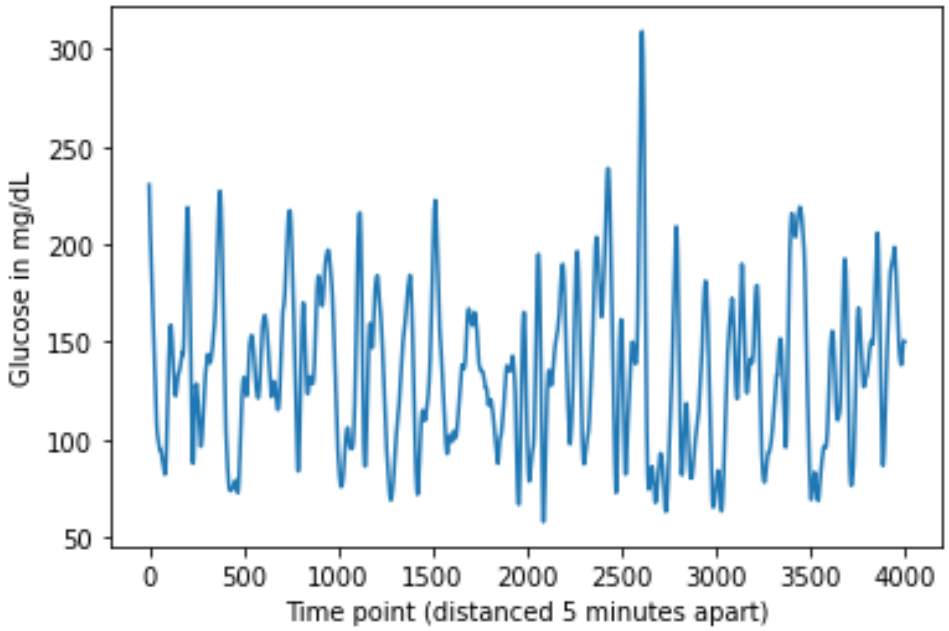

Figure 4.3: Plot of 4000 temporally aligned points from patient 563's filtered CGM signal

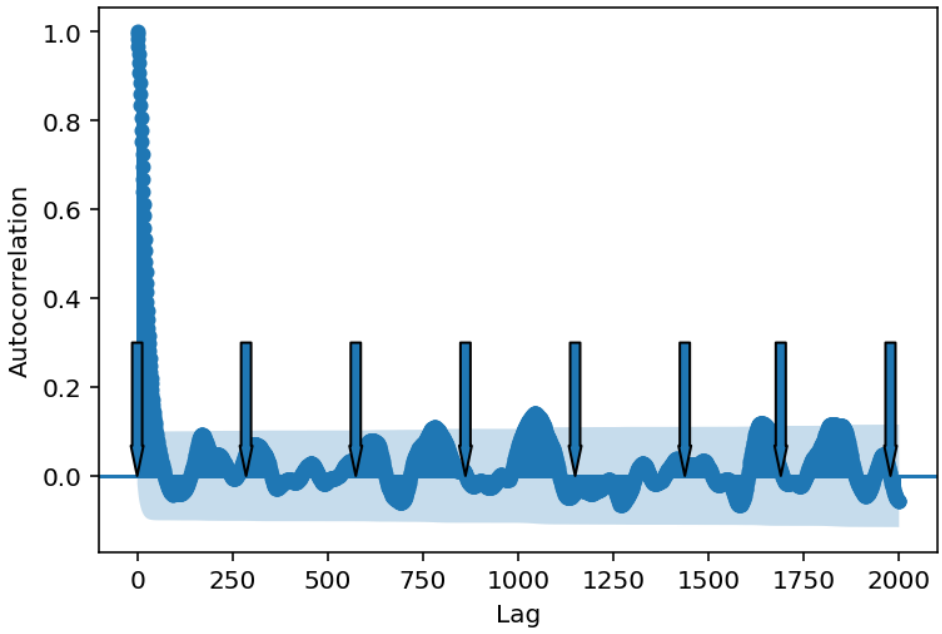

Figure 4.4: Plot of the autocorrelation of patient 563's CGM signal for 2000 lags. The arrows represent the beginning of a new day and the blue shade corresponds to the 95% confidence interval

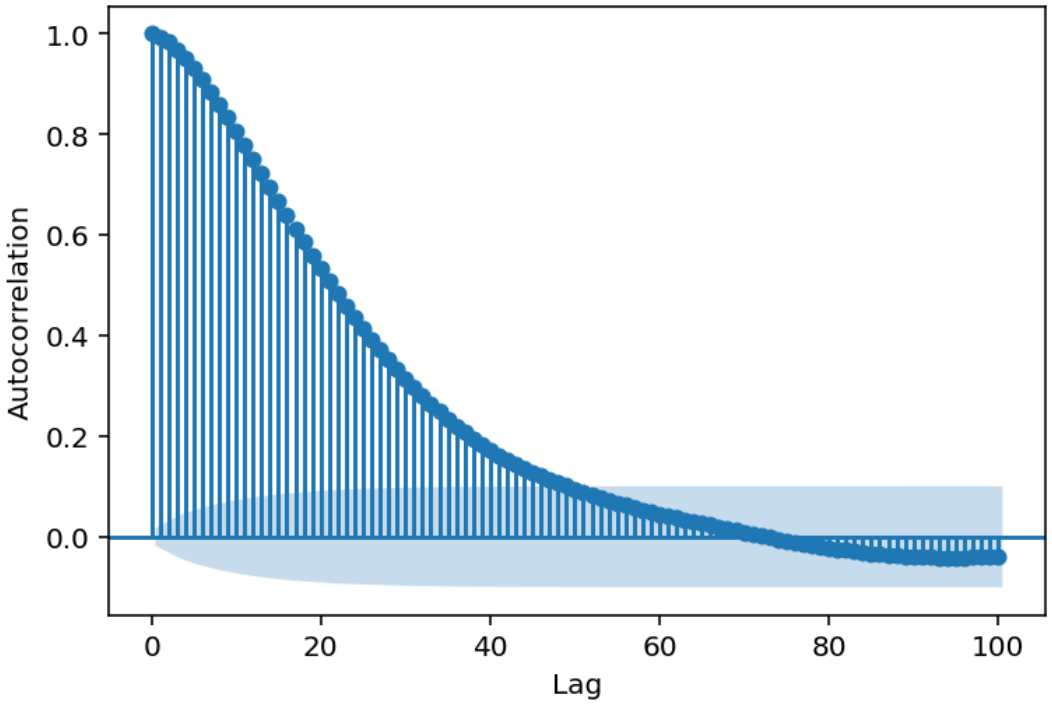

Figure 4.5: Plot of the autocorrelation of patient 563's CGM signal for 100 lags. The blue shade corresponds to the 95% confidence interval

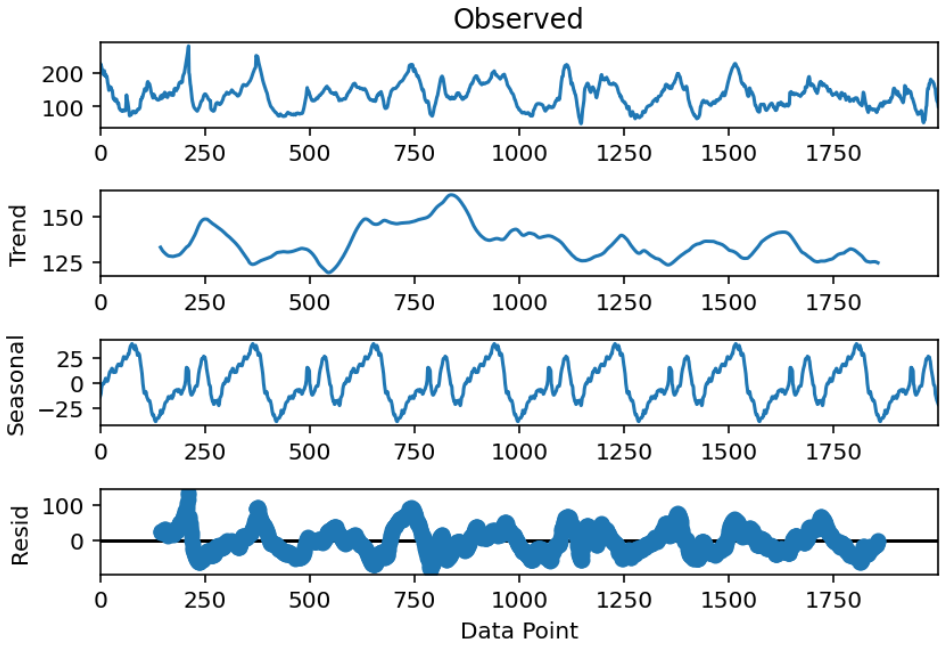

Figure 4.6: Patient 563's observed CGM signal and its decomposition in trend, seasonal and residual parts

extracted using *seasonal_decompose* function from the statsmodels API [84][83]. As one can tell both from the observed signal and its trend component, there is no clear directed signal trend, which confirms our expectations, given that the CGM signal will have a defined range of values in which it will operate. Thus, there is no need to differentiate the signal [30].

4.2 Experimental Work Report

4.2.1 Model Architecture

The model's implementation was done using TensorFlow's API [13]. Due to its high predictive power in time series data, a specific architecture of RNNs, the LSTM Networks, as well as a bidirectional version of it, were used throughout this work and combined with dense layers with a variable number of neurons, activation functions and depths. The use of a LSTM layer or a bidirectional LSTM was mutually exclusive.

Given that the majority of the proposed problems consisted on binary classification (hypoglycaemia and other glycaemic values), the output layer is comprised of a single neuron and a sigmoid activation function. However, in the sole case when a multiclass (hypoglycaemia, euglycaemia and hyperglycaemia) version of this problem was experimented, where we used 3 neurons and a softmax activation function. The LSTM layer, as well as its bidirectional version in the cases where it was used, has the same number of neurons as the number of time points of the input signal and distinct activation functions depending on the case. TensorFlow's Adam optimizer was

used, with a learning rate and decay of 10^{-5}, and the chosen loss function to minimise was TensorFlow's binary cross entropy, except in the multiclass classification problem, where TensorFlow's categorical cross entropy was used.

Given the rarity of hypoglycemic events, hypoglycaemias and hyperglycaemias were weighted based on the value proposed for each class by Scikit-Learn's compute_class_weight function [72]. However, analysis of model performances with the computed weights led us to scale the hypoglycaemic class by a factor of 0.4.

4.2.2 Training and Testing

Based on other experiences performed using the Ohio Dataset, as well as the information gathered during the literature review, it was decided that rather than testing a universal model trained on all available training data, the first step should be testing personalized models for each patient, given that these were described as more adequate. The training and validation phases of the personalized models were performed using patient 563's data.

Each patient's data was split into training and testing on 80:20 proportions. As mentioned, in the personalized models, patient 563 was used as an "experimental training set", using both the training and testing data of that patient to experiment with different architectures and validate each model's performance. The training phase included a 5-fold cross-validation and early stopping, i.e., the training part of the dataset was split into five folds of 80% training and 20% validation and, in each of the folds the trained models would be only saved if they were improving the loss function's value. The selected model for testing was the one which presented the best AUC results in one of the folds. In order to decrease the computational resources allocated to this research problem, we started by comparing models using solely 5 epochs, only after this pre-selection, did we increase the number of epochs, once we felt that the low amount of epochs could affect the results.

To actually test the pipeline's performance, the best performing model architectures will be trained and tested in three new patients, simulating the pipeline's desired application.

Though personalized models were considered the best option in the literature, at a given point we felt like generalized models could be beneficial, as each patient contained few data. In this approach, the training data from 9 out of the 12 patients was fed to the model as training data. We then proceeded to use the test data from these patients as validation data. Afterwards, the goal was using the same test patients as in the personalized model's approach, and use their training data to perform transfer learning on the baseline generalized model. Finally, we compared the difference between the personalized models, the generalized model and the transfer learning model on the test patient's testing data.

4.2.3 Experiments and Results

During this section, the experiments follow an exploratory model where the findings that are made in a given model will be used in the following one, unless specifically mentioned otherwise.

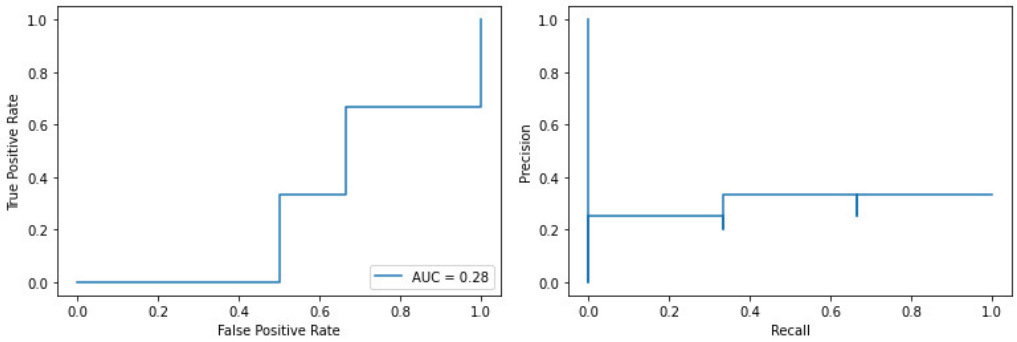

Figure 4.7: ROC curve and AUC value (left) and Precision-Recall curve (right) for LSTM layer using ReLU as activation function on hourly summarised carbohydrates intake, basal and bolus insulin doses, heart rate and glucose values trained for 5 epochs, using test data

All model inputs were standardized between 0.005 and 1, according to formula **??**. The value zero was assigned to missing values. In order to keep this problem viable for data outside the training scope, we decided to define biological boundaries for each variable. Glucose minimum and maximum values were set between 0 and 500, heart rate between 0 and 300, steps between 0 and 1000, insulin between 0 and 100 and carbohydrates between 0 and 300.

4.2.3.1 Personalized Models

Preliminary experiments were made using a hourly frequency of the data for a whole day, having 24 values for each feature, and predicting the existence of a hypoglycaemia the following day. The fundamental hypothesis for this approach lies in the supra mentioned periodicity of the auto-correlation function. In this approach, the carbohydrates intake, the basal and bolus insulin doses (where the metrics used to summarise the data were minimum, maximum and sum), as well as the heart rate and the glucose values (where the used metrics were minimum, maximum, mean and standard deviation) were used. The model was trained for 5 epochs and used ReLU as the LSTM layer's activation function; however, the performance of this model would match that of a random classifier (figure 4.7).

Analysing figure 4.7, one assumption of the possible reasons for the poor performance of this model could simply be the low amount of data obtained when performing hourly summarizations.

Given the poor results obtained with this rather complex problem, we decided to make the problem quite simpler by using temporally organised raw continuous glucose monitoring as the only input. For this new problem, 72 data points of glucose data, corresponding to 6 hours of CGM, were used to train a model to predict whether there would be a hypoglycaemia episode in the next 12 data points, corresponding to the next hour.

The models that will now be described follow the same core architecture: an initial LSTM or Bi-LSTM layer followed by a 60 neuronal units dense layer, followed by an output layer. A diagram of this architecture is shown in figure 4.8.

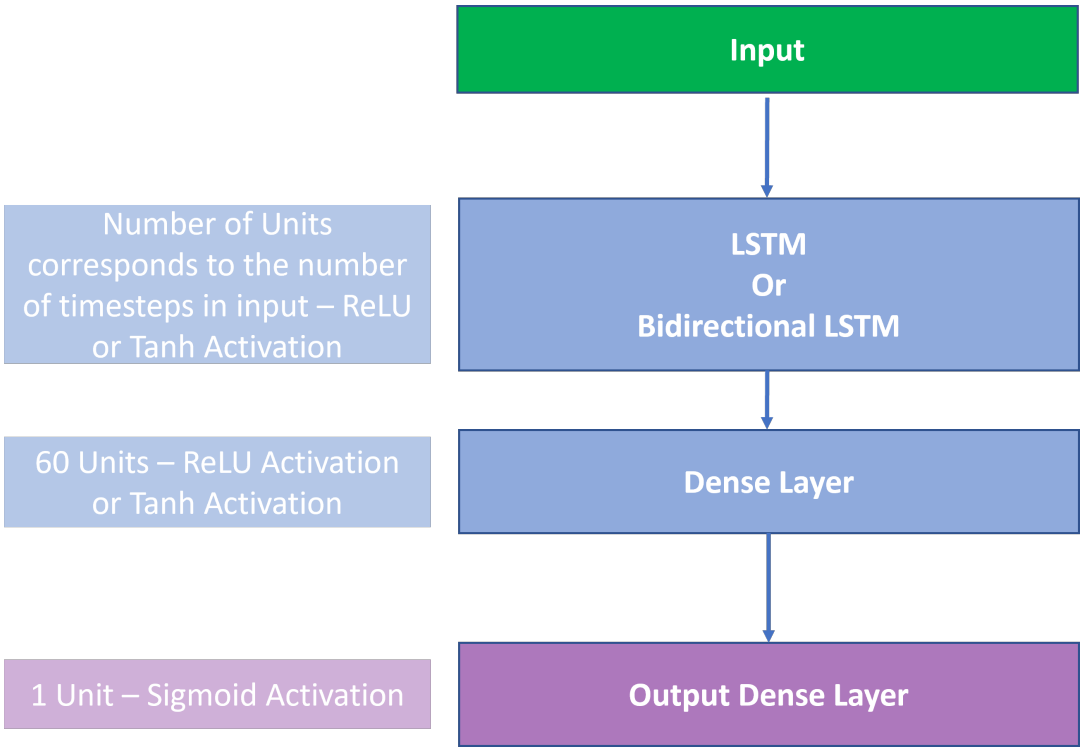

Figure 4.8: Base-models' architecture diagram

We began by comparing the results of two distinct approaches. We compared models with the same normal LSTM layer and the raw CGM as the model's input, but we varied this layer's activation function between a ReLU and a hyperbolic tangent (tanh). The dense layer's activation function was a ReLU. Both models were trained for 5 epochs. We evaluated the models' performance on test data based on ROC and Precision-Recall curves (figures 4.9 and 4.11), as well as the plots of the models' probability of hypoglycaemia in function of the actual minimum and mean values of glucose in the next hour, depicted in figures 4.10 and 4.12.

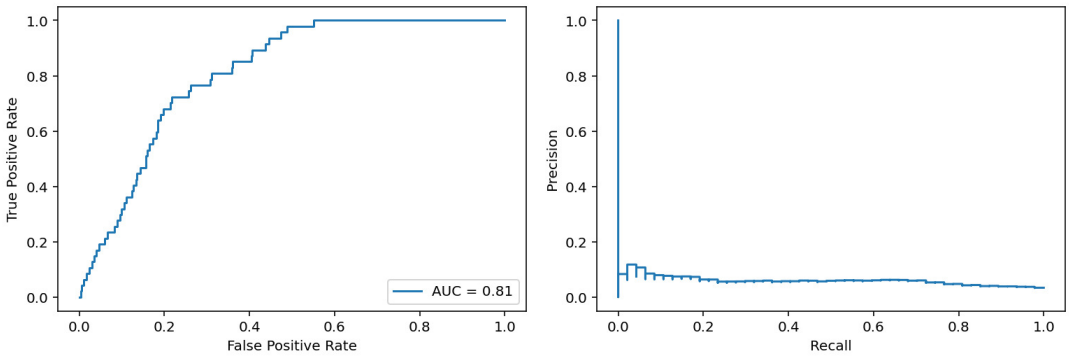

Figure 4.9: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Raw CGM with LSTM layer using ReLU as activation function trained for 5 epochs, using test data

To choose the prediction probability threshold, a combination of two methods was used [28].

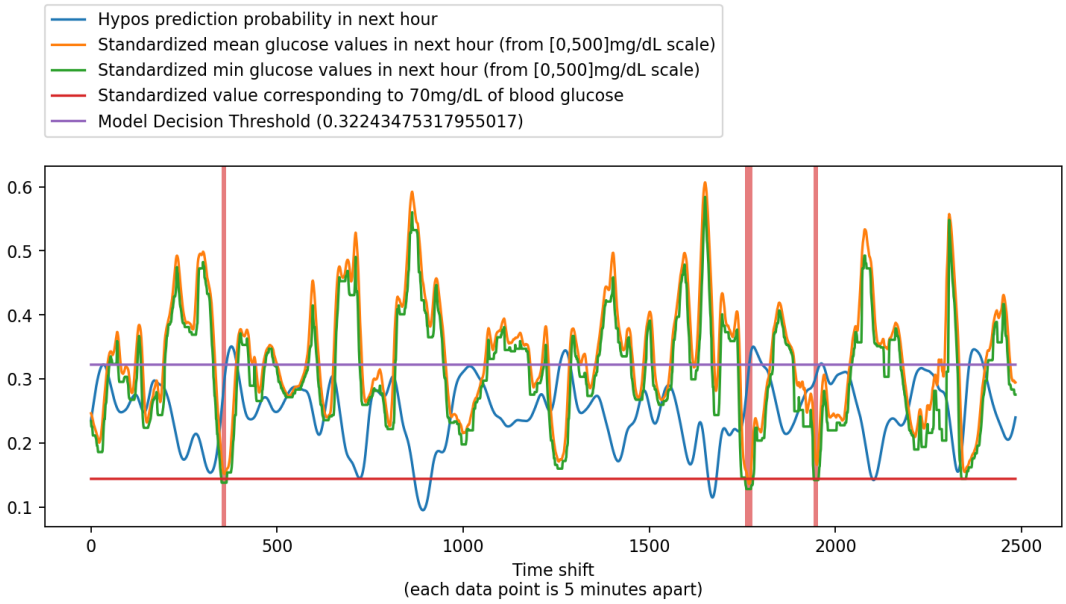

Figure 4.10: Plot of the probability of hypoglycaemia predicted by the model on Raw CGM with LSTM layer using ReLU as activation function trained for 5 epochs. The moment when the blue line crosses above the purple line corresponds to a prediction of hypoglycaemia in the next hour; and the moment when the green line crosses under the red line corresponds to the ground truth of a hypoglycaemia in the next hour, also marked by the regions of the plot highlighted in red

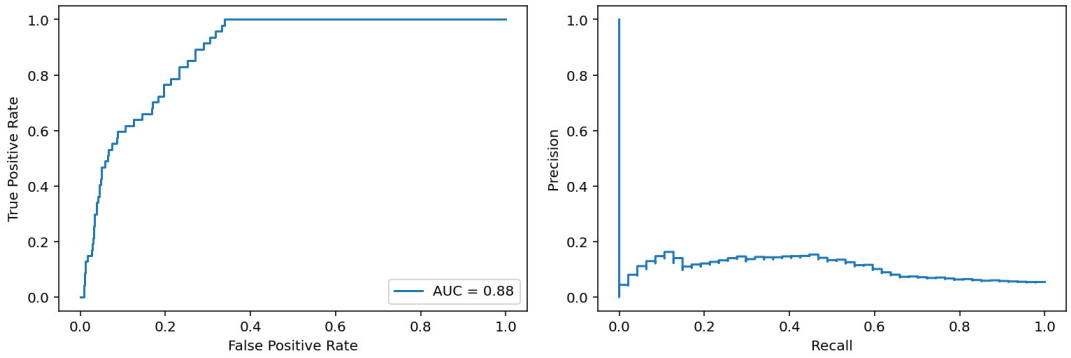

Figure 4.11: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Raw CGM with LSTM layer using tanh as activation function trained for 5 epochs, using test data

From the ROC curve, the G-Mean metric values were computed as follows:

$$G-Mean = \sqrt{Sensitivity * Specificity} = \sqrt{TruePositiveRate * (1 - FalsePositiveRate)} \quad (4.1)$$

From the Precision-Recall curve, the F1-score metric was computed as follows:

$$F1-score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.2)$$

From there, for each of the various Precision-Recall curve thresholds and its closest threshold

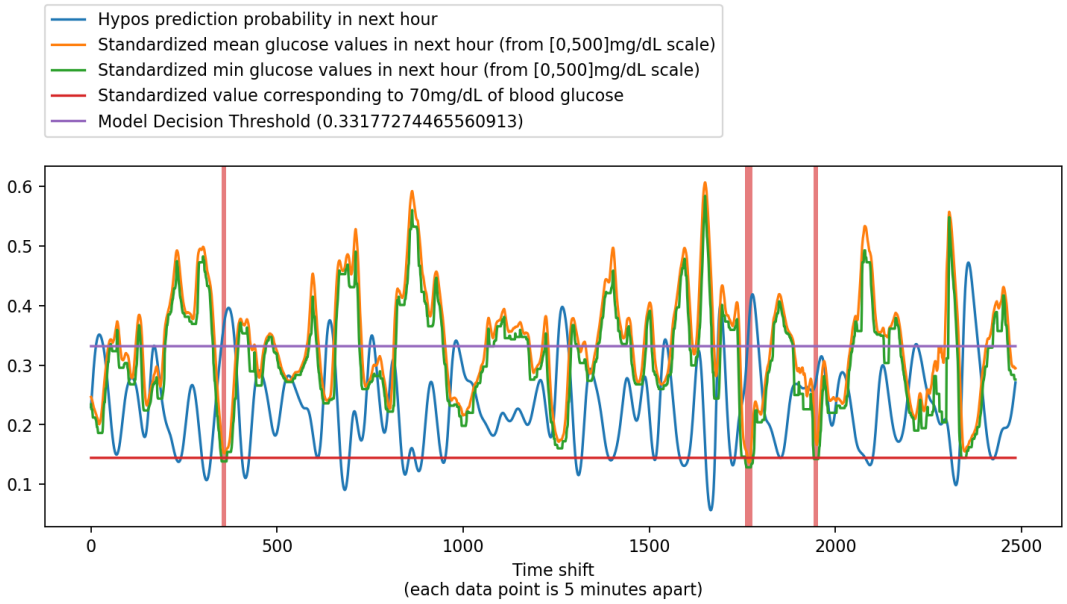

Figure 4.12: Plot of the probability of hypoglycaemia predicted by the model on Raw CGM with LSTM layer using tanh as activation function trained for 5 epochs

on the ROC curve, the sum of the F1-Score and G-Mean was computed and the chosen threshold would be the mean of the thresholds which maximised the mentioned sum.

The next step involved comparing the performance of the LSTM layer against a Bidirectional LSTM layer. As mentioned, adding bidirectionality to the LSTM layer provides an input flow in both directions, allowing both the future and past information to be preserved. This new architecture was run for 5 epochs, varying the activation function once again between the ReLU (figures 4.13 and 4.14) and the tanh (figures 4.15 and 4.16). The addition of bidirectionality to the LSTM layer provided better results, as there were enhancements both in the AUC value, ROC and Precision-Recall curves and a reduction in the number of false positives. Given the quicker computation of the tanh activation function when compared to the ReLU function (when using the ReLU activation function, Tensorflow would not use *cuDNN*, becoming much slower) in the LSTM layer, as well as the slightly better results of the tanh activation function, we decided to use it as the Bidirectional LSTM layer's activation function.

Furthermore, if we analyse the metrics reported in table 4.3, it becomes obvious that there is much more potential in the Bi-LSTM and tanh activation function combo, as it is the one having the lowest balanced accuracy and specificity. In the specific case of this problem, we value the most a correct identification of the hypoglycaemias than one of the the non-hypoglycaemias, as explained previously via the Clarke Error Gird.

CGM devices are often affected by a random noise component, which affects the signal at high frequencies [80]. In order to evaluate the impact of the noise in the model's prediction, we applied the Savitzky-Golay filter (savgol_filter from the SciPy API [91]), using a time-window of 71 points and polynomial order of 5, to the input, evaluating the results using the best working

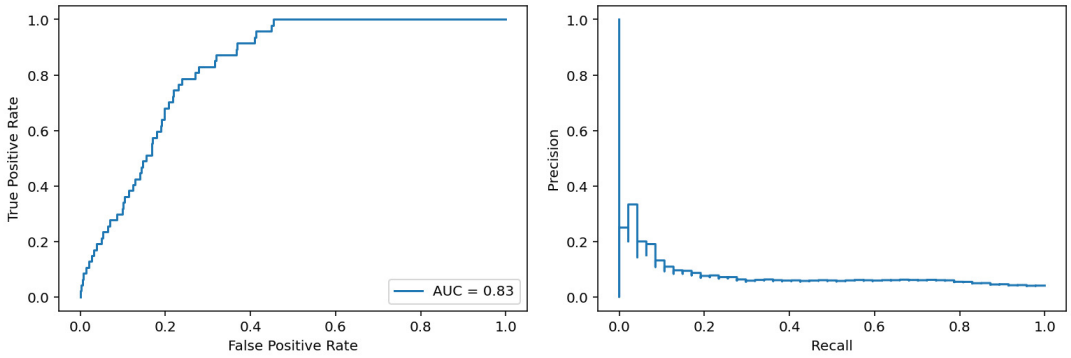

Figure 4.13: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Raw CGM with Bidirectional LSTM layer using ReLU as activation function trained for 5 epochs, using test data

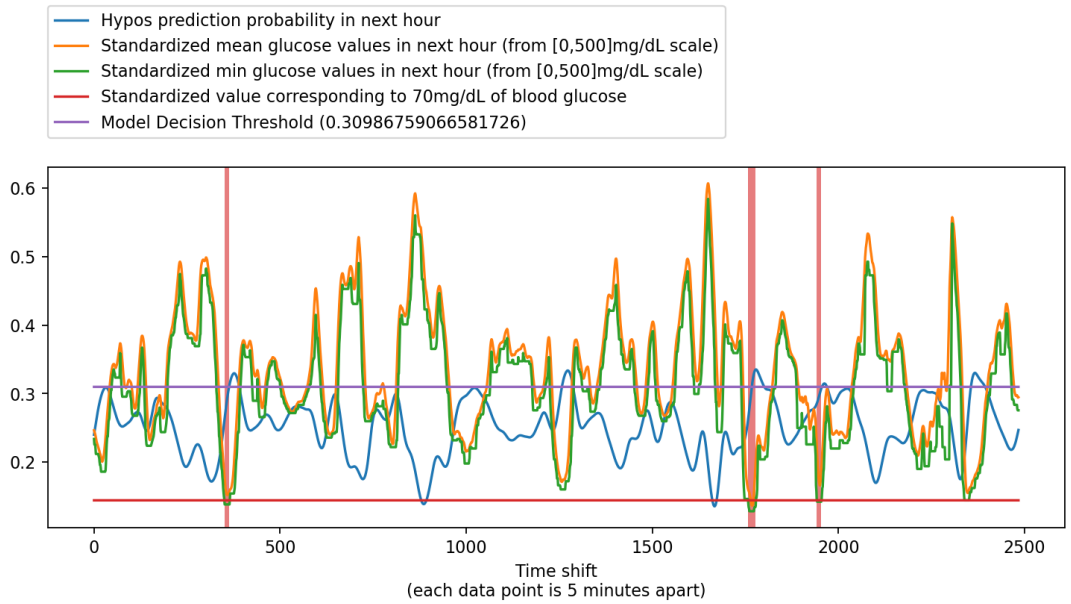

Figure 4.14: Plot of the probability of hypoglycaemia predicted by the model on Raw CGM with Bidirectional LSTM layer using ReLU as activation function trained for 5 epochs

model so far, the Bidirectional LSTM layer with a tanh activation function. This model was also trained for 5 epochs and its results are presented in figures 4.17 and 4.18. However, the results of the filtered CGM signal did not surpass those of the RAW CGM, with accuracy, sensitivity, specificity, precision and MCC values of 0.878, 0.638, 0.883, 0.095 and 0.213, respectively, and poorer ROC and Precision-Recall curves.

Nonetheless, given the better results described in literature using filtered signals, we decided to try this approach on a less underfitted model. Thus, we run the same Bidirectional LSTM layer with a tanh activation function for both the RAW and the filtered CGM inputs, but this time for 60 epochs. The results for the model using the RAW input are presented in figures 4.19 and 4.20, while those for the model using the filtered input are shown in figures 4.21 and 4.22. We

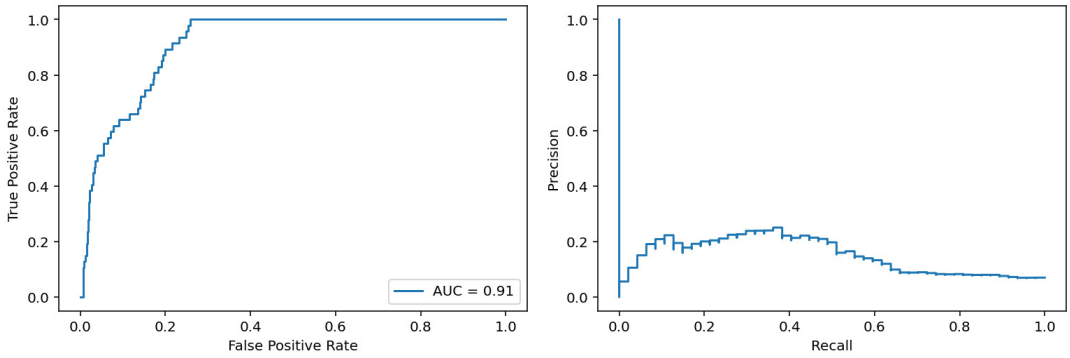

Figure 4.15: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Raw CGM with Bidirectional LSTM layer using tanh as activation function trained for 5 epochs, using test data

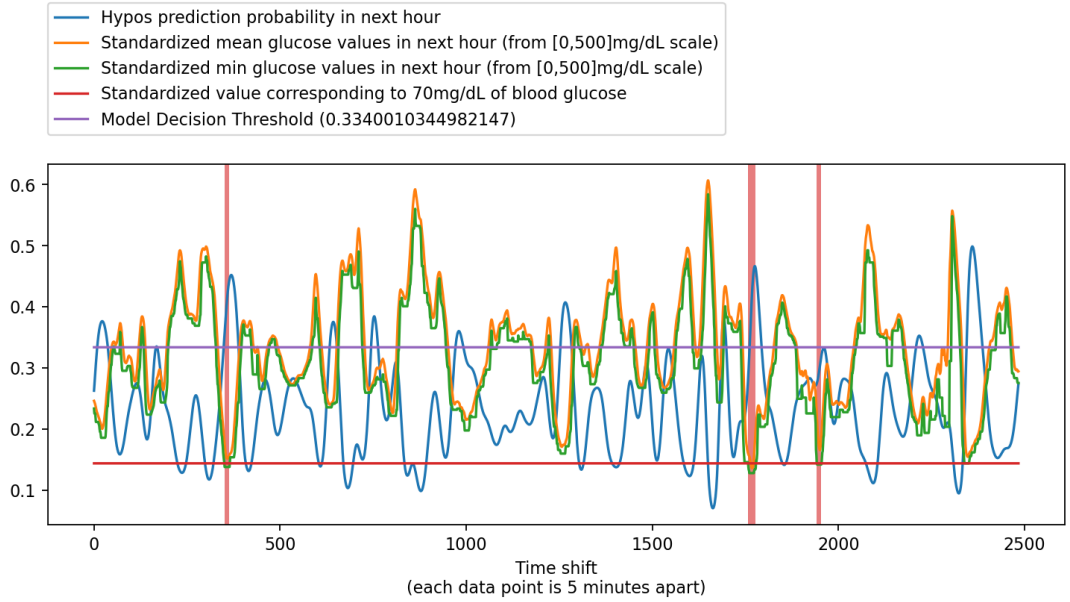

Figure 4.16: Plot of the probability of hypoglycaemia predicted by the model on Raw CGM with Bidirectional LSTM layer using tanh as activation function trained for 5 epochs

considered the model using the filtered input as the better one, as it was closer to classifying the 3 existing hypoglycaemic events than the other model. Furthermore, analysing table 4.4, filtering the input seemed, in this case, to have increased a bit the sensitivity, which interests us the most according to the clinical severity explanation already provided in section 2.1, while maintaining the other metrics' values.

Analysing table 4.4, one might wonder why the choice of 60 epochs as the best number of epochs, as this will be maintained from this point forward. The reason lies both in the pretended application for the pipeline we are trying to create, where it would be advantageous to have a lower number of epochs to decrease computational time; but also in the fact that the choice of the decision threshold lies in the use of both the ROC and Precision-Recall curves, thus the odd

Table 4.3: Evaluation metrics comparing LSTM vs Bi-LSTM and tanh vs ReLU models trained for 5 epochs with Raw CGM as input. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
LSTM_tanh	0.734	0.553	0.914	0.111	0.218
LSTM_ReLU	0.568	0.191	0.945	0.063	0.080
Bi-_LSTM_ReLU	0.590	0.234	0.946	0.077	0.105
Bi-_LSTM_tanh	0.770	0.638	0.902	0.111	0.236

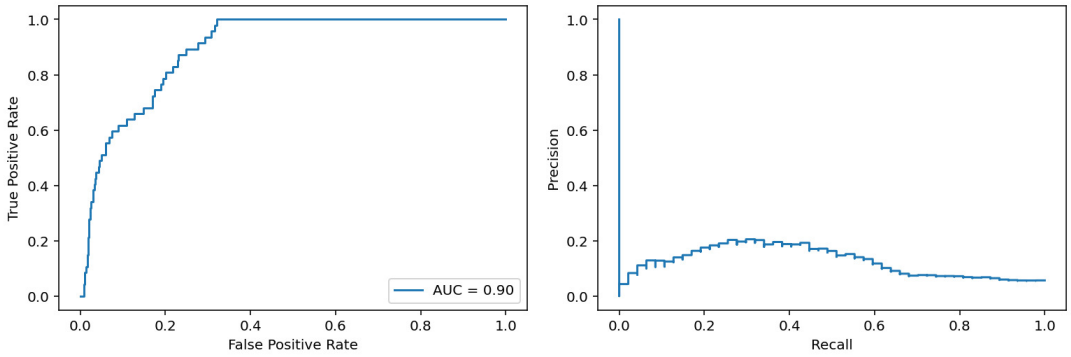

Figure 4.17: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function trained for 5 epochs, using test data

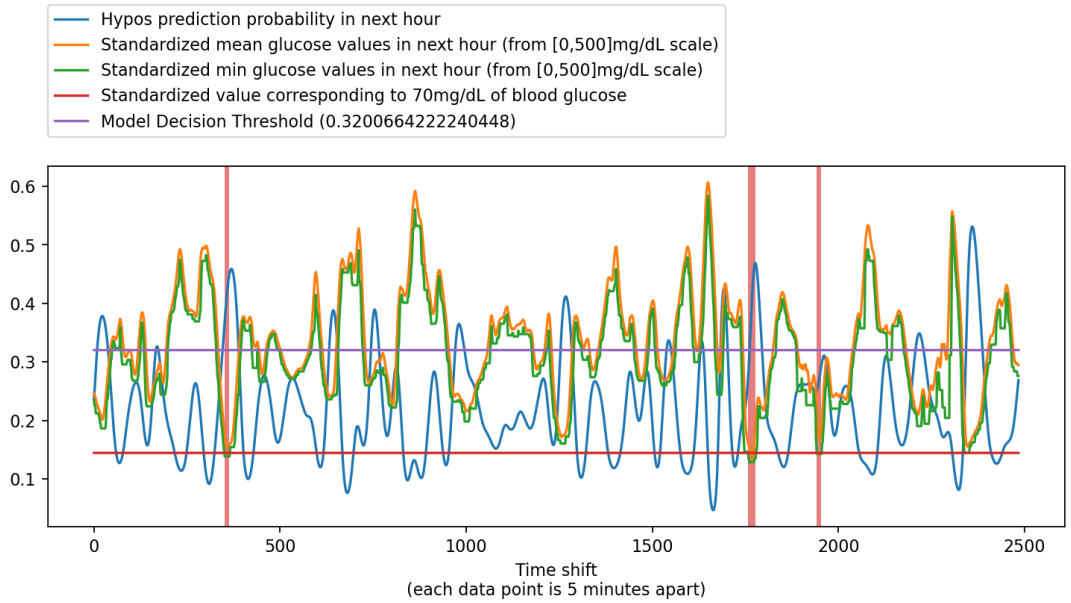

Figure 4.18: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function trained for 5 epochs

shapes of the ROC and Precision-Recall curves produced by the 75 and 100 epochs trained models (figures 4.23 and 4.24) led us to discard these amounts of training.

Given that during night time hypoglycaemic episodes are specially prone to occur, and the

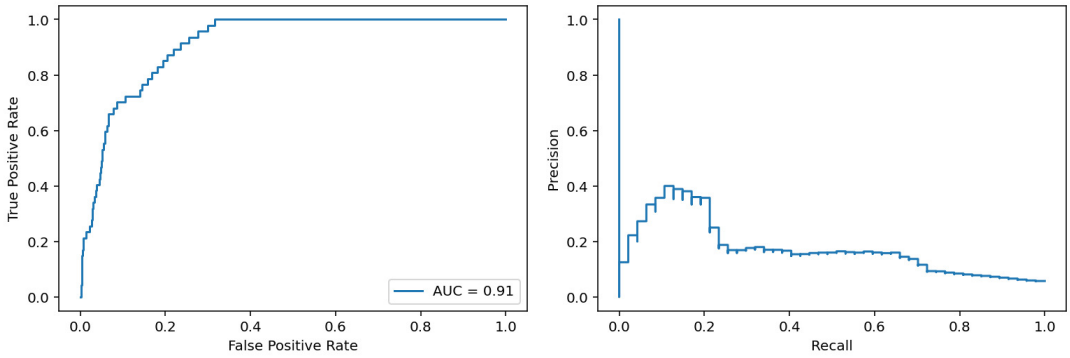

Figure 4.19: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on RAW CGM with Bidirectional LSTM layer using tanh as activation function trained for 60 epochs, using test data

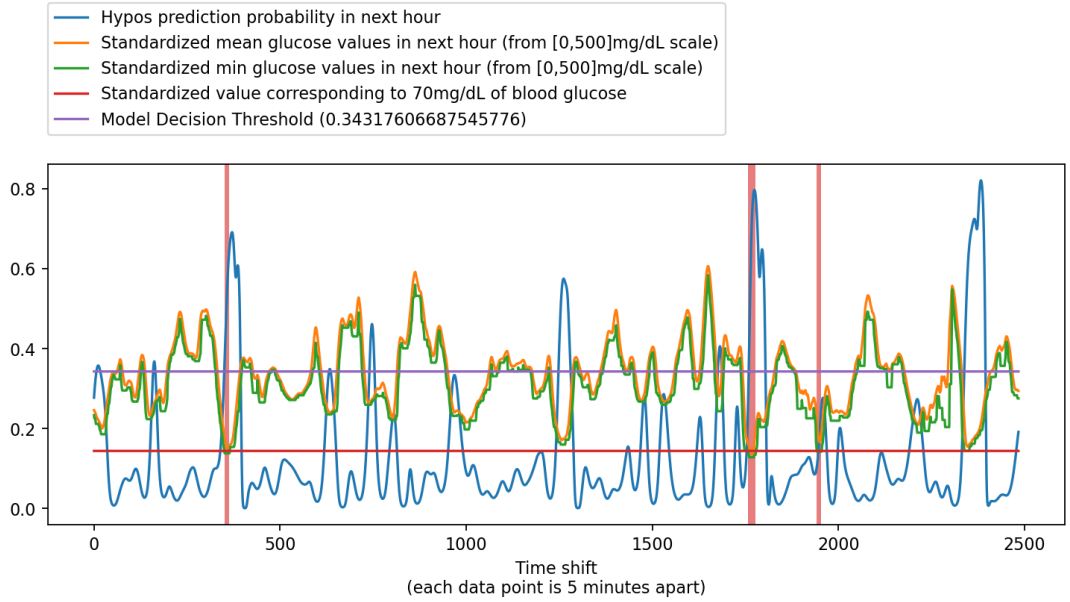

Figure 4.20: Plot of the probability of hypoglycaemia predicted by the model on RAW CGM with Bidirectional LSTM layer using tanh as activation function trained for 60 epochs

role that stress may also play in these events, we hypothesised that adding the hour of the day and the weekday to the input could bring some increase in performance. However, we could not have been more wrong, as there was a huge drop in sensitivity. The following results were produced by this approach: 0.957 of accuracy; 0.340 of sensitivity; 0.968 of specificity; 0.172 of precision; and 0.222 of MCC.

We were, then, concerned that some of the information could have been lost after passing the middle layer with a ReLU activation function, as this function will zero every negative value fed to it. Thus, we decided to try also using the tanh activation function in the middle layer. This change seemed to be positive, with every evaluation metric having increased values and the sensitivity having remained the same. These results are shown in figures 4.25 and 4.26 and in table 4.5.

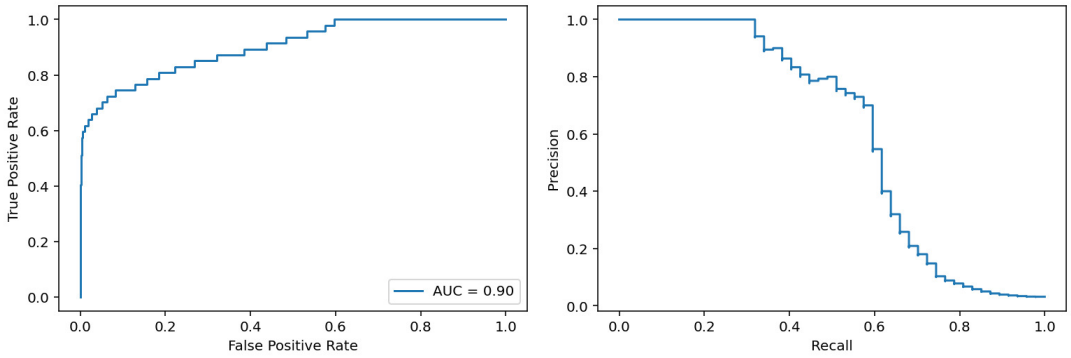

Figure 4.21: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function trained for 60 epochs, using test data

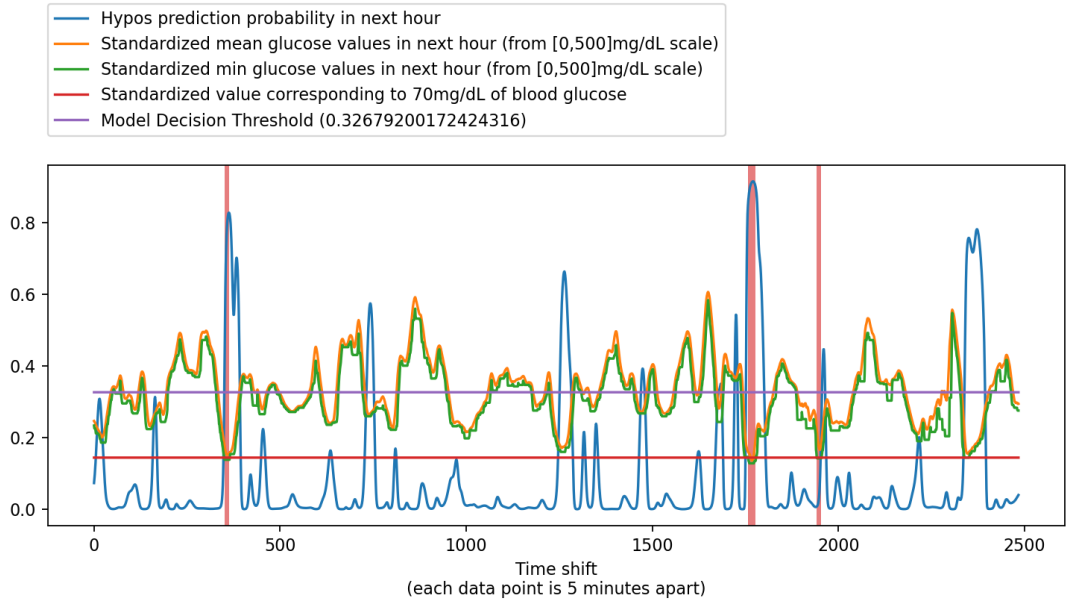

Figure 4.22: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function trained for 60 epochs

As described previously, heart rate variations can lead to fluctuations in blood glucose. Consequently, we decided to try to add the heart rate values to the model's input, while keeping the blood glucose values. The used heart rate value may, sometimes, be an approximation of the real one, as there was sometimes a time shift of 2 or 3 minutes when compared to the blood glucose values. Nonetheless, this addiction to the input seemed to be positive, as there was an increase in every single evaluation metric (table 4.5). The ROC and Precision-Recall curves are shown in figure 4.27 and the model's probability plot is shown in figure 4.28. Though it did improve the results, only half the dataset had available heart rate data, which would impede the transfer learning methodology we aimed to do further ahead. Therefore, we decided to exclude this feature from the model's input.

Table 4.4: Evaluation metrics comparing a RAW vs a filtered CGM input, using Bi-LSTM and variable epochs. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
Filtered_CGM_3_epochs	0.735	0.553	0.918	0.115	0.223
Filtered_CGM_5_epochs	0.760	0.638	0.883	0.095	0.213
Filtered_CGM_15_epochs	0.721	0.596	0.847	0.070	0.164
Filtered_CGM_50_epochs	0.805	0.702	0.908	0.128	0.273
Filtered_CGM_60_epochs	0.821	0.723	0.918	0.146	0.300
Filtered_CGM_75_epochs	0.849	0.745	0.954	0.236	0.402
Filtered_CGM_100_epochs	0.857	0.745	0.970	0.324	0.478
RAW_CGM_60_epochs	0.801	0.681	0.921	0.143	0.286

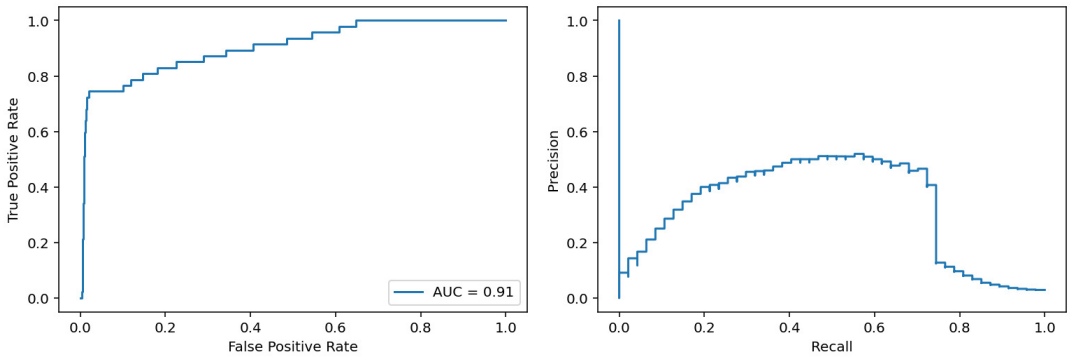

Figure 4.23: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function trained for 75 epochs, using test data

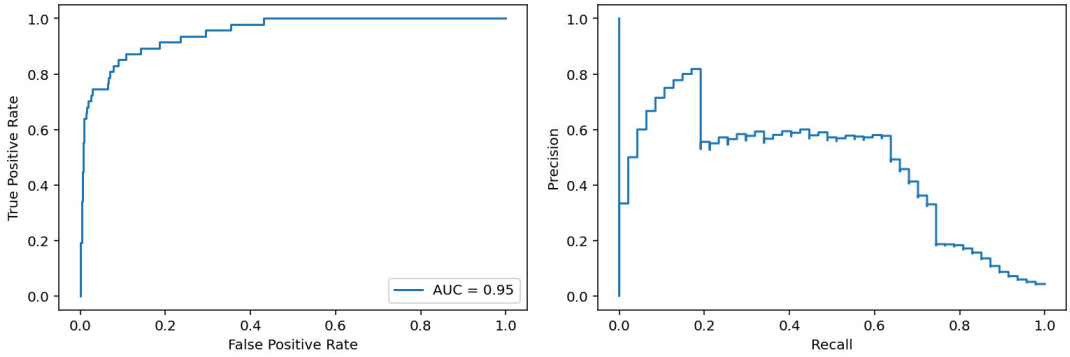

Figure 4.24: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function trained for 100 epochs, using test data

Though, for the same reason as the heart rate, we did not intend to use the amount of steps as an input for our final model, we felt like it would be relevant to assess the influence the amount of steps given by the patient had on the prediction of hypoglycaemias. As the steps were already continuously recorded by the band, we used the raw steps signal as input, while also maintaining the heart rate and blood glucose values mentioned in the previous model. The results of this

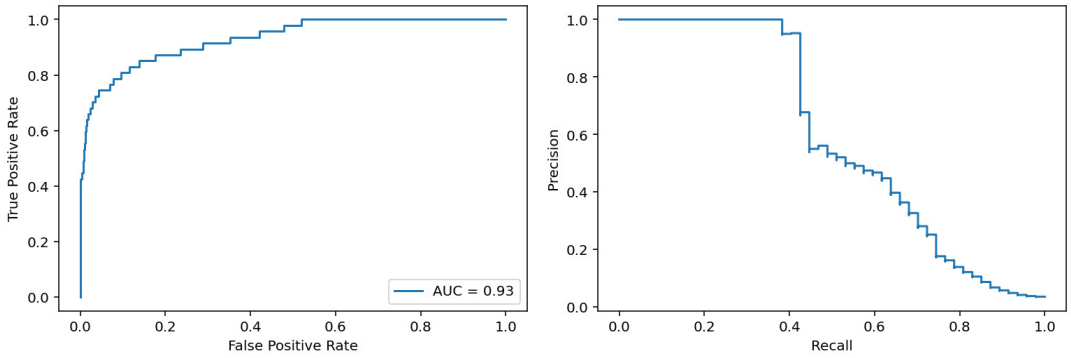

Figure 4.25: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the middle layer trained for 60 epochs, using test data

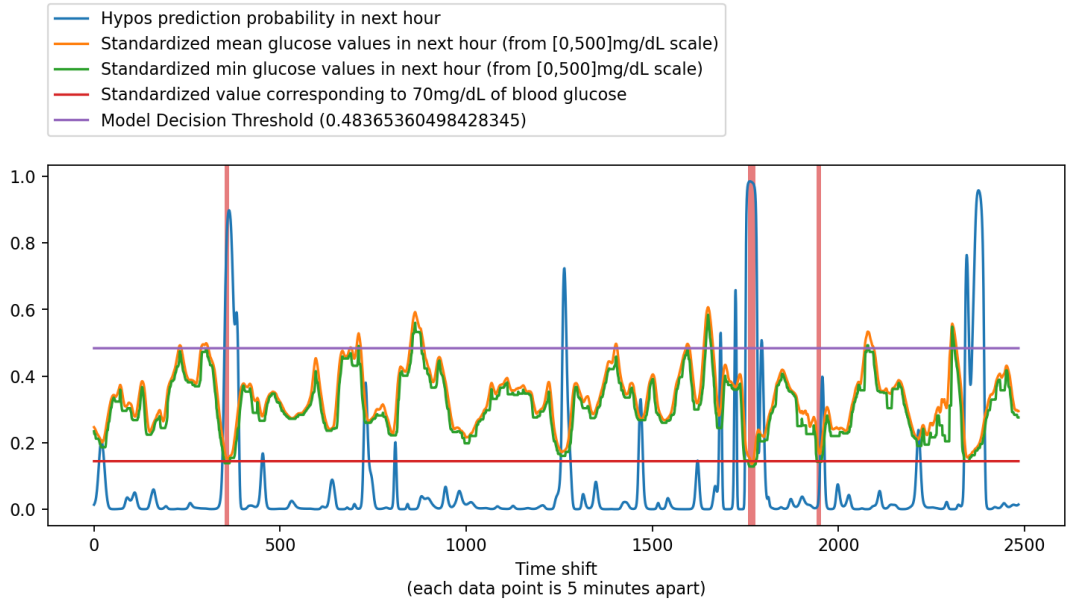

Figure 4.26: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the middle layer trained for 60 epochs, using test data

approach are shown in figures 4.29 and 4.34 and in table 4.5. The results produced by this approach seem to be worse than those of the simpler heart rate and blood glucose input. One possible reason for this decrease in performance is the fact that while the patient is resting, even if he has just had vigorous activity, the steps value will be zero. Though we believed that the model could figure out some sort of relation between the step count and the physical activity effect on the body, perhaps this type of input should be fed previously to a physiological model that could figure out this relation, only then to be used as an input to the neural network model.

At this point, we found that two new approaches needed to be taken. The first was increasing the deepness of the network, using four middle layers with tanh as activation functions and

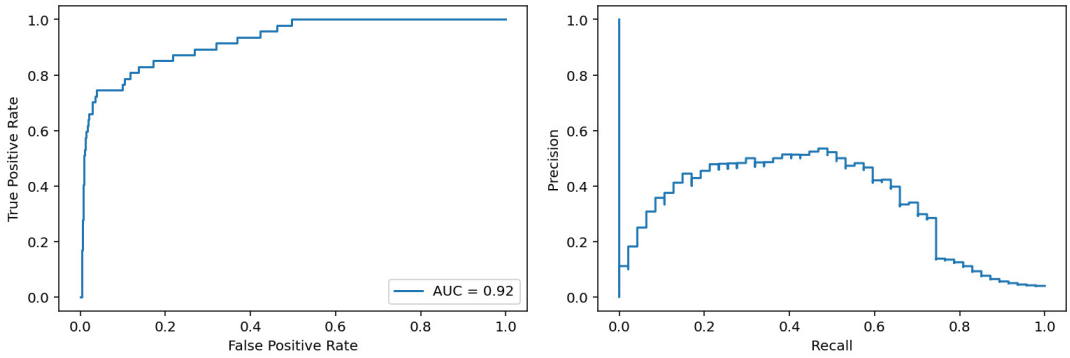

Figure 4.27: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM and Heart Rate data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the middle layer trained for 60 epochs, using test data

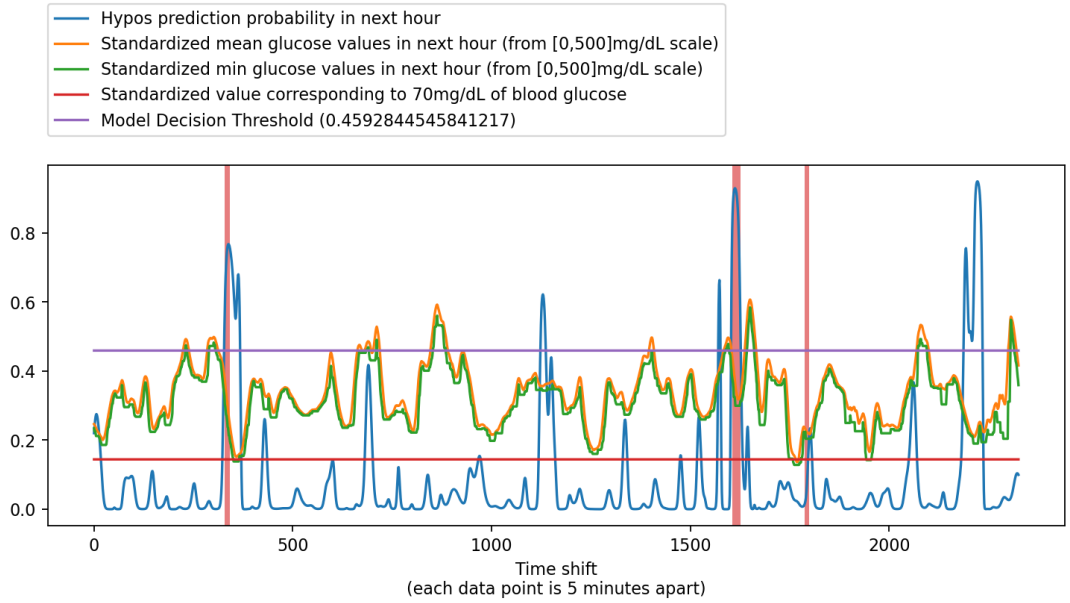

Figure 4.28: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM and Heart Rate data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the middle layer trained for 60 epochs, using test data

stacked in the following order: 60, 40, 15 and 5 units. This new architecture is depicted in figure 4.31. It proved its efficacy in increasing the model's sensitivity and MCC while maintaining performance in the remaining metrics. This model's ROC and Precision-Recall curves are shown in figure 4.32, its confusion matrix in figure 4.33 and the model probabilities' plot in figure 4.34. Given the amount of hidden-layers the model currently possessed, we tried to avoid over-fitting by adding 20% dropout layers between the four mentioned middle layers. This model's ROC and Precision-Recall curves are shown in figure 4.35, its confusion matrix in figure 4.36 and the model probabilities' plot in figure 4.37. The results showed a decrease from 0.851 to 0.809 in the model's sensitivity, but if we look closely to this model's confusion matrix, we can tell that there

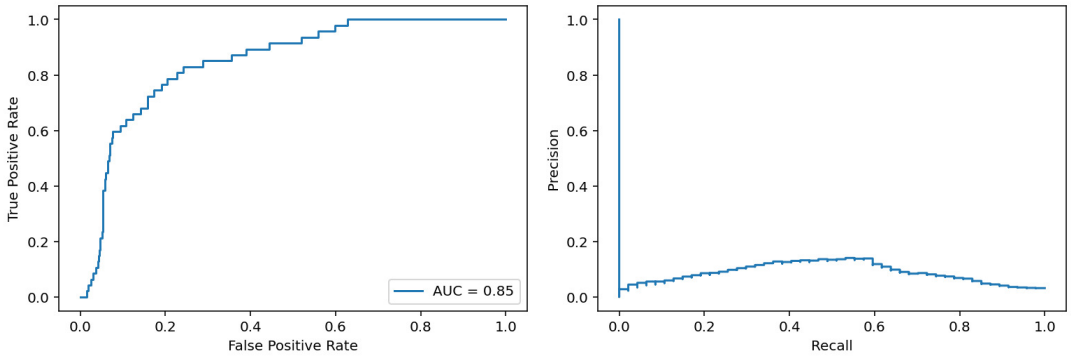

Figure 4.29: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM and Heart Rate data, as well as unfiltered Steps data, with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the middle layer trained for 60 epochs, using test data

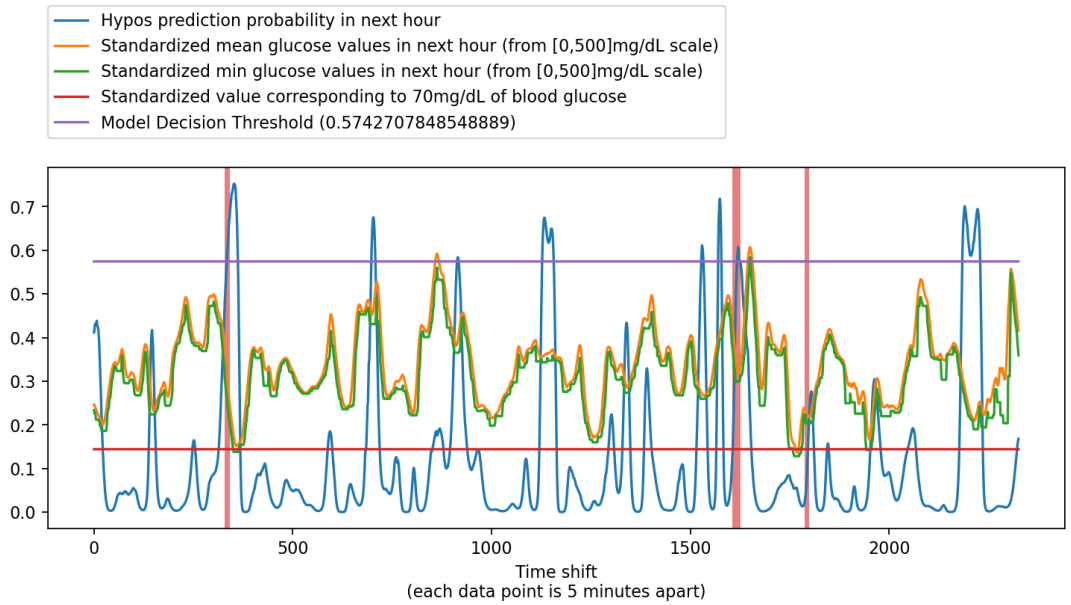

Figure 4.30: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM and Heart Rate data, as well as unfiltered Steps data, with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the middle layer trained for 60 epochs, using test data

was great progress in the reduction of false positives, without having that big of an effect in the true positives. The performance metrics of both models are summarized in table 4.6. Hence, the dropout layers were considered useful and maintained in the model's architecture.

Given that the blood glucose values are very much correlated with both the ingestion of carbohydrates and the use of insulin, we decided to include this information in the model's input. However, these data were manually inserted by the patient and very sparse throughout the day. Thus, just like with the steps data previously described, inputting these data in its raw form would simply not work. The solution, was to find a physiological model which described the amount of

Table 4.5: Evaluation metrics using tanh as the middle layer's activation function and showing the addition of heart rate and steps to the filtered CGM input, using Bi-LSTM and 60 epochs. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
Filtered_CGM_mLTanh	0.841	0.723	0.959	0.256	0.413
Filtered_CGM_HR_mLTanh	0.851	0.745	0.957	0.265	0.427
Filtered_CGM_HR_Steps_mLTanh	0.633	0.319	0.946	0.109	0.159

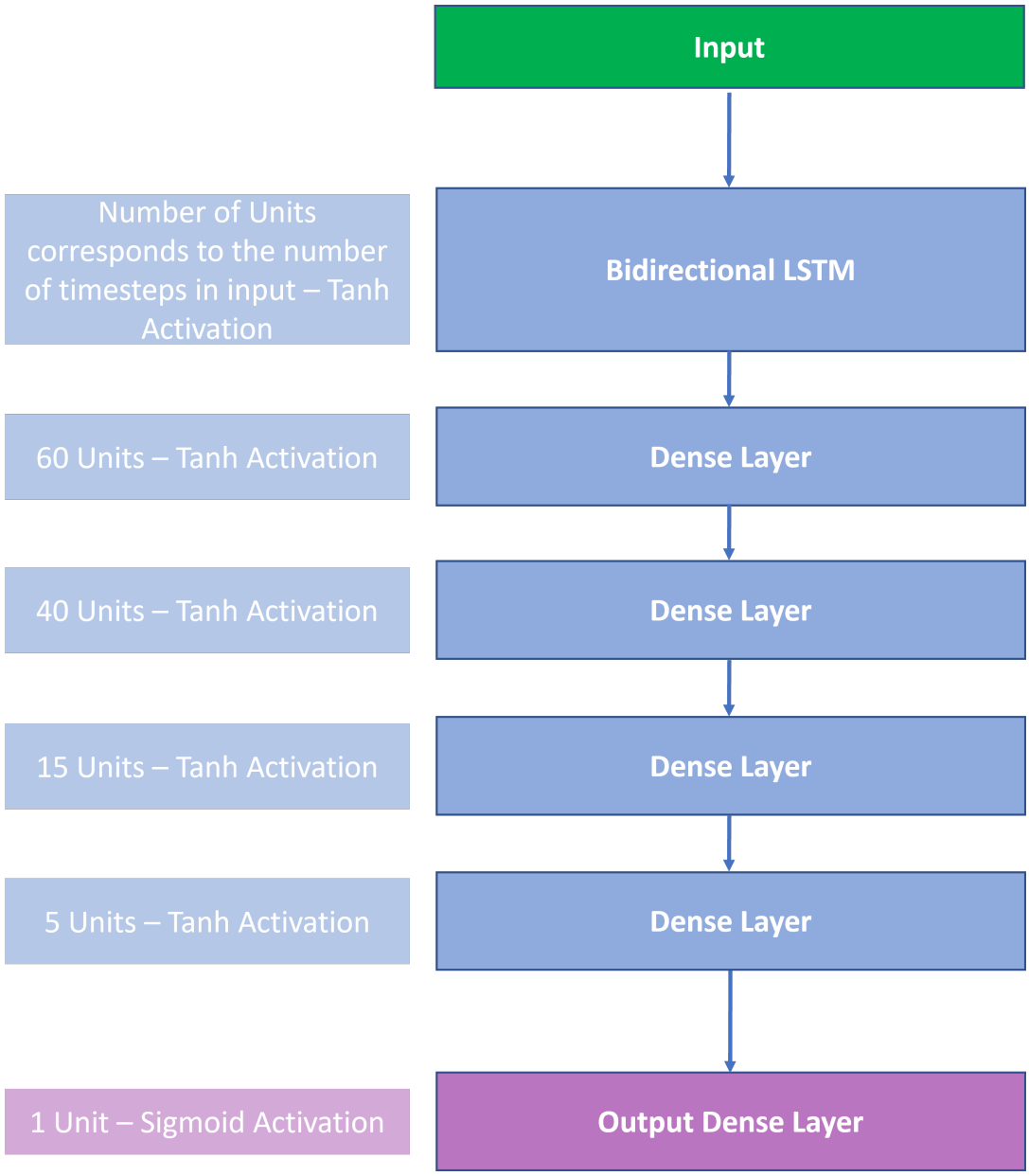

Figure 4.31: Deeper-models' architecture diagram

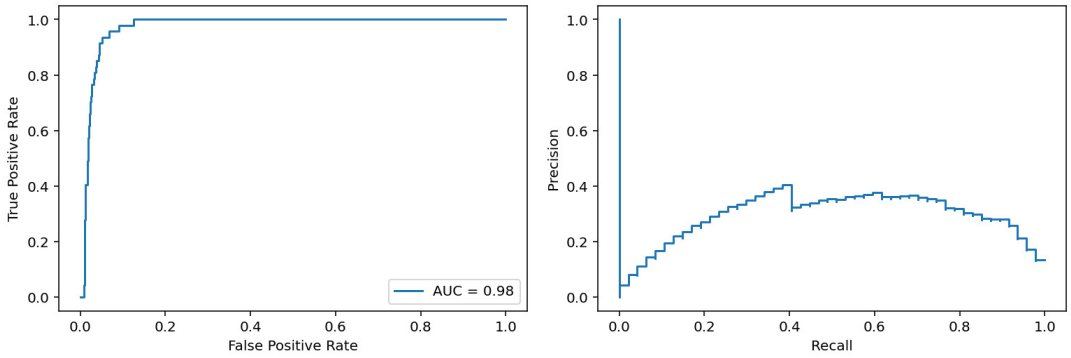

Figure 4.32: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs, using test data

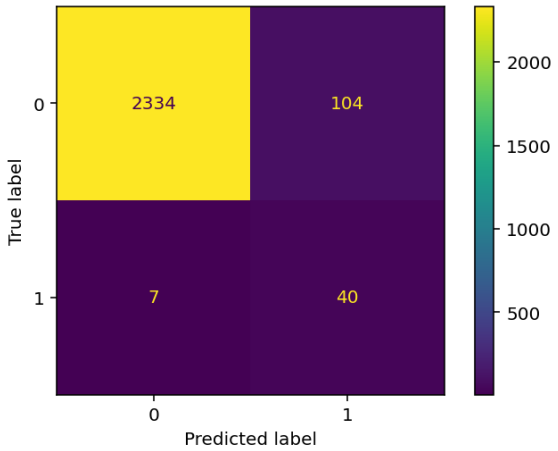

Figure 4.33: Confusion matrix of the model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs, using test data

insulin in the blood after a bolus and its decay throughout time and another to model the amount of ingested carbohydrates and its decay throughout time. These models were based on equations 4.3, 4.4 and 4.5, where t represents the time instant, the compartments C_1 and C_2 have initial values of zero, $u(t)$ is the insulin dose and $K_{DIA} = 0.0195$ is a constant related to the duration of insulin action; and equation 4.6, where t is the time instant, C_{in} is the amount of carbohydrates ingested in a meal (in grammes), $C_{bio} = 0.8$ is the carbohydrate bioavailability, and $t_{max,G} = 50(min)$ denotes the time of the maximum appearance rate of glucose in the accessible glucose compartment.

$$\frac{dC_1(t)}{dt} = u(t) - K_{DIA}C_1(t) \tag{4.3}$$

$$\frac{dC_2(t)}{dt} = K_{DIA}(C_1(t) - C_2(t)) \tag{4.4}$$

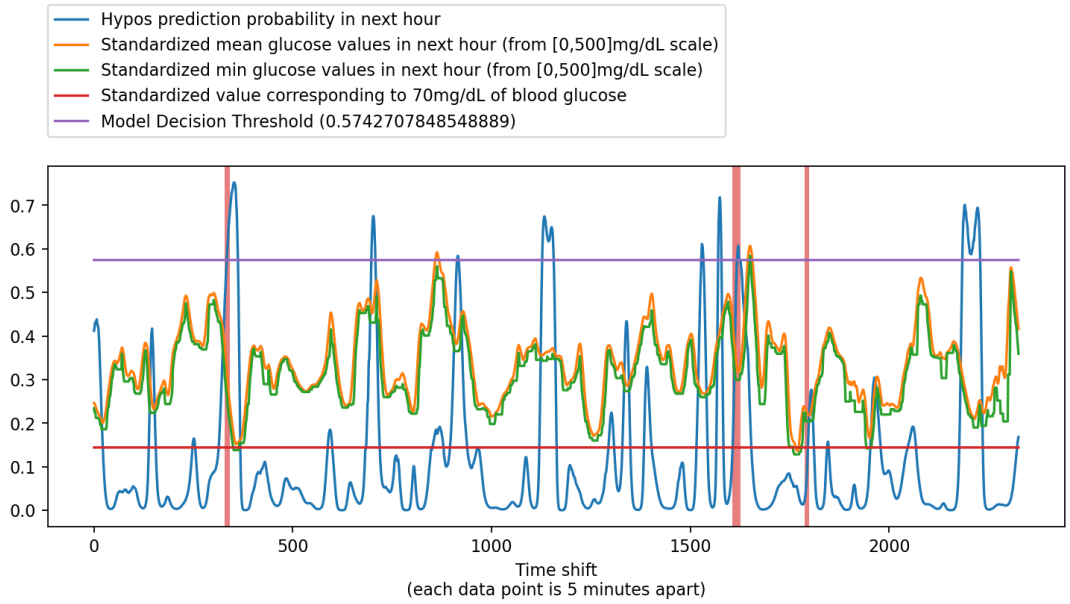

Figure 4.34: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs, using test data

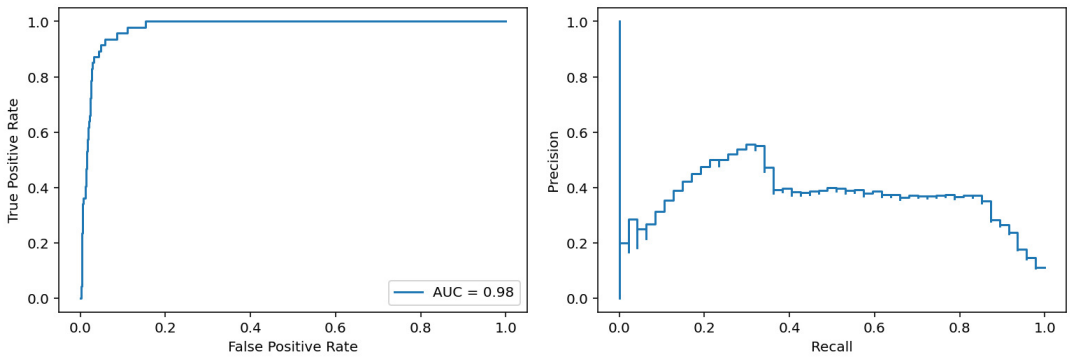

Figure 4.35: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

$$InsulinonBoard = C_1(t) + C_2(t) \tag{4.5}$$

$$GlucoseAbsorptionRate = \frac{C_{in}C_{bio}te^{(-t/t_{max,G})}}{t_{max,G}^2} \tag{4.6}$$

Usually, there was a gap between the beginning of the CGM data collection and the carbohydrates and insulin ones. The solution was to eliminate these data points. However, a few times, there would be full days worth of missing manual inputs, but this time we simply ignored those, as

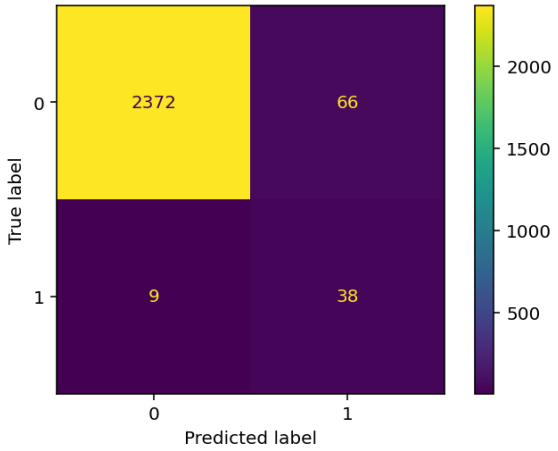

Figure 4.36: Confusion matrix of the model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

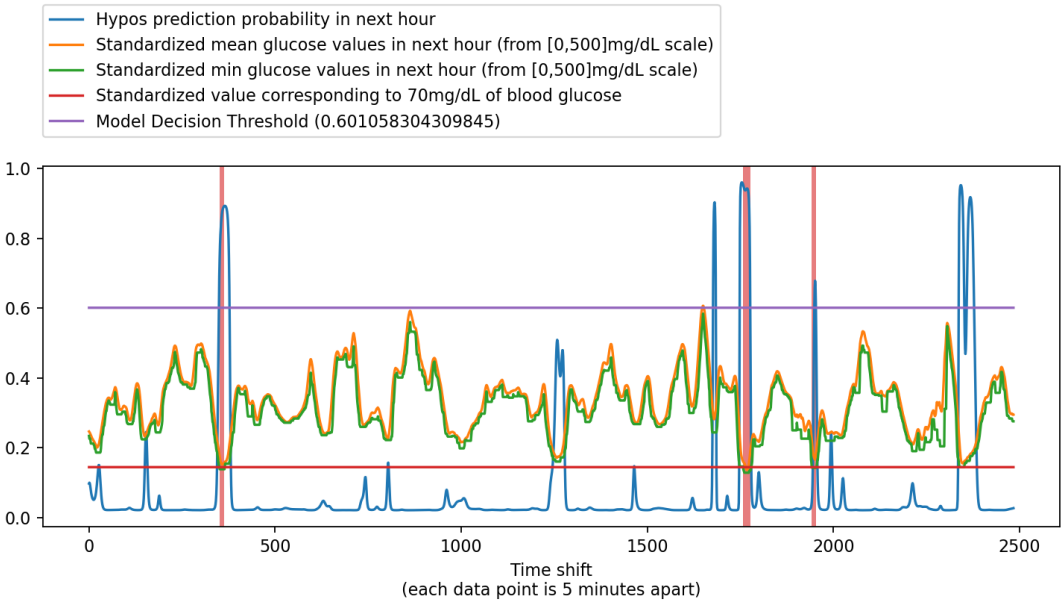

Figure 4.37: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

we considered it to be normal and would possibly add robustness to the model when dealing with this type of situations. It is also important to highlight that the time delay between the patient's manual insertion of the insulin boluses and carbohydrates intakes may also impact the model's performance.

Firstly, we assessed the influence of adding the insulin values, output by its physiological model, to the model's input. This model's performance is depicted in figures 4.38 and 4.39 and in table 4.7. Secondly, we added to the filtered CGM input both the insulin and the carbohydrate

Table 4.6: Evaluation metrics comparing deeper networks with and without dropout layers using tanh as the middle layer's activation function using Bi-LSTM and 60 epochs. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
middleLayerTanh	0.904	0.851	0.957	0.278	0.471
middleLayerTanh_Dropout_20_%	0.891	0.809	0.973	0.365	0.532

values. This model's results are shown in figures 4.40 and 4.41 and in table 4.7. Comparing the two, it does not become quite clear which model is best, however, the addition of the carbohydrates does seem to help increase a bit the sensitivity. Even so, in personalized models, it seems like the best model input is just the filtered CGM data, given that these new additions led to a decrease in every single performance metric showed in table 4.7. Moreover, figures 4.39 and 4.41 seem to indicate some overfitting, with quite high model probabilities output in situations where it does not appear to be a reason for that to happen. This makes it very hard to compute a threshold that can perfectly separate hypoglycaemias from the rest of the glycaemic values.

The reason why we did not decide to try a model solely using filtered CGM and carbohydrates data is due to the fact that though the addition of carbohydrates data theoretically aids the model to decide when to avoid a false positive, is does not actually enhance the prediction of true positives, given that most of the times in type 1 diabetics, hypoglycaemias are caused by a insulin bolus.

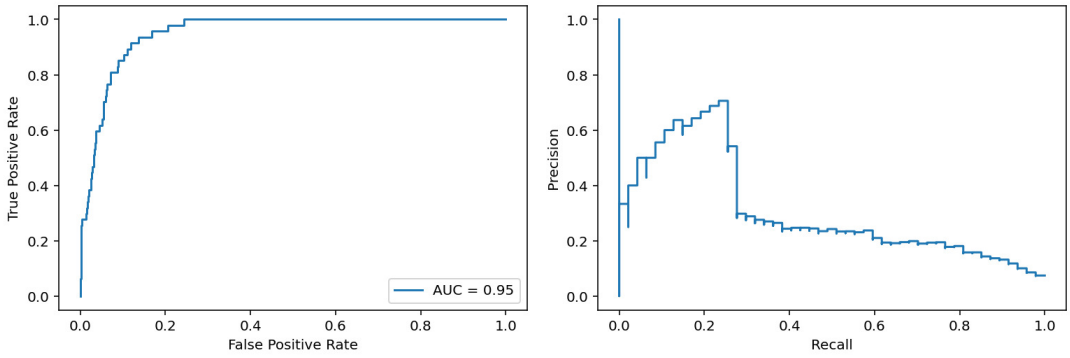

Figure 4.38: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM and Insulin data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

Table 4.7: Evaluation metrics comparing deeper networks with insulin and carbohydrates data added to the input, using Bi-LSTM, tanh as the middle layer's activation function, 20% dropout layers and 60 epochs. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
Filtered_CGM_with_Ins	0.777	0.596	0.958	0.219	0.341
Filtered_CGM_with_Ins_Carb	0.775	0.660	0.891	0.107	0.234

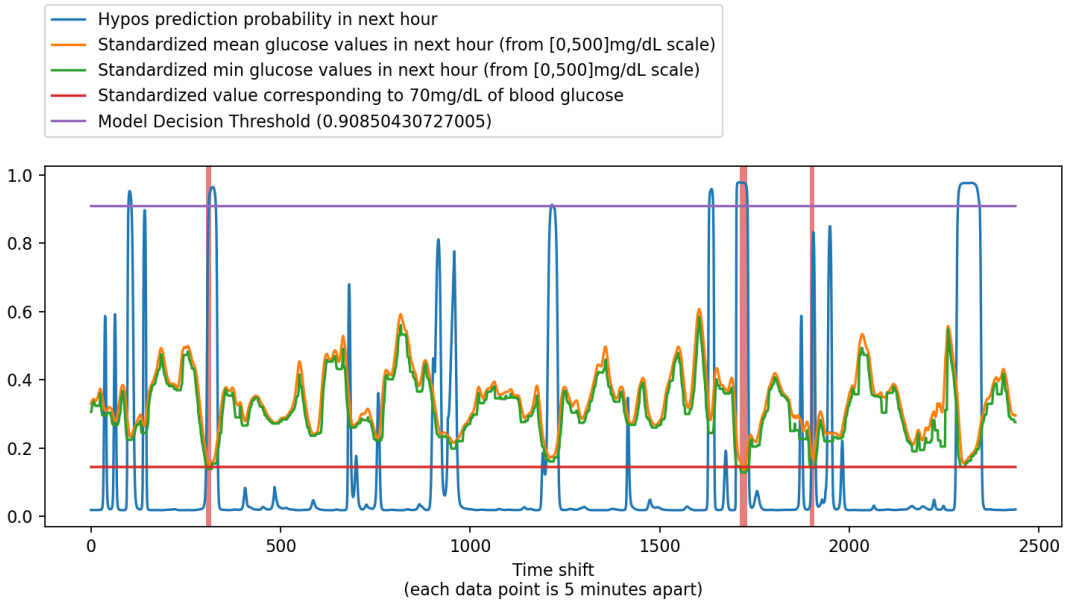

Figure 4.39: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM and Insulin data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

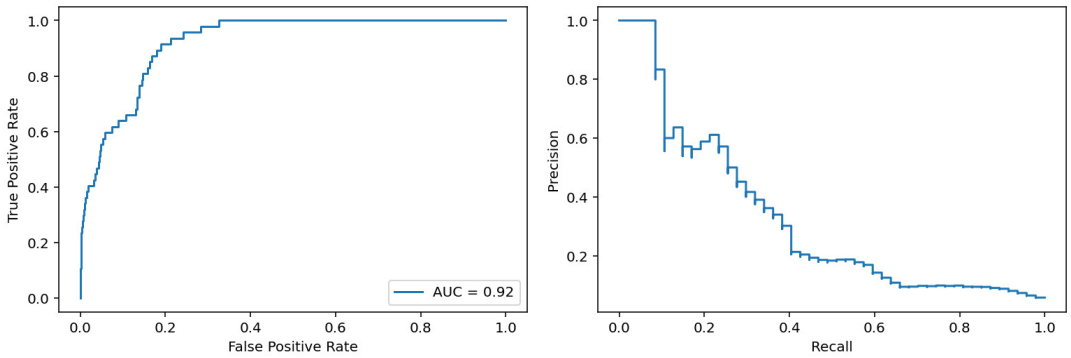

Figure 4.40: ROC curve and AUC value (left) and Precision-Recall curve (right) for model on Filtered CGM, Insulin and Carbohydrates data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

4.2.3.2 Generalized Models

Though this approach did not work when using personalized models, we hypothesised that the under-performance of models with the added insulin and carbohydrates data could be due to a lack of enough training data, given the added complexity of the input. Consequently, we decided to give a try to generalized models and transfer learning. To train the generalized model, the training data from 9 patients was used. These models were validated using the testing data from the same 9 patients. Despite the fact that the model using filtered CGM and insulin did perform

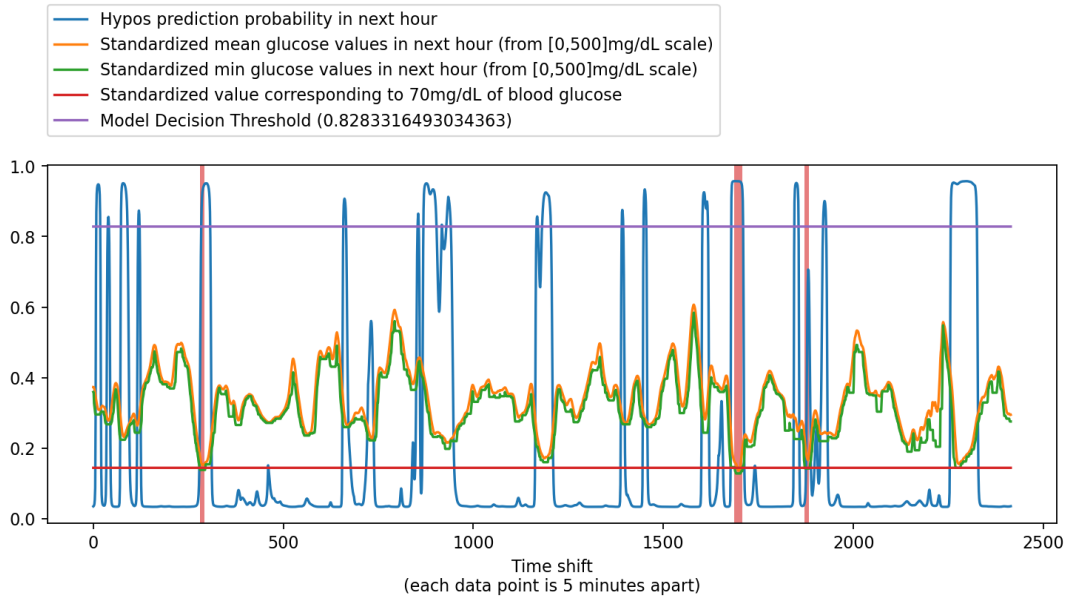

Figure 4.41: Plot of the probability of hypoglycaemia predicted by the model on Filtered CGM and Insulin and Carbohydrates data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

better in a few metrics than that using a filtered CGM, insulin and carbohydrates input, the fact is that the latter did provide a better sensitivity and, in a context where further data is available, may provide better results.

Firstly, we trained a generalized model using solely the filtered CGM data, in order to have a baseline model to compare the following ones. This model's ROC and Precision-Recall curves are shown in figure 4.42 and its performance metrics are summarized in table 4.8. In the case of a sole filtered CGM input, the generalize model shows a decrease in performance when compared to a personalized one. We then proceeded to test our hypothesis and trained the models with a filtered CGM, insulin and carbohydrates input using the training data from these 9 patients. Using this input, we compared the performance of the deep network architecture described in figure 4.31 with 20% dropout layers interleaved in between the hidden layers and that of a model alike but using successively the following number of units in the middle layers: 150, 120, 100, 80, 60, 50, 40, 20, 10 and 5, all interleaved with 20% dropout layers. The model's ROC and Precision-Recall curves are shown in figures 4.43 and 4.44, respectively, and their performance metrics are summarized in table 4.8. As predicted, a bigger training set did increase the performance of models using insulin and carbohydrates data when using these same patient's testing data. Given the higher balanced accuracy and sensitivity, coupled with a low decrease in the remaining metrics, we found the even deeper model architecture to be best.

We hypothesised that transforming the problem into a multiclass classification one, with a class for euglycaemia, another for hypoglycaemia and another for hyperglycaemia, could detect some underlying patterns and enhance the hypoglycaemia classification performance. We used

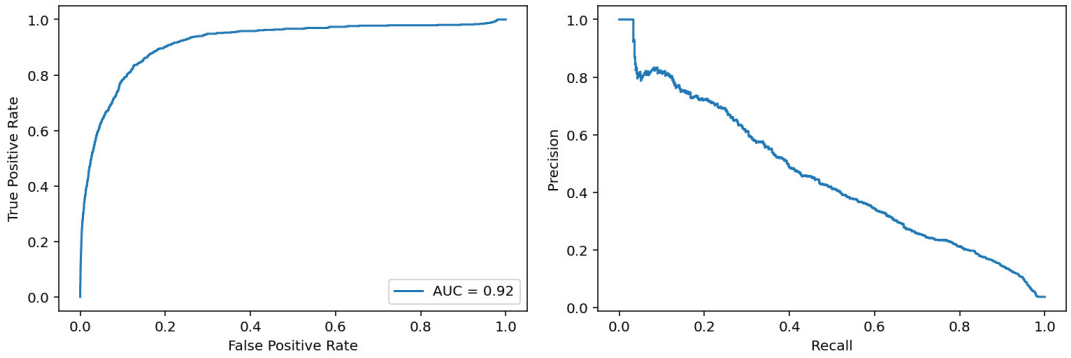

Figure 4.42: ROC curve and AUC value (left) and Precision-Recall curve (right) for generalized model on Filtered CGM and Insulin with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

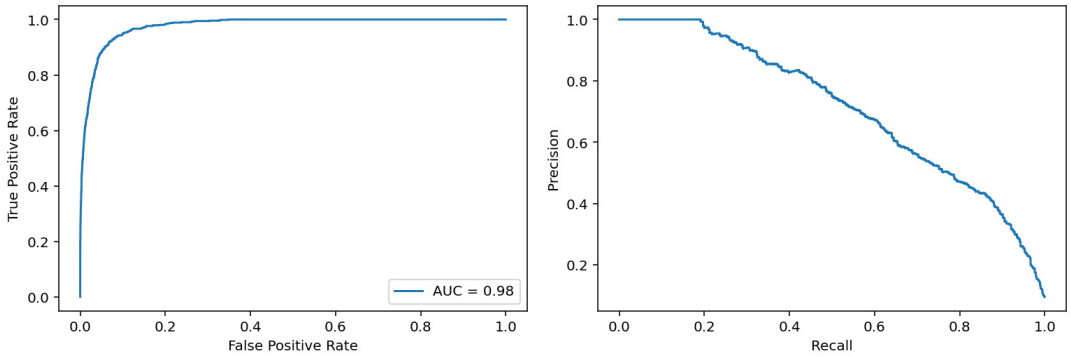

Figure 4.43: ROC curve and AUC value (left) and Precision-Recall curve (right) for generalized model on Filtered CGM, Insulin and Carbohydrates data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

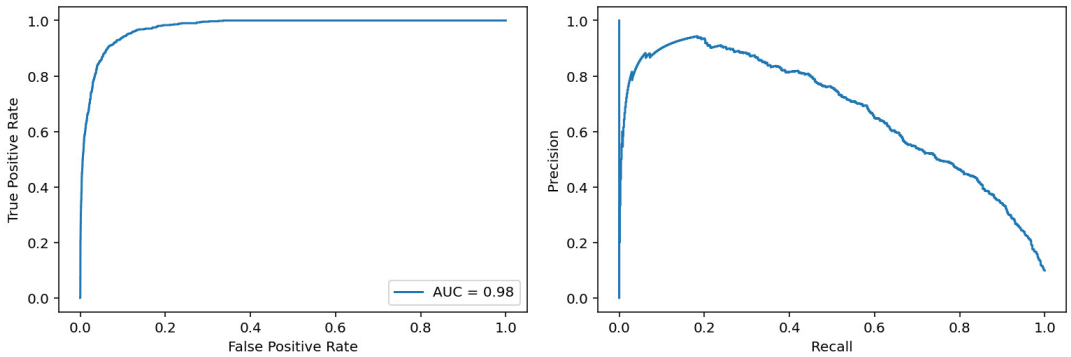

Figure 4.44: ROC curve and AUC value (left) and Precision-Recall curve (right) for generalized model on Filtered CGM, Insulin and Carbohydrates data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the even deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

Table 4.8: Evaluation metrics comparing two distinct network depths and several inputs, as well as multiclass labels, using Bi-LSTM, tanh as the middle layer's activation function, 20% dropout layers and 60 epochs. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
Filtered_CGM_Deep	0.802	0.669	0.936	0.278	0.401
Filtered_CGM_Ins_Carb_Deep	0.885	0.804	0.967	0.470	0.597
Filtered_CGM_Ins_Carb_Deeper	0.897	0.834	0.960	0.437	0.585
Multiclass	0.869	0.773	0.965	0.470	0.583

this last best model in this approach. However, when we computed the metrics relatively to the hypoglycaemia class, the performance was in fact worse, as can be seen in table 4.8. So, we discarded this approach.

Lastly, we also tried these two depths of neural networks to test the relevance of having patient specific information, in detail, the insulin type, the gender and the age range, added to the filtered CGM, insulin and carbohydrates inputs. The model with an architecture alike 4.31 with 20% dropout layers interleaved in between the hidden layers presented the following ROC and Precision-Recall curves (fig 4.45) and the one with the previously described deeper architecture, the ones found in figure 4.46. The summary of all evaluation metrics can be found in table 4.9. The model using the architecture described in figure 4.31 with 20% dropout layers interleaved in between the hidden layers and having the added patient info to the input did seem to present the best performance.

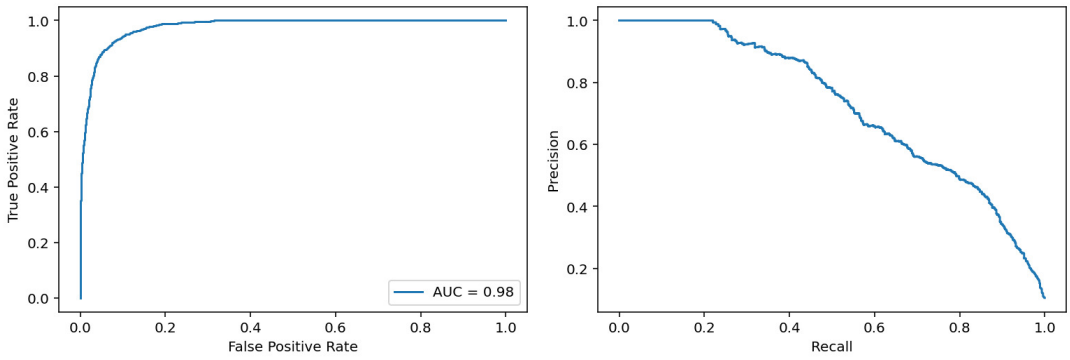

Figure 4.45: ROC curve and AUC value (left) and Precision-Recall curve (right) for generalized model on Filtered CGM, Insulin, Carbohydrates and Patient Information data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

4.2.3.3 Best Models' Testing

Once finalized the model selection process, we proceeded to evaluate the models effectiveness in new patients. So, we tested three different models on the three test patients we had separated earlier. The tested models were the personalized model with filtered CGM data as input and using the architecture depicted in figure 4.31 and 20% dropout layers interleaved in between the hidden

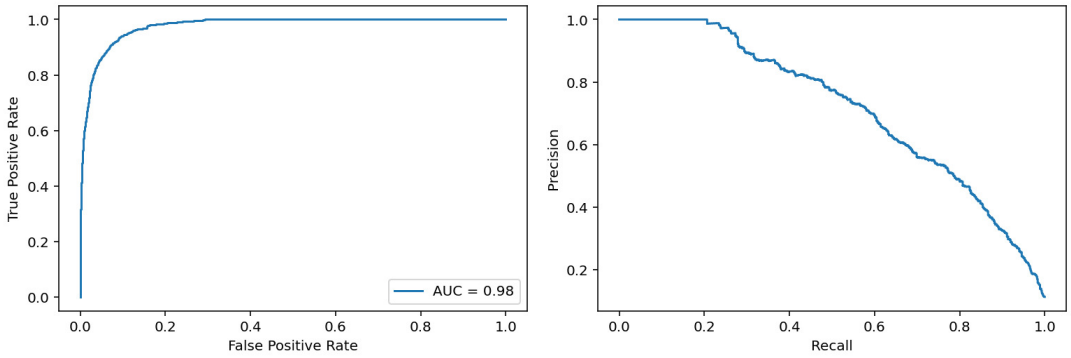

Figure 4.46: ROC curve and AUC value (left) and Precision-Recall curve (right) for generalized model on Filtered CGM, Insulin, Carbohydrates and Patient Information data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the even deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

layers (model 1); the generalized model using the same model architecture as the one described for the previous model, but having as input the filtered CGM, insulin, carbohydrates and patient information data (model 2); and a transfer learning version of this last model trained on the training data of each specific test patient (model 3).

In order to perform the transfer learning task, we removed the four last trained layers and replaced then with a 15 and a 5 units dense layers using tanh as the activation function and interleaved with a 20% dropout layer, and a single unit output dense layer using a sigmoid activation function.

These models' performances are summarized via mean values across all test patient's testing data and the standard deviation in table 4.10. Given the rather similar performance between models 2 and 3, it seems like there is little to no advantage in allocating the computational power to perform the transfer learning task.

Thus, we conclude that the best performing model is model 2. Their output probability plots are shown in figures 4.47, 4.48 and 4.49. It is interesting to notice the fact that the model, in the case of patient 596, only outputs higher probabilities for cases where there exists indeed a hypoglycaemic episode. We believe, that, in such cases, a fine tuning of the model's decision threshold would greatly benefit the model's performance. It is also important to highlight that, if we consider each red shaded area as a single hypoglycaemic episode, virtually all hypoglycaemic episodes are detected by the model, independently of the patient.

Table 4.9: Evaluation metrics comparing two distinct network depths of models on Filtered CGM, Insulin, Carbohydrates and Patient Information data, using Bi-LSTM, tanh as the middle layer's activation function, 20% dropout layers and 60 epochs. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
With_Patient_Info_Deep	0.906	0.852	0.961	0.446	0.598
With_Patient_Info_Deeper	0.886	0.804	0.968	0.482	0.605

Table 4.10: Evaluation metrics comparing the performance of the final selected models in 3 distinct test patients. The reported values correspond to a mean across patients and the respective standard deviation. The reported metrics are relative to the test data

	Bal_Acc	**Sensitivity**	**Specificity**	**Precision**	**MCC**
Model 1	0.855 ± 0.019	0.756 ± 0.038	0.953 ± 0.006	0.603 ± 0.072	0.639 ± 0.033
Model 2	0.876 ± 0.017	0.810 ± 0.050	0.942 ± 0.018	0.568 ± 0.023	0.643 ± 0.019
Model 3	0.875 ± 0.009	0.804 ± 0.031	0.946 ± 0.015	0.585 ± 0.019	0.651 ± 0.011

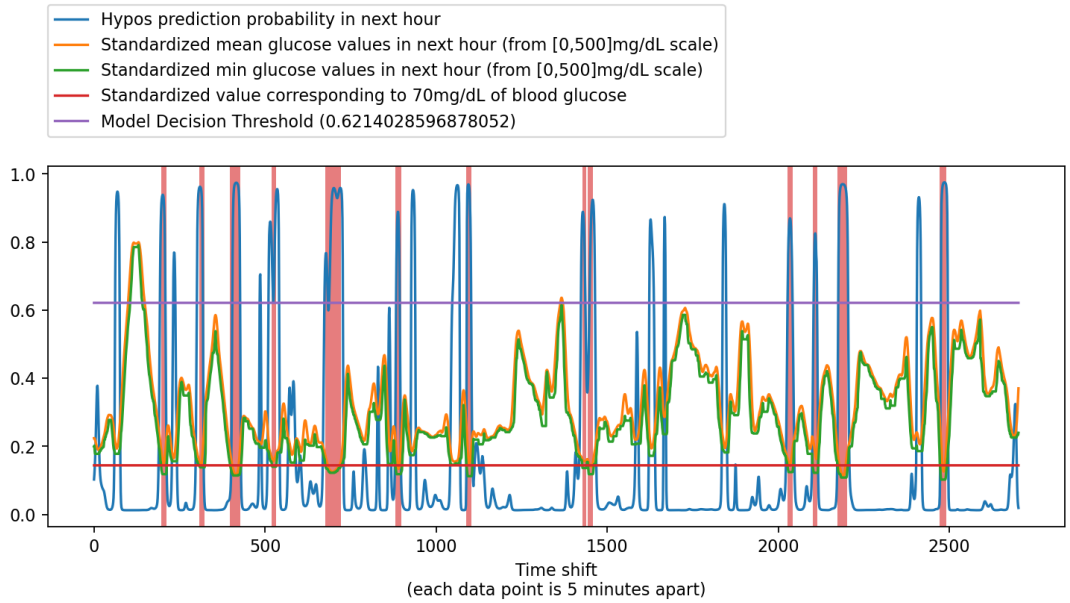

Figure 4.47: Patient 540's plot of the probability of hypoglycaemia predicted by the model on Filtered CGM, Insulin, Carbohydrates and Patient Information data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

Comparing our model's performance with those reported in the scientific literature, our model holds quite lower precision values than those described in Quan et al., though outperforming their model in the sensitivity values, which, as previously explained, seem to be more clinically relevant. When comparing with Zhu et al. described metrics, which are exactly the same as those used throughout our work (except for the accuracy, in which case we used the balanced accuracy metric), it outperforms their model's performance in every single described metric. Comparing to the sensitivity and specificity values described in Cichosz et al., our model does present a lower specificity than the one described, but it must be considered that our PH is 40 minutes bigger than the one they used.

Additionally, our results appear to contradict the higher effectiveness of personalized models reported in the scientific literature, which may reduce the computational power and time required to implement such a solution in a real life application. Though, as described in literature, there is relevance in patient specific information.

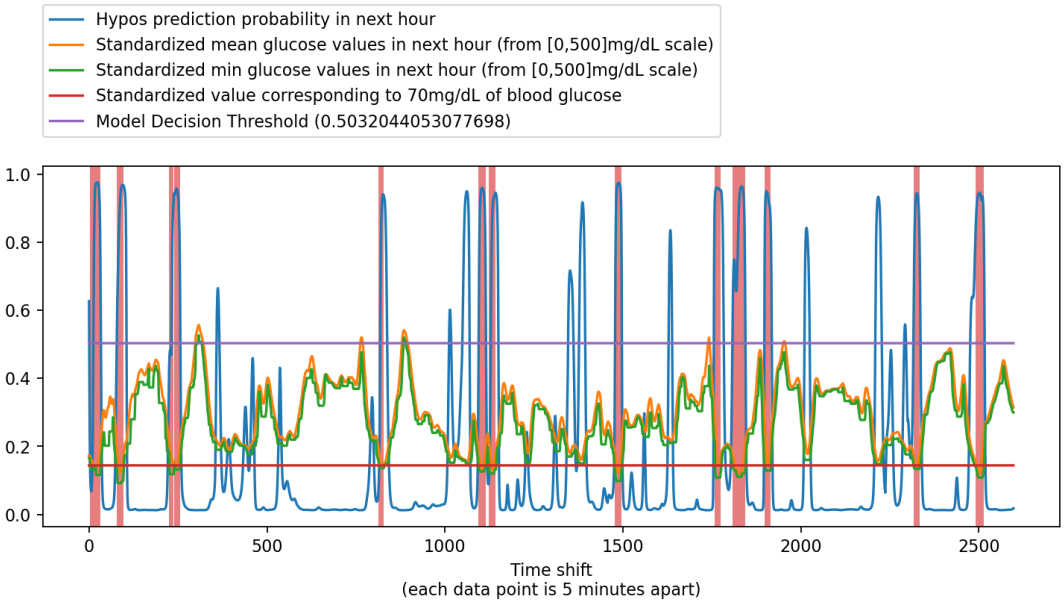

Figure 4.48: Patient 591's plot of the probability of hypoglycaemia predicted by the model on Filtered CGM, Insulin, Carbohydrates and Patient Information data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

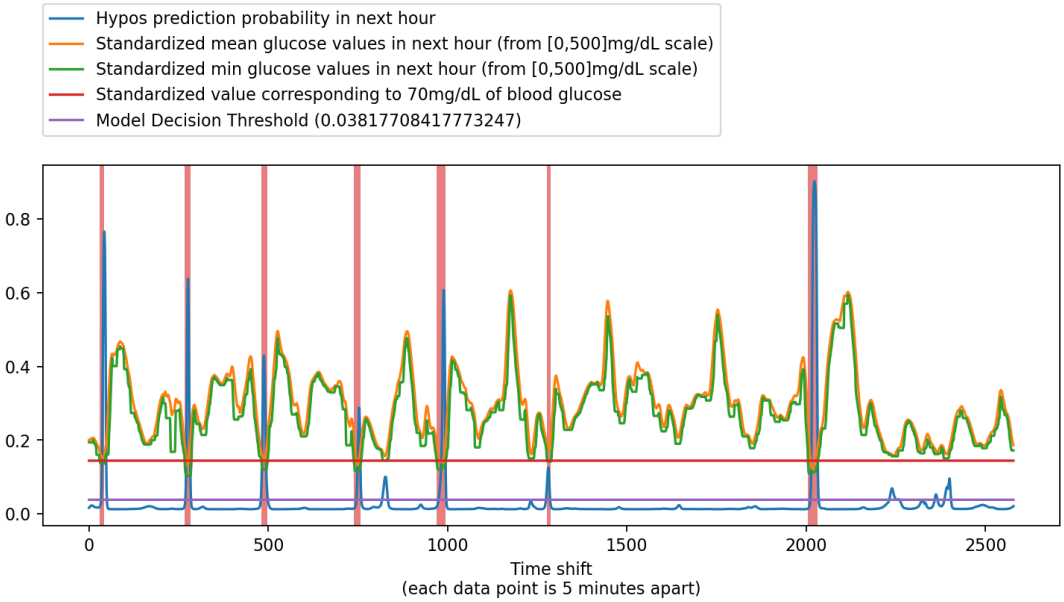

Figure 4.49: Patient 596's plot of the probability of hypoglycaemia predicted by the model on Filtered CGM, Insulin, Carbohydrates and Patient Information data with Bidirectional LSTM layer using tanh as activation function for both the Bidirectional LSTM and the deeper middle layers trained for 60 epochs with 20% dropout layers, using test data

Even though clinically speaking, there are few consequences to the false positive model outputs, there must be a better performance in this area, as they may lead to a discredit, from the

patient, of the good performance of the model.

Chapter 5

Conclusion

Throughout this work we aimed to create a pipeline for hypoglycaemia prediction with a short to medium PH, which could be useful in the management of type 1 diabetes.

There seems to exist relevance in inputting into the model filtered CGM data, accompanied by the available insulin and carbohydrates quantities, as well as relevant patient information (in this case, we used the insulin type, the gender and the age range). In the presented case, the use of personalized or transfer learning models did not appear to be relevant.

The trained model using data from nine distinct patients performed well when forced to generalize what it had learned to the three remaining test patients' testing data. The reported metrics for this model were a balanced accuracy of 0.876 ± 0.017, a sensitivity of 0.810 ± 0.050, a specificity of 0.942 ± 0.018, a precision of 0.568 ± 0.023 and a MCC of 0.843 ± 0.019. When looking closely to the model's probability plots, we can confirm that there is a coverage of virtually every single hypoglycaemic episode. However, the model still outputs too many false positives, which may make a patient discredit the created alerts.

Nonetheless, we were able to create a solid working pipeline capable of helping type 1 diabetes patients better manage their disease. It performs almost equally well between the training and testing patients, and surpasses, to the best of our knowledge, the models described in the current scientific literature for type 1 diabetes patients, reaching the primary goals that we proposed ourselves to.

Regarding the limitations of the work that was developed, the inherently imbalanced distribution between hypoglycaemic values and the remaining blood glucose values, makes it hard to approach this classification problem, given that it reduces the number of distinct cases the model deals with. However, we did use class weights in order to try to minimize this problem. The low amount of patients available in the dataset also make for a less robust model. Ideally, there should be an increase in the number of patients in the dataset, so that the model can be trained to perform even better. We could also tell that, in some cases, the computed decision threshold could be optimized.

As for future developments in this project, we would like to try out our pipeline in new type 1 diabetes datasets. It could also be worth trying attention layers or substituting the Bi-LSTM layer by some other types of neural networks, such as GRU. If available, we would like to consider new model inputs, given that the glycaemic oscillations can be caused by various factors, as described throughout this work. We would also like to continue the already explored approach of classification of both hypo and hyperglycaemic events, in order to fully aid the patients. Once the pipeline has a more robust performance, we plan on collecting type 2 diabetes patient's data and try to generalize this type of pipeline to those patients.

References

[1] Apresentamos o novo dexcom g6. https://www.dexcom.com/pt-PT.

[2] Diabetes: Overview. https://www.who.int/health-topics/diabetes#tab=tab_1.

[3] Guardian connect continuous glucose monitoring system. https://www.medtronicdiabetes.com/products/guardian-connect-continuous-glucose-monitoring-system.

[4] Epidemiology of severe hypoglycemia in the diabetes control and complications trial. *The American Journal of Medicine*, 90(1):450–459, 1991.

[5] Hypoglycemia in the diabetes control and complications trial. the diabetes control and complications trial research group. *Diabetes*, 46(2):271–286, 1997.

[6] Economic costs of diabetes in the u.s. in 2002. *Diabetes Care*, 26(3):917–932, 2003.

[7] The bglp challenge: Results, papers, and source code, Aug 2020.

[8] Deep dive into bidirectional lstm. https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/, May 2020.

[9] Diabetes: Factos e números 2016, 2017 e 2018*. http://www.revportdiabetes.com/wp-content/uploads/2020/05/RPD-Março-2020-Revista-Nacional-págs-19-27.pdf, 2020.

[10] Relatório programa nacional para a diabetes: Desafios e estratégias 2019. https://www.dgs.pt/portal-da-estatistica-da-saude/diretorio-de-informacao/diretorio-de-informacao/por-serie-1184293-pdf.aspx?v===DwAAABLCAAAAAAABAArySzItzVUy81MsTU1MDAFAHzFEfkPAAAA, Feb 2020.

[11] The global diabetes compact. https://www.who.int/publications/m/item/the-global-diabetes-compact, Apr 2021.

[12] About GATEKEEPER. https://www.gatekeeper-project.eu/about-gatekeeper/#block-missionandvision, Mar 2022.

[13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya

Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, 2015. Software available from tensorflow.org.

[14] Hyacinth Ampadu. Dropout in deep learning: Understanding dropouts in deep learning to reduce overfitting, Apr 2021.

[15] American Diabetes Association. Hyperglycemia (high blood glucose). https://www.diabetes.org/healthy-living/medication-treatments/blood-glucose-testing-and-control/hyperglycemia.

[16] American Diabetes Association. Hypoglycemia (low blood glucose). https://www.diabetes.org/healthy-living/medication-treatments/blood-glucose-testing-and-control/hypoglycemia.

[17] American Diabetes Association. Economic costs of diabetes in the U.S. in 2017. *Diabetes Care*, 2018.

[18] American Diabetes Association. Cost-effectiveness of diabetes interventions. https://www.cdc.gov/chronicdisease/programs-impact/pop/diabetes.htm, May 2021.

[19] Baeldung. Epoch in neural networks, Feb 2021.

[20] Kiran Bhageshpur. Council post: Data is the new oil – and that's a good thing, Nov 2019.

[21] Elizabeth Boland, Teresa Monsod, Maria Delucia, Cynthia A. Brandt, Sanjay Fernando, and William V. Tamborlane. Limitations of Conventional Methods of Self-Monitoring of Blood Glucose: Lessons learned from 3 days of continuous glucose sensing in pediatric patients with type 1 diabetes. *Diabetes Care*, 24(11):1858–1862, 11 2001.

[22] Swapnil P. Borse, Abu Sufiyan Chhipa, Vipin Sharma, Devendra Pratap Singh, and Manish Nivsarkar. Management of type 2 diabetes: Current strategies, unfocussed aspects, challenges, and alternatives. *Medical Principles and Practice*, 30(2):109–121, 2020.

[23] Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: Point estimates and confidence intervals. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 451–466, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[24] Sara Brown. Mit: Machine learning, explained, Apr 2021.

[25] Jason Brownlee. A gentle introduction to transfer learning for deep learning, Sep 2019.

[26] Jason Brownlee. How to choose loss functions when training deep learning neural networks, Aug 2020.

[27] Jason Brownlee. What is machine learning?, Aug 2020.

[28] Jason Brownlee. How to choose an activation function for deep learning, Jan 2021.

[29] Daniela Bruttomesso. Toward automated insulin delivery. *New England Journal of Medicine*, 381(18):1774–1775, 2019.

[30] Vitor Cerqueira. 12 things you should know about time series, Mar 2022.

[31] Chris Chatfield. *Time-series forecasting*. CRC Press, 2000.

[32] Simon Lebech Cichosz, Jan Frystyk, Ole K. Hejlesen, Lise Tarnow, and Jesper Fleischer. A novel algorithm for prediction and detection of hypoglycemia based on continuous glucose monitoring and heart rate variability in patients with type 1 diabetes. *Journal of Diabetes Science and Technology*, 8(4):731–737, 2014. PMID: 24876412.

[33] Ivan Contreras and Josep Vehi. Artificial intelligence for diabetes management and decision support: Literature review. *Journal of Medical Internet Research*, 20(5), 2018.

[34] Stefania Cristina. Calculus in action: Neural networks, Mar 2022.

[35] Philip E. Cryer, Stephen N. Davis, and Harry Shamoon. Hypoglycemia in diabetes. *Diabetes Care*, 26(6):1902–1912, 2003.

[36] Anjali D Deshpande, Marcie Harris-Hayes, and Mario Schootman. Epidemiology of Diabetes and Diabetes-Related Complications. *Physical Therapy*, 88(11):1254–1264, 11 2008.

[37] Dexcom. Glycemic urgency assessment and alerts interface. patentus AU2015244291B2, Dexcom, 2017.

[38] Omar Diouri, Monika Cigler, Martina Vettoretti, Julia K. Mader, Pratik Choudhary, Eric Renard, and HYPO-RESOLVE Consortium. Hypoglycaemia detection and prediction techniques: A systematic review on the latest developments. *Diabetes/Metabolism Research and Reviews*, 37(7):e3449, 2021.

[39] Sebai Dorsaf. Comprehensive synthesis of the main activation functions pros and cons, Apr 2020.

[40] Christopher Duckworth, Matthew J. Guy, Anitha Kumaran, Aisling Ann O'Kane, Amid Ayobi, Adriane Chapman, Paul Marshall, and Michael Boniface. Explainable machine learning for real-time hypoglycemia and hyperglycemia prediction and personalized control recommendations. *Journal of Diabetes Science and Technology*, 0(0):19322968221103561, 0. PMID: 35695284.

[41] Chloe L. Edridge, Alison J. Dunkley, Danielle H. Bodicoat, Tanith C. Rose, Laura J. Gray, Melanie J. Davies, and Kamlesh Khunti. Prevalence and incidence of hypoglycaemia in 532,542 people with type 2 diabetes on oral therapies and insulin: A systematic review and meta-analysis of population based studies. *PLOS ONE*, 10(6):1–20, 06 2015.

[42] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[43] International Diabetes Federation. Diabetes in europe – 2021, 11 2021.

[44] Li Fei-Fei. Neural networks and lecture 4: Backpropagation. http://cs231n.stanford.edu/slides/2020/lecture_4.pdf.

[45] E. A. Feigenbaum, B. G. Buchanan, and J. Lederberg. On generality and problem solving A case study using the DENDRAL program. In Bernard Meltzer and Donald Michie, editors, *Machine Intelligence 6*, pages 165–190. Edinburgh University Press, 1971.

[46] Virginie Felizardo, Nuno M. Garcia, Nuno Pombo, and Imen Megdiche. Data-based algorithms and models using diabetics real data for blood glucose and hypoglycaemia prediction – a systematic literature review. *Artificial Intelligence in Medicine*, 118:102120, 2021.

[47] Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2206–2212. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[48] Samuele Fiorini, Chiara Martini, Davide Malpassi, Renzo Cordera, Davide Maggi, Alessandro Verri, and Annalisa Barla. Data-driven strategies for robust forecast of continuous glucose monitoring time-series. volume 2017, pages 1680–1683, 07 2017.

[49] Carlo B. Giorda, Alessandro Ozzello, Sandro Gentile, Alberto Aglialoro, Anna Chiambretti, Fabio Baccetti, Francesco M. Gentile, Giuseppe Lucisano, Antonio Nicolucci, Maria Chiara Rossi, and et al. Incidence and risk factors for severe and symptomatic hypoglycemia in type 1 diabetes. results of the hypos-1 study. *Acta Diabetologica*, 52(5):845–853, 2015.

[50] Tushar Gupta. Deep learning: Feedforward neural network, Dec 2018.

[51] N. Hex, C. Bartlett, D. Wright, M. Taylor, and D. Varley. Estimating the current and future costs of type 1 and type 2 diabetes in the uk, including direct health costs and indirect societal and productivity costs. *Diabetic Medicine*, 29(7):855–862, 2012.

[52] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[53] Mike Hoskins. All about the dexcom g6 continuous glucose monitor, Jun 2021.

[54] Angel Igareta. The million-dollar question: When to stop training your deep learning model, Jun 2021.

[55] Shruti Jadon. Introduction to different activation functions for deep learning. *Medium, Augmenting Humanity*, 16, 2018.

[56] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[57] Dianna Magliano and et al. IDF Diabetes Atlas 2021, 2021.

[58] Cindy Marling and Razvan C. Bunescu. The ohiot1dm dataset for blood glucose level prediction: Update 2020. In Kerstin Bach, Razvan C. Bunescu, Cindy Marling, and Nirmalie Wiratunga, editors, *Proceedings of the 5th International Workshop on Knowledge Discovery in Healthcare Data co-located with 24th European Conference on Artificial Intelligence, KDH@ECAI 2020, Santiago de Compostela, Spain & Virtually, August 29-30, 2020*, volume 2675 of *CEUR Workshop Proceedings*, pages 71–74. CEUR-WS.org, 2020.

[59] Anne B. Martin, Micah Hartman, Benjamin Washington, Aaron Catlin, and The National Health Expenditure Accounts Team . National health care spending in 2017: Growth slows to post–great recession rates; share of gdp stabilizes. *Health Affairs*, 38(1):10.1377/hlthaff.2018.05085, 2019. PMID: 30521399.

[60] Michael Mayo, Lynne Chepulis, and Ryan G. Paul. Glycemic-aware metrics and oversampling techniques for predicting blood glucose levels using machine learning. *PLOS ONE*, 14(12):1–19, 12 2019.

[61] Rory J. McCrimmon and Robert S. Sherwin. Hypoglycemia in Type 1 Diabetes. *Diabetes*, 59(10):2333–2339, 10 2010.

[62] Medtronic. Method , system and computer program product for CGM - based prevention of hypoglycemia via hypoglycemia risk assessment and smooth reduction insulin delivery. patentus US10842419B2, Medtronic, 2020.

[63] Hrushikesh Mhaskar, Sergei Pereverzyev, and Maria Van der Walt. A deep learning approach to diabetic blood glucose prediction. *Frontiers in Applied Mathematics and Statistics*, 3, 07 2017.

[64] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

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

[66] Christopher Olah. Neural networks, manifolds, and topology, Apr 2014.

[67] Christopher Olah. Understanding LSTM networks, Aug 2015.

[68] Motunrayo Olugbenga. Balanced accuracy: When should you use it?, Jul 2022.

[69] Artem Oppermann. Regularization in deep learning, l1, l2, and dropout. https://towardsdatascience.com/regularization-in-deep-learning-l1-l2-and-dropout-377e75acc036, Aug 2020.

[70] Silvia Oviedo, Josep Vehí, Remei Calm, and Joaquim Armengol. A review of personalized blood glucose prediction strategies for t1dm patients. *International Journal for Numerical Methods in Biomedical Engineering*, 33(6):e2833, 2017. e2833 CNM-Jul-16-0155.R1.

[71] Antonio Parmezan, Vinícius Alves de Souza, and Gustavo Batista. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences*, 01 2019.

[72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[73] Christo Petrov. 25+ impressive big data statistics for 2022, Aug 2022.

[74] Timothée Proix, Wilson Truccolo, Marc G Leguia, Thomas K Tcheng, David King-Stephens, Vikram R Rao, and Maxime O Baud. Forecasting seizure risk in adults with focal epilepsy: a development and validation study. *The Lancet. Neurology*, 20(2):127—135, February 2021.

[75] Tran Minh Quan, Takuyoshi Doike, Cong Dang Bui, Kenya Hayashi, Shigeki Arata, Atsuki Kobayashi, Md. Zahidul Islam, and Kiichi Niitsu. Ai-based edge-intelligent hypoglycemia prediction system using alternate learning and inference method for blood glucose level data with low-periodicity. *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 201–206, 2019.

[76] Ignacio Rodríguez-Rodríguez, Ioannis Chatzigiannakis, José-Víctor Rodríguez, Marianna Maranghi, Michele Gentili, and Miguel-Ángel Zamora-Izquierdo. Utility of big data in predicting short-term blood glucose levels in type 1 diabetes mellitus through machine learning techniques. *Sensors*, 19(20), 2019.

[77] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[78] Kristina I. Rother. Diabetes treatment — bridging the divide. *New England Journal of Medicine*, 356(15):1499–1501, 2007.

[79] Stuart Russell and Peter Norvig. *Artificial Intelligence: A modern approach*. Pearson, 3rd edition, 2020.

[80] Fahreddin Sadıkoglu and Cemal Kavalcıoğlu. Filtering continuous glucose monitoring signal using savitzky-golay filter and simple multivariate thresholding. *Procedia Computer Science*, 102:342–350, 2016. 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria.

[81] R.C. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Earlbaum Assoc, 1977.

[82] R.C. Schank and C.K. Riesbeck. *Inside Computer Understanding Five Programs Plus Miniatures*. Lawrence Erlbaum Associates, Hillsdale, N.J., 1981.

[83] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[84] PhD Sigmundo Preissler Jr. Seasonality in python: Additive or multiplicative model?, Nov 2018.

[85] Andrew Smith and Chelsea Harris. Type 1 diabetes: Management strategies. *American Family Physician*, 98:154–162, 08 2018.

[86] Adrian Tam. Training-validation-test split and cross-validation done right, Sep 2021.

[87] Eve Van Cauter, Kenneth S. Polonsky, and Andre J. Scheen. Roles of Circadian Rhythmicity and Sleep in Human Glucose Regulation*. *Endocrine Reviews*, 18(5):716–738, 10 1997.

[88] William P. T. M. van Doorn, Yuri D. Foreman, Nicolaas C. Schaper, Hans H. C. M. Savelberg, Annemarie Koster, Carla J. H. van der Kallen, Anke Wesselius, Miranda T. Schram, Ronald M. A. Henry, Pieter C. Dagnelie, Bastiaan E. de Galan, Otto Bekers, Coen D. A. Stehouwer, Steven J. R. Meex, and Martijn C. G. J. Brouwers. Machine learning-based glucose prediction with use of continuous glucose and physical activity monitoring data: The Maastricht Study. *PLOS ONE*, 16(6):1–17, 06 2021.

[89] Josep Vehí, Iván Contreras, Silvia Oviedo, Lyvia Biagi, and Arthur Bertachi. Prediction and prevention of hypoglycaemic events in type-1 diabetic patients using machine learning. *Health Informatics Journal*, 26(1):703–718, 2020. PMID: 31195880.

[90] Yugesh Verma. Complete guide to bidirectional lstm (with python codes), Jul 2021.

[91] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[92] Ashenafi Zebene Woldaregay, Eirik Årsand, Ståle Walderhaug, David Albers, Lena Mamykina, Taxiarchis Botsis, and Gunnar Hartvigsen. Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes. *Artificial Intelligence in Medicine*, 98:109–134, 2019.

[93] Nicola N. Zammitt and Brian M. Frier. Hypoglycemia in type 2 diabetes. *Diabetes Care*, 28(12):2948–2961, 2005.

[94] Nicola N. Zammitt and Brian M. Frier. Hypoglycemia in Type 2 Diabetes: Pathophysiology, frequency, and effects of different treatment modalities. *Diabetes Care*, 28(12):2948–2961, 12 2005.

[95] 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(3):308–324, 2020.

[96] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Deep learning for diabetes: A systematic review. *IEEE Journal of Biomedical and Health Informatics*, 25:2744–2757, 2021.

[97] Taiyu Zhu, Chukwuma Uduku, Kezhi Li, Pau Herrero, Nick Oliver, and Pantelis Georgiou. Enhancing self-management in type 1 diabetes with wearables and deep learning. *npj Digital Medicine*, 5(1), 2022.

[98] Taiyu Zhu, X Yao, Kenneth Li, Pau Herrero, and P Georgiou. Blood glucose prediction for type 1 diabetes using generative adversarial networks. *CEUR Workshop Proceedings*, 2675, 01 2020.