

# Machine learning-assisted continuous glucose and ketone monitoring for diabetic ketoacidosis

by

Subhamoy Biswas

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Master of Applied Science  
in  
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Subhamoy Biswas 2024

## **Author's Declaration**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

Type 1 diabetes has affected millions of people worldwide, and rigorous tracking of blood glucose levels is critical for providing care and avoiding severe complications like hyperglycemia and diabetic ketoacidosis. Continuous glucose monitoring (CGM) devices have emerged as an effective tool to detect glucose levels from interstitial fluid (ISF) instead of blood and offer real-time treatment. ISF usually serves as a rich source of biomarkers, enabling minimally-invasive detection for continuous health monitoring through ISF-based sensors like CGM. Various machine learning works in the past have used these CGM measurements to forecast glucose levels and predict events like hypoglycemia or any potential risks. However, despite their effective performances, most have focused on short-term predictions, neglecting the importance of long-term forecasting for better insulin therapy. In the first half of this thesis, we present an encoder-decoder architecture for long-term forecasting of future BG levels that expands the forecasting horizon from conventional 1 hour up to 3 hours. This work has the potential to improve the precision of state-of-the-art insulin delivery platforms for effective diabetes management.

In addition to this, despite the advantages offered by ISF-based sensors in general, they encounter a significant challenge related to the sensing delay in the transferring of target analytes like glucose from blood to ISF. Particularly, this delay can vary significantly from subject to subject, and if not estimated properly, can impact the accuracy of the sensor measurements. Prior machine learning frameworks for continuous measurement of glucose from ISF have not adequately accounted for this existing delay between blood and ISF. Therefore, in the second half of this thesis, we investigate and quantify sensing delays in the transfer of glucose and ketone bodies from blood to ISF using decision-tree-based algorithms by considering a case study of diabetic rats that emulate the conditions of diabetic ketoacidosis. Accounting for this delay in the measurement process can eventually improve the accuracy of such sensors and offer a more personalized response during continuous monitoring of glucose and ketone bodies for better insulin dosing.

## Acknowledgments

I would like to extend my sincere gratitude to my supervisor, Dr. Mahla Poudineh, for introducing me to her research group in the IDEATION Lab at the University of Waterloo. Her constant enthusiasm, faith, and patience have always been a source of motivation during my graduate research. She has always supported me in ensuring proper directions for my work, evaluating and reviewing my results, and writing manuscripts.

I would like to thank Dr. Ali Etemad (Queen's University) and Ahmad Beirami (Google) for their guidance in implementing a multitude of machine learning algorithms during the course of my graduate studies. I also want to thank my lab members Irfani Ausri, Sadegh Sadeghzadeh, Peyman GhavamiNejad, and Pouyan Keshavarz Motamed for their notable suggestions, advice, and review of my work on continuous glucose and ketone monitoring. Lastly, I would like to thank the members of my thesis review committee, Dr. Mohammad Kohandel and Dr. George Shaker, for reviewing this thesis and providing their valuable suggestions.

## **Dedication**

I would like to dedicate this thesis to my family, who has always been the cornerstone of support and love in every second of my life.

# Table of Contents

<b>Author's Declaration</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Dedication</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Abbreviations</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Thesis structure at a glance . . . . .	2
<b>2 Background</b>	<b>4</b>
2.1 Type 1 diabetes and diabetic ketoacidosis: a brief review . . . . .	4
2.2 Continuous glucose monitoring: a review of data-driven approaches . . . . .	6

2.2.1	Classical statistical algorithms . . . . .	8
2.2.2	Artificial neural networks . . . . .	9
2.2.3	Support vector machines . . . . .	11
2.2.4	Recurrent neural networks . . . . .	11
2.2.5	Decision-tree based architectures . . . . .	12
2.2.6	Sequence-to-sequence frameworks . . . . .	13
2.3	Continuous ketone monitoring: significance and challenges . . . . .	15
<b>3</b>	<b>Long-term glucose forecasting for type 1 diabetes</b>	<b>17</b>
3.1	A review of time-series forecasting models for glucose monitoring: why are long-term predictions important? . . . . .	17
3.2	RNN and LSTM: a brief introduction . . . . .	23
3.3	Understanding sequence-to-sequence modeling and attention mechanism . . . . .	28
3.4	Problem formulation and data preprocessing . . . . .	32
3.4.1	Mathematical background . . . . .	32
3.4.2	Data preprocessing . . . . .	34
3.5	Experiments . . . . .	36
3.5.1	Experimental settings and model architecture . . . . .	36
3.5.2	Results . . . . .	39
3.6	A summary of challenges addressed . . . . .	41
<b>4</b>	<b>Quantification of personalized sensing delays for a novel CGM-CKM device</b>	<b>42</b>
4.1	Sensing delays in interstitial fluid-based biosensing . . . . .	42
4.2	Decision tree-based algorithms: a brief introduction . . . . .	44
4.2.1	Decision trees and random forests . . . . .	44

4.2.2	Boosting algorithms . . . . .	47
4.3	Mathematical background . . . . .	51
4.3.1	Problem formulation . . . . .	51
4.3.2	Quantifying time-lags between blood and ISF . . . . .	51
4.4	Experiments . . . . .	53
4.4.1	Data collection . . . . .	53
4.4.2	Data preprocessing and input features . . . . .	55
4.4.3	Performance metrics . . . . .	55
4.5	Results . . . . .	57
4.6	A summary of challenges addressed . . . . .	61
<b>5</b>	<b>Final remarks</b>	<b>66</b>
	<b>References</b>	<b>68</b>

# List of Figures

2.1	A schematic of using a CGM platform to interpret blood glucose levels . . .	7
2.2	A simple illustration of the autoregressive model . . . . .	9
2.3	A simple illustration of an artificial neural network architecture with 2 hidden layers . . . . .	10
2.4	A simple illustration of support vector regression . . . . .	12
3.1	A simple illustration of RNN with input $x$ forwarded to a hidden state $h$ , resulting in an output $o$ . . . . .	24
3.2	A simple illustration of LSTM with cell state $c^t$ , hidden state $h^t$ , input $x^t$ , and output $o^t$ . . . . .	27
3.3	An illustration of a sequence-to-sequence machine learning framework with its encoder and decoder each functionalized with a simple recurrent neural network layer and with an input vector of length $n$ and an output vector of length $m$ . . . . .	30
3.4	A summary of the data preprocessing steps implemented before learning the glucose forecasting framework . . . . .	35
3.5	An illustration of our long-term glucose forecasting architecture . . . . .	38
4.1	A simple illustration of a decision tree . . . . .	45
4.2	A simple illustration of the gradient boosting algorithm . . . . .	49

4.3	(a): A sample CKM output - black and blue curves indicate preoxidation and detection profiles; (b) the corresponding blood ketone graph; (c): a sample CGM current-voltage loop; (d): the corresponding glucose current	54
4.4	A sample plot for outlier detection . . . . .	56
4.5	An illustration of our framework for evaluating sensing delays and using them to provide personalized responses for our CGM-CKM device . . . . .	63
4.6	Sample plots for variation of metrics with delays for (a) CKM and, (b) CGM	64
4.7	Overall correlation for CKM (left) and CGM (right) . . . . .	65

# List of Tables

3.1	A review of time series models for glucose forecasting . . . . .	19
3.2	Short-term forecasting metrics of our model versus the baselines considered in this study . . . . .	40
3.3	Long-term forecasting metrics of our model versus the baselines considered in this study . . . . .	40
4.1	A summary of the best possible individual delays $\tau_k$ and their corresponding metrics for CKM . . . . .	59
4.2	A summary of the best possible individual delays $\tau_k$ and their corresponding metrics for CGM . . . . .	59
4.3	A summary of the overall <i>MARD</i> and <i>MAD</i> for all the algorithms . . . . .	60
4.4	A literature summary of the <i>MARD</i> performances of some widely-used CGM systems implemented on human subjects . . . . .	60
4.5	Comparison of the overall <i>MAD</i> and <i>MARD</i> obtained while implementing our ketone sensing framework with a state-of-the-art study [6] . . . . .	61

# List of Abbreviations

- AID** automated insulin delivery [14](#)
- ANN** artificial neural networks [9](#), [10](#), [23](#)
- ARIMA** autoregressive integrated moving average [8](#), [18](#)
- BHB** beta-hydroxy butyrate [15](#), [16](#)
- BPTT** back-propagation through time [25](#)
- CatBoost** Categorical boosting [50](#)
- CGM** continuous glucose monitoring [5–8](#), [13–15](#), [32](#)
- CKM** continuous ketone monitoring [15](#)
- DKA** diabetic ketoacidosis [5](#), [6](#), [15](#), [44](#)
- GRU** gated recurrent units [29](#)
- IDF** International Diabetes Federation [4](#)
- ISF** interstitial fluid [5](#), [42](#)
- LSTM** Long short-term memory networks [11](#), [26](#)
- MAD** mean absolute difference [55](#)

**MAPE** mean absolute percentage error 37

**MARD** mean absolute relative difference 56

**MLP** multi-layer perceptron 39, 40

**RF** random forest 39, 40

**RMSE** root mean squared error 18, 37

**RNN** recurrent neural networks 11, 18, 23

**Seq2Seq** Sequence-to-sequence 28

**SVR** support vector regression 11

**T1D** Type 1 Diabetes 4–6, 13, 15, 32

**XGBoost** extreme gradient boosting 48

# Chapter 1

## Introduction

### 1.1 Motivation

Millions of people are affected by type 1 diabetes globally, and patients often require rigorous blood glucose tracking to prevent further complications like hyperglycemia and diabetic ketoacidosis. Continuous glucose monitoring (CGM) devices based on interstitial fluid-based sensing and leveraged with machine learning algorithms have emerged as an effective technique to provide real-time treatment, forecast glucose levels for a reasonable amount of time, and predict any possible future risks. Previous efforts, however, have mainly focused on short-term predictions. A long-term forecasting horizon allows better management and precision of insulin therapies for concerned patients. In addition to this, ISF-based sensors, in general, face challenges with sensing delays in transferring bioanalytes like glucose and ketone from blood to ISF. These delays significantly vary between subjects and affect sensor accuracy. Previous frameworks dealing with glucose and ketone sensing have not adequately addressed this delay. Hence, the primary motivation of this thesis is two-fold - (a) to explore long term forecasting of glucose by expanding the prediction horizon from the conventional 1 hour to 3 hours to improve insulin dosing, and (b) to investigate and quantify sensing delays during continuous glucose and ketone monitoring and how they can be used to provide personalized responses for any subject and ensure a better overall management of diabetic ketoacidosis.

## 1.2 Thesis structure at a glance

The structure of this thesis is summarized as follows:

1. Chapter 2 gives a brief introduction to type 1 diabetes and its complication, diabetic ketoacidosis. It explores how continuous glucose monitoring has emerged as a popular approach to mitigate unwanted risks like hypoglycemia and improve existing insulin therapies. It further explores how a multitude of machine learning algorithms have impacted the performance of continuous glucose monitoring devices. Finally, it touches upon the relatively newer avenue of continuous ketone monitoring and how it is beneficial for tracking the conditions of diabetic ketoacidosis patients. Above all, this chapter focuses on explaining the importance of continuous monitoring of analytes and leveraging them with the power of artificial intelligence to improve existing remedies for type 1 diabetes and diabetic ketoacidosis.
2. Chapter 3 is focused on a preliminary work on a long-term forecasting architecture for predicting glucose levels up to 3 hours. The chapter begins with a review of how most of the prior machine learning works have focused on short-term glucose forecasting, ranging from 15 minutes up to 1 hour, and why long-term forecasting beyond 1 hour is essential. It further gives a brief introduction to recurrent neural networks and their variant, long short-term memory networks, and also attention mechanisms and sequence-to-sequence models that were used to develop the framework we used here. Eventually, the chapter describes the mathematical formulation of the problem, the details of the architecture used in this thesis for long-term forecasting, and how it performs when compared with baseline models and a state-of-the-art paper. Above all, this chapter sheds light on how a time series forecasting framework for predicting blood glucose typically works.
3. Chapter 4 is focused on a more specific problem of quantifying sensing delays or time lags observed in interstitial fluid-based sensing approaches like continuous glucose and ketone monitoring. It begins with a brief discussion on the importance of interstitial fluid and the associated advantages and disadvantages of using it as the medium of detection compared to invasive blood-based techniques, as well as how

sensing delays during transmission of analytes from blood to interstitial fluid pose a burden on the precision and accuracy of such sensors. It further describes a few decision tree-based algorithms, like boosting, that were used later in the chapter. Following this, it describes the mathematical formulation for quantifying such time lags and using them to adjust and provide accurate glucose and ketone responses that are personalized for every subject. It further evaluates the scheme using three different boosting algorithms and compares their results with other state-of-the-art approaches. Above all, this chapter delves deep into the issue of time lags in interstitial fluid-based sensors and how computational approaches help evaluate them and improve the sensors' performances. It also sheds light on the application of machine learning algorithms for the first time in the case of continuous ketone monitoring devices to report personalized (subject-specific) ketone responses in exact mM units, which allows for a more reasonable comparison of interstitial fluid-based ketone outputs with blood-based references to measure their accuracy.

# Chapter 2

## Background

### 2.1 Type 1 diabetes and diabetic ketoacidosis: a brief review

Type 1 Diabetes (T1D) is primarily a chronic autoimmune disease where the immune system mistakenly attacks and destroys insulin-producing beta cells in the pancreas. The incidence of T1D is increasing globally, particularly among children. According to the International Diabetes Federation (IDF), currently around 8.7 million people worldwide are estimated to have T1D with approximately 3.9 million lives lost due to T1D so far, accounting for around 32 years of healthy life lost on average per person [30]. The exact trigger for the autoimmune response behind T1D is not fully understood, but genetic predisposition like family history and environmental factors, such as viral infections, dietary factors and early childhood nutrition, are believed to play significant roles and increase the risk of T1D onset [49][64]. Insufficient production of insulin leads to high blood glucose levels (hyperglycemia) and persistent hyperglycemic conditions cause multiple symptoms such as increased thirst, frequent urination, fatigue, and weight loss [46]. Long-term effects of uncontrolled T1D include cardiovascular disease [76], neuropathy [83], retinopathy and nephropathy [100]. The treatment for T1D requires lifelong management, including intensive insulin therapy, blood glucose monitoring at regular intervals, and major lifestyle modifications [44].

One acute complication of T1D is [diabetic ketoacidosis \(DKA\)](#). DKA occurs when there is an abnormal deficiency of insulin, leading to a situation where glucose cannot enter cells to be used for energy. Without adequate insulin, the body instead breaks down fat for energy, producing acidic ketones as a byproduct [27]. Besides this, skipping insulin injections, conditions such as urinary tract infections or respiratory infections as well as other factors like stress, any past trauma or surgery, and certain medications can increase the body's need for insulin and contribute to the development of DKA. The effects generally include excessive electrolyte imbalances, metabolic acidosis and decreased blood volume, leading to a potential shock [29].

The management of T1D and its potential complications has evolved greatly, especially with the widespread adoption of [continuous glucose monitoring \(CGM\)](#) devices. Regular glucose monitoring is crucial to provide clinical care during any form of diabetes. While traditional blood glucose monitoring (BGM) systems have improved in accuracy and usability, they are only known to provide static blood glucose information. In contrast, CGM technology can also track glucose changes over time, fluctuations, trends, and several other patterns, allowing for a more thorough analysis of glycemic trends to effectively guide insulin-based therapeutic approaches [52]. Over many years, CGM platforms have demonstrated their accuracy in achieving glycemic targets, reducing hypoglycemia, which is characterized by abnormally low blood glucose concentrations (glucose level  $< 72\text{mg/dL}$ ) and other risks, and enhancing the overall quality of life of the patients [12, 67].

A CGM platform is essentially a wearable device that continuously and automatically records glucose levels at regular intervals, ranging from every 5 to even 15 minutes, from the [interstitial fluid \(ISF\)](#), a fluid that surrounds the body's cells. CGMs primarily involve enzymatic reactions that occur within the sensor. When glucose molecules in the ISF come into contact with the enzyme within the sensor, they undergo a reduction-oxidation (redox) reaction that releases electrons that are transferred to an electrode within the sensor, generating an electric current, such that the magnitude of the current is directly proportional to the concentration of glucose in the ISF [19]. A standard CGM platform typically comprises three main components - (a) the wearable sensor that is attached to the body, usually on the abdomen or arm, and contains an electrode that penetrates the skin to measure glucose levels in the ISF, (b) a transmitter, which wirelessly sends the glucose

readings to a nearby receiver, and (c) a receiver or reader, which may be a dedicated device or a smartphone app, that receives the data transmitted by the sensor, displays them in real-time for further visualization and provides information on current glucose levels, trends, and alerts for future risks like hyperglycemia [51]. Healthcare practitioners and patients, with the help of artificial intelligence (AI)-guided decision support frameworks [88], gain insights from these information to make informed decisions regarding adjustments to medication as well as lifestyle, diet, or exercise. Figure 2.1 gives an illustration of interpreting glucose information using CGM biosensors.

As for the treatment and management of T1D and its complications like DKA, insulin infusion is a commonly used technique used to reduce the rapidly increasing levels of blood glucose and suppress the resulting blood ketone production. Modern insulin analogs range from rapid-acting dosages, usually taken after meals, that are effective very quickly but can only last a few hours, to long-acting insulin that requires a longer duration of time to start working but can provide a baseline concentration of insulin to control blood glucose levels for almost up to an entire day [3]. While these traditional insulin injections can meet the basal and bolus insulin requirements of prospective patients, state-of-the-art methods like insulin pump therapy offers a more dynamic approach by delivering insulin continuously and more effectively, mimicking the body’s natural insulin secretion pattern [47]. However, the precision of insulin pump therapies heavily depends on accurate insulin dosing. In this context, CGM devices with their ability to provide real-time continuous glucose measurements, and predictive algorithms to forecast glucose levels and automate insulin delivery, both are indispensable for precise insulin dosing and consequently, improving the performance and convenience of insulin therapy [95].

## 2.2 Continuous glucose monitoring: a review of data-driven approaches

In this thesis, the primary focus will be on leveraging the power of predictive analytics into the CGM information obtained from part (c) as described in Section 2.1, that is, the receiver that accepts the glucose information from the sensor via a transmitter. Over the

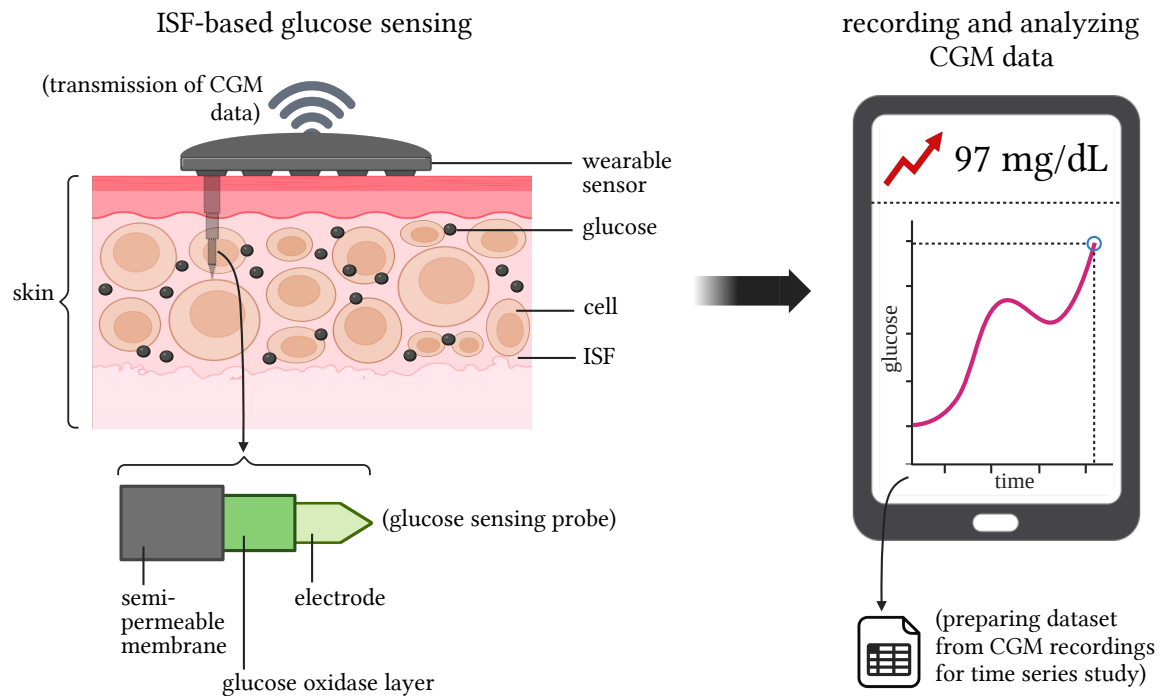


Figure 2.1: A schematic of using a CGM platform to interpret blood glucose levels

past decade, there has been multiple approaches that used AI-assisted decision support systems to analyze a patient’s glucose information [94].

CGM devices have undergone a transformative evolution through the integration of machine learning (ML) algorithms, revolutionizing their functionality and efficacy in managing diabetes. By leveraging their capabilities of analyzing extensive datasets encompassing glucose levels, insulin dosages, meal times, physical activity, and other pertinent factors and recognizing complex patterns and anomalies in glucose fluctuation, ML algorithms have not only increased the accuracy of state-of-the-art CGM devices, but also paved ways for providing personalized insights and predictive alerts tailored to individual users, facilitating proactive management of blood glucose levels and reducing the risk of hypo- or hyperglycemic events, false alarms and user anxiety. Furthermore, adaptive ML models continuously learn from user feedback, refining their predictions and recommenda-

tions over time to further optimize accuracy and personalization. The integration of ML with closed-loop systems represents another milestone, enabling real-time adjustment of insulin delivery based on CGM data, thereby achieving tighter glycemic control. Overall, ML algorithms have not only improved the performance and reliability of CGM devices but has also empowered individuals with diabetes to take proactive control of their health, leading to enhanced quality of life and better management outcomes. The purpose for implementing these algorithms can be broadly classified into two primary categories - (1) blood glucose prediction, and (2) optimal insulin dosage selection to counter any abnormalities in glucose levels. Using CGM biosensors' data for glucose prediction not only explores the classification of glycemic control profiles and alerts prospective patients of any risk of events like post-prandial hypoglycemia (glucose < 72mg/dl) [17, 70, 92], but also includes time series models and other relevant algorithms that forecast glucose levels for the next 15 minutes up to 1 hour as their prediction horizon. In the following subsections, we discuss a few commonly used algorithms for time series forecasting of glucose levels.

### 2.2.1 Classical statistical algorithms

The most primitive approaches to develop frameworks for glucose prediction involve the use of classical statistical algorithms like autoregression [81]. An autoregressive model, as shown in Figure 2.2, is essentially a statistical method used for representing time-varying processes or data, where the value of the random variable under study, at any given time instant, is predicted based on its previous values in the form of a stochastic difference equation. Mathematically, an autoregressive model of order  $p$  is usually denoted as  $AR(p)$  that expresses the current value of the variable as a linear combination of its past  $p$  values, along with a stochastic error term [9]. Methods such as least squares [42] or maximum likelihood estimation [80] are further used to determine the optimal parameters of the autoregressive model  $AR(p)$ . One primary characteristic of an autoregressive model is that it considers a stationary or constant relationship between the variable's current and past values. However, due to the inherent nonlinearity and fluctuations in a patient's day-to-day blood glucose levels, classical approaches like the autoregressive model, which assumes linearity, struggle to effectively model the dynamics in glucose measurements. A more improvised version, called as the [autoregressive integrated moving average \(ARIMA\)](#)

model, modifies the autoregressive model  $AR(p)$  with a differencing term to handle non-stationary time series and a moving average component to perform autocorrelation with past error terms for better prediction. A recent study in 2024 [53] used the ARIMA model for time series-based non-invasive and real-time tracking of glucose and cholesterol levels in diabetic patients.

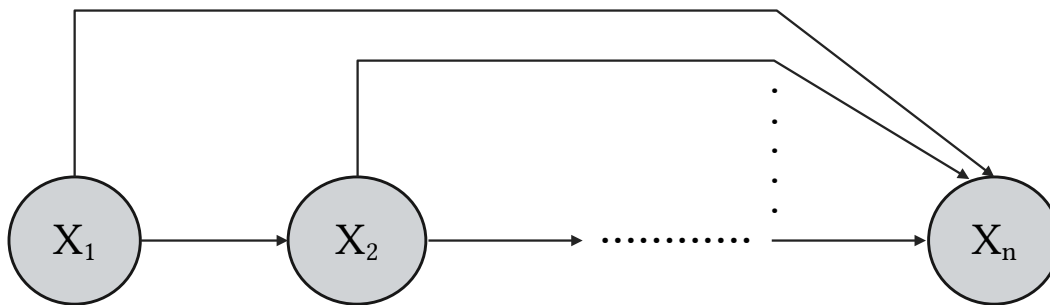


Figure 2.2: A simple illustration of the autoregressive model

### 2.2.2 Artificial neural networks

Other machine learning techniques like neural networks, on the flip side, excel at adaptively learning and capturing the complex, nonlinear, time-varying patterns in glucose metabolism. The most fundamental feedforward **artificial neural networks (ANN)** are inspired from the structure and functioning of the biological neural networks in human body. As in Figure 2.3, they primarily consist of interconnected nodes (neurons) organized in layers, where each node performs a weighted sum of inputs followed by applying an activation function to introduce non-linearity, allowing the network to model complex patterns in the data. For optimization, the weights of the connections between neurons are updated based on the error of the output compared to the ground truth, using algorithms like backpropagation. Over the years, ANNs and other forms of neural networks have proven to be useful for handling datasets from larger and more diverse diabetic patient cohorts, and can manage extensive historical information alongside glucose recordings like insulin

dosages, meal timings and parameters associated with day-to-day physical activities, to improve prediction accuracy. [54]. A recent study [5] used an ANN-based prediction model that considered glucose measurements as a single input and explored time-domain features as additional attributes with the primary aim of reducing the number of inputs required without affecting the model's forecasting accuracy.

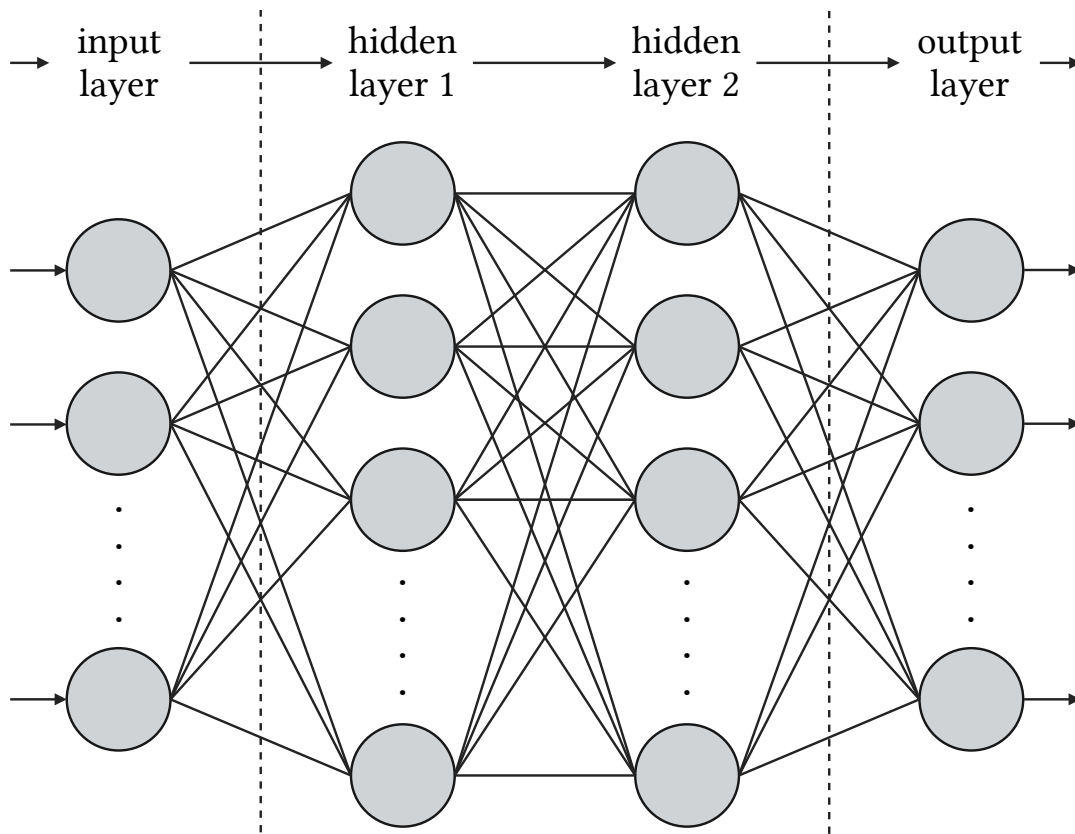


Figure 2.3: A simple illustration of an artificial neural network architecture with 2 hidden layers

### 2.2.3 Support vector machines

Support vector regression is another machine learning technique that has been used for multiple studies focusing on time-series based glucose prediction. It primarily works by learning a mapping that approximates the output with a tolerance for error, which is defined by a margin around the predicted mapping. As shown in Figure 2.4, it uses a set of data points known as support vectors lying outside the margin or on its boundary, that determine the position and orientation of the predicted regression line or the hyperplane (a higher dimensional space, usually involves dimensions  $> 3$ ) while minimizing the model's error. It further employs a loss function, typically known as the epsilon-insensitive loss function, that ignores any error within the margin, thereby enabling the predicted function to maintain a flexibility between its accuracy and its ability to generalize to unseen test data. To further understand nonlinear patterns, it uses the help of nonlinear kernel functions. A comparative study [35] implemented [support vector regression \(SVR\)](#) to address the challenge of subcutaneous glucose predictions by comparing the forecasting performances of two different types of conditions - one with glucose recordings as a single input and the other, where the model is enriched with additional sources of historical information like past insulin dosages.

### 2.2.4 Recurrent neural networks

Some of the other machine learning frameworks applied for blood glucose forecasting include [recurrent neural networks \(RNN\)](#) . RNNs are an adaptation of the conventional feedforward ANNs that focus on retaining memory of previous inputs through internal states for better understanding of long-term temporal patterns and dependencies in sequential data like glucose recordings from CGM devices. An advantage offered by RNNs over ANNs is they can additionally handle input sequences of varying lengths. [Long short-term memory networks \(LSTM\)](#), an improvement over RNNs, further use additional states that allow researchers to mitigate problems like vanishing gradients and limited memory in conventional RNNs, while dealing with forecasting problems based on time series datasets. A state-of-the-art article [60] used a framework based on LSTM networks for predicting blood glucose levels and further provided a measure of certainty of its forecast by learn-

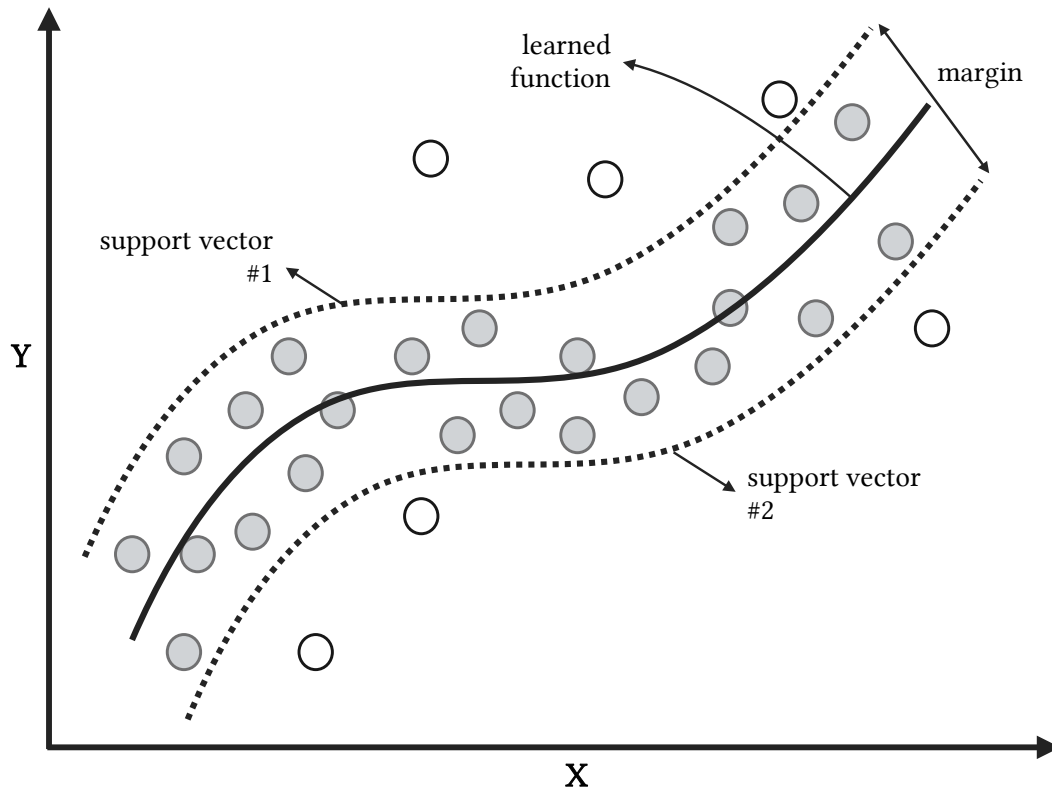


Figure 2.4: A simple illustration of support vector regression

ing and parameterizing a univariate Gaussian distribution over the output. A detailed description of these models is available in Section 3.2.

### 2.2.5 Decision-tree based architectures

Decision tree-based frameworks have been widely used for time series-based prediction of blood glucose levels for both classification and regression tasks. A decision tree is essentially a single, standalone graph-based model that splits the input data based on feature values to create a tree-like structure of decisions, aiming to maximize purity at each node using criteria like Gini impurity, entropy, or mean squared error. While decision trees are simple

to understand and interpret, they are often prone to overfitting, especially with complex datasets, as they tend to capture noise along with the underlying data patterns. Random forest [15] is a popular decision tree-based model that is primarily an ensemble learning method that builds multiple decision trees using random subsets of the input training data and features. This approach focusing on parallel tree training and aggregation is known as bootstrapping which ensures that each tree is different and captures various aspects of the data. The predictions from all the trees are then aggregated by majority voting for classification or averaging for regression tasks like time series forecasting, allowing it to be more robust and accurate than a single decision tree and mitigating the overfitting problem inherent in individual decision trees. Boosting algorithms, another variant of decision tree-based frameworks, develop trees sequentially and in an iterative fashion, where each new tree is trained to correct the errors made by the previous ones. In boosting, each tree focuses more on the hard-to-predict examples by adjusting the weights of training instances based on the errors of the previous trees. This sequential approach thereby allows the model to significantly reduce bias, leading to higher accuracy. A study from 2015 [36] focused on developing a framework based on random forests for short-term subcutaneous glucose prediction. It identified and ranked the best set of features, or predictors, in the order of their importance, from various personal, medical and lifestyle self-monitoring data sources like the time of the day, plasma insulin, food intake, physical activity, etc. of T1D patients, that could be used alongside CGM measurements for better forecasting of glucose levels. Many approaches, over the years, have used additional inputs in their models in addition to CGM recordings, such as carbohydrates intake [48], day-to-day physical activities [89], and one-hot encoding vectors labeling every unique patient in the CGM data used [7] to allow for a more personalized feedback to the patient. Sections 4.2 delves into further details on decision tree-based models.

### 2.2.6 Sequence-to-sequence frameworks

Sequence-to-sequence (seq2seq) models have been instrumental in the machine learning domain in the recent years, particularly for challenges related to time-series forecasting and natural language processing. Seq2seq models primarily consist of two components - an encoder and a decoder. The encoder processes an input sequence and summarizes it into

a fixed-size context vector. This vector is then passed to the decoder, which generates the output sequence in a step-by-step fashion. To construct their encoders and decoders, traditional seq2seq models often rely on RNNs or their variants like LSTM networks, which can handle sequences of variable lengths by maintaining a hidden state that allow information to persist. Further details on seq2seq models are discussed in Section 3.3. Transformer models, first introduced in 2017 [91] for applications in natural language processing, revolutionized seq2seq modeling by using an attention mechanism that allows the model to weigh different parts of the input sequence simultaneously, rather than sequentially. This approach allows the model to focus on the most relevant and important information of the sequence only and further enables parallel processing, which significantly speeds up model training and improves the understanding of long-range dependencies in the data. With respect to time series problems, multiple variants of Transformers like informer [107], temporal fusion transformer [56] and autoformer [101] have been developed in the recent years. A state-of-the-art study [77] explored the use of transformer architectures in forecasting blood glucose and how they offer improved accuracy and precision over conventional machine learning techniques.

On the other hand, for optimal insulin dosing, the most state-of-the-art method of treatment is the use of [automated insulin delivery \(AID\)](#) devices, often nicknamed as artificial pancreas systems. Broadly speaking, these platforms consist of - a continuous glucose monitor, a subcutaneous insulin pump, and a feedback control algorithm that links the information or predictions obtained from the [CGM](#) to decide the amount of insulin to be administered to the patient at any given time instant [43]. A majority of the artificial pancreas systems developed over the years are "hybrid" in nature, meaning the algorithm interprets the glucose predictions from the [CGM](#) data and automatically predicts the patient's basal insulin requirement, but still needs the user to regularly check the meal carbohydrate and bolus insulin intake [2]. The state-of-the-art focus is, therefore, to develop fully-automated artificial pancreas device frameworks that do not involve any form of human intervention. Across multiple studies based on improving the speed and accuracy of automatically adjusting insulin dosages, the algorithms used range from classical mathematical models using standard equations in the form of insulin bolus calculators [38], to advanced machine learning techniques like reinforcement learning [13, 87], Bayesian

networks [39] and convolutional neural networks [21].

## 2.3 Continuous ketone monitoring: significance and challenges

Despite several advances in T1D management, complications like DKA still pose a heavy toll in terms of morbidity and mortality in concerning patients. The abrupt deficiency in insulin during DKA often results in a surge in the levels of counter-regulatory hormones like glucagon, cortisol, and epinephrine [79]. Consequently, this hormonal imbalance causes a chain of events, starting from severe hyperglycemia to generating significant levels of ketone bodies through the breakdown of fats, resulting in an overall metabolic acidosis inside the body [29]. Hence, the simultaneous control of both glucose and ketone levels is essential in the prevention, detection, and management of DKA. Continuous monitoring of ketone concentrations can help in not only determining the early of DKA but also detecting any interruption or failure in the patient’s insulin therapy resulting in subsequent hyperglycemic conditions.

Traditional blood-based evaluation of ketone bodies generally involves the use of strip-based, capillary blood ketone meters, measuring beta-hydroxy butyrate (BHB) levels, an important biomarker in diagnosing DKA. In total, there are three ketone bodies, namely, acetoacetate, BHB, and acetone, that are produced in significant levels during DKA. BHB sensing is usually associated with low costs and lesser requirements for hospitalizations [26]. Just like glucose measurement, a disadvantage of using strip-based blood ketone monitors is they only provide static information about the BHB levels or any possible implications of ketonemia at any fixed instant of time [66]. The current state-of-the-art approach is, therefore, fabricating subcutaneous continuous ketone monitoring (CKM) platforms, just like CGM devices, that can also track the dynamics in ketone levels at recurring intervals of time.

For ketone monitoring, however, scientific research is only limited to the design and fabrication of minimally-invasive CKM biosensors [86, 6]. There is a lack of comprehensive machine learning approaches that can optimize their overall performance and predict or

quantify [BHB](#) levels from ISF in a point-of-care, personalized fashion with a greater degree of precision and accuracy. A recent review [\[18\]](#) suggests how wearable ketone platforms and automated insulin delivery systems should be merged with data processing techniques, machine learning algorithms, and cloud-based architectures for better and efficient management of diabetic ketoacidosis.

# Chapter 3

## Long-term glucose forecasting for type 1 diabetes

### **3.1 A review of time-series forecasting models for glucose monitoring: why are long-term predictions important?**

Long-term forecasting of blood glucose, characterized by a prediction horizon spanning more than an hour, offers multiple advantages over short-term glucose predictions. It enables health practitioners with more effective baseline insulin planning by anticipating future glucose trends for extended periods of time and improves the precision of any preemptive insulin dosages provided to prospective patients to avoid any possible risks for hyperglycemic or hypoglycemic events in the future [57]. Another advantage is that it facilitates better dietary management by offering patients ample time to plan their meals and bolus insulin intakes in advance based on the predicted glucose trends and, subsequently, control their overall carbohydrate levels. Last but not the least, it helps in better and more strategic scheduling of physical activities to prevent any hypoglycemic risks during or after exercise and adjusting treatment plans by taking delayed effects of factors like sleep, stress or hormonal changes, into consideration. Hence, by offering a broader window for decision-

making, long-term forecasting optimizes personalized treatment tailored to every patient’s needs, resulting in better glycemic control, reduced risk of T1D-related complications, and improved quality of life.

Table 3.1 gives a brief review of some of the state-of-the-art approaches in time series forecasting of glucose levels in type 1 diabetic patients. The algorithms used range from classical statistical techniques like autoregressive models [14, 25, 74] and autoregressive integrated moving average (ARIMA) models [7] to more robust machine learning techniques like support vector machines [74, 75, 97, 99, 103], random forests [61, 90], recurrent neural networks (RNN) [7, 14, 32, 62, 72], extreme learning machines [97] and convolutional neural networks [103] to name a few. A study from 2018 [105] implemented a recursive least squares-based filtering fusion and decision-making mechanism that forecasts glucose concentrations for both in silico and clinical subjects. Another work from 2019 [106] explores the use of an empirical mode decomposition technique that enables the modeling of the fluctuations and trends in glucose time series data separately, followed by a genetic algorithm-inspired backpropagation neural network scheme that achieved root mean squared error (RMSE) as low as 6.13mg/dl and 8.21mg/dl respectively for prediction horizons 15 and 30 minutes respectively. A state-of-the-art paper from 2019 [55] proposes a multi-layer convolutional RNN that uses temporal convolution and kernel filters to find the most important set of input features for glucose prediction followed by an LSTM-inspired network for forecasting. The work achieved RMSE values of 9.38mg/dL and 18.87mg/dL for 30 and 60 minutes prediction respectively on simulated patient data and values of 21.07mg/dL and 33.27mg/dL for 30 and 60 minutes respectively on real, clinical human subjects. More improvised deep learning frameworks like Transformers and infinite mixture models are being implemented in state-of-the-art glucose forecasting research in modern times as in the case of a recent study [77] from 2023 that was able to achieve a mean absolute percentage error as low as 7.78% and RMSE of 14.73mg/dL for a 60 minute forecast. A common theme in all these studies is that they have primarily focused on short-term predictions ( $< 1$  hour). However, studies have shown that symptoms of various complications of T1D like DKA can develop within 24 hours in human beings [1]. Hence extending this horizon beyond 1 hour is necessary to accommodate better and more precise insulin remedies.

Time series forecasting schemes for glucose are primarily of two types - (a) a single output scheme, that works in a one-step-ahead fashion, or in other words, one step of glucose is first predicted and then fed back to the input to generate the next step and so on [72]; or (b) a multi-output scheme, which directly predicts the glucose values for the entire prediction horizon, all at once [32]. Furthermore, many studies [25, 45, 55, 61, 62] have focused on using historical information like insulin dosages, meal intakes, body temperature and even heart rates to improve the understanding of the CGM data by time series forecasting models. However, with the primary target of the state-of-the-art insulin delivery platforms to be fully-automated in the recent years, it is more favorable to use CGM as a single source of input as it removes any requirement of human intervention or supervision while such platforms predict blood glucose levels and subsequently estimate the optimal baseline insulin concentration to be delivered to the patient under study.

Above all, the focus of this chapter is two-fold - (a) to achieve competitive performance in terms of accuracy and robustness for long-term forecasting scenarios beyond 1 hour, and (b) maintaining CGM as the only source of input data for the long-term prediction, in accordance with state-of-the-art fully-automated insulin delivery frameworks.

Table 3.1: A review of time series models for glucose forecasting

Reference (year)	Model(s) used	Prediction horizon (mins)	Patients	# of days or samples	Inputs used
[7] (2021)	ARIMA, random forest and RNN	15, 30, 45, 60	38	399302 samples	CGM
[14] (2014)	autoregression, RNN	15, 30, 45	23	$5.30 \pm 1.4$	CGM
[25] (2013)	ANN, autoregression	15, 30, 45	23	$2695 \pm 670$ samples per subject	CGM and insulin

Continued on next page

Table 3.1: A review of time series models for glucose forecasting (Continued)

Reference (year)	Model(s) used	Prediction horizon (mins)	Patients	# of days or samples	Inputs used
[32] (2018)	RNN (gated recurrent unit)	30	40	1900	CGM
[45] (2019)	XGBoost	30	6	52	CGM, insulin, meal intakes, physical activity, heart rate, sleep and stress data
[55] (2019)	convolutional RNN	30	20	180 or 360	CGM, insulin and meal intakes
[61] (2020)	random forest and boosting algorithms	30, 60	12	8 weeks	CGM, insulin, meal intakes, physical activity, sleep, stress and work data, heart rate and body temperature
[62] (2019)	LSTM	30, 60	6	56	CGM, insulin and meal intakes
[72] (2020)	bi-directional LSTM	30, 60	106	> 15 million samples	CGM, insulin, sleep data and activity levels

Continued on next page

Table 3.1: A review of time series models for glucose forecasting (Continued)

Reference (year)	Model(s) used	Prediction horizon (mins)	Patients	# of days or samples	Inputs used
[74] (2020)	autoregression and SVR	30, 45, 60	6	30 or 1 days	CGM, insulin, meal intakes and physical activity
[75] (2018)	autoregression, SVR and ANN	30, 45, 60	16	at least 30 days	CGM, insulin, meal intakes and physical activity
[77] (2023)	infinite mixture models and transformers	60	38	399302 samples	CGM
[82] (2012)	autoregression, kernel-based and state-space models	40	6	unspecified	CGM, insulin, meal intakes and physical activity
[90] (2018)	random forest	30	93	194891 samples	CGM, insulin, meal intakes, physical activity, heart rate and energy expenditure
[93] (2017)	grammatical evolution	60	1	1152 samples	CGM, insulin and meal intakes

Continued on next page

Table 3.1: A review of time series models for glucose forecasting (Continued)

Reference (year)	Model(s) used	Prediction horizon (mins)	Patients	# of days or samples	Inputs used
[97] (2013)	autoregression, extreme learning machines, SVR	30	10	860 samples per subject	CGM
[99] (2019)	SVR	30	1	863 samples	CGM
[102] (2018)	random forest, boosting, SVR, LSTM, lasso and ridge regression and temporal convolutional networks	30	6	8 weeks	CGM, insulin, meal intakes and heart rate
[103] (2020)	elastic, lasso and ridge regression, SVR, decision trees, autoregression, LSTM and temporal convolutional neural networks	5, 30	6	8 weeks	CGM, insulin, meal intakes and heart rate
[104] (2019)	linear regression and decision trees	unspecified	unspecified	7642 samples	CGM, blood test results, blood lipid, liver index and biochemical index

Continued on next page

Table 3.1: A review of time series models for glucose forecasting (Continued)

Reference (year)	Model(s) used	Prediction horizon (mins)	Patients	# of days or samples	Inputs used
[105] (2018)	recursive least squares	30	39	> 4000 samples	CGM
[106] (2019)	genetic algorithm-backpropagation neural network	15, 30	1	4 days, 1152 samples	CGM
[108] (2023)	SVR, LSTM, dilated RNN, transformers	30, 60	12	unspecified	CGM

## 3.2 RNN and LSTM: a brief introduction

The most traditional feedforward artificial neural network (ANN) [41] functions by implementing sequential and unidirectional propagation of the input data through interconnected layers of neurons, which mimics the biological neurons present in a human body. It starts with an input layer that accepts the raw information, passes it through hidden layers (if present), and finally terminates at the output layer. At each neuron in any layer, a weighted sum of inputs with a bias term is first computed and then transformed using a nonlinear activation function, allowing the network to explore the intricate patterns in the input data. By leveraging backpropagation, which enables the prediction errors to be propagated backward through the network, the model updates its weights and biases to minimize the errors and achieve optimal performance. On the other hand, recurrent neural networks (RNN) are an extension of the conventional feedforward neural networks consisting of loops in their architecture that allow information to persist. Figure 3.1 shows a simple illustration of RNNs.

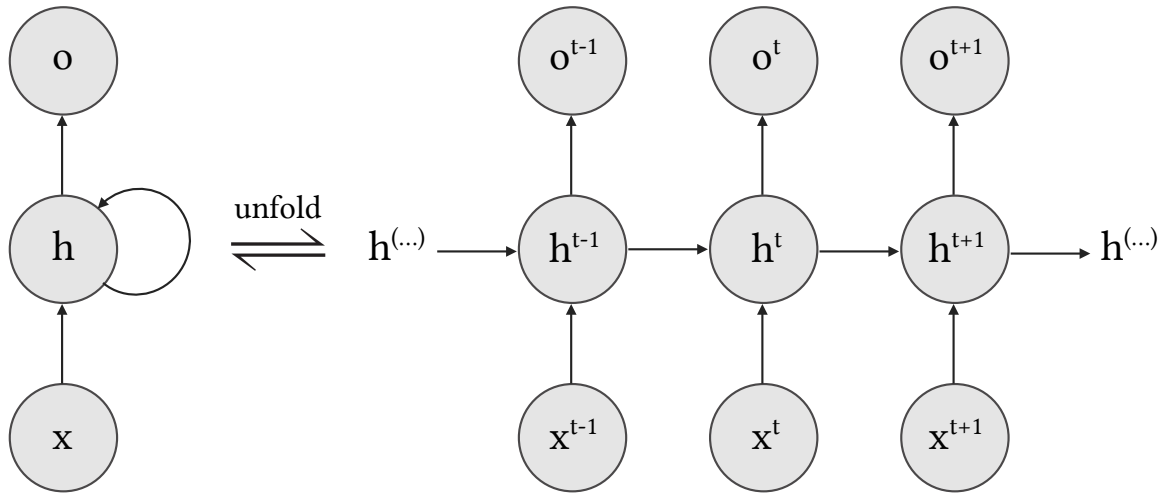


Figure 3.1: A simple illustration of RNN with input  $x$  forwarded to a hidden state  $h$ , resulting in an output  $o$

RNNs, in general, offer multiple advantages over conventional ANNs. First, they excel in modeling sequential data due to their ability to maintain internal state or memory. Unlike feedforward neural networks, which independently process each input data point, RNNs process sequences of data by iteratively updating their hidden states as they receive each input. This enables them to capture any temporal dependencies and contextual information within the sequential data, making them well-suited for tasks where the order of inputs matters, as in the case of time series prediction of glucose. Secondly, ANNs can only manage fixed-size inputs producing fixed-size output, making them unsuitable for tasks involving sequences of varying lengths, whereas RNNs can handle sequences of variable lengths since they process inputs sequentially, adjusting their internal state based on the length of the input sequence. Finally, in RNNs, the same set of parameters are shared across all time steps, allowing them to leverage learned representations from earlier time stamps when processing subsequent inputs. This parameter sharing reduces the com-

putational cost of training RNNs compared to feedforward neural networks with similar numbers of parameters, as the later require independent parameters for each layer.

Figure 3.1 shows an illustrative representation of an RNN operating on an input sequence  $x^t$  at any given time step index  $t$ , such that  $t = 1, \dots, T$ ,  $T$  is the maximum number of iterations. The cycle shown in the computational graph in the figure indicates the impact of a variable's past value on the present time step. The graph has a repetitive structure that unfolds into a recursive chain of events that directs the flow of information in the RNN. The direction is forward at the time of evaluating the outputs and the corresponding losses and backward at the time of computing the gradients to update the parameters of the RNN. The unfolded computational graph follows the equation:

$$h^t = f(h^{t-1}, x^t | W), \tag{3.1}$$

where  $f$  is a mapping,  $h^t$  represents the network's hidden state at time step  $t$ , and  $W$  denotes the weights of the network connections. Primarily, there are three kinds of weights in the RNN graph: input-to-hidden, hidden-to-hidden, and hidden-to-output connection weights. To train the RNN graph, the first step involves the gradient computation of the loss function with respect to the parameters involved in forward propagation from left to right of the unfolded graph. This is followed by a back-propagation step, implemented using an algorithm called [back-propagation through time \(BPTT\)](#) [63], moving from right to left through the graph to update the new set of weight matrices. The forward propagation step to calculate the output  $o^t$  and the losses usually involves the following equations (using hyperbolic tangent as the nonlinear activation function and softmax function [16] to calculate probabilities):

$$a^t = b + Wh^{t-1} + Ux^t, \tag{3.2}$$

$$h^t = \tanh(a^t), \tag{3.3}$$

$$o^t = c + Vh^t, \tag{3.4}$$

$$\hat{y}^t = \text{softmax}(o^t), \tag{3.5}$$

Here,  $b$  and  $c$  indicate the biases and  $U$ ,  $V$  and  $W$  denote the weight matrices for input-to-hidden, hidden-to-output, and hidden-to-hidden connections respectively. Considering a differentiable loss function  $\mathcal{L}(\cdot, \cdot)$ , the loss for sequence  $x$  corresponding to its ground truth  $y$  is determined as:

$$\sum_{t=1}^T \mathcal{L}^t = \mathcal{L}(x^1, x^2, \dots, x^T, y^1, y^2, \dots, y^T), \quad (3.6)$$

Finally, during back-propagation, the gradient of the function above is computed for every iteration with respect to the parameters  $U, V, W, b, c, x^t, h^t, o^t$ , and  $\mathcal{L}^t$  to minimize the loss.

Long short-term memory networks (**LSTM**) are an extension to the traditional RNNs that use gated mechanisms to improve long-term dependencies in the input time series data, especially for random variables like glucose concentration. LSTMs are known to offer several advantages when compared to RNNs. First, the traditional RNNs are prone to a problem called vanishing gradients, where the gradients, calculated during the back-propagation step, diminish exponentially over time, thereby, hindering the learning of long-term dependencies in the input data. LSTMs mitigate this issue through the use of a gating mechanism, which enables them to selectively retain or discard information over multiple sequential time steps and, thus, propagate gradients more effectively over time. Second, RNNs often struggle to retain information over long sequences owing to the limited capacity of their hidden state, resulting in difficulties in capturing long-term dependencies in the time series data. LSTMs are specifically designed to capture long-term dependencies in sequential data. Traditional LSTMs address this issue by incorporating a cell state, enabling them to capture complex temporal patterns and dependencies in the data more effectively.

Figure 3.2 shows a sample illustration of an LSTM cell. As evident from the representation, the inner self loop present in LSTM, in addition to the original architecture of a RNN allows the gradients to flow for longer duration of time and are not fixed.

There are three gating units in an LSTM cell — the external input gate, the forget gate, and the output gate - that control the flow of data inside the cell. The forget gate at

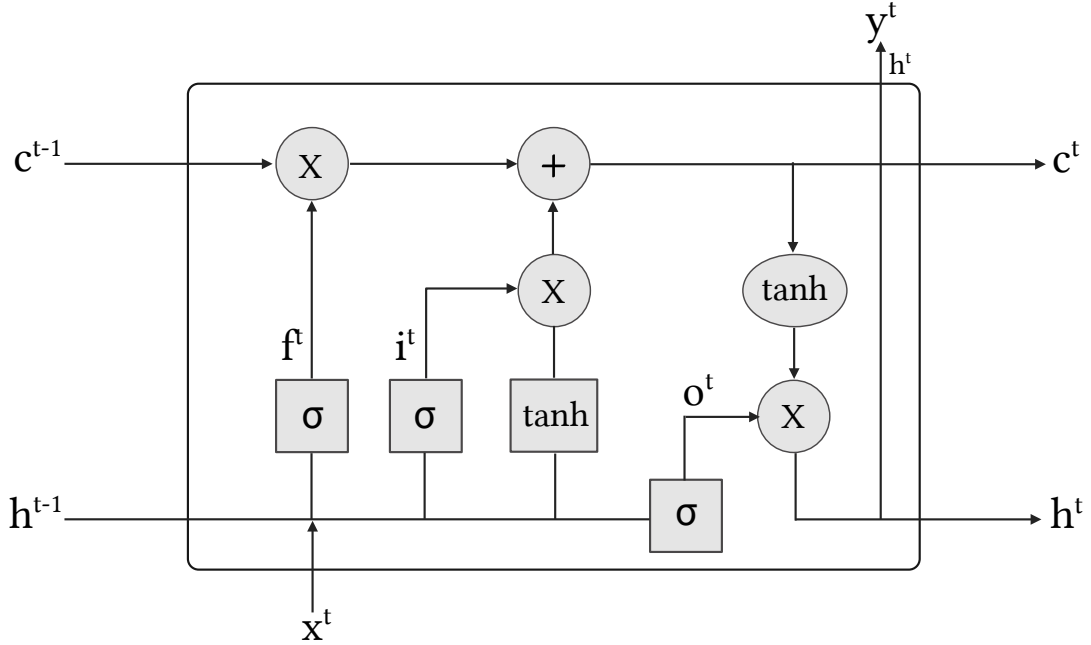


Figure 3.2: A simple illustration of LSTM with cell state  $c^t$ , hidden state  $h^t$ , input  $x^t$ , and output  $o^t$

a given time  $t$  and the internal state decide whether the information should be discarded from the cell state or not. It controls the self-loop by considering a nonlinear sigmoid function  $\sigma$  and setting the value of the weight between 0 and 1. If the value tends to 1, the past information is retained, or else, if it is close to 0, then rejected. Following this, with the help of the external input gate layer, it is decided which new information should be added to the internal state. Finally, the output gate of the LSTM, which also implements a sigmoid function unit, determines which output values should be passed or discarded.

To mathematically represent the working of an LSTM cell, the following equations are considered:

$$f^t = \sigma(W_f x^t + U_f h^{t-1} + b_f) \dots(\text{forget gate}), \quad (3.7)$$

$$i^t = \sigma(W_i x^t + U_i h^{t-1} + b_i) \dots(\text{input gate}), \quad (3.8)$$

$$o^t = \sigma(W_o x^t + U_o h^{t-1} + b_o) \dots(\text{output gate}), \quad (3.9)$$

$$c^t = f^t c^{t-1} + i^t \cdot \tanh(W_c x^t + U_c h^{t-1} + b_c) \dots(\text{cell state update}), \quad (3.10)$$

$$h^t = o^t \cdot \tanh(c^t) \dots(\text{hidden layer state}), \quad (3.11)$$

$$y^t = h^t \dots(\text{output}), \quad (3.12)$$

Here,  $x^t$  indicates the input,  $h^t$  represents the hidden layer state,  $c^t$  refers to the flow of information from one LSTM to another.  $U_f, U_i, U_o, U_c$  and  $W_f, W_i, W_o, W_c$  indicate the input and recurrent weight matrices respectively, and  $b_f, b_i, b_o, b_c$  denote the bias terms, to be learned while training the LSTM network.

### 3.3 Understanding sequence-to-sequence modeling and attention mechanism

[Sequence-to-sequence \(Seq2Seq\)](#) models in machine learning have evolved as a progressive step towards handling a wide range of problems involving sequential data. Their applications have widely spanned across areas like machine translation, where sentences in one language are converted to another; text summarization, where a long article is shortened into a brief summary; speech recognition, where the spoken language is transcribed into text and in recent years, in better understanding of time series datasets. They usually offer multiple advantages over conventional machine learning techniques. Firstly, these models are efficient in being able to interpret input and output sequences of varying lengths. Traditional fixed-size input-output models cannot handle this variability precisely, but Seq2Seq models deploy an encoder-decoder architecture that encodes the input sequence into a meaningful, intermediate representation, which is then decoded into the output sequence, allowing them to manage varying lengths effectively. Secondly, Seq2Seq models, particularly along with mechanisms like attention, excel at capturing long-range dependencies and contextual information from within the input sequences. The mechanism allows us to focus on the most relevant parts of the input sequence, improving the quality of the

generated output.

The Seq2Seq architecture fundamentally consists of two primary components: an encoder and a decoder. The encoder's role is to process the input sequence, one element at a time, and transform it into a fixed-size context vector. This context vector is essentially a compressed representation of the entire input sequence. The encoder is typically implemented using recurrent neural networks (RNN), long short-term memory networks (LSTM), or [gated recurrent units \(GRU\)](#). These networks are particularly suited for sequential data because they can maintain and update a hidden state that captures the information from previous elements in the sequence. As the encoder processes each element of the input, it updates this hidden state, and the final hidden state becomes the context vector that encapsulates the information from the entire input sequence.

Once the encoder has processed the input sequence and generated the context vector, the decoder then takes over to generate the output sequence. The decoder also uses RNNs, LSTMs, or GRUs for its architecture and works step-by-step to produce the output elements. It starts with the context vector from the encoder and generates the first element of the output sequence. This element, along with the context vector, is then used to produce the next element in an autoregressive fashion, and this process continues until the entire output sequence is generated. A key challenge here is that the decoder needs to rely heavily on the context vector, which may not always capture all the nuances of longer input sequences.

Figure 3.3 shows a sample illustration of how a Seq2Seq model works. The encoder and the decoder are both represented as a single layer of RNN cells. It should be noted that the most fundamental structure of a Seq2Seq model uses only one context vector learned from the encoder, as shown in the figure. However, to mitigate the limitation of relying on a single context vector as the only source of information for the decoder, Seq2Seq models use a mechanism called attention that allows the decoder to dynamically focus on different parts of the input sequence at each step of generating the output. Instead of using only the fixed context vector, the decoder can access the entire sequence of encoder hidden states, assigning different weights to different parts of the input sequence based on their relevance to the current step of the output sequence.

Traditional sequence-to-sequence models face many limitations while handling long

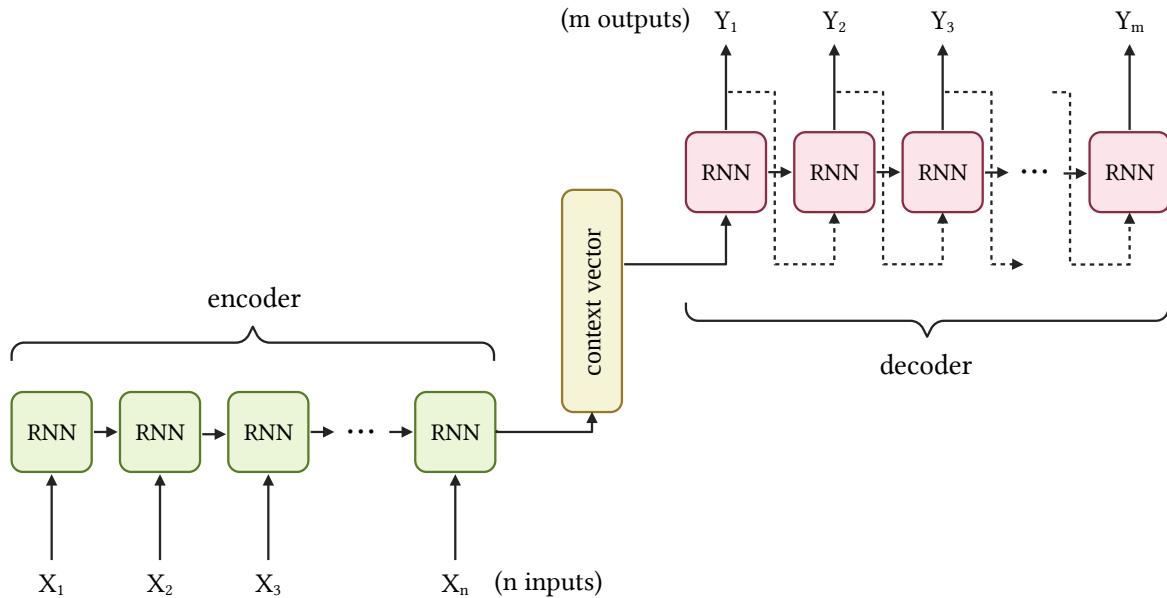


Figure 3.3: An illustration of a sequence-to-sequence machine learning framework with its encoder and decoder each functionalized with a simple recurrent neural network layer and with an input vector of length  $n$  and an output vector of length  $m$

input sequences due to a fixed-size context vector. The primary function of the context vector as an output of the encoder is to capture all the information of the input sequence, but with the gradual increase in the length of the input sequence, using a single vector to summarize everything about the input becomes inefficient and leads to information loss. The attention mechanism addresses this issue by allowing the model to focus on different parts of the input sequence at each step of the output generation rather than relying on a single fixed-size context vector. To understand this better, we refer to Figure 3.3. We have an input sequence of length  $n$  fed to the encoder, which consists of a single layer of  $n$  RNN cells. Let the hidden states obtained as output from the encoder's cells be  $h_1, h_2, \dots, h_n$ . Following this, for every time step  $t$ , the decoder generates the output sequence  $y_t = \{Y_1, Y_2, \dots, Y_m\}$  from these hidden states, given  $m$  as the length of the output sequence

desired. As per the definition of attention, at the first step, for every time step  $t$ , a set of attention scores  $e_{t,i}$  is calculated for each encoder hidden state  $h_i$ . These scores represent the relevance, or in other words, the importance of each hidden state to the current decoding step. Typically, these scores are computed using a function of the decoder hidden state  $s_{t-1}$  obtained at the previous time step  $t - 1$  and the encoder hidden states  $h_i$ . This function could range from a simple dot-product operation to more advanced frameworks involving learnable functions like feed-forward neural networks. Mathematically, for a function  $f$  to calculate these scores, we have,

$$e_{t,i} = f(s_{t-1}, h_i) \tag{3.13}$$

Following this, we normalize these scores to eventually get the attention weights, defined as  $\alpha_{t,i}$ . Assuming softmax function for normalization, we have,

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^n \exp(e_{t,j})} \tag{3.14}$$

Furthermore, the context vector, defined as  $c_t$  for time step  $t$  is computed as a weighted sum of the encoder hidden states, using the attention weights as,

$$c_t = \sum_{i=1}^n \alpha_{t,i} h_i \tag{3.15}$$

Eventually, the context vector  $c_t$  is combined with the decoder hidden state  $s_{t-1}$  to produce the output for the current time step  $y_t$ .

There are multiple types of attention depending on how the attention scores are calculated. One type of attention is the additive attention, also known as, the Bahdanau attention, named after its study [10]. In this case, the set of attention scores are calculated using a feedforward neural network with a single hidden layers as per the equation:

$$e_{t,i} = v^T \tanh(W_1 s_{t-1} + W_2 h_i) \tag{3.16}$$

where  $v$ ,  $W_1$ , and  $W_2$  are all learnable parameters.

Another form of attention, called as the dot-product attention, or the Luong attention [58] computes the scores using a simple dot product between the previous decoder hidden

state and the encoder hidden states.

$$e_{t,i} = s_{t-1}^T h_i \tag{3.17}$$

A more improvised version, called as the scaled dot-product attention performs the conventional dot-product attention but scales the scores by the square root of the dimension of the hidden states to counteract the effect of large dimensions of the architecture, as shown below.

$$e_{t,i} = \frac{s_{t-1}^T h_i}{\sqrt{k}} \tag{3.18}$$

where  $k$  is the dimension of the hidden states.

With the introduction of attention mechanisms, Seq2Seq models can now show an improved performance while handling longer input-output sequences by focusing on relevant parts of the input. Furthermore, it has improved the interpretability of these models as the attention weights provide insights into which parts of the input the model is mainly focusing on and are comparatively more significant compared to others.

## 3.4 Problem formulation and data preprocessing

### 3.4.1 Mathematical background

The primary aim of this study is to develop a machine learning framework that performs a long-term forecasting of glucose in patients with **T1D**. For this, we formulate our problem as follows. Let  $H \in \mathbf{N}$  (set of natural numbers) denote the number of patients considered in the cohort with each patient indicating a unique glucose sequence in the study. We define a notation  $[H] := \{1, 2, \dots, H\}$ . Let  $h$  indicate any glucose sequence chosen at random from  $[H]$  and let  $\tau_h$  indicate the number of **CGM** recordings performed in the study for the sequence  $h$ . Let  $U_h$  be the set of CGM measurements for  $h$  given as  $U_h = \{u_h^i\}_{i \in [\tau_h]}$ . Essentially  $U_h \in \mathbf{R}^{\tau_h}$ . Because the glucose profiles vary from one patient to another, we normalize every  $U_h$  to the same range. Let  $U_h^*$  indicate the normalized version of  $U_h$  for patient  $h$ , such that  $U_h^* \in [0, 1]^{\tau_h}$ ,  $[0, 1]$  indicating the set of all real numbers ranging from

0 to 1. Here 0 represents the lowest glucose value recorded by the CGM device, after normalization, across the entire cohort of  $H$  patients, while 1 indicates the highest value measured by the device.

After data preprocessing (explained in the next subsection), let  $X_h := \{x_h^i\}_{i \in [\tau_h]}$  be the filtered, or in other words, the denoised glucose sequence obtained from  $U_h^*$ , for each  $h \in [H]$ . Following this, we perform an operation on each  $X_h$  called the seasonal-trend-dispersion decomposition as explained in a recent study [28]. In this case, we extract three different components from the filtered glucose sequence  $X_h$ . First, we have the trend component sequence  $T_h$  whose primary function is to capture the long-term patterns, or trends in the glucose sequence  $X_h$ . This could indicate any long-term changes influenced by meal timings or exercises or any other similar factor, at regular intervals, on the patients' glucose levels. We obtained  $T_h$  by applying a moving average smoothing operation with a pre-specified window size on  $X_h$ . Let  $z$  be the window size. Second, we have the seasonal component sequence  $S_h$ , which mainly focuses on capturing the high-frequency, short-term fluctuations that occurs in the glucose profile  $X_h$ . We considered an assumption that  $X_h = T_h + S_h$  and obtained the seasonal component  $S_h$  by subtracting the trend sequence from  $X_h$ . Lastly, we considered a dispersion sequence  $D_h$  that indicates the measure of dispersion and its variation with time for the sequence  $X_h$ , as a measure of controlling the problem of heteroscedasticity in glucose time series problems. (explained further in the next subsection)

Let  $m$  be the number of past instants to consider for the input timeframe for the forecasting model to interpret and  $n$  be the number of future instants we want to predict for the output prediction horizon. Essentially,  $m, n < \tau_h$  for every  $h \in [H]$ . Having obtained the individual components from each filtered sequence  $X_h$ , we proceed towards defining the inputs for our model. For this, we consider a window of size  $m$  and slide it across  $T_h$  to obtain a subset or segment of the form  $T_h^{i:i+m-1} := \{t_h^j\}_{j=i, i+1, \dots, i+m-1}$  where  $t_h^j$  indicates any particular value in the trend sequence. We applied the same operation on  $S_h$  and  $D_h$  to obtain subsets of the form  $S_h^{i:i+m-1}$  and  $D_h^{i:i+m-1}$ . We further consider a fourth subset called as the lookback input, that considers the most recent information about the filtered glucose sequence  $X_h$  from the entire input timeframe ranging from  $i$  to  $i + m - 1$  for any given  $i$ . For this, we consider  $q$  as the number of instants of

the input timeframe to include in the lookback period. Essentially  $q < m$ . Thereby, from definition, the lookback input is given as  $X_h^{i+m-q:i+m-1} := \{x_h^j\}_{j=i+m-q, \dots, i+m-1}$ . Finally, to learn the forecasting model, we consider a mapping  $M$  such that it accepts  $T_h^{i:i+m-1}, S_h^{i:i+m-1}, D_h^{i:i+m-1}$  and  $X_h^{i+m-q:i+m-1}$  as its inputs. The predicted output vector will consist of  $n$  elements, as per the definition of our output horizon. We define the output glucose vector as  $Y_{h,i,n} := M(T_h^{i:i+m-1}, S_h^{i:i+m-1}, D_h^{i:i+m-1}, X_h^{i+m-q:i+m-1} | \theta)$ , where  $\theta$  represents the set of learned parameters. The main reason to consider a lookback input for the model is to ensure that not only the errors in predicting the glucose values are minimized, but also the predicted glucose trajectory moves in the correct direction and validates the expected long-term trends in the glucose levels of the patient. To finally evaluate the performance of the forecast, we compare the predicted vector  $Y_{h,i,n}$  with its corresponding ground truth, extracted from the original filtered glucose sequence  $X_h$  as  $X_h^{i+m:i+m+n-1}$ . Therefore, for our problem, we construct a dataset as  $\mathcal{D} = \{(T_h^{i:i+m-1}, S_h^{i:i+m-1}, D_h^{i:i+m-1}, X_h^{i+m-q:i+m-1}, X_h^{i+m:i+m+n-1})\}_{i \in [\tau_h - m - n + 1], h \in [H]}$ . We further partitioned  $\mathcal{D}$  into two parts, one for training the model  $M$  and the other for evaluating its performance.

### 3.4.2 Data preprocessing

The first step of data preprocessing involves the removal of noise from the glucose sequences for each patient recorded by the CGM device. After normalization of the recorded sequences, we implemented a low pass filter that removes the high-frequency noise from the normalized sequences  $U_h^*, h \in [H]$ . As defined earlier, the resulting filtered sequence obtained from  $U_h^*$  is  $X_h$ . In Figure 3.4, for a sample patient, the sequences  $U_h^*$  and  $X_h$  are represented as "normalized glucose sequence" and "filtered glucose input" respectively.

In the next step, we consider a framework that considers the issue of heteroscedasticity in glucose sequences to improve the forecasting performance in the long term scenario. Heteroscedasticity primarily refers to a condition in which the variance of the errors in a regression model is not constant across all levels of the independent variables considered in the model. In the context of time series forecasting, in a heteroscedastic time series, the variability of the error or deviation changes with time and is never stationary. Almost

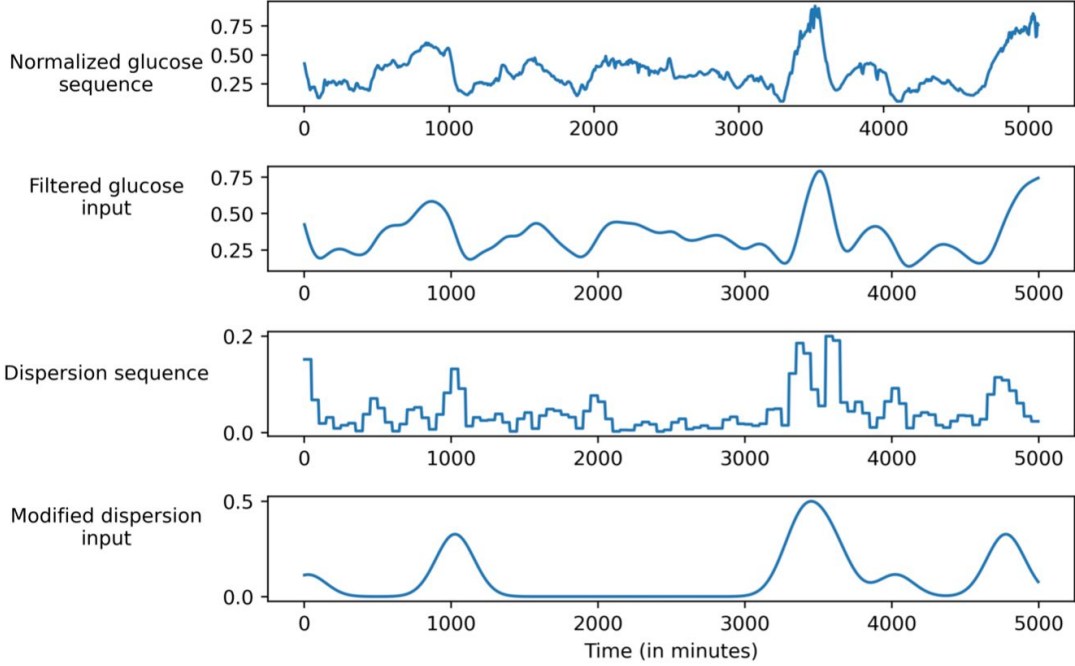


Figure 3.4: A summary of the data preprocessing steps implemented before learning the glucose forecasting framework

all real life time series datasets, like in the case of CGM recordings, are heteroscedastic in nature. However, a majority of algorithms used in time-series forecasting applications, like an autoregressive model, assume a constant variance or deviation with time, resulting in a failure to deal with seasonal variations in time dependent sequences like glucose recordings, and thereby, resulting in a failure to provide a precise and accurate forecast. For this, we consider the seasonal-trend-dispersion decomposition technique [28] to eventually find a sequence that keeps track of the variance or dispersion with respect to time corresponding to each filtered sequence  $X_h$ . First, we assume  $X_h$  has a seasonality of time period  $w$ , and its length  $\tau_h$  is a multiple of  $w$ . Considering  $z = \frac{\tau_h}{w}$ ,  $z \in \mathbb{N}$ , we partition  $X_h$  into  $z$  equal-sized vectors of length  $w$ . We consider each such vector as  $X_h^s \in (0, 1)^w$ ,  $s = 1, 2, \dots, z$ . Let  $\tilde{x}_s \in \mathbb{R}$  be the standard deviation of the  $s^{th}$  seasonal cycle  $X_h^s$ . Now, we define a new sequence  $P_h \in (0, 1)^{\tau_h}$  of same seasonality period  $w$ , such that its  $s^{th}$  seasonal cycle

$P_h^s \in (0, 1)^w$  is given as  $P_h^s = \{\tilde{x}_s, \tilde{x}_s, \dots, \tilde{x}_s\}$  ( $w$  times).

In Figure 3.4, 'dispersion sequence' refers to the sequence  $P_h$ . As evident,  $P_h$  at any given time instant is not always zero, indicating a change in variance with time. This further proves that the filtered sequence  $X_h$  is not a constant variance signal, or in other words, it is heteroscedastic in nature. Let  $P_h = \{p_h^i\}_{i \in [\tau_h]}$ . However,  $P_h$  is noisy. Hence, we focused on its significant peaks only by binarizing it. Let the new binary sequence created be  $P_h^{bin} = \{p_h^{bin,i}\}_{i \in [\tau_h]}$ , where  $F$  is a function  $\mathbf{R} \rightarrow \mathbf{R}$ , and,

$$p_h^{bin,i} = F(p_h^i) = \left( \sum_{i=1}^{\tau_h} \frac{x_h^i}{\tau_h} \right) \cdot \max \left( p_h^i - \frac{\max(P_h)}{2}, 0 \right) \quad (3.19)$$

where  $i \in [\tau_h]$  and,  $h \in [H]$ .  $\left( \sum_{i=1}^{\tau_h} \frac{x_h^i}{\tau_h} \right)$  is a constant used for scaling  $P_h^{bin}$ . The binary dispersion sequence  $P_h^{bin}$  is fundamentally, an aperiodic square wave signal of varying amplitude, and the jump discontinuities at the corner points may disrupt the model in accurately predicting future glucose trajectories. Hence, we further implement a smoothing operation on  $P_h^{bin}$  by defining  $G$  as a zero-mean, discrete Gaussian waveform and performing a discrete-time convolution of  $P_h^{bin}$  with  $G$  to finally obtain the dispersion component sequence  $D_h$ , as,

$$D_h = k \cdot \mathbf{conv}(P_h^{bin}, G) = k \cdot (P_h^{bin} * G) \quad (3.20)$$

where,  $k$  is a scaling constant used for bringing  $D_h$  and the trend and seasonal components  $T_h$  and  $S_h$  all within a comparable range.  $*$  denotes discrete-time convolution operation between two sequences. In Figure 3.4,  $D_h$  is represented as "modified dispersion input".

## 3.5 Experiments

### 3.5.1 Experimental settings and model architecture

We utilized the CGM data collected in a previous study [32], currently available open-source in GitHub [31], and recently explored in multiple glucose forecasting-based studies [7, 78]. The cohort consists of 38 patients in total, that is  $H = 38$ . The CGM measurements were taken after every 5 minutes. For model training and evaluation, we partitioned the

dataset  $\mathcal{D}$  into three distinct parts - training, validation, and testing sets. We used a ratio of 20 : 1 : 1 respectively. The validation set was used exclusively for hyperparameter tuning for our model, with the aim of preventing overfitting and optimizing the model’s performance further. In hyperparameter tuning, we implemented a random search method to find an optimal set of values for the learnable parameters  $\theta$  of our model. Essentially, the validation data was not used during the training of the model but was limited to finding the set of learnable parameters  $\theta$  and was not used in the test dataset, ensuring an unbiased evaluation of our architecture and all the baselines considered.

The performance metrics used for evaluation were [mean absolute percentage error \(MAPE\)](#) and root mean squared error ([RMSE](#)). For our problem, as per the definitions in Section 3.4.1, we considered  $m = 180$ , the number of past instants to consider in the input timeframe. We evaluated our model under two conditions - a long-term scenario of 3 hours and a short-term scenario of 1 hour. Hence, for our problem,  $n = 12$ , or 36 (12 instants refers to 1 hour). For the lookback input, we considered  $q = 60$ . To derive the dispersion sequence, we assumed a seasonality of time period equivalent to  $w = 10$  instants.

Figure 3.5 illustrates the model architecture we used in our study. We describe our framework in the following steps:

1. To begin with, for each example in the dataset  $\mathcal{D}$ , we considered 4 inputs - a trend input  $T_h^{i:i+m-1}$ , a seasonal input  $S_h^{i:i+m-1}$ , a dispersion input  $D_h^{i:i+m-1}$ , and a lookback input  $X_h^{i+m-q:i+m-1}$ . Next, we defined a separate encoder that generates a unique representation for each of 4 inputs.
2. The encoder consists of a self attention layer followed by a layer of LSTM networks. During self attention, the simplest form of attention, the query, key and value of the conventional attention mechanism are assumed all equal to the input sequence received. LSTM networks were chosen here owing to their superior performance in handling time series problems compared to conventional RNNs.
3. We further consider the 2 encoders associated with the trend and seasonal inputs and generate two sequences. The first sequence is obtained by concatenating only the final context vectors from the 2 encoders. The second sequence is obtained by concatenating the hidden and cell states of the LSTM networks used in the 2

encoders in all the previous time steps as one whole sequence. The purpose of the first sequence is to feed the trend and seasonal information to the decoder together after being studied separately by the encoders. We implemented one more step of self attention followed by a layer of LSTM networks to summarize the information further, ensuring the most relevant features in the long-term trends and short-term seasonal fluctuations are only considered. The second sequence is fed to a self attention layer, whose output is used to initialize the LSTM used in studying the first sequence.

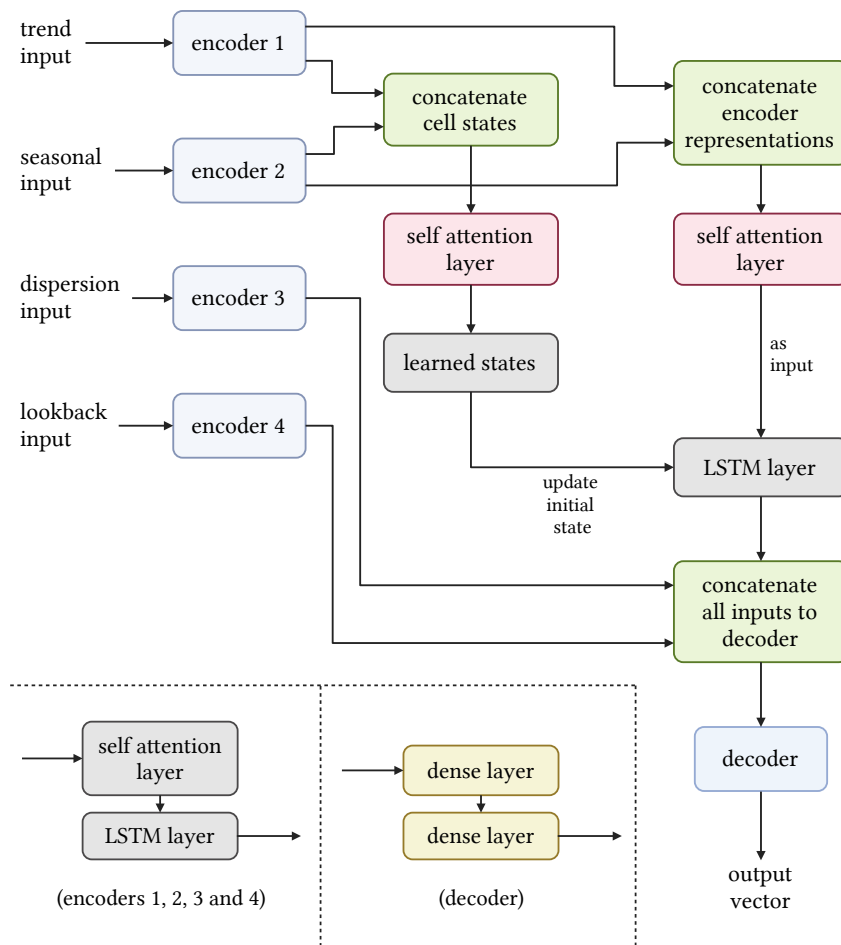


Figure 3.5: An illustration of our long-term glucose forecasting architecture

4. The resulting information obtained after applying the concatenated trend and seasonal context vectors to the LSTM network, together with the encoder representations obtained from dispersion and lookback inputs, are further concatenated to be finally fed to the decoder. The decoder used in our framework is primarily based on a multi-layer perceptron architecture with two hidden layers to generate the final output glucose vector  $Y_{h,i,n}$ .

### 3.5.2 Results

To compare the performance of our proposed framework with state-of-the-art approaches, we considered the following baselines for the study:

1. The first baseline considered was a [random forest \(RF\)](#) model, a standard time series forecasting algorithm that focuses on building multiple decision trees simultaneously to summarize information from the input sequences and then compares or aggregates the responses from each tree to generate the final output glucose vector. Parameters like the number of trees constructed and the maximum depth of each tree were varied to achieve optimized forecasting predictions. Each tree was constructed with all the training examples and no sub-sampling of the training dataset was involved.
2. The second baseline considered was a [multi-layer perceptron \(MLP\)](#) model. This baseline primarily represents a very primitive form of an artificial neural network that uses 1 hidden layer to map the filtered input glucose sequence to the predicted glucose vector. Parameters like the number of iterations, the size of the hidden layer, the loss function used, and the learning rate were varied to achieve optimized forecasting predictions.
3. For the third baseline, we considered a state-of-the-art study [78] that suggests the use of Transformers as a Seq2Seq approach and infinite mixture models to quantify uncertainties in predictions for accurate forecasting of future glucose trajectories. The study focused on short-term glucose predictions up to 1 hour, and considering the importance of long-term forecasting, we implemented the same model for the long-term scenario of 3 hours.

In Tables 3.2 and 3.3, we summarize the forecasting errors (MAPE and RMSE) obtained with the 3 baselines and our proposed framework for both short-term (1 hour) and long-term scenarios (3 hours) respectively. The predictions from all the 4 cases were executed in multi-output condition, that is, we predict the entire output timeframe all at once.

For both short-term (1 hour) and long-term (3 hours) forecasting scenarios, our framework gave the best performance, followed by Gluformer for the 3 hours case and random forest for the 1 hour case. The MLP and RF baselines along with our framework were implemented on the filtered glucose sequences, whereas, for Gluformer, the original recordings were used as proposed by the protocol in the article [78]. The fact that Gluformer performed worse than RF and MLP in short-term also shows how filtering of glucose sequences improved forecasting performance whilst it removes noise from the original CGM recordings.

Table 3.2: Short-term forecasting metrics of our model versus the baselines considered in this study

Model	MAPE	RMSE
Random forest	4.578	6.004
Multi-layer perceptron	11.45	9.0041
Gluformer	9.599	18.142
Proposed framework	3.893	3.789

Table 3.3: Long-term forecasting metrics of our model versus the baselines considered in this study

Model	MAPE	RMSE
Random forest	49.739	33.714
Multi-layer perceptron	44.171	40.061

Continued on next page

Table 3.3: Long-term forecasting metrics of our model versus the baselines considered in this study (Continued)

Model	MAPE	RMSE
Gluformer	18.604	36.44
Proposed framework	10.416	10.868

### 3.6 A summary of challenges addressed

The challenges addressed during this work with respect to the theme of long-term forecasting of blood glucose are summarized as follows:

1. The architecture proposed in this thesis aims at forecasting blood glucose in the long term for improved performance in applications like insulin dosing for T1D patients. A comparison of our model’s performance with a state-of-the-art study and a few baselines showed that the model performs better in the long-term scenario (for 3 hours) and even in a short-term scenario of just 1 hour.
2. Glucose profiles of patients recorded using conventional CGM devices often suffer from unwanted noise. Especially if we are working on forecasting glucose trajectories, it corrupts and paves the way for errors that are hard to disentangle from the actual predictions. During the data processing stage, we used a low-pass filter that removes this noise from the glucose profiles in the dataset we considered.
3. A glucose sequence fundamentally consists of two components - a high frequency, or seasonal component, that represents the short-term fluctuations, and a low frequency, or trend component, that captures the long-term patterns or trends in glucose levels. Feeding filtered glucose sequences directly to a time series model makes it very difficult for the framework to understand the long-term patterns and the short-term fluctuations simultaneously. We divided the filtered glucose sequences into trend and seasonal sequences that allowed our architecture’s encoders to understand them separately before feeding them to the decoder to produce the final output.

# Chapter 4

## Quantification of personalized sensing delays for a novel CGM-CKM device

### 4.1 Sensing delays in interstitial fluid-based biosensing

Interstitial fluid (ISF) is the fluid underneath the skin that surrounds the cells in body tissues and provides a medium through which nutrients, oxygen, and waste products are exchanged between blood and cells. It constitutes a significant part of the extracellular fluid, occupying the spaces between cells and accounting for about 16% of an individual's body weight. ISF is essential for maintaining the homeostasis of tissues, acting as a buffer and intermediary for the transport of substances. Its composition closely resembles that of plasma but lacks the high protein concentration in blood. Over the years, ISF has proved to be an easily accessible source of biomarkers, as it reflects the metabolic state of the surrounding body tissues. Wearable sensors, developed for analyzing ISF and measuring these biomarkers, offer a minimally invasive approach to continuous health monitoring [73]. These wearable sensors, which are increasingly becoming prevalent, can potentially replace the invasive blood-based measurements [59]. Continuous glucose monitoring (CGM) devices, leveraged with microneedles or minimally invasive probes, are an excellent example of ISF sensors that have transformed diabetes care by facilitating real-time tracking of

glucose levels in human bodies [85] by offering a less painful and more convenient method of monitoring compared to traditional blood-based sampling techniques. However, a significant limitation of wearable ISF sensors is the presence of a delay between the actual concentration of biomarkers in blood and the concentration recorded by the ISF sensors [71]. The delay is mainly caused by variations in transport efficiencies between the ISF and the blood in circulation [50]. Studies have shown a 5 to 20-minute delay in detecting changes in blood glucose compared to the measurements performed by CGM [11]. Yet, the delay for other crucial biomarkers, such as ketone bodies and insulin, remains unknown. Furthermore, the delay can vary among individuals, emphasizing the need to determine personalized delays. To successfully incorporate continuous wearable ISF monitoring into clinical practice, it is essential to be able to understand the delay between ISF and blood levels for clinically significant analytes in a precise and personalized manner.

Over the past years, several machine learning-guided platforms were developed to analyze a patient’s ISF glucose information obtained from CGM devices [94]. As discussed before in Chapter 2, these ISF-based glucose models primarily explored two different problems - classification of glycemic control profiles for alerting patients of the risk of events like post-prandial hypoglycemia (glucose < 72mg/dl) [17, 70, 92] and, introducing time series models that predict glucose levels for the next 15 minutes up to 1 hour. The algorithms used by these models ranged from classical statistical methods like autoregressive models [81] to more advanced artificial neural network frameworks [5], long short-term memory networks [60], Gaussian processes [36], support vector machines [35, 102] and transformer-inspired encoder-decoder models [7, 78] to name a few. However, one downside of these approaches is that they have not accounted for the delays between the blood and ISF measurements and are solely limited to the evaluation of glucose levels. The sensing delays occurring between blood-based and ISF-based biosensor measurements for various analytes primarily arise from the physiological processes involved in their transport. ISF, which surrounds the cells in tissues, receives the target analyte from capillary blood through a diffusion process. The analyte must first leave the capillary blood vessels and diffuse through the capillary walls into the interstitial space. However, this diffusion is relatively slower when compared to the rapid transport of the same analyte in the bloodstream. This eventually results in the introduction of a sensing delay or a time lag in the measurement of ISF levels in comparison to blood-based measurements. The slow diffusion is influenced by multiple factors,

like the permeability of the capillary walls, the distance between the blood capillaries and the interstitial space, and the concentration gradient of the target analyte.

Here, we developed a new wearable platform for continuous monitoring of glucose and ketone bodies (called CGM-CKM device [8, 37]) in ISF. Specifically, patients with type 1 diabetes need to track their ketone levels in addition to their glucose to avoid the risk of diabetes ketoacidosis (DKA). The primary aim is to calculate sensing delays between the levels of glucose and ketones in blood and ISF, personalized for every individual subject under study, and thereby improve the performance of this real-time monitoring device, leading to a better quality of life for people living with diabetes, and specifically, DKA. Subsequently, for our purpose, we explored the use of decision tree-based algorithms to quantify such delays between the ISF-based detection from the CGM-CKM device and blood-based measurements of glucose and ketone levels in different diabetic rats. We achieved this by applying three different boosted decision-tree algorithms that estimated an ISF-level time series corresponding to a reference blood-based time series. Following this, we cross-correlated the predicted ISF response from the device against the blood levels and estimated a delay specific to each tested animal. Determining the sensing delays between ISF and blood will eventually enable personalized auto-adjustment of the CGM-CKM device for future human studies.

## 4.2 Decision tree-based algorithms: a brief introduction

### 4.2.1 Decision trees and random forests

A decision tree is essentially a supervised learning technique that finds its application in both classification and regression problems in machine learning. Figure 4.1 represents a simple illustration of a decision. It primarily constitutes a graph-based structure that works by splitting the data into subsets based on the value of input features, making decisions at each node to reach a conclusion at the leaf nodes. Here, the root node, or the topmost node, represents the entire dataset; internal nodes indicate the nodes where the

data is split based on feature values, and leaf nodes denote the terminal nodes that hold the output (continuous values for regression problems like time series forecasting). Based on the importance of the branches splitting the nodes, a technique called pruning is used to prevent overfitting scenarios.

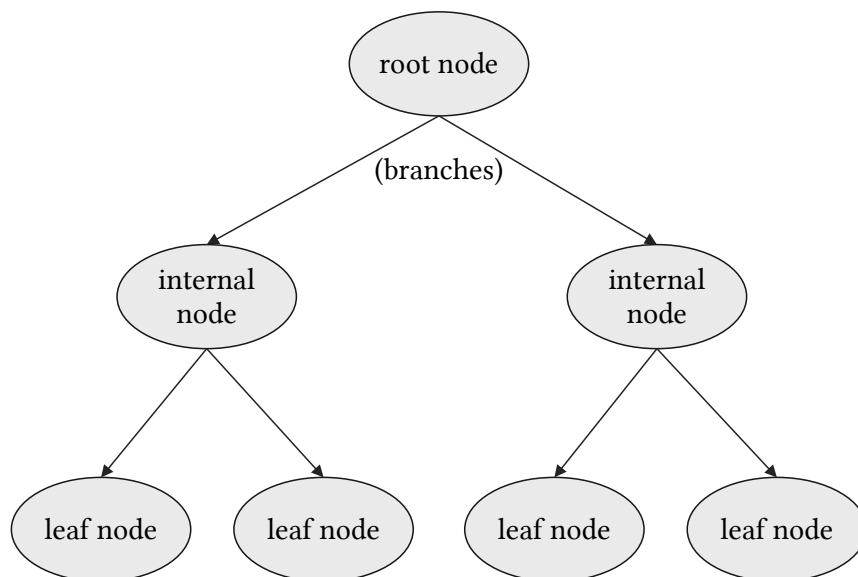


Figure 4.1: A simple illustration of a decision tree

The splitting of nodes in a decision tree model involves the use of an impurity measure (or measure of randomness). For regression problems like forecasting, one common impurity measure used is variance reduction, or in other words, mean squared error, given as,

$$\text{Var}(t) = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (4.1)$$

where  $\bar{y}$  represents the mean target value at any given node  $t$ . Furthermore, to develop a decision tree model, a metric called information gain is used to measure the effectiveness of a feature split. It quantifies the reduction in entropy (or uncertainty) achieved by splitting the data based on that feature. Essentially, it compares the entropy of the dataset before and after the split, with a higher information gain indicating a more effective feature. The

decision tree algorithm uses information gain to select the feature that best separates the data, thus improving the tree’s ability to make accurate predictions. It is given as,

$$\text{IG}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (4.2)$$

where  $S$  represents the dataset,  $A$  indicates the feature, and  $S_v$  is the subset of  $S$  for which feature  $A$  has value  $v$ .

However, applying decision tree models for forecasting problems comes with a few downsides. First, decision trees are prone to overfit the training data, especially when they are allowed to grow to their full depth without any constraints, making them sensitive to noise and resulting in poor generalization to new test data. Second, any slight change in the data can lead to a very different tree structure, which means that decision trees are pretty unstable. This usually occurs because the splitting process in decision trees is based on a greedy approach and is locally optimal, which may not always result in achieving the global optimum. To solve these issues, random forest models offer a solution by considering an ensemble learning approach that combines multiple decision trees together to improve the overall performance and reduce the chances of overfitting. To construct a random forest model, at first, a technique called bootstrapping aggregating or bagging is used that randomly samples the entire dataset with replacement to train each individual tree. Let  $B$  be the number of bootstrapped training datasets  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_B$  chosen from the original dataset  $\mathcal{D}$ . Furthermore, to add feature randomness, a subset of features is randomly selected for each split in each tree. Finally, a voting process is performed that aggregates the individual outputs of all the trees to generate a single final output. In case of forecasting problems, usually the average of the prediction is considered, given as,

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B T_b(x) \quad (4.3)$$

where  $T_b$  represents the output of the tree using the training set  $\mathcal{D}_b$ . With this approach, random forests offer multiple advantages over decision trees. They not only offer reduced chances of overfitting compared to individual trees but also handle larger datasets with higher dimensionalities. Additionally, they are also robust to noisy data and outliers due

to better generalization to newer data points, when compared with decision trees. However, this does not ignore the fact that random forests could be computationally more intensive and less interpretable than decision trees.

## 4.2.2 Boosting algorithms

Boosting algorithms are another class of ensemble learning models that are based on decision trees. Unlike random forests that consider individual decision trees and aggregate their predictions together, boosting algorithms focus on sequentially constructing multiple decision trees to achieve optimal performance. In the following subsections, we discuss a few commonly used boosting algorithms for time series modeling.

### Gradient boosting

In gradient boosting [33], just like random forests, an ensemble of decision tree models are built. However, they are introduced in a sequential manner such that every new decision tree corrects the errors in prediction of the previous tree. The main idea is to minimize a loss function by adding models that predict the residuals, or in other words, errors of the previous models. To construct a gradient boosting ensemble, we first assume  $n$  as the number of trees to construct and  $N$  as the number of training examples. Following this, we define a "weak" learner, or a weak predictor,  $F_0$  to initialize the model. A common way to define  $F_0$  for regression problems is to consider the mean or average of target values, given as,

$$F_0 = \frac{1}{N} \sum_{i=1}^N y_i \quad (4.4)$$

Now, let  $i = 1, 2, \dots, n$  indicate any given step of iteration and  $j = 1, 2, \dots, N$  as the index for any fixed training example. For every  $i$ , we first compute the residuals for each training example based on the model constructed in the previous step as,

$$r_j^{(i)} = -\frac{\delta L(y_j, F_{i-1}(x_j))}{\delta F_{i-1}(x_j)} \quad (4.5)$$

where,  $L(., .)$  is any differentiable loss function. Following this, we fit a new model  $h_i(x)$  to the residuals  $r_j^{(i)}$ . The aim of this model tries is to predict the residuals. Finally we update the model  $F_i$  by the following additive equation,

$$F_i(x) = F_{i-1}(x) + \alpha h_i(x) \tag{4.6}$$

where,  $\alpha$  represents the learning rate that controls the importance of the update the newer model  $F_i$  makes over  $F_{i-1}$ .

The choice of the loss function  $L$  to learn the residuals could range from simpler functions like squared loss to advanced measures like quantile loss. For example, in case of squared loss, we have,

$$L(y_j, F_{i-1}(x_j)) = \frac{1}{2}(y_j - F_{i-1}(x_j))^2 \tag{4.7}$$

In this particular scenario, the residuals will be given as,

$$r_j^{(i)} = y_j - F_{i-1}(x_j) \tag{4.8}$$

that is, the difference between the actual ground truth and the predicted values by the model  $F_{i-1}$  at the previous step  $i - 1$ . For practical considerations, the parameters that are varied to find the best possible gradient boosting ensemble for the given dataset include learning rate  $\alpha$ , the number of iterations  $n$ , the depth of each tree built, the loss function used to learn the residuals. In many cases, instead of considering the entire training dataset for learning each step, a fixed percentage (called as the subsampling fraction) of all the training examples chosen at random with replacement are considered for every iteration to prevent any further chances of overfitting. Figure 4.2 shows a simple illustration of the working of a gradient boosting model.

## XGBoost

Implementing gradient boosting for forecasting problems, however, comes with a few downsides. Due to sequential learning of multiple decision trees, gradient boosting is often intensive and requires a large training time if the number of iterations is significantly high. In this regard, [extreme gradient boosting \(XGBoost\)](#) [20], a widely used variant of

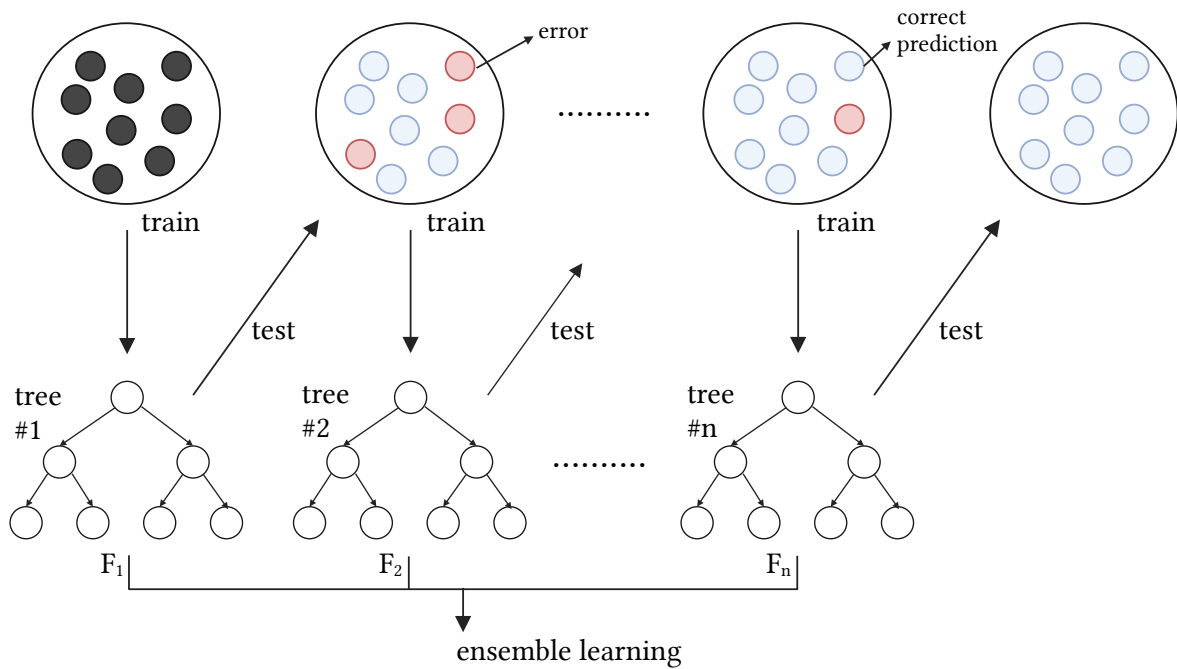


Figure 4.2: A simple illustration of the gradient boosting algorithm

gradient boosting, offers an improvised solution to this problem. XGBoost extends the basic framework of gradient boosting with several enhancements that make it faster, more efficient, and more robust. One key improvisation lies in the optimization technique employed by XGBoost, which includes parallel processing and hardware optimization that results in significantly reduced training times compared to traditional gradient boosting. In addition to this, XGBoost also incorporates advanced regularization methods (L1 and L2 regularization) that help prevent overfitting and enable better generalization to newer data points.

In contrast to the loss function used in gradient boosting, the objective term used for XGBoost can be represented of the form,

$$\mathcal{L} = \sum_{j=1}^N L(y_j, \hat{y}_j) + \sum_{i=1}^n \Omega(F_i) \quad (4.9)$$

where,  $n$  is the number of trees or iterations,  $N$  represents the number of training examples considered,  $L$  is the differentiable loss term used earlier for gradient boosting and  $F_i$  represents the  $i$ -th tree constructed, given  $i = 1, 2, \dots, n$ .  $\Omega$  represents the regularization term, given as,

$$\Omega(F_i) = \gamma T_i + \frac{1}{2} \lambda \sum_{k=1}^{T_i} w_k^2 \quad (4.10)$$

where,  $T_i$  indicates the number of leaves in the tree  $F_i$ ,  $w_k$  denotes the weight of the  $k$ -th leaf in the tree  $F_i$  and  $\lambda$  is the regularization constant. In simpler terms, the regularization term  $\Omega(F_i)$  includes the effect of the weights learned in every iteration, and is eventually added to the loss term  $L$  to constitute the minimization objective.

Another significant difference in comparison to gradient boosting is XGBoost’s ability to handle missing values and its implementation of shrinkage (learning rate reduction) and column subsampling, which often contributes to better model accuracy and robustness. The built-in support for early stopping and integrated cross-validation further streamlines the model tuning process, making XGBoost more user-friendly and efficient than traditional gradient boosting methods. Overall, these enhancements make XGBoost a more robust and versatile tool for many machine learning tasks.

## CatBoost

[Categorical boosting \(CatBoost\)](#) [68] is another widely-used decision tree-based framework inspired by the traditional gradient boosting algorithm that finds its application in both classification and regression problems in machine learning. CatBoost primarily extends the framework used in gradient boosting and includes the objective function functionalized with a regularization term  $\Omega$ , implemented in XGBoost algorithms, to handle any given dataset that involves the use of categorical features. It uses multiple techniques like Bayesian smoothing and one-hot encoding to transform categorical targets into numerical values.

To further improve computational efficiency, CatBoost considers two more factors. First, it implements ordered boosting, which means the decision trees are constructed using ordered statistics to mitigate overfitting. In traditional gradient boosting or XGBoost, the same data is used to train and evaluate trees, but in ordered boosting, the

data is split into two parts, one which is used to determine the splitting of the nodes and the other to evaluate it. Second, CatBoost grows symmetric trees, which means that the same splitting criterion is applied to both branches of a node. This ensures a balanced growth of trees in the ensemble and a stable training of the model, improving the overall performance.

## 4.3 Mathematical background

### 4.3.1 Problem formulation

The primary aim of this study is to quantify sensing delays between blood-based and ISF measurements, personalized for every subject under study. For this, we consider  $n \in \mathbb{N}$  as the number of tested subjects in the dataset for glucose or ketone measurements. We defined  $[n] := \{1, 2, \dots, n\}$ . Let  $k$  indicate any subjects chosen randomly from the set  $[n]$  and  $m_k$  be the number of times that subject  $k$  was tested. A dataset with examples of the form

$$\mathcal{D} = \{(\mathbf{x}_{i,k}, y_{i,k})\}_{i \in [m_k], k \in [n]} \quad (4.11)$$

was defined.  $\mathbf{x}_{i,k} \in \mathbb{R}^s$  was the vector of all input features extracted from the CKM or CGM sensor platform,  $s \in \mathbb{N}$  was the number of features in  $\mathbf{x}_{i,k}$ , and  $y_{i,k} \in \mathbb{R}$  indicated the corresponding output analyte concentration. Here, the primary objectives were: (a) to construct a mapping (or function)  $F_k : \mathbb{R}^s \rightarrow \mathbb{R}$  for every individual  $k$  that learns from the vector of input features from CKM or CGM sensor  $\mathbf{x}_{i,k}$  to make an estimation for the ketone or glucose levels, respectively, in blood  $y_{i,k}$ , and (b) to find an optimal delay  $\tau_k$  by which the subject  $k$ 's predicted ISF levels  $F_k(\mathbf{x}_{i,k}; \theta_k)$  should be delayed to match with its corresponding blood concentration  $y_{i,k}$  as close as possible.  $\theta_k$  was the set of learned parameters of the model  $F_k$  for subject  $k$ .

### 4.3.2 Quantifying time-lags between blood and ISF

To learn the personalized ISF prediction models  $F_k$  and consequently find the delay  $\tau_k$ , we considered a leave-one-subject-out scheme, where, at a time, we fixed only 1 subject (i.e.,

$k$ ). For this  $k$ , let  $k^*$  represent any other subject in the dataset  $\mathcal{D}$  other than  $k$ , that is,  $k^* \in [n] \setminus \{k\}$ . Next, we divided  $\mathcal{D}$  into two partitions - a training dataset  $\mathcal{D}_{train}^k$  and a test dataset  $\mathcal{D}_{test}^k$ . We considered,

$$\mathcal{D}_{train}^k = \mathcal{D} \setminus k = \{(\mathbf{x}_{i,k^*}, y_{i,k^*})\}_{i \in [m_{k^*}], k^* \in [n] \setminus \{k\}} \text{ and,} \quad (4.12)$$

$$\mathcal{D}_{test}^k = \mathcal{D}^k = \{(\mathbf{x}_{i,k}, y_{i,k})\}_{i \in [m_k]} \quad (4.13)$$

It should be noted that  $\mathcal{D}_{train}^k$  and  $\mathcal{D}_{test}^k$  were mutually exhaustive and disjoint. We then used all the examples in  $\mathcal{D}_{train}^k$  to train  $F_k$  using a learner algorithm. It should be noted that  $F_k$  varies for every  $k$  based on the number of trees and loss function, emphasizing on personalized predictions.

To test our model  $F_k$  and find the delay for subject  $k$ , we defined two time series. First, let

$$\mathbf{y}_k^{blood} := \{y_{i,k}\}_{i \in [m^k]} \quad (4.14)$$

be the series of blood-based levels of  $k$ , to be used as ground truth for comparison with ISF levels. Second, we consider,

$$\mathbf{y}_k^{ISF} := \{F_k(\mathbf{x}_{i,k}; \theta_k)\}_{i \in [m^k]} \quad (4.15)$$

as the series of the ISF-level predictions for  $k$ . Essentially, we define  $\mathbf{t}_k = \{t_{i,k}\}_{\forall i \in [m^k]}$  as the set of all time instants at which an ISF experiment was performed for  $k$ . For positive scalar  $\tau \in \mathbb{R}^+$ , we defined an operation, element-wise addition given as  $\{t_{i,k} + \tau\}_{i \in [m^k]}$  (element-wise addition). By definition of ISF delays,  $\mathbf{y}_k^{ISF}$  ideally should be a delayed version of  $\mathbf{y}_k^{blood}$ . To find this delay, first, we linearly interpolated  $\mathbf{y}_k^{ISF}$  with respect to  $\mathbf{t}_k$ . From the interpolated curve for ISF response, we re-sampled it, but instead, at times  $\{t_{i,k} + \tau\}_{i \in [m^k]}$ , such that  $\tau \in [\tau_{max}] \cup \{0\}$ ,  $\tau_{max}$  being the physiologically acceptable limit in minutes for ISF delays. By re-sampling at the modified time instants, the original ISF response  $\mathbf{y}_k^{ISF}$  was now delayed by  $\tau$  minutes to give us a delayed response - we labeled it as  $\mathbf{y}_{k,\tau}^{ISF}$ . We chose  $\tau_{max}$ , the maximum limit of any delay to be 50 minutes. Finally, we compared  $\mathbf{y}_{k,\tau}^{ISF}$  with  $\mathbf{y}_k^{blood}$  and simultaneously, varied  $\tau$  in  $[\tau_{max}] \cup \{0\}$  to find the optimal delay  $\tau = \tau_k$  for subject  $k$  that minimizes the error metrics (see Section 4.4.3 for more

information). We propose that this  $\tau_k$  is the delay during the transmission of bioanalytes from blood to ISF in subject  $k$ . If  $\tau_k = 0$ , it means transport of the analyte from blood to ISF takes place in negligible time, whereas if  $\tau_k > 0$ , it means transmission in  $k$  requires a significantly higher duration.

## 4.4 Experiments

### 4.4.1 Data collection

For both ketone and glucose monitoring, in our study, the number of diabetic rats tested was  $n = 4$ . In case of CKM, the device output for a certain subject  $k$  at every time of experiment  $t_{i,k}$  consisted of two time series - a pre-oxidation current  $\mathbf{p}_{i,k}$  representing the amount of ketone recognition element present at the start of ketone detection, and a detection current  $\mathbf{d}_{i,k}$ , indicating the amount of recognition element left after detection. We annotated the 4 subjects for ketone sensing as - Rat 1, Rat 2, Rat 3 and Rat 4 as in Table . Figure 4.3(a) gives a sample pair of  $\mathbf{p}_{i,k}$  and  $\mathbf{d}_{i,k}$  profiles for Rat 2 ( $k = 2$  for ketone sensor) at time  $t_{i,2} = 93$  minutes. For CKM, however, due to experimental restrictions, the blood-based measurements were performed at a different set of time of experiments for each subject  $k$  in comparison to ISF experiments. Let these time instants for blood ketone recordings be  $\hat{\mathbf{t}}_k = \{\hat{t}_{j,k}\}_{j \in [m_k^{blood}]}$  (in minutes),  $m_k^{blood}$  being the number of times blood based ketone measurements taken for subject  $k$ . In other words, cardinality of  $\mathbf{t}_k \cup \hat{\mathbf{t}}_k > m_k$  and  $> m_k^{blood}$ , for every subject  $k$ .

To solve this disparity, the blood ketone levels, at first, were linearly interpolated with respect to  $\hat{\mathbf{t}}_k$ , as in Figure 4.3(b). From this curve, the blood ketone values were then re-sampled, but instead, at times  $\mathbf{t}_k$ , for comparison with our ISF-based ketone predictions. The profiles were recorded every 0.1s.  $\mathbf{p}_{i,k}$  was measured up to 20s and  $\mathbf{d}_{i,k}$  up to 50s for every  $k$ .

For CGM, the device output for each subject  $k$  had two steps. First, a time-varying current  $\mathbf{g}_{i,k}$  indicating the glucose levels in  $k$  at time  $t_{i,k}$  recorded every 0.1s for up to 60s, and second, a current-voltage loop for every  $k$  to interpret the scale of sensor response in  $k$  and thereby calibrate  $\mathbf{g}_{i,k}$ . We annotated the 4 animal subjects on which the glucose

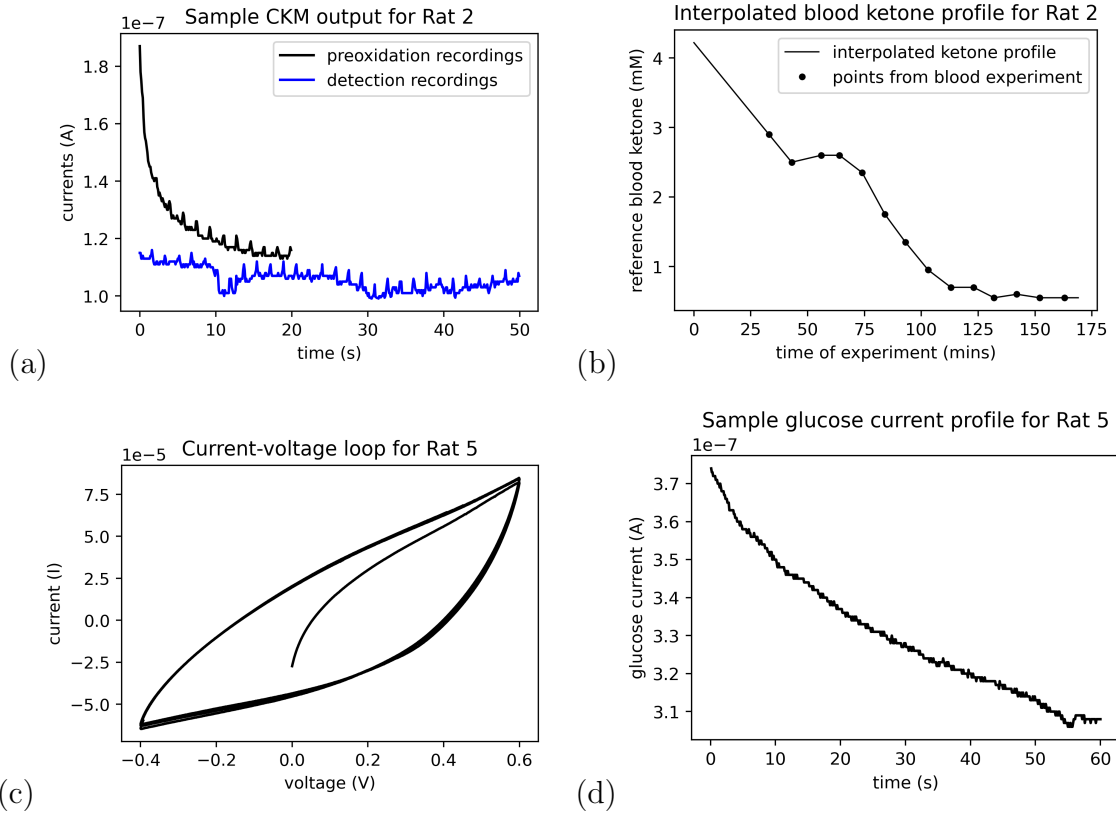


Figure 4.3: (a): A sample CKM output - black and blue curves indicate preoxidation and detection profiles; (b) the corresponding blood ketone graph; (c): a sample CGM current-voltage loop; (d): the corresponding glucose current

sensing was performed as - Rat 5, Rat 6, Rat 7 and Rat 8. Figures 4.3(c) and (d) show a sample current-voltage loop and its corresponding  $\mathbf{g}_{i,k}$  profile. Because the  $\mathbf{g}_{i,k}$  current ranges varied from one in vivo rat to another, they were brought to the same level by defining an adjustment factor. For this, from the current-voltage loop for every  $k$ , we recorded the average current value corresponding to the maximum voltage level  $0.6V$  for every rat. Let this sampled current be  $G_{0.6,k}$ . Choosing  $G_{0.6,1}$  for the Rat 5 ( $k = 1$  for glucose) as the reference, we divided the  $G_{0.6,k}$  values for all the other rats 6, 7 and 8 ( $k = 2, 3, 4$  for glucose respectively) by  $G_{0.6,1}$  to find their individual scaling, or adjustment factors. We later used these factors to re-scale their individual  $\mathbf{g}_{i,k}$  profiles to a comparable

range. Also, unlike the ketone measurements, in case of CGM, the reference blood glucose levels were recorded at the same instants as the times of ISF experiments, that is, for our glucose experiments,  $\mathbf{t}_k = \hat{\mathbf{t}}_k$  and  $m_k = m_k^{blood}$  in this case.

#### 4.4.2 Data preprocessing and input features

The varying physiological conditions inside each diabetic rat incorporated noise into the  $\mathbf{p}_{i,k}$  and  $\mathbf{d}_{i,k}$  profiles during ketone sensing. As such, we used a moving average filter to smooth both the current profiles. The  $\mathbf{g}_{i,k}$  profiles for glucose however had negligible noise and did not require any such low-pass filters. To define the input features in  $\mathbf{x}_{i,k}$ , we take samples from the output current profiles from the CGM-CKM platform. For ketone, we sampled  $\mathbf{p}_{i,k}$  and  $\mathbf{d}_{i,k}$  profiles at their end points. Let these samples be  $P_{i,k}$  and  $D_{i,k}$  respectively. For glucose, we sampled  $\mathbf{g}_{i,k}$  at 10s and 40s. Let these samples be  $g_{i,k,10}$  and  $g_{i,k,40}$ . For CKM, the ratio  $\frac{D_{i,k}}{P_{i,k}}$  indicates the percentage of recognition element left after detecting ketone bodies - directly proportional to ketone concentration in subject  $k$ . Administration of insulin doses as a remedial action against DKA during our ISF experiments ensured a drop in ketone levels with time and hence,  $\frac{D_{i,k}}{P_{i,k}}$  ideally should decrease monotonically with time  $t_{i,k}$ . Similarly for CGM,  $g_{i,k,10} + g_{i,k,40}$  being directly proportional to glucose levels, should also decrease with time. Presence of outliers often disrupts these trends, and so, we used a robust framework to remove such outliers. To remove outliers for CKM, we first fixed any rat  $k$  and performed a linear fit for  $\left\{ \frac{D_{i,k}}{P_{i,k}} \right\}_{i \in [m_k]}$  versus  $\mathbf{t}_k$  as in Figure 4.4 and rejected those ratios and their time instants which lied outside the 95% confidence interval (shown in red) of the linear fit. For CGM, we performed a similar task using a linear fit for  $\{g_{i,k,10} + g_{i,k,40}\}_{i \in [m_k]}$  versus  $\mathbf{t}_k$  and defining similar confidence boundaries for each rat  $k$  to remove the outliers. Following this, we moved on to train  $F_k$  for every rat  $k$  for both CGM-CKM. Finally, for ketone, we considered  $\mathbf{x}_{i,k} = [P_{i,k} \ D_{i,k} \ t_{i,k}]$  and for glucose,  $\mathbf{x}_{i,k} = [g_{i,k,10} + g_{i,k,40} \ t_{i,k}]$ .

#### 4.4.3 Performance metrics

To test the performance of  $F_k$  for subject  $k$ , we applied it on an exclusively new set of test examples  $\mathcal{D}_{test}^k$ . To compare  $\mathbf{y}_k^{ISF}$  with  $\mathbf{y}_k^{blood}$ , we considered two metrics, [mean absolute](#)

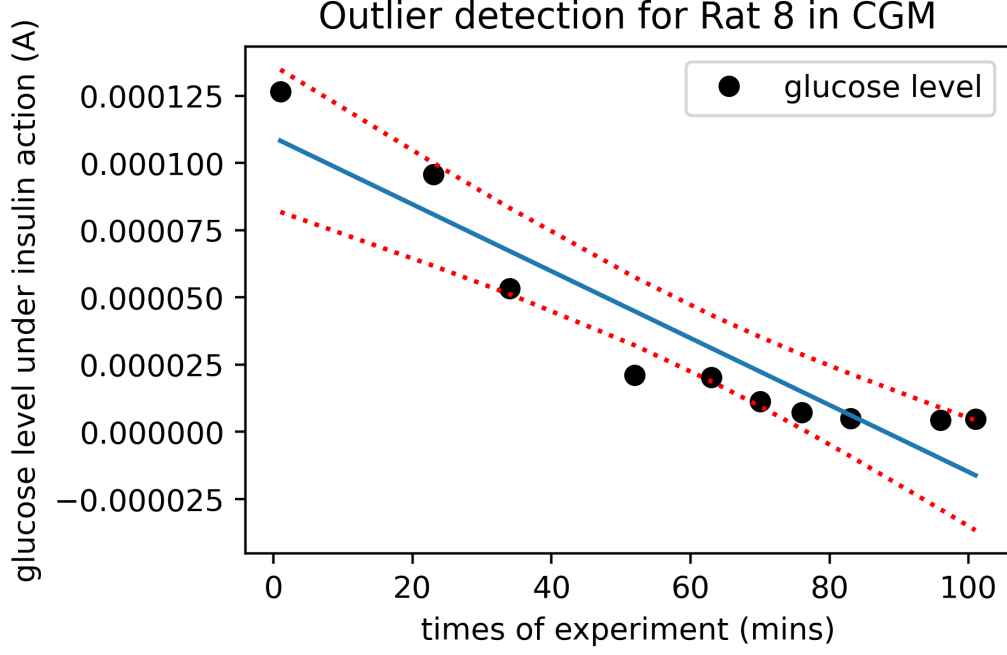


Figure 4.4: A sample plot for outlier detection

difference (MAD) (absolute error), defined here as  $MAD_k$  and **mean absolute relative difference (MARD)** (percentage error), defined here as  $MARD_k$ . Both of these metrics are widely-accepted to measure the accuracy of continuous monitoring biosensors, especially for glucose [40, 109]. While comparing delayed ISF  $\mathbf{y}_{k,\tau}^{ISF}$  with  $\mathbf{y}_k^{blood}$ , we annotated the metrics as  $MAD_{k,\tau}$  and  $MARD_{k,\tau}$  respectively. For every  $k$ , we varied  $\tau$  to find the optimal delay  $\tau = \tau_k$  that minimizes  $MARD_{k,\tau}$  and  $MAD_{k,\tau}$  between  $\mathbf{y}_{k,\tau}^{ISF}$  and ground truth  $\mathbf{y}_k^{blood}$ . Here,

$$MARD_k = \frac{1}{m_k} \sum_{i=1}^{m_k} \left| \frac{y_{i,k} - F_k(\mathbf{x}_{i,k}; \theta_k)}{y_{i,k}} \right| \quad (4.16)$$

and,

$$MAD_k = \frac{1}{m_k} \sum_{i=1}^{m_k} |y_{i,k} - F_k(\mathbf{x}_{i,k}; \theta_k)| \quad (4.17)$$

For CKM, we were also interested if the model can differentiate between normal ketosis ( $F_k(\mathbf{x}_{i,k}; \theta_k) \leq 1.5mM$ ) and hyperketonemia ( $F_k(\mathbf{x}_{i,k}; \theta_k) > 1.5mM$ ). Hence, for ketone, we considered a third metric,  $mistakes_{k,\tau}$ , that counts the number of misclassified examples in the binary classification between normal ketosis and hyperketonemia. Here,

$$mistakes_k = \sum_{i=1}^{m_k} \left( \mathbf{1}_{y_{i,k} > 1.5} \neq \mathbf{1}_{F_k(\mathbf{x}_{i,k}; \theta_k) > 1.5} \right) \vee \left( \mathbf{1}_{y_{i,k} \leq 1.5} \neq \mathbf{1}_{F_k(\mathbf{x}_{i,k}; \theta_k) \leq 1.5} \right) \quad (4.18)$$

where  $\vee$  indicates the logical "OR" operation. The function  $\mathbf{1}$  would equate to 1 if the condition in the suffix is true, otherwise 0.

## 4.5 Results

Figure 4.5 represents an overall schematic of our approach. As described previously in Section 4.3, we individually learn  $F_k$  for every  $k$ . For this, to begin with, we implemented 3 different decision-tree based algorithms - gradient boosting, CatBoost and XGBoost. Parameters like the number of trees constructed, the maximum depth of each tree, and the loss function used were varied to achieve optimized glucose and ketone predictions. Another parameter that was also varied was the subsampling function, which indicates that a certain percentage of all training examples was chosen at random with replacement to construct each tree in every iteration sequentially. We then computed the performance metrics,  $MAD$  and  $MARD$ , for different values of  $\tau$  with 1 minute interval for both CKM and CGM. By varying  $\tau$ , we chose the optimum delay as the time lag that minimized  $MAD$  as shown in Figure 4.6. Tables 4.1 and 4.2 list the estimated optimal sensing delays for each rat for the three different learning algorithms for both CKM and CGM respectively. For every individual rat, the delays varied approximately in the range 9 – 44 minutes for the CKM device and in the range 0 – 22 minutes for the CGM device. The differences between the delays in individual rats clearly establishes the fact that there is an underlying variability in ketone and glucose kinetics, even under controlled conditions with genetically similar animals. This has important implications for human patients, who are genetically diverse and have different environmental conditions, intensifying this inter-individual variability. At the same time, the delays  $\tau_k$  obtained for every rat  $k$  for both the CKM and CGM devices

for all the three algorithms were within the same range, which indicates the consistency of our frameworks in quantifying delays for the CGM-CKM device. Additionally, the range within which the delays varied for our CKM evaluation differed from the range in the case of CGM. This can be attributed to how sensing delays occur while using ISF-based biosensors. During the diffusion process of analytes from blood capillaries to ISF, the time taken for diffusion is proportional to the square of the distance traveled to diffuse [84]. Simultaneously, this diffusion time also depends on features reflecting the nature of the analyte, such as molecular weight, size, shape, solubility, and so on, and features reflecting the location of the ISF measurement like the nature of the medium, temperature, and other surrounding physical conditions varying across the specific subject. All these factors account for two separate ranges of sensing delays for CKM and CGM.

For both CGM and CKM, we combined the delayed ISF predictions and the reference blood values for all subjects and determined the overall *MAD* and *MARD* for each algorithm as mentioned in Table 4.3. We also compare the overall *MAD* and *MARD* obtained for gradient boosting with state-of-the approaches. Table 4.4 gives a list of popular CGM sensors available in market currently and their reported *MARD* scores for varying T1D subjects. As evident, our overall best *MARD* of 10.89% is at par with these prior works. Similarly, we compare our performance for CKM with a recent paper that used similar ISF-based technologies for detecting ketone levels in a minimally-invasive fashion. As prescribed in its protocol, we divided all the predicted points into normal ketosis range ( $< 1.5mM$ ) and hyperketonemic range ( $\geq 1.5mM$ ). We calculated *MAD* for the first range and *MARD* for the second. As such, the scores for both the paper and our framework were similar as shown in Table 4.5.

We further correlated the delayed ISF predictions obtained from applying gradient boosting with the blood references for both CKM and CGM as shown in Figure 4.7. From this, we obtained correlation values of 0.941 for CKM and 0.790 for the CGM device, when compared with blood-based measurements.

Table 4.1: A summary of the best possible individual delays  $\tau_k$  and their corresponding metrics for CKM

Target	Rat	MAD (mM) / MARD (%)			$\tau_k$ (mins)		
		Grad. Boost	CatBoost	XGBoost	Grad. Boost	CatBoost	XGBoost
Ketone	Rat 1	0.100 / 7.714	0.164 / 10.982	0.192 / 12.803	31	23	18
	Rat 2	0.144 / 8.500	0.184 / 9.739	0.114 / 10.254	9	11	1
	Rat 3	0.352 / 20.953	0.376 / 24.542	0.541 / 35.701	15	7	11
	Rat 4	0.246 / 6.379	0.328 / 7.109	0.254 / 6.162	29	44	31

Table 4.2: A summary of the best possible individual delays  $\tau_k$  and their corresponding metrics for CGM

Target	Rat	MAD (mM) / MARD (%)			$\tau_k$ (mins)		
		Grad. Boost	CatBoost	XGBoost	Grad. Boost	CatBoost	XGBoost
Glucose	Rat 5	4.147 / 18.849	3.643 / 20.966	4.000 / 23.428	0	0	0
	Rat 6	1.208 / 13.502	3.104 / 32.851	4.384 / 45.599	13	9	9
	Rat 7	1.143 / 5.378	0.799 / 3.457	2.497 / 12.374	22	19	10
	Rat 8	0.661 / 4.075	0.682 / 4.477	0.655 / 4.070	3	10	10

Table 4.3: A summary of the overall *MARD* and *MAD* for all the algorithms

Target	Algorithms used	Performance metrics			
		Overall (mM)	MAD	Overall (%)	MARD
Ketone	Gradient Boosting	0.2261		10.8887	
	XGBoost	0.2915		16.3618	
	CatBoost	0.2849		13.4851	
Glucose	Gradient Boosting	1.5272		8.3838	
	XGBoost	2.4439		16.1028	
	CatBoost	1.5588		10.6467	

Table 4.4: A literature summary of the MARD performances of some widely-used CGM systems implemented on human subjects

CGM system used	Reference	Number of subjects	Overall MARD
Abbott Freestyle Libre	[4]	12	13.2%
Medtronic Guardian 3	[22]	88	9.6%
Senseonics Eversense	[23]	90	8.8%
Abbott Navigator I	[24]	6	11.8%
Dexcom G4 Platinum	[65]	10	10.9%
Dexcom G6	[96]	262	10%
Dexcom G7	[34]	318	8.2%
Medtronic SofSensor	[98]	Adults: 71,	Adults: 9.9%,
		children: 61	children: 10%

Table 4.5: Comparison of the overall *MAD* and *MARD* obtained while implementing our ketone sensing framework with a state-of-the-art study [6]

Condition	[6]	Our approach
overall <i>MAD</i> for ketone levels < 1.5mM	<b>0.129</b>	0.1799
overall <i>MARD</i> for ketone levels $\geq$ 1.5mM	14.4%	<b>10.1083%</b>

## 4.6 A summary of challenges addressed

The challenges addressed during this work with respect to the quantification of sensing delays for glucose and ketone monitoring in DKA patients are summarized as follows:

1. The framework proposed in this chapter aims to evaluate the sensing delays or time lags that occur during the prediction of analyte concentrations during minimally invasive ISF-based monitoring when compared with blood-based standards. The calculated time lags varied from one subject to another, proving that these delays are often influenced by the genetic variabilities across a multitude of subjects within the same study cohort. The delays also remained in close range for all three algorithms implemented, showing the consistency of our framework.
2. The work explores the feasibility of using machine learning frameworks as a time series problem to estimate blood ketone levels from the ISF for the first time. Future implications of this include having more subjects in the study to include more data points per individual and extending the framework to other algorithms.
3. Personalization is a key aspect of our approach, and it has significant implications for clinical studies. By adjusting treatments to suit the individual patient’s unique genetic constitution, lifestyle, and health conditions, it is possible to enhance treatment accuracy, reduce adverse effects, and improve patient outcomes. Our use of the

quantified sensing delays to calculate personalized sensor responses for every rat is a clear example of this. As the delay plots show, the variation of our performance metrics with time had a global minimum, indicating improved precision. This is a powerful testament to the potential of personalization in improving the quality of care.

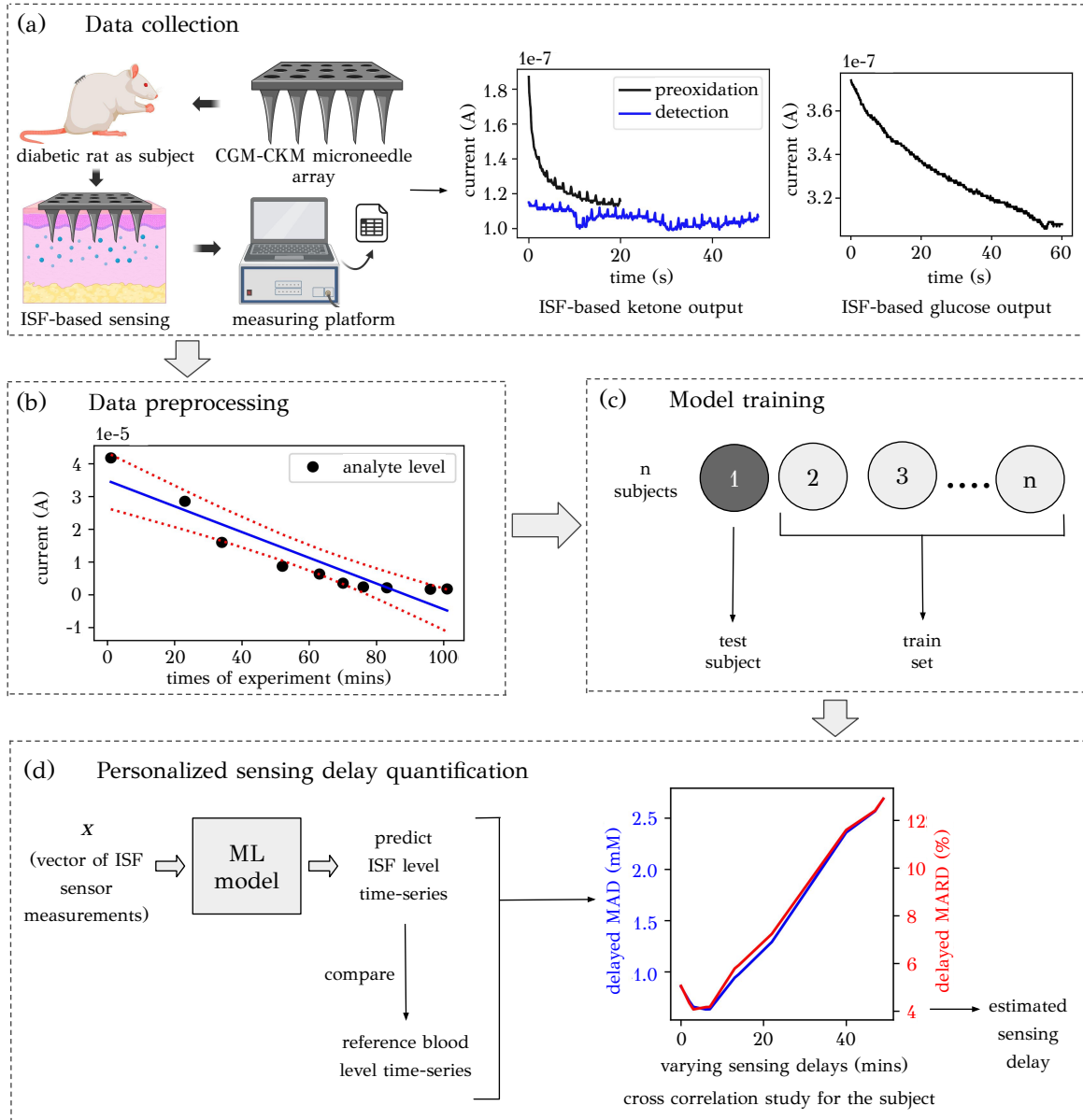
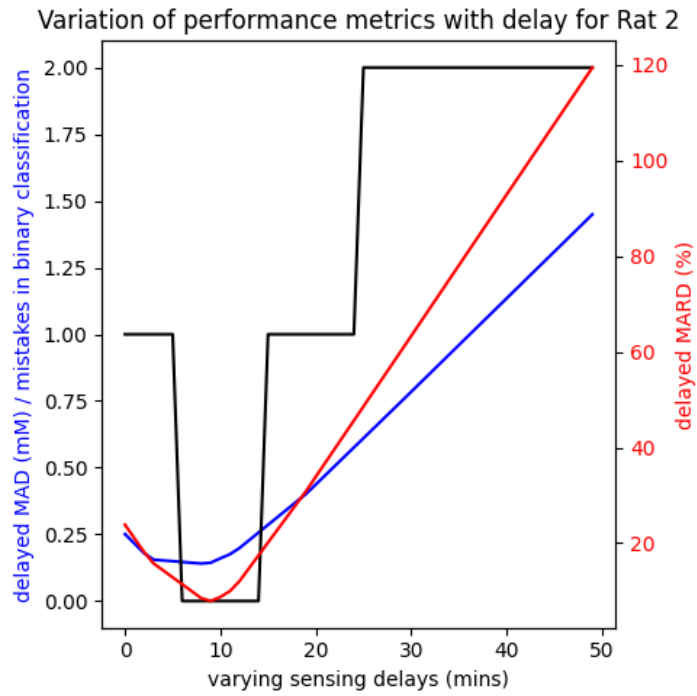
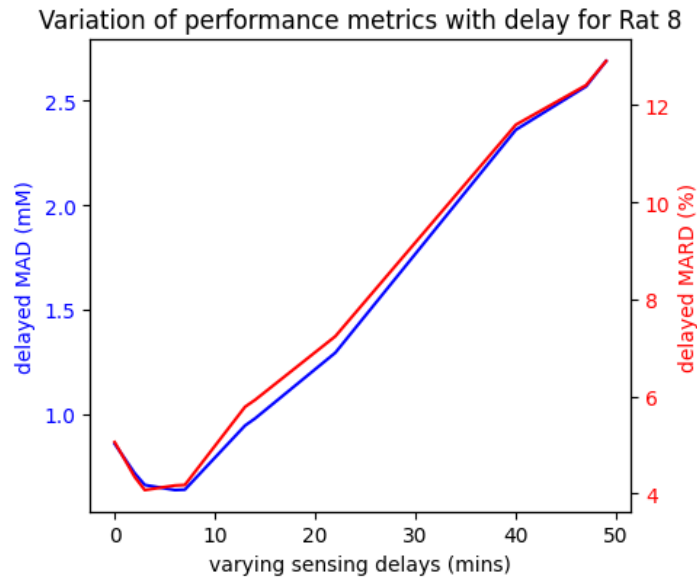


Figure 4.5: An illustration of our framework for evaluating sensing delays and using them to provide personalized responses for our CGM-CKM device



(a)



(b)

Figure 4.6: Sample plots for variation of metrics with delays for (a) CKM and, (b) CGM

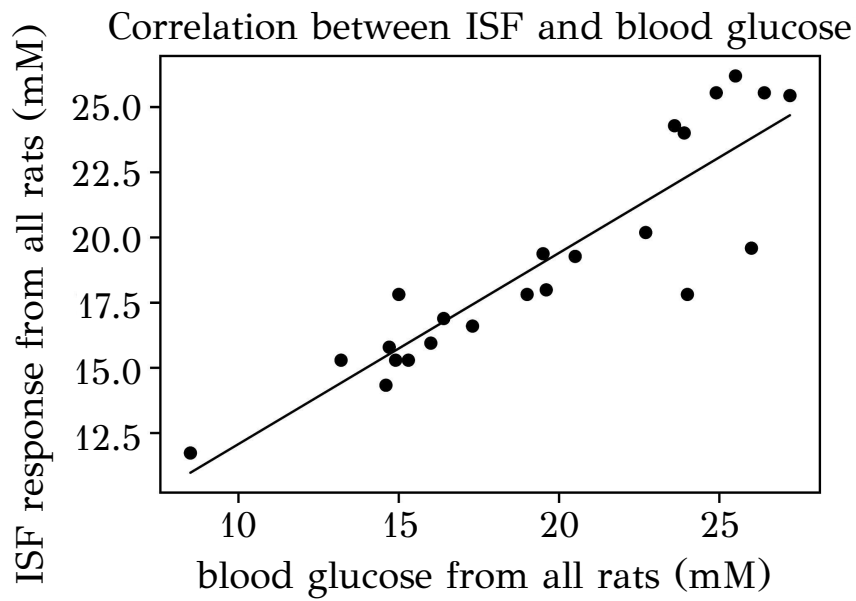
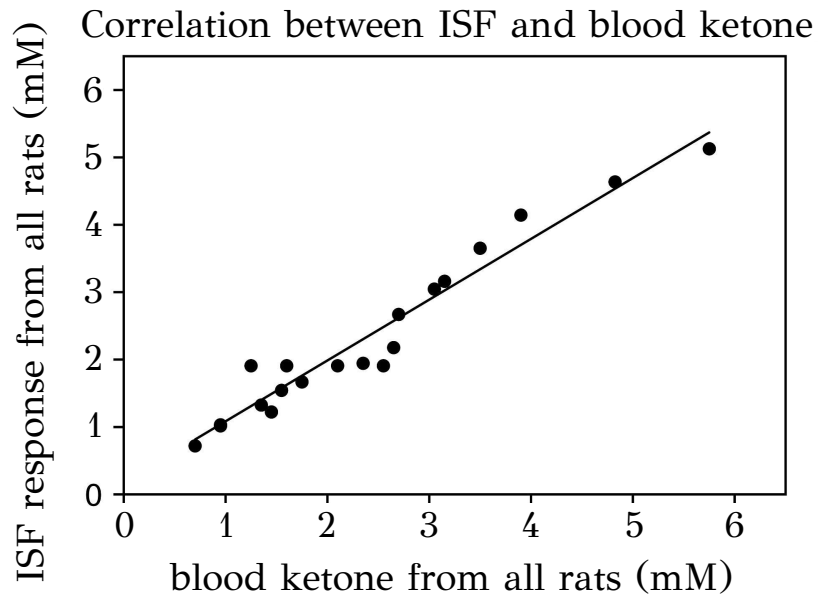


Figure 4.7: Overall correlation for CKM (left) and CGM (right)

# Chapter 5

## Final remarks

Through this thesis, we have primarily explored how machine learning frameworks have impacted and influenced existing therapies like continuous monitoring of analytes for the prevention of type 1 diabetes and its complication, diabetic ketoacidosis. In the first half, we discussed the use of a multitude of algorithms that have been used for forecasting blood glucose levels and predicting any future risks based on continuous glucose monitoring data. However, these prior works only focused on short-term prediction up to 1 hour, and in this regard, we introduce a robust, sequence-to-sequence framework that extends this prediction horizon to 3 hours. These long-term glucose predictions will give health professionals and existing insulin delivery platforms more room to make better and more precise decisions regarding insulin dosing for the concerned patients. However, this work has many scopes for improvement. First, during comparison, we showed how our model gives a superior performance with respect to a few baselines and a state-of-the-art article in both long-term and short-term scenarios. However, we want to extend this by comparing it with other sequence-to-sequence frameworks that have also been used for applications outside glucose forecasting, like informer [107] and autoformer [101]. Secondly, for our study, we used a data cohort that included only 38 patients. In the future, we want to make a similar application and comparison of performance using other datasets recruiting type 1 diabetic patients. Another avenue could be pre-training our model on one dataset and tuning it based on another to allow our long-term glucose forecasting framework to learn from a more diverse population.

In this thesis, we further discussed about quantifying sensing delays or time lags observed in interstitial fluid-based sensing compared to blood-based standards. We implemented our framework with respect to continuous glucose and ketone monitoring for diabetic ketoacidosis patients. However, there are multiple plans to extend this work forward. First, the evaluation was limited to only four diabetic rats for both glucose and ketone. We are currently expanding our efforts behind data collection with our CGM-CKM device to obtain longer time-series sequences from a larger number of diabetic rats in our study cohort. The increased number of collected measurements will be used to enhance the precision of our algorithms for assessing these sensing delays and developing new representation learning algorithms for time-series data obtained from such wearable devices. Secondly, we plan to extend this framework to evaluate the performance of the CGM-CKM device in human patients and utilize the developed predictive framework to measure time lags in human subjects. Lastly, we quantified the personalized sensing delays using only decision tree-based algorithms. In the future, we aim at expanding this to other state-of-the-art machine learning algorithms like recurrent neural networks, to improve the generalizability of the work.

# References

- [1] *Diabetes-Related Ketoacidosis (DKA)*. Cleveland Clinic.
- [2] *Artificial Pancreas*. National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), 2021.
- [3] *Types of Insulin*. Centers for Disease Control and Prevention (CDC), 2022.
- [4] Felix Aberer, Martin Hajnsek, Markus Rumpler, Sabine Zenz, Petra M. Baumann, Hesham Elsayed, Adelheid Puffing, Gerlies Treiber, Thomas R. Pieber, Harald Sourij, and Julia K. Mader. *Evaluation of subcutaneous glucose monitoring systems under routine environmental conditions in patients with type 1 diabetes*. *Diabetes, Obesity and Metabolism*, 19(7), 1051-1055, 2017.
- [5] Ganjar Alfian, Muhammad Syafrudin, Muhammad Anshari, Filip Benes, Fransiskus Tatas Dwi Atmaji, Imam Fahrurrozi, Ahmad Fathan Hidayatullah, and Jongtae Rhee. *Blood glucose prediction model for type 1 diabetes based on artificial neural network with time-domain features*. *Biocybernetics and Biomedical Engineering*, 40(4), 1586-1599, 2020.
- [6] Shridhara Alva, Kristin Castorino, Hyun Cho, and Junli Ou. *Feasibility of Continuous Ketone Monitoring in Subcutaneous Tissue using a Ketone Sensor*. *Journal of Diabetes Science and Technology*, 15(4), 768-774, 2021.
- [7] Mohammadreza Armandpour, Brian Kidd, Yu Du, and Jianhua Z. Huang. *Deep Personalized Glucose Level Forecasting Using Attention-based Recurrent Neural Networks*. *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2021.

- [8] Irfani Rahmi Ausri, Sadegh Sadeghzadeh, Subhamoy Biswas, Hanjia Zheng, Peyman GhavamiNejad, Michelle Dieu Thao Huynh, Fatemeh Keyvani, Erfan Shirzadi, Fasih A. Rahman, Joe Quadrilatero, Amin GhavamiNejad, and Mahla Poudineh. *Multifunctional Dopamine-Based Hydrogel Microneedle Electrode for Continuous Ketone Sensing*. *Advanced Materials*, 36(32), 2024.
- [9] Amazon Web Services (AWS). *What Are Autoregressive Models?*
- [10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. International Conference on Learning Representations (ICLR), 2015.
- [11] Ananda Basu, Simmi Dube, Sona Veettil, Michael Slama, Yogish C. Kudva, Thomas Peyser, Rickey E. Carter, Claudio Cobelli, and Rita Basu. *Time Lag of Glucose From Intravascular to Interstitial Compartment in Type 1 Diabetes*. *Journal of Diabetes Science and Technology*, 9(1), 63-68, 2015.
- [12] Roy W. Beck, Tonya Riddlesworth, Katrina Ruedy, Andrew Ahmann, Richard Bergenstal, Stacie Haller, Craig Kollman, Davida Kruger, Janet B. McGill, William Polonsky, Elena Toschi, Howard Wolpert, and David Price. *Effect of Continuous Glucose Monitoring on Glycemic Control in Adults With Type 1 Diabetes Using Insulin Injections: The DIAMOND Randomized Clinical Trial*. *JAMA*, 317(4), 371-378, 2017.
- [13] Melanie K. Bothe, Luke Dickens, Katrin Reichel, Arn Tellmann, Björn Ellger, Martin Westphal, and Ahmed A. Faisal. *The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas*. *Expert Review of Medical Devices*, 10(5), 661-673, 2014.
- [14] Ransford Henry Botwey, Elena Daskalaki, Peter Diem, and Stavroula G. Mougiakakou. *Multi-model data fusion to improve an early warning system for hypo-/hyperglycemic events*. 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2014.
- [15] Leo Breiman. *Random Forests*. *Machine Learning*, 45, 5-32, 2001.

- [16] John Bridle. *Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters*. Advances in Neural Information Processing Systems (NIPS), 1989.
- [17] Giacomo Cappon, Andrea Facchinetti, Giovanni Sparacino, Pantelis Georgiou, and Pau Herrero. *Classification of Postprandial Glycemic Status with Application to Insulin Dosing in Type 1 Diabetes—An In Silico Proof-of-Concept*. Sensors, 19(14), 2019.
- [18] Rafael Del Caño, Tamoghna Saha, Chochanon Moonla, Ernesto De la Paz, and Joseph Wang. *Ketone bodies detection: Wearable and mobile sensors for personalized medicine and nutrition*. Trends in Analytical Chemistry, 159, 2023.
- [19] Cheng Chen, Xue-Ling Zhao, Zhan-Hong Li, Zhi-Gang Zhu, Shao-Hong Qian, and Andrew J. Flewitt. *Current and Emerging Technology for Continuous Glucose Monitoring*. Sensors, 17(1), 182, 2017.
- [20] Tianqi Chen and Carlos Guestrin. *XGBoost: A Scalable Tree Boosting System*. KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794, 2016.
- [21] M. Kalpana Chowdary, C. Ganesh, P. Michael Preetam Raj, M. Girish Kumar, M. Sridhar, and S. Sandhya. *An Expert System for Insulin Dosage Prediction using Machine Learning and Deep Learning Algorithms*. 8th International Conference on Communication and Electronics Systems (ICCES), 2023.
- [22] Mark P. Christiansen, Satish K. Garg, Ronald Brazg, Bruce W. Bode, Timothy S. Bailey, Robert H. Slover, Ashley Sullivan, Suiying Huang, John Shin, Scott W. Lee, and Francine R. Kaufman. *Accuracy of a Fourth-Generation Subcutaneous Continuous Glucose Sensor*. Diabetes Technology and Therapeutics, 19(8), 446-456, 2017.
- [23] Mark P. Christiansen, Leslie J. Klaff, Ronald Brazg, Anna R. Chang, Carol J. Levy, David Lam, Douglas S. Denham, George Atiee, Bruce W. Bode, Steven J. Walters, Lynne Kelley, and Timothy S. Bailey. *A Prospective Multicenter Evaluation of the Accuracy of a Novel Implanted Continuous Glucose Sensor: PRECISE II*. Diabetes Technology and Therapeutics, 20(3), 197-206, 2018.

- [24] Edward R. Damiano, Firas H. El-Khatib, Hui Zheng, David M. Nathan, and Steven J. Russell. *A comparative effectiveness analysis of three continuous glucose monitors*. *Diabetes Care*, 36(2), 251-259, 2013.
- [25] Elena Daskalaki, Kirsten Nørgaard, Thomas Züger, Aikaterini Prountzou, Peter Diem, and Stavroula Mougiakakou. *An Early Warning System for Hypoglycemic/Hyperglycemic Events Based on Fusion of Adaptive Prediction Models*. *Journal of Diabetes Science and Technology*, 7(3), 689-698, 2013.
- [26] Ketan Dhatariya. *Blood Ketones: Measurement, Interpretation, Limitations, and Utility in the Management of Diabetic Ketoacidosis*. *The Review of Diabetic Studies*, 13(4), 217–225, 2016.
- [27] Ketan K. Dhatariya, Nicole S. Glaser, Ethel Codner, and Guillermo E. Umpierrez. *Diabetes ketoacidosis*. *Nature Reviews Disease Primers*, 6, 2020.
- [28] Grzegorz Dudek. *STD: A seasonal-trend-dispersion decomposition of time series*. *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [29] Azza B. El-Remessy. *Diabetic Ketoacidosis Management: Updates and Challenges for Specific Patient Population*. *Endocrines*, 3(4), 801-812, 2022.
- [30] International Diabetes Federation. *Type 1 Diabetes Index*. <https://www.t1dindex.org>.
- [31] Ian Fox. *multi-output-glucose-forecasting*. Github Repository, <https://github.com/igfox/multi-output-glucose-forecasting>, 2018.
- [32] Ian Fox, Lynn Ang, Mamta Jaiswal, Rodica Pop-Busui, and Jenna Wiens. *Deep Multi-Output Forecasting: Learning to Accurately Predict Blood Glucose Trajectories*. 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.
- [33] Jerome H. Friedman. *Greedy function approximation: a gradient boosting machine*. *Annals of statistics*, 1189-1232, 2001.

- [34] Satish K. Garg, Mark Kipnes, Kristin Castorino, Timothy S. Bailey, Halis Kaan Akturk, John B. Welsh, Mark P. Christiansen, Andrew K. Balo, Sue A. Brown, Jennifer L. Reid, and Stayce E. Beck. *Accuracy and Safety of Dexcom G7 Continuous Glucose Monitoring in Adults with Diabetes*. *Diabetes Technology and Therapeutics*, 24(6), 373-380, 2022.
- [35] Eleni I. Georga, Vasilios C. Protopappas, Diego Ardigo, Michela Marina, Ivana Zavaroni, Demosthenes Polyzos, and Dimitrios I. Fotiadis. *Multivariate Prediction of Subcutaneous Glucose Concentration in Type 1 Diabetes Patients Based on Support Vector Regression*. *Journal of Biomedical and Health Informatics*, 17(1), 71-81, 2013.
- [36] Eleni I. Georga, Vasilios C. Protopappas, Demosthenes Polyzos, and Dimitrios I. Fotiadis. *Evaluation of short-term predictors of glucose concentration in type 1 diabetes combining feature ranking with regression models*. *Medical and Biological Engineering and Computing*, 53, 1305-1318, 2015.
- [37] Peyman GhavamiNejad, Amin GhavamiNejad, Hanjia Zheng, Karan Dhingra, Melisa Samarikhalaj, and Mahla Poudineh. *A Conductive Hydrogel Microneedle-Based Assay Integrating PEDOT:PSS and Ag-Pt Nanoparticles for Real-Time, Enzyme-Less, and Electrochemical Sensing of Glucose*. *Advanced Healthcare Materials*, 12(1), 2022.
- [38] Todd M. Gross, David Kayne, Allen King, Carla Rother, and Suzanne Juth. *A Bolus Calculator Is an Effective Means of Controlling Postprandial Glycemia in Patients on Insulin Pump Therapy*. *Diabetes Technology and Therapeutics*, 5(3), 365-369, 2004.
- [39] Guillermo Edinson Guzman Gómez, Luis Eduardo Burbano Agredo, Veline Martínez, and Oscar Fernando Bedoya Leiva. *Application of Artificial Intelligence Techniques for the Estimation of Basal Insulin in Patients with Type I Diabetes*. *International Journal of Endocrinology*, 1-8, 2020.
- [40] Lutz Heinemann, Michael Schoemaker, Günther Schmelzeisen-Redecker, Rolf Hinzmann, Adham Kassab, Guido Freckmann, Florian Reiterer, and Luigi Del Re. *Benefits and Limitations of MARD as a Performance Parameter for Continuous Glucose*

*Monitoring in the Interstitial Space.* Journal of Diabetes Science and Technology, 14(1), 135-150, 2019.

- [41] John A. Hertz, Anders Krogh, and Richard G. Palmer. *Introduction To The Theory Of Neural Computation.* Taylor and Francis Group LLC, 1991.
- [42] Aapo Hyvarinen, Kun Zhang, Shohei Shimizu, and Patrik O. Hoyer. *Estimation of a Structural Vector Autoregression Model Using Non-Gaussianity.* Journal of Machine Learning Research, 11, 1709-1731, 2010.
- [43] Peter G. Jacobs, Pau Herrero, Andrea Facchinetti, Josep Vehi, Boris Kovatchev, and Marc D. Breton. *Artificial Intelligence and Machine Learning for Improving Glycemic Control in Diabetes: Best Practices, Pitfalls, and Opportunities.* IEEE Reviews in Biomedical Engineering, 17, 19-41, 2023.
- [44] Andrej Janež, Cristian Guja, Asimina Mitrakou, Nebojsa Lalic, Tsvetalina Tankova, Leszek Czupryniak, Adam G. Tabák, Martin Prazny, Emil Martinka, and Lea Smircic-Duvnjak. *Insulin Therapy in Adults with Type 1 Diabetes Mellitus: a Narrative Review.* Diabetes Therapy, 11(2), 387–409, 2020.
- [45] Jouhyun Jeon, Peter J. Leimbiger, Gaurav Baruah, Michael H. Li, Yan Fossat, and Alfred J. Whitehead. *Predicting Glycaemia in Type 1 Diabetes Patients: Experiments in Feature Engineering and Data Imputation.* Journal of Healthcare Informatics Research, 4, 71-90, 2020.
- [46] Lindy Kahanovitz, Patrick M. Sluss, and Steven J. Russell. *Type 1 Diabetes – A Clinical Perspective.* Point of Care: The Journal of Near-Patient Testing and Technology, 16(1), 39-40, 2017.
- [47] Beate Karges, Anke Schwandt, Bettina Heidtmann, Olga Kordonouri, Elisabeth Binder, Ulrike Schierloh, Claudia Boettcher, Thomas Kapellen, Joachim Rosenbauer, and Reinhard W. Holl. *Association of Insulin Pump Therapy vs Insulin Injection Therapy With Severe Hypoglycemia, Ketoacidosis, and Glycemic Control Among Children, Adolescents, and Young Adults With Type 1 Diabetes.* JAMA, 318(14), 1358–1366, 2017.

- [48] Rebaz A.H. Karim, István Vassányi, and István Kósa. *After-meal blood glucose level prediction using an absorption model for neural network training*. Computers in Biology and Medicine, 125, 2020.
- [49] Anastasia Katsarou, Soffia Gudbjörnsdóttir, Araz Rawshani, Dana Dabelea, Ezio Bonifacio, Barbara J. Anderson, Laura M. Jacobsen, Desmond A. Schatz, and Åke Lernmark. *Type 1 diabetes mellitus*. Nature Reviews Disease Primers, 3, 2017.
- [50] D. Barry Keenan, John J. Mastrototaro, Gayane Voskanyan, and Garry M. Steil. *Delays in Minimally Invasive Continuous Glucose Monitoring Devices: A Review of Current Technology*. Journal of Diabetes Science and Technology, 3(5), 1207-1214, 2009.
- [51] David C. Klonoff and David Ahn ad Andjela Drincic. *Continuous glucose monitoring: A review of the technology and clinical use*. Diabetes Research and Clinical Practice, 133, 178-192, 2017.
- [52] Jelena Kravarusic and Grazia Aleppo. *Diabetes Technology Use in Adults with Type 1 and Type 2 Diabetes*. Endocrinology and Metabolism Clinics of North America, 49(1), 37-55, 2020.
- [53] Umapathi Krishnamoorthy, V Karthika, M K Mathumitha, Hitesh Panchal, Vijay Kumar S Jatti, and Abhinav Kumar. *Learned prediction of cholesterol and glucose using ARIMA and LSTM models – A comparison*. Results in Control and Optimization, 14, 2024.
- [54] Sanjay Kumar, Meenakshi Srivastava, and Vijay Prakash. *Comparative Analysis of ARIMA, Deep Learning, and Lasso Regression Models for Time Series Forecasting: Assessing Accuracy, Robustness, and Computational Efficiency*. 1st International Workshop on Artificial Intelligence: Empowering Sustainable Development (AISD), 2023.
- [55] Kezhi Li, John Daniels, Chengyuan Liu, Pau Herrero, and Pantelis Georgiou. *Convolutional Recurrent Neural Networks for Glucose Prediction*. IEEE Journal of Biomedical and Health Informatics, 24(2), 603-613, 2019.

- [56] Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*. International Journal of Forecasting, 37(4), 1748-1764, 2019.
- [57] Chengyuan Liu, Josep Vehí, Parizad Avari, Monika Reddy, Nick Oliver, Pantelis Georgiou, and Pau Herrero. *Long-Term Glucose Forecasting Using a Physiological Model and Deconvolution of the Continuous Glucose Monitoring Signal*. Sensors, 19(19), 2019.
- [58] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. *Effective Approaches to Attention-based Neural Machine Translation*. Empirical Methods in Natural Language Processing (EMNLP), 2015.
- [59] Julia Madden, Conor O'Mahony, Michael Thompson, Alan O'Riordan, and Paul Gavin. *Biosensing in dermal interstitial fluid using microneedle based electrochemical devices*. Sensing and Bio-Sensing Research, 29, 2020.
- [60] John Martinsson, Alexander Schliep, Björn Eliasson, and Olof Mogren. *Blood Glucose Prediction with Variance Estimation Using Recurrent Neural Networks*. Journal of Healthcare Informatics Research, 4, 1-18, 2020.
- [61] Richard McShinsky and Brandon Marshall. *Comparison of Forecasting Algorithms for Type 1 Diabetic Glucose Prediction on 30 and 60-Minute Prediction Horizons*. 5th International Workshop on Knowledge Discovery in Healthcare Data (KDH), 2020.
- [62] Sadegh Mirshekarian, Hui Shen, Razvan Bunescu, and Cindy Marling. *Lstms and neural attention models for blood glucose prediction: Comparative experiments on real and synthetic data*. 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019.
- [63] Michael C. Mozer. *A Focused Backpropagation Algorithm for Temporal Pattern Recognition*. Complex Systems, 3, 349-381, 1989.
- [64] Jill M. Norris. *Infant and Childhood Diet and Type 1 Diabetes Risk: Recent Advances and Prospects*. Current Diabetes Reports, 10, 345-349, 2010.

- [65] Stefan Pleus, Michael Schoemaker, Karin Morgenstern, Günther Schmelzeisen-Redeker, Cornelia Haug, Manuela Link, Eva Zschornack, and Guido Freckmann. *Rate-of-Change Dependence of the Performance of Two CGM Systems During Induced Glucose Swings*. Journal of Diabetes Science and Technology, 9(4), 801-807, 2015.
- [66] Annette Plüddemann, Carl Heneghan, Christopher P. Price, Jane Wolstenholme, and Matthew Thompson. *Point-of-care blood test for ketones in patients with diabetes: primary care diagnostic technology update*. British Journal of General Practice, 61(589), 530-531, 2011.
- [67] William H. Polonsky, Danielle Hessler, Katrina J. Ruedy, and Roy W. Beck. *The Impact of Continuous Glucose Monitoring on Markers of Quality of Life in Adults With Type 1 Diabetes: Further Findings From the DIAMOND Randomized Clinical Trial*. Diabetes Care, 40(6), 736-741, 2017.
- [68] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. *CatBoost: unbiased boosting with categorical features*. Advances in Neural Information Processing Systems (NeurIPS), 2018.
- [69] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*. Advanced Lectures on Machine Learning, Lecture Notes in Computer Science, 3176, 2003.
- [70] Ravi Reddy, Navid Resalat, Leah M. Wilson, Jessica R. Castle, Joseph El Youssef, and Peter G. Jacobs. *Prediction of Hypoglycemia During Aerobic Exercise in Adults With Type 1 Diabetes*. Journal of Diabetes Science and Technology, 13(5), 919-927, 2019.
- [71] Paolo Rossetti, Jorge Bondia, Josep Vehí, and Carmine G. Fanelli. *Estimating Plasma Glucose from Interstitial Glucose: The Issue of Calibration Algorithms in Commercial Continuous Glucose Monitoring Devices*. Sensors, 10(12), 2010.
- [72] Harry Rubin-Falcone, Ian Fox, and Jenna Wiens. *Deep Residual Time-Series Forecasting: Application to Blood Glucose Prediction*. 5th International Workshop on Knowledge Discovery in Healthcare Data (KDH), 2020.

- [73] Khaled Mohammed Saifullah and Zahra Faraji Rad. *Sampling Dermal Interstitial Fluid Using Microneedles: A Review of Recent Developments in Sampling Methods and Microneedle-Based Biosensors*. *Advanced Materials Interfaces*, 10(10), 2023.
- [74] Kyriaki Saiti, Martin Macaš, Lenka Lhotská, Kateřina Štechová, and Pavlína Pithová. *Ensemble methods in combination with compartment models for blood glucose level prediction in type 1 diabetes mellitus*. *Computer Methods and Programs in Biomedicine*, 196, 2020.
- [75] Kyriaki Saiti, Martin Macaš, Kateřina Štechová, Pavlína Pit'ňová, and Lenka Lhotská. *A Combined-Predictor Approach to Glycaemia Prediction for Type 1 Diabetes*. *IFMBE Proceedings, World Congress on Medical Physics and Biomedical Engineering*, 753-756, 2018.
- [76] Jonathan Schofield, Jan Ho, and Handrean Soran. *Cardiovascular Risk in Type 1 Diabetes Mellitus*. *Diabetes Therapy*, 10(3), 773–789, 2019.
- [77] Renat Sergazinov, Mohammadreza Armandpour, and Irina Gaynanova. *Gluformer: Transformer-Based Personalized Glucose Forecasting with Uncertainty Quantification*. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.
- [78] Renat Sergazinov, Mohammadreza Armandpour, and Irina Gaynanova. *Gluformer: Transformer-Based Personalized Glucose Forecasting with Uncertainty Quantification*. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2023.
- [79] Wajeeha Shahid, Faria Khan, Aamir Makda, Vinesh Kumar, Sidra Memon, and Amber Rizwan. *Diabetic Ketoacidosis: Clinical Characteristics and Precipitating Factors*. *Cureus*, 12(10), 2020.
- [80] Jitkomut Songsiri, Joachim Dahl, and Lieven Vandenberghe. *Maximum-likelihood estimation of autoregressive models with conditional independence constraints*. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.

- [81] Giovanni Sparacino, Francesca Zanderigo, Stefano Corazza, Alberto Maran, Andrea Facchinetti, and Claudio Cobelli. *Glucose Concentration can be Predicted Ahead in Time From Continuous Glucose Monitoring Sensor Time-Series*. IEEE Transactions on Biomedical Engineering, 54(5), 931-937, 2007.
- [82] Fredrik Stahl, Rolf Johansson, and Eric Renard. *Bayesian combination of multiple plasma glucose predictors*. 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 2012.
- [83] Kari Anne Sveen, Bassam Karimé, Ellen Jørum, Svein Ivar Mellgren, Morten Wang Fagerland, Vincent M. Monnier, Knut Dahl-Jørgensen, and Kristian F. Hanssen. *Small- and Large-Fiber Neuropathy After 40 Years of Type 1 Diabetes*. Diabetes Care, 36(11), 3712–3717, 2013.
- [84] Geoffrey I. Taylor. *Diffusion by Continuous Movements*. Proceedings of the London Mathematical Society, s2-20(1), 196-212, 1922.
- [85] Evelyn Teo, Norasyikin Hassan, Wilson Tam, and Serena Koh. *Effectiveness of continuous glucose monitoring in maintaining glycaemic control among people with type 1 diabetes mellitus: a systematic review of randomised controlled trials and meta-analysis*. Diabetologia, 65, 604-619, 2022.
- [86] Hazhir Teymourian, Chochanon Moonla, Farshad Tehrani, Eva Vargas, Reza Aghavali, Abbas Barfidokht, Tanin Tangkuaram, Patrick P. Mercier, Eyal Dassau, and Joseph Wang. *Microneedle-Based Detection of Ketone Bodies along with Glucose and Lactate: Toward Real-Time Continuous Interstitial Fluid Monitoring of Diabetic Ketosis and Ketoacidosis*. Analytical Chemistry, 92(2), 2291-2300, 2020.
- [87] Kamuran Turksoy, Iman Hajizadeh, Nicole Hobbs, Jennifer Kilkus, Elizabeth Littlejohn, Sediqeh Samadi, Jianyuan Feng, Mert Sevil, Caterina Lazaro, Julia Ritthaler, Brooks Hibner, Nancy Devine, Laurie Quinn, and Ali Cinar. *Multivariable Artificial Pancreas for Various Exercise Types and Intensities*. Diabetes Technology and Therapeutics, 20(10), 662-671, 2018.
- [88] Nichole S. Tyler and Peter G. Jacobs. *Artificial Intelligence in Decision Support Systems for Type 1 Diabetes*. Sensors, 20(11), 3214, 2020.

- [89] Nichole S. Tyler, Clara Mosquera-Lopez, Gavin M. Young, Joseph El Youssef, Jessica R. Castle, and Peter G. Jacobs. *Quantifying the impact of physical activity on future glucose trends using machine learning*. *iScience*, 25(3), 2022.
- [90] Mohammad Reza Vahedi, Koenrad B. MacBride, Woo Wunsik, Yosep Kim, Chris Fong, Andrew J. Padilla, Mohammad Pourhomayoun, Alex Zhong, Sameer Kulkarni, Siddharth Arunachalam, and Boyi Jiang. *Predicting Glucose Levels in Patients with Type1 Diabetes Based on Physiological and Activity Data*. 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop, 4, 1-5, 2018.
- [91] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [92] 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.
- [93] Jose Manuel Velasco, Oscar Garnica, Sergio Contador, Jose Manuel Colmenar, Esther Maqueda, Marta Botella, Juan Lanchares, and J. Ignacio Hidalgo. *Enhancing Grammatical Evolution Through Data Augmentation: Application to Blood Glucose Forecasting*. *European Conference on the Applications of Evolutionary Computation*, 142-157, 2017.
- [94] Martina Vettoretti, Giacomo Cappon, Andrea Facchinetti, and Giovanni Sparacino. *Advanced Diabetes Management Using Artificial Intelligence and Continuous Glucose Monitoring Sensors*. *Sensors*, 20, 2020.
- [95] Martina Vettoretti and Andrea Facchinetti. *Combining continuous glucose monitoring and insulin pumps to automatically tune the basal insulin infusion in diabetes therapy: a review*. *BioMedical Engineering OnLine*, 18(37), 2019.
- [96] R. Paul Wadwa, Lori M. Laffel, Viral N. Shah, and Satish K. Garg. *Accuracy of a Factory-Calibrated, Real-Time Continuous Glucose Monitoring System During 10 Days of Use in Youth and Adults with Diabetes*. *Diabetes Technology and Therapeutics*, 20(6), 395-402, 2018.

- [97] Youqing Wang, Xiangwei Wu, and Xue Mo. *A Novel Adaptive-Weighted-Average Framework for Blood Glucose Prediction*. *Diabetes Technology and Therapeutics*, 15(10), 2013.
- [98] John B. Welsh, Francine R. Kaufman, and Scott W. Lee. *Accuracy of the Sof-Sensor Glucose Sensor with the iPro Calibration Algorithm*. *Journal of Diabetes Science and Technology*, 6(2), 2012.
- [99] S. Wen, H. R. Li, H. H. Han, and X. Yu. *A Glucose Prediction Model based on Variational Mode Decomposition and Least Squares Support Vector Regression*. 3rd International Conference on Artificial Intelligence Applications and Technologies (AIAAT), 646, 2019.
- [100] Neil H. White. *Long-term Outcomes in Youth with Diabetes Mellitus*. *Pediatric Clinics of North America*, 62(4), 889–909, 2015.
- [101] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [102] Jinyu Xie and Qian Wang. *Benchmark Machine Learning Approaches with Classical Time Series Approaches on the Blood Glucose Level Prediction Challenge*. *CEUR-WS*, 2148, 2018.
- [103] Jinyu Xie and Qian Wang. *Benchmarking Machine Learning Algorithms on Blood Glucose Prediction for Type I Diabetes in Comparison With Classical Time-Series Models*. *IEEE Transactions on Biomedical Engineering*, 67(11), 2020.
- [104] Han Yu, Jianmin Lu, Yue Jin, Binglei Yue, and Xiao Ma. *The Prediction Model of Blood Glucose Concentration for Smart Health*. *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2019.
- [105] Xia Yu, Kamuran Turksoy, Mudassir Rashid, Jianyuan Feng, Nicole Hobbs, Iman Hajizadeh, Sediqeh Samadi, Mert Sevil, Caterina Lazaro, Zacharie Maloney, Elizabeth

- Littlejohn, Laurie Quinn, and Ali Cinar. *Model-fusion-based online glucose concentration predictions in people with type 1 diabetes*. Control Engineering Practice, 71, 129-141, 2018.
- [106] Tianqi Zhao, Xia Yu, Yue Cui, and Jianchang Liu. *Blood Glucose Prediction Based on Empirical Mode Decomposition and GA-BP Neural Network*. Chinese Control and Decision Conference (CCDC), 2019.
- [107] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. AAAI, 2021.
- [108] Taiyu Zhu, Tianrui Chen, Lei Kuangt, Junming Zeng, Kezhi Li, and Pantelis Georgiou. *Edge-Based Temporal Fusion Transformer for Multi-Horizon Blood Glucose Prediction*. IEEE International Symposium on Circuits and Systems (ISCAS), 2023.
- [109] Thomas Zueger, Peter Diem, Stavroula Mougiakakou, and Christoph Stettler. *Influence of Time Point of Calibration on Accuracy of Continuous Glucose Monitoring in Individuals with Type 1 Diabetes*. Diabetes Technology and Therapeutics, 14(7), 2012.