PREDICTING HYPOGLYCEMIA IN DIABETIC PATIENTS

USING MACHINE LEARNING TECHNIQUES

by

Khouloud Abdel Aziz Safi Eljil

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Computer Engineering

Sharjah, United Arab Emirates

June 2014

Approval Signatures

We, the undersigned, approve the Master's Thesis of Khouloud Safi Eljil.

Thesis Title: Predicting Hypoglycemia in Diabetic Patients using Machine Learning Techniques

Signature **Date of Signature** (dd/mm/2014)

_____ _____

Dr. Ghassan Qaddah
Associate Professor, Department of Computer Science and Engineering
Thesis Advisor

_____ _____

Dr. Michel Pasquier
Associate Professor, Department of Computer Science and Engineering
Thesis Co-Advisor

_____ _____

Dr. Fadi Aloul
Associate Professor, Department of Computer Science and Engineering
Thesis Committee Member

_____ _____

Dr. Mohammad Jaradat
Visiting Associate Professor, Department of Mechanical Engineering
Thesis Committee Member

_____ _____

Dr. Assim Sagahyroon
Professor, Department Head
Department of Computer Science and Engineering

_____ _____

Dr. Hany El-Kadi
Associate Dean
College of Engineering

_____ _____

Dr. Leland Blank
Dean
College of Engineering

_____ _____

Dr. Khaled Assaleh
Director of Graduate Studies

Acknowledgments

First and foremost I thank Allah, the compassionate, the almighty Merciful, who kindly helped me to complete my thesis and blessed me throughout my life.

I also offer my profoundest gratitude to my thesis advisors, Dr. Ghassan Qaddah, and Dr. Michel Pasquier, for their help and guidance in directing me toward the completion of my thesis.
It would have been impossible to finish this thesis without their direction.

I am also indebted to Professor Kevin Loughlin who squeezed time from his busy schedule to help me with my thesis.

I would like to offer my special thanks to my family for supporting me spiritually throughout my life and during my graduate study. My father who was always a good example for me, supported me throughout my life and believes in me. He also taught me by having faith in almighty God, I can overcome any obstacle. My mother, may Allah have mercy on her, though she passed away when I was a child, she was capable of teaching me good morals and values. God willing she is living happily ever after in Paradise. I am also grateful to my sisters and brothers, to whom I am so blessed to have them in my life.

Last but not least, I would like to thank my friends who prayed for my success, encouraged me, and were there for me in both good times and bad.

To my mother's soul, may Allah grant her the highest of Paradise,
and to my father for his endless support

Abstract

Diabetes is a chronic disease that needs continuous blood glucose monitoring and self-management. The improper control of blood glucose levels in diabetic patients can lead to serious complications such as kidney and heart diseases, strokes, and blindness. The proper treatment of diabetes, on the other hand, can help a person live a long and normal life. On the other hand, tighter glycemic controls increase the risk of developing hypoglycemia, a sudden drop in a patients' blood glucose levels that can lead to coma and possibly death if proper action is not taken immediately. Continuous Glucose Monitoring (CGM) sensors placed on a patient body measure glucose levels every few minutes. They are also capable of detecting hypoglycemia. Yet detecting hypoglycemia sometimes is too late for a patient to take proper action, so a better approach is predicting the hypoglycemia event before it occurs. Recent research efforts have been made in predicting subcutaneous glucose levels at specific points in the future. Moreover, the models developed used are ill suited for predicting out-of-range glucose values, namely, hypoglycemia and hyperglycemia. Hence, in this research, we use machine learning techniques suitable for predicting hypoglycemia within a prediction horizon of thirty minutes. This period should be long enough to enable the diabetes patients to avoid hypoglycemia by taking proper action. In specific, we use and compare two approaches to perform the hypoglycemia prediction, namely, a time sensitive artificial neural networks (TS-ANN) and tree based temporal classification (TBTC) by applying feature extraction from the patient glucose signal. While the TS-ANN performed reasonably well (with average sensitivity= 80.19%, average specificity= 98.2%, and average accuracy= 97.6%), nevertheless, the TBTC approach outperformed the TS-ANN one with the ability to predict hypoglycemia events accurately (with average sensitivity= 93.9%, average specificity= 98.8, average accuracy= 98.16%) using three aggregate global features; mean, minimum, and difference, and two parameterized event primitives (PEPs), namely the negative slope and local minimum of the glucose signal.

Search Terms: machine learning, hypoglycemia prediction, decision tree, neural network, CGM sensors

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AH: American hospital

ANN: artificial neural network

ARMAX: autoregressive moving average with exogenous inputs

AUC: area under curve

BMI: body mass index

CAD: computer-aided diagnosis

CBGM: capillary blood glucose measurements

CHD: coronary heart disease

DCCT: diabetes control and complications trial

Distributed TDNN: the distributed time delay neural network

DT: decision tree classifier

FBG: fasting blood glucose

FDA: Food and Drug Administration

FN: false negative

FNN: fuzzy neural network

FP: false positive

GA: genetic algorithm

HBA: homogeneity-based algorithm

JCIA: Joint Commission International Accreditation

MLNN: multilayer neural network

MSE: mean squared error

NARX: nonlinear autoregressive network with exogenous inputs

ODE: ordinary differential equations

PDE: partial differential equation

PEP: parameterized event primitives

PEV: prediction error variance

PH: prediction horizon

PID: Pima Indians diabetes

PW: prediction window

r : correlation coefficient

RMSE: root mean squared error

SVM: support vector machine classifier

SVR: support vector machines for regression

T1DM: Type 1 diabetes mellitus

T2DM: Type 2 diabetes mellitus

TDL: tapped delay line

TDNN: time delay neural network

TP: true positive

TBTC: tree based temporal classification

TS-ANN: time sensitive artificial neural network

WEKA: waikato environment for knowledge analysis

Chapter 1

Introduction

1.1. Background

Diabetes is a chronic disease that occurs when the human pancreas does not produce enough insulin, or when the body cannot effectively use the insulin it produces, which leads to an increase in blood glucose levels [1]. Normally, after a meal, the body breaks the food down into glucose, which is carried by the blood to cells throughout the body. The cells use insulin, a hormone made in the pancreas, to convert the blood glucose into energy [2]. People with diabetes have problems in doing such a conversion leading to fatigue and many other serious complications.

Late diagnosis and/or improper control of diabetes can lead to many serious complications: damage to the eye (leading to blindness), kidney (leading to renal failure), and nerves (leading to impotence and foot disorders with possible amputation). As well, it increases the risk of heart disease, stroke, and reduces life expectancy [3]. Diabetes has recently become one of the most common diseases around the world, where in year 2000, 171 million people worldwide suffered from diabetes [4]. This number is expected to increase to 366 million by the year 2030. Figure 1 shows the prevalence of diabetes around the world in year 2000 and the expected numbers in year 2030. The prevalence of diabetes in the Middle East is expected to increase 164% [5]. It is interesting to note here that within the UAE, where this research is carried out, diabetes currently affects 19.2 per cent of its population [6], which makes it the tenth highest prevalence ratio worldwide. In addition, diabetes in the Gulf is a regional challenge. Gulf countries such as Kuwait, Qatar, Saudi Arabia, and Bahrain all feature in the top ten countries in diabetes prevalence worldwide [6]. Therefore, finding an effective management process for this disease is of most importance, especially for those countries listed above.

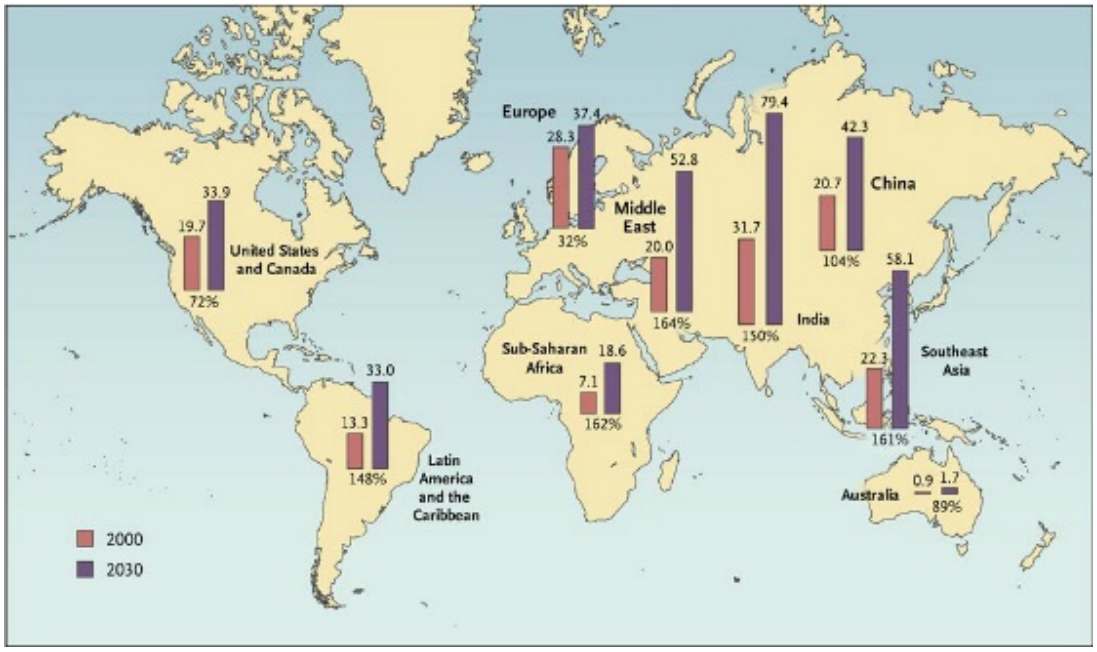

Figure 1. Global prevalence of diabetes in 2000 and 2030 [5]

There are three types of diabetes [1], namely:

1. Type 1 diabetes mellitus (T1DM) or sometimes called juvenile diabetes, is the type of diabetes that results from stopping the insulin generation by the pancreas beta cells. It is the most severe type of diabetes among all of the other ones. It is prevalent among children and requires several insulin injections each day to bring the patient's glucose levels under control.
2. Type 2 diabetes mellitus (T2DM), or the so-called adult-onset diabetes, is the most common type of diabetes, where it compromises 90% of diabetic population worldwide. People can develop T2DM at any age. This form of diabetes usually starts with insulin resistance, which eventually leads to the loss of the pancreas ability to produce enough insulin in response to food intake.
3. Gestational diabetes [7] is the type of diabetes affecting some women during pregnancy only.

There is no cure for diabetes as of yet, nevertheless, an early diagnosis of this disease, followed by a suitable medication, a balanced diet, and regular physical activity go a long way in controlling blood glucose levels and decreasing the risk of developing complications. Controlling the blood glucose level of diabetic patients and keeping it within the normal range (70 mg/dL -120 mg/dL) is therefore the focal goal of physicians [8]. However, the main challenge in blood glucose control is the desire

to keep its level as close to the normal range as possible, while keeping the number of hypoglycemia events to a minimum. Hypoglycemia [9] is a condition that occurs when blood glucose drops dangerously low (below 60 mg/dL). This event may occur due to a number of reasons, such as taking insulin at the wrong time or taking extra dosages, not eating enough during meals or delaying them, and excessive exercising or changing the time of exercising. Effects of hypoglycemia vary from mild dysphoria to conditions that are more dangerous; these present as seizures, unconsciousness, and possibly permanent brain damage or death. Hypoglycemia is treated simply by the oral intake of carbohydrate food by the diabetic patient [9]. It is interesting to note here that the Diabetes Control and Complications Trial (DCCT) [10] found the occurrence of hypoglycemia is three times higher in intensively insulin-treated group of diabetic patients when compared to the group receiving standard treatment [11].

Traditionally, self-monitoring of blood glucose requires the drawing of a blood sample several times a day. The need for reducing the number of daily Capillary Blood Glucose Measurements (CBGM) is due to the pain associated with the needles' use. Nowadays, the availability of Continuous Glucose Monitoring (CGM) devices, placed on the patient body, makes it possible to obtain subcutaneous glucose reading information in real time, i.e., every few minutes. These real time measurements reduce the need for the painful CBGM. Many types of CGM sensors have been approved by the Food and Drug Administration (FDA), including, the Freestyle Navigator (Abbott Diabetes Care) [12], the Seven Plus (DexCom) [13], and the Guardian Real-Time (Medtronic Diabetes) [14]. CGM are a minimally invasive device whereby it measures the glucose level in the patient's interstitial fluid, but not in his/her blood streams[1]. As a result, the measured glucose levels lag 8-10 minutes behind the capillary blood glucose values [15]. The CGM readings, therefore, must be calibrated occasionally against the readings of the traditional glucose meter (finger stick) to reflect the actual blood glucose values.

One of the main advantages of CGM is its ability to detect hypoglycemia and sound an alarm in such conditions. The CGM device detects hypoglycemia using a predefined threshold of glucose level (for example < 60 mg/dl). Sometimes, detecting hypoglycemia is too late for a patient to take corrective actions; therefore, a better approach is to predict such a condition before its occurrence. It is interesting to note

[1] Interstitial fluid is a thin layer of fluid that surrounds the cells of the body.

here that the development of CGM is considered a breakthrough in diabetes treatment and management fields, due to the wealth of useful information it supplies to the diabetic patients [16].

1.2 Problem Statement

Controlling the blood glucose levels for diabetic patients has attracted significant research interest. Researchers used sensors and machine learning algorithms to help patients in maintaining their blood glucose level within the normal range. However, this tight glycemic control results in threefold increases in severe hypoglycemia occurrences [11]. On the other hand, predicting hypoglycemia using machine learning techniques has received less research interest.

In this thesis, we investigate the hypoglycemia predication problem using machine learning techniques. In particular, two approaches are developed and compared, namely, time sensitive artificial neural network (TS-ANN) and tree based temporal classification (TBTC) techniques. In the TS-ANN, the ANN predicts the future subcutaneous glucose measurements, using previous values of blood glucose and other parameters, and then uses those predicted values to decide on the occurrence or non-occurrence of hypoglycemia events within a specific horizon. In the TBTC approach, features that best discriminate hypoglycemia from non-hypoglycemia events are extracted from the patient's glucose signal, and then fed to a decision tree for predicting the occurrence of the hypoglycemia events.

1.3 Summary of research results

- In this work, we were able to predict the near future occurrence of hypoglycemia accurately (average sensitivity= 93.93%, average specificity= 98.8, and accuracy= 98.15%) using the TBTC approach.

- We have used different tree-based techniques for predicting hypoglycemia events, namely, J4.8, REPTree, and Bagging. The Bagging algorithm has outperformed the other models in predicting hypoglycemia.

- We showed the ability of the recent history of subcutaneous glucose measurements to predict short-term hypoglycemia events precisely.

- The TBTC approach has a better predictive power than the time sensitive artificial neural network one, i.e., training the tree-based temporal classification model on the hypoglycemic events directly is better than training it on future subcutaneous glucose levels and then deciding on hypoglycemic events.

- In order to reduce the bias of the TBTC model toward non-hypoglycemia events prediction, we used subsampling of the dataset, i.e., deleting some instances of non-hypoglycemia events such that the proportion of hypoglycemia is increased to be 15% of the dataset.

- Adding the mathematical models to simulate the absorption of insulin and carbohydrates intake, in the TS-ANN approach, increased the performance of the model to predict future subcutaneous glucose measurements as compared to using recent history of subcutaneous glucose measurements alone for prediction. Although a higher increase in performance was expected, the limitation of the used mathematical models and the intra- and inter-individual differences has restricted the increase.

- The TS-ANN approach resulted in a BG prediction model that is accurate in predicting euglycemic range, but not in forecasting hypoglycemia. Where the RMSE for predicting subcutaneous glucose in a prediction horizon of thirty minutes is very good at 13.29, the average sensitivity for predicting hypoglycemia is 62.05% only.

- Moreover, the use of the cost function that penalizes the false negative misclassification more than the false positive one, in TS-ANN approach, has increased the average sensitivity of the prediction models in forecasting hypoglycemia; this is due to the rarity of the hypoglycemic events.

1.4 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 presents a literature review of related work, namely, diabetes diagnosis, glucose prediction, and hypoglycemia prediction using machine learning techniques. The acquisition of data

needed for this study is also presented in Chapter 2. Chapter 3, on the other hand, outlines the development of the time sensitive artificial neural network approach in the prediction of hypoglycemia. Chapter 4 illuminates the tree based temporal classification using the feature extraction approach. The experimentations with the developed models, as well as the comparison between their predictive accuracy, are presented in Chapter 5. Finally, Chapter 6 presents some concluding remarks and future work suggestions.

Chapter 2

Literature Review

Machine learning techniques [17] proved their effectiveness in different fields including medicine [18], business [19], marketing [20], education [21], and many others. During the last decade, there has been an explosion of interest in medical machine learning research, due to the vast amount of data collected from individual patients and stored in databases. In addition, hospital errors are becoming a real concern, where errors and carelessness of hospital staff were the reason behind 87% of hospital deaths in U.S. in [22]. By applying machine learning techniques to patients' databases, errors can be discovered and avoided. This research is about diabetes, thus the use of machine learning techniques in diabetes diagnosis, glucose, and hypoglycemia prediction is surveyed in the following.

2.1 Diabetes diagnosis:

In a diabetes diagnosis, machine learning models are used to discriminate between diabetic and non-diabetic individuals. Classification is used extensively in medicine as a prescreening technique. Different classifiers are used in literature for diabetes diagnosis; namely, neural networks, support vector machines, k-nearest neighbors, naïve Bayes, and decision trees [17]. The use of these classifiers in diabetes diagnostics is outlined in the following.

2.1.1 The Decision Tree classifier (DT) - based diabetes diagnosis

DT uses a tree-like model that maps observations to conclusions [17]. In this model, the leaves represent class labels and the branches represent conjunction of features' conditions that result in one of the class labels. The DT model is a white box; therefore, it can be easily understood and interpreted. It is robust and only requires small amount of time to analyze huge volumes of data. For these reasons, we have used the DT classifiers in developing TBTC models to predict hypoglycemia events in the research presented in Chapter 4 of this thesis.

In [23], the author used a WEKA open source machine learning tool to generate a decision tree classifier using the Pima Indians diabetes (PID) dataset available online at the UCI website [24]. This dataset contains 768 females of Pima Indian culture, 21-

years-old or more. 500 of the females are non-diabetic (65.1%) and the rest (34.9%) are diabetic. There are eight attributes that describe each female within this dataset, as well as, the class attribute. These attributes are listed in Table 1.

Table 1. Attributes of PID Dataset [24]

Attribute ID	Attribute Definition
1	number of times pregnant
2	plasma glucose concentration after two hours in an oral glucose tolerance test
3	diastolic blood pressure (mm Hg)
4	triceps skin fold thickness (mm)
5	two-hour serum insulin (mu U/ml)
6	body mass index (weight in kg/ (height in m)^2)
7	diabetes pedigree function (It is a measure of the expected genetic influence of relatives on the subject's eventual diabetes risk)
8	age (years)
9	class variable (0 or 1)

Having '0' in the class variable (attribute ID = 9) indicates a healthy female, while '1' indicates a diabetic one. The study reported in [23] handled the values missing from PID using different methods. After numerical attribute discretization, the J48 algorithm of WEKA tool has been used to build a decision tree classifier. It is interesting to note here that the J48 algorithm is WEKA's implementation of the well-known C4.5 algorithm [17]. Ten-fold cross validation is used to assess the performance of the generated tree. The obtained accuracy is 78.1768% is a reasonable result given the non-linearity nature of the PID dataset.

On the other hand, the authors of [25] addressed the problem of overfitting and overgeneralization of a given classification algorithm when it is applied to the PID dataset. It uses the Homogeneity-based algorithm (HBA) [26] to control these behaviors of the classification algorithm. As is noticed in Table 2, the HBA, when it is added to the original classifiers (SVM, DT, and ANN), improves its accuracy tremendously. The obtained accuracy, when applying HBA in conjunction with a DT,

is 91.67%. This is the best accuracy obtained for the PID dataset, as far as we know. In Table 2, RATE_FP is the false positive rate, RATE_FN is the false negative rate, and the last column is the percentage of improvement of classification accuracy when using HBA algorithm along with the three different classifiers.

Table 2. The improvement of classifiers' accuracy when using HBA on PID dataset [25]

Algorithm	% Accuracy	% of Improvements
SVM-HBA	94.79%	16.57%
ANN-HBA	94.79%	16.57%
DT-HBA	91.67%	13.45%

2.1.2 Support vector machine classifier (SVM)

The SVM model uses a feature vector space to represent the different input examplesin which each example is mapped as a point in that space, in such a way as to maximize the distance between examples of the different class categories. The main disadvantage of SVM and artificial neural network (ANN) is their underlying models are of black box type. As a result, the classification rules can not be seen or interpreted easily. Interpreting the classifiers' rules is important, especially in medical diagnosis when we need to convince the doctors with the obtained results [27].

In [28], an SVM classifier is used to extract rules. Two methods for rules extraction have been proposed, namely:

1) **SQRex-SVM**: Sequential Covering Approach for Rule Extraction, which extracts rules using a modified sequential covering algorithm [27], where feature selection is used to prune irrelevant features from the rules. Rules performance is measured using true positives and false positives along with the area under curve (AUC) [17]. These measures are used to determine and select the most accurate and comprehensive rules [29].
2) **Eclectic Rule Extraction [30]:** the SVM classifier is trained using a labeled dataset until an acceptable accuracy is reached, then support vectors are constructed using the predicted class of SVM classifier as the target class. Finally, a C5 decision tree is used to extract rules from the newly constructed dataset.

The rules developed above have been used to diagnose diabetes patients using the by Oman diabetes dataset [31] collected using a specially designed questionnaire. It contains 3014 patients of at least 20 years old. The Omani dataset includes attributes such as age, gender, Body Mass Index (BMI), and so on. The prevalence of diabetes in this dataset is 9%, thus the dataset is skewed toward the non-diabetic class. To overcome such a drawback, Barakat et al [28] used subsampling and k-means clustering to choose representative instances of the non-diabetic class.

The SVM model is constructed, trained, and then tested to classify the independent test dataset. Accuracy, true positives, and false positive rates were then calculated. Later, rules were extracted using the two proposed methods presented above. As can be noticed from Table 3, the performance of the two proposed extraction rules methods is respectable and almost similar to one another.

Table 3. Comparison of TP rate, FP rate and AUC of the different methods [28]

Method	TP rate	FP rate
SVM	0.95	0.13
SQRex-SVM	0.93	0.06
Eclectic	0.96	0.09

2.1.3 Neural networks classifier – based diabetes diagnosis

The Artificial Neural Network (ANN) [32] is a computational model composed of interconnected nodes called neurons that are arranged in layers: input, hidden, and output layers. Each interconnection is associated with a weight, which changes during the training phase until satisfactory results are reached. In order to extract meaningful patterns, ANN is used to model complex/non-linear relationships between inputs and outputs.

In [33], ANN-based classifier is applied to the PID dataset [24]. The proposed ANN classifier [32] has m-n-p configuration, where m = 8 (the number of inputs "attributes" to the model), n is the number of neurons in the hidden layer (a single hidden layer is used), and p is the number of outputs which is one. n is selected using a Genetic Algorithm (GA) [34]. The optimal value of n is found to be five. The

classifier is trained using a back propagation algorithm [32]; the accuracy of this model is 73.4%.

In [35], on the other hand, two different types of neural networks are used to classify the PID dataset. The first neural network is multilayer and trained using the Levenberg-Marquardt algorithm [36]. This MLNN consists of an input layer, two hidden layers, and an output layer in which there are eight inputs, 50 neurons in each hidden layer and two outputs (the index of two classes). The accuracy obtained using this 10-fold cross validation [37] is 79.62%. The second ANN used in this study is a probabilistic one. It consists of an input layer, a single hidden layer, and an output layer in which the input is the attributes' vector and the output layer has two outputs (the index of the two classes). The obtained accuracy of this classifier is 78.05%.

Kahramanli and Allahverdi [38] used fuzzy logic to improve the classification accuracy of the PID dataset. A hybrid system is designed consisting of two neural networks, an Artificial neural network (ANN), and another Fuzzy neural network (FNN) that is trained using a back propagation algorithm [32]. The inputs are divided into two groups: fuzzy-like age and blood pressure; the rest are considered as crisp data. The first stage of the proposed model is standardizing the crisp input values [39] and feeding them to the first ANN (ANN1). After fuzzifying the fuzzy data, their values are presented to FNN and the obtained result is defuzzyfied. The attained results of ANN1 and FNN are fed to the second ANN (ANN2) to calculate the final output. If the output value is different from the actual value, weights of these networks will change. This process is repeated until satisfactory results are reached. The accuracy of this model using the 10-fold cross validation is 84.2%, a good accuracy when compared to the ones obtained from the different classifiers implemented for the pima dataset that range between 60% and 79.62% [35], with the exception of [25] where the accuracy is 91.67%.

2.2 Blood Glucose Prediction:

Predicting future glucose levels can help diabetic patients to manage their daily life in a more healthy way, i.e., insulin doses, meal amounts and time, and exercising. Two main approaches to predict blood glucose in diabetic patients have been identified [40]-[43] and [45], namely, mathematical-based and data-driven based models. A mathematical model is a description of a system using mathematical formulas [44]. This approach will not be covered in this literature

survey because it is out of this study's scope. In the data-driven models-based approach, on the other hand, the glucose is predicted using a set of input-output data. The data-driven approach can be divided into two groups, namely, machine learning and time series based methods. Both of these techniques are presented next.

2.2.1. Machine learning based blood glucose prediction

In [45], a machine learning technique was used to predict next morning (fasting) blood glucose (FBG) levels. The authors ran the study on four insulin treated diabetic patients for a period of three months; they provided these patients with three portable commercial devices namely, a blood glucose monitor, a metabolic rate monitor, and a laptop as food intake monitor. The portable blood glucose monitor was used to measure the fasting blood glucose (FBG) level once a day. The patients were attached to the metabolic rate monitors to measure the rate at which the body burns calories [46]. Calories calculating software was designed and implemented to calculate the calories of the food intake. It provided the patient with different meals displayed on the laptop screen, to choose the appropriate one for breakfast, lunch, and dinner. After collecting the data, a machine-learning technique [45] was used to estimate blood glucose levels. However, the estimated FBG level turned out to be inaccurate after the eighth week of data collection,; the error rates for the four subjects were 8.9%, 26.1%, 13.5%, and 22.2%. This was due to the small number of blood glucose measurements taken per day (only one glucose reading per day is used).

In comparison, the authors of [43] used both compartmental (mathematical) models along with a support vector machines for regression (SVR) to predict the future glucose levels. They used, as shown in Figure 2, three compartmental models as already found in the literature of [47] and [48]: firstly, a meal model that simulates the consumption and absorption of carbohydrates by the human body [47]; secondly, the insulin model that simulates the absorption and the pharmacokinetics/pharmacodynamics of subcutaneously administered insulin [48]; and finally, the exercise model that simulates the impact of exercising on glucose–insulin interaction.

The glucose predictive model utilized the SVR to predict the subcutaneous glucose concentration y at the time t+k, given that t is the current time and k (the prediction horizon) is set to 15, 30, 60, and 120 minutes. The input variables to SVR, as shown in Figure 2, are the rate of glucose appearance in plasma after a meal (Ra),

the plasma insulin concentration (Ip), the subcutaneous glucose measurements (CGM data), and a set of physical activity related variables. The SVR is built using a linear kernel function [17], and is trained using each patient individually.

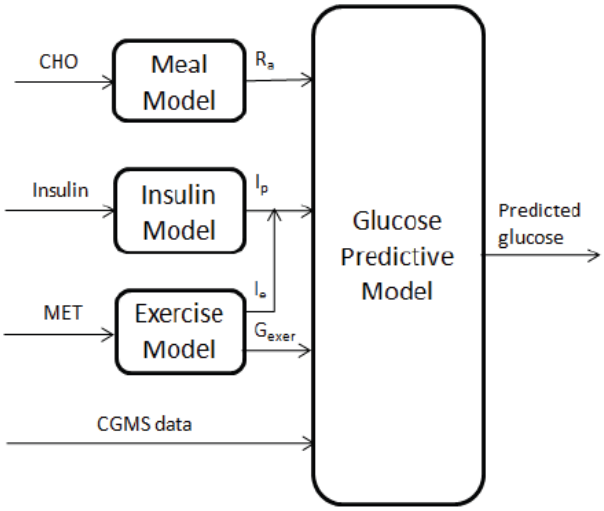

Figure 2. The glucose prediction technique proposed in [43]

In order to collect data, seven diabetic patients were monitored for 10 days, using several monitoring devices/techniques. Patients wore the Guardian Real-Time CGM system [14], which was used to measure the subcutaneous glucose concentration every five minutes. For monitoring physical activities, a SenseWear body-monitoring device [49] was used. Finally, a special paper-based diary was used to track the food intake, dosage, and time of insulin injections.

The predictive accuracy of the proposed system was assessed by calculating the Root Mean Squared Error (RMSE) and the correlation coefficient (r) between the predicted and actual subcutaneous glucose values. RMSE for prediction horizons of 15 and 30 minutes were low, namely, 9.51, and 16.02, respectively, and the correlation factors (r) with the real glucose values were high, namely, 0.95 and 0.88, respectively. Nevertheless, as the prediction horizon increases in length, the prediction accuracy significantly decreases.

Another study, presented in [50], uses, as shown in Figure 3, an artificial neural network (ANN) for predicting the subcutaneous glucose concentration at prediction horizons of 15, 30, and 45 minutes. The inputs to the ANN are the current and previous subcutaneous glucose concentrations obtained from a CGM sensor during the last 20 minutes. Their dataset is composed of nine subjects who used the

Medtronic Guardian device [14] (group I) and six subjects that used the Abbott Navigator device [12] (group II). Regarding the structure of the ANN, it is composed of two hidden layers with the first hidden layer of 10 neurons and a second layer of five neurons; the transfer function in both layers is sigmoidal. The output layer has one neuron with linear transfer function. The ANN was trained using the back propagation Levenberg-Marquardt optimization training algorithm [51].

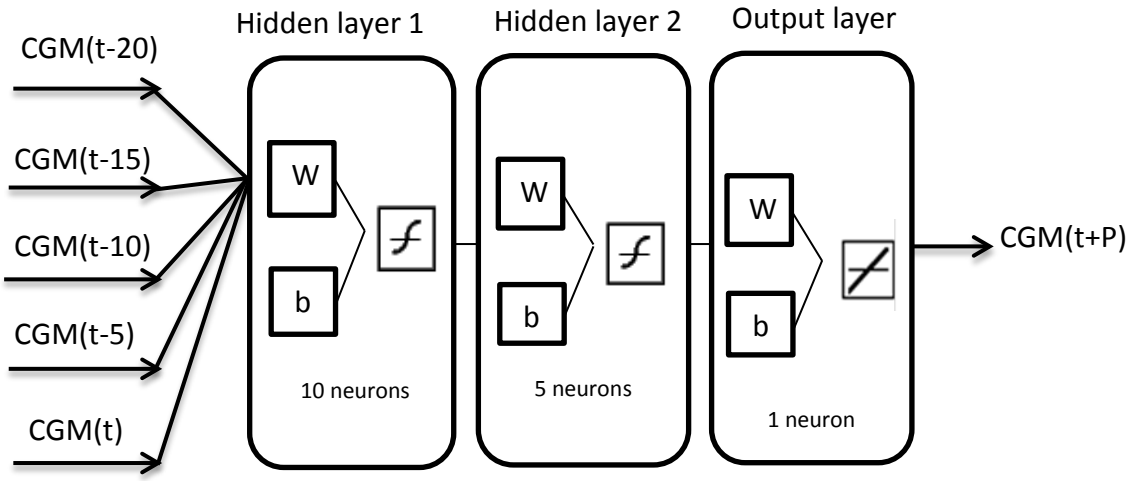

Figure 3. The ANN predictive architecture of [50]

The authors used a root mean square error (RMSE) [17] for performance assessment of the predictive system. The results of the ANN in the two groups of patients are as follows: The RMSE results for group I are 9.74, 17.45, and 25.08 for the prediction horizon of 15, 30, and 45 minutes respectively. The RMSE results for group II are 10.38, 19.51, and 29.07 for the prediction horizon of 15, 30, and 45 minutes respectively. One can notice that the performance results for group I are better than those of group II. Moreover, they were able to reach good glucose prediction results when using only the recent history of glucose as input to the ANN model.

2.2.2. Time series based blood glucose level prediction

Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Time series forecasting, on the other hand, is the process of using a model to generate predictions (forecasts) for future events based on known current and past ones [52]. In [53], the autoregressive moving average with exogenous inputs (ARMAX) [54] model is used to predict the future values of blood glucose. The inputs to the model, as shown in Figure 4, are plasma glucose rates of appearance after carbohydrate (CHO) absorption and plasma insulin

after subcutaneous injection. The output is the interpolated blood glucose from the drawn blood samples.

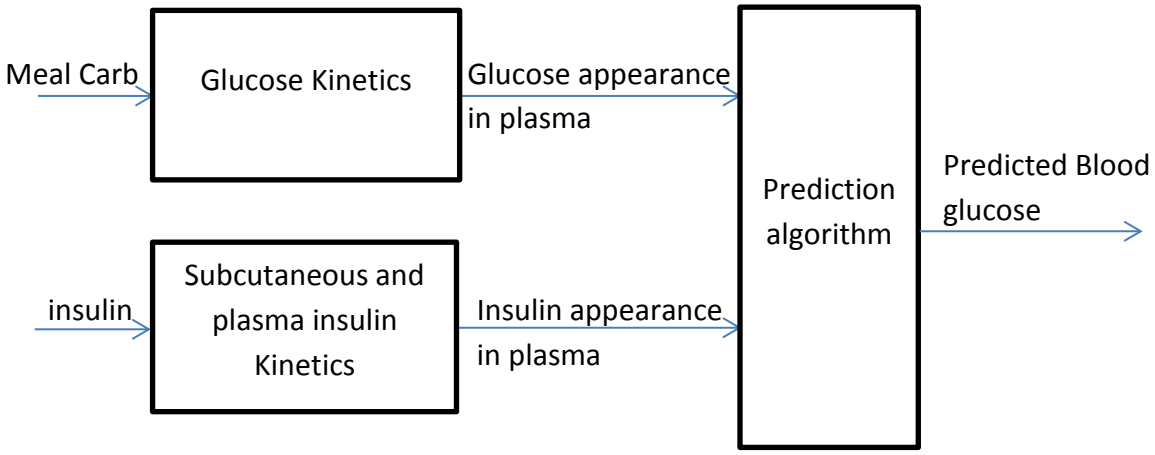

Figure 4. Structure of blood glucose prediction model of [53]

The dataset used in the study presented above is composed of nine Type 1 Diabetes Mellitus (T1DM) subjects. Blood samples were collected from these patients in order to measure plasma glucose concentrations for a total of 37 blood samples per day. They used the models already available in the literature [55] for both meal (Glucose Kinetics) and insulin (plasma insulin kinetics) models. They found some variability between the actual plasma insulin and the ones obtained using the simulation model. Their findings showed they used the actual (interpolated) plasma insulin as input for their prediction model. The performance of the model reported in [53] has been assessed using a prediction error variance (PEV), for the prediction horizon of 30, 60, 90, and 120 minutes. According to their defined equation, PEV equals to the mean squared error (MSE), which measures the average of the squares of the "errors". The error is the difference between the predicted and actual values. We have calculated the average RMSE error rate of the proposed model of [53]. The RMSE is the square root of Mean Square Error (MSE) from which we obtained the following results, namely, RMSE of 16.34, 26.55, 39.08, and 44.44 for 30, 60, 90, and 120 minutes respectively, as shown in Figure 5. Again, here the performance results of short-term predictions (30 and 60 minutes) are reasonable, but longer predictions cannot be trusted.

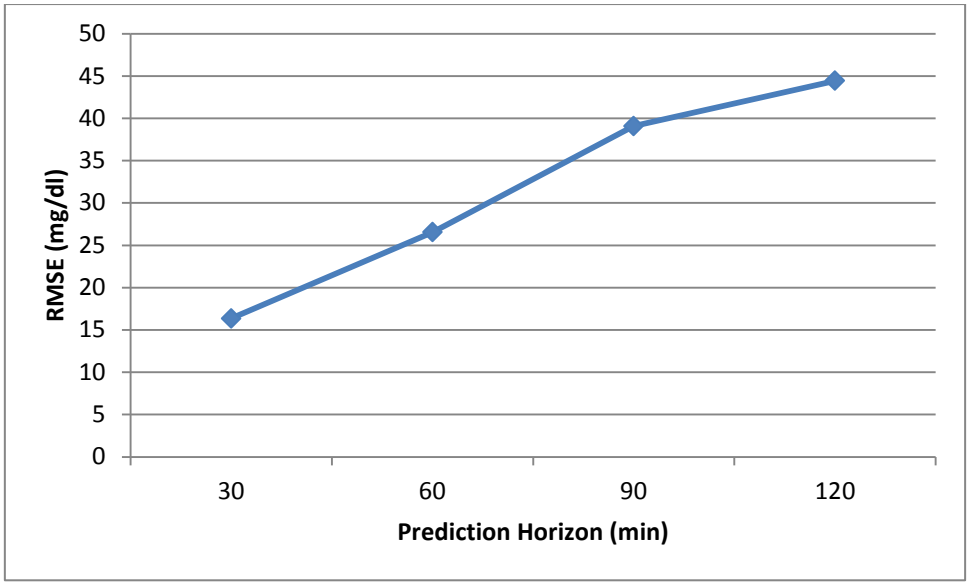

Figure 5. Average RMSE error rate for the prediction model of [53] vs prediction horizon

Table 4 presents a summary of studies concerning glucose prediction in diabetes, along with the methods used, input variables, and the number of patients. Regarding the performance, the model proposed in [45] resulted in the worst performance due to the small number of blood glucose measurements taken per day (only one glucose reading per day), as mentioned earlier. The best performance was obtained from the model of [43], where the RMSE for prediction horizon of 30 minutes is 16.02. Yet, the performance of [50] and [53] models were close to that. The RMSE for group the I diabetic patients of [50] for prediction horizon of 30 minutes is 17.45, and 19.51 for group II. Finally, the RMSE of [53] is 16.34 for prediction horizons of 30 minutes. It is worth mentioning here that [50] has showed the ability of its model to predict future subcutaneous glucose measurements using only the recent history of subcutaneous glucose measurements, without the need for other parameters, i.e., insulin and meals.

Table 4. Summary of glucose prediction studies using data-driven techniques

Study	Year	Number of Patients	Input variables	Method	RMSE
[45]	2003	4	fasting blood glucose, metabolic rate, and food intake	data mining technique	-
[43]	2011	7	CGM data, insulin, carbohydrates	SVR	16.02
[50]	2010	9	CGM data	ANN	18.48
[53]	2011	9	carbohydrates, insulin	time series based method [ARMAX]	16.34

In our time series based ANN approach, presented in Chapter 3, we modify the approach presented in [43], the best reviewed model, to predict hypoglycemia events. We replaced the SVR with ANN as our machine learning technique; moreover, for the insulin model we simulated the absorption of monomeric insulin, since the patients participating in our study used pumps to infuse rapid acting insulin (monomeric insulin). The model of [43], and all the above reviewed glucose prediction models, predicts the glucose at a specific point (t+n). However, in our case we predict the glucose at several steps, namely, t+1, t+2, …, all the way till t+n. We also added the hypoglycemia prediction function and the cost function to the developed model as presented in Chapter 3.

2.3 Hypoglycemia Prediction

Hypoglycemia prediction, unlike glucose prediction, has not received much of attention by researchers. However, similar to glucose prediction work, research found in the literature can be categorized into two groups: hypoglycemia prediction using mathematical models and data driven models, including time series and machine learning types.

Table 5 provides a summary of some of the studies conducted for hypoglycemia event prediction, along with methods used, input variables, and the number of patients participated.

Table 5. Summary of research on hypoglycemia prediction in diabetic patients

Study	Year	Number of Patients	Input Variables	Method
[56]	2010	22	CGM data	mathematical model
[58]	2010	54	CGM data	time series analysis method
[60]	2008	3116	blood glucose, intravenous insulin	classification tree

2.3.1 Mathematical model based hypoglycemia prediction

The authors of [56] addressed the problem of nocturnal hypoglycemia, since according to DCCT [10], 55% of severe hypoglycemia occur during sleeping hours. They claim that CGM alarms may be inefficient during sleep, as they monitored the patients who were wearing GlucoWatch CGM [57] during the night. [56] found that 71% of the patients did not respond to the alarm at night. In order to circumvent the aforementioned problem, [56] suggested that the CGM sensor sends a signal to the patient's pump to stop infusing insulin whenever a hypoglycemia event is predicted. They used a mathematical model to predict hypoglycemia. This model utilizes a system composed of five prediction algorithms: 1) linear projection, 2) Kalman filtering, 3) hybrid infinite impulse, 4) statistical prediction, and 5) numerical logical algorithm. These five algorithms predict hypoglycemia events using current and previous glucose levels. The alarm will sound according to a voting system when the number of algorithms that predict a hypoglycemia event is above the predefined voting threshold (3 and 4 were used as voting thresholds), and the hypoglycemia threshold used is $\leq$70 mg/dl. When three algorithms were used to trigger the insulin pump suspension, nocturnal hypoglycemia was prevented in 60% (sensitivity = 0.6) of the cases. However, by using only two prediction algorithms to predict the hypoglycemia in the results found 84% prevention of hypoglycemia events (sensitivity = 0.84). In addition, this study found that the prediction rate decreases

when the voting threshold increases. Contrary, false alarms generated with lower voting threshold are more prevalent, and as a result, the objective of the developed system is to achieve a balance between the prediction and false alarm rates.

2.3.2 Time series based hypoglycemia prediction

In [58], three times series-based hypoglycemia prediction methods were proposed, namely, A) absolute predicted glucose values, B) a cumulative-sum (CUSUM) control chart, and C) an exponentially weighted moving-average (EWMA) control chart. To validate the proposed methods, the authors of [58] used a DirectNet dataset [59] that contains 54 patients with type I diabetes (7-18 years of age). Each patient wore a Medtronic CGMS™ Continuous Glucose Monitoring system [14] that measures glucose level every five minutes. They used 60 mg/dl as a hypoglycemia threshold. For the hypoglycemia alarm, they combined the sensor built-in alarm and the predicted hypoglycemia one using one of the three proposed methods - A, B, or C. For performance evaluation, they used 30 min as the prediction horizon; method (A) correctly predicted 89% of hypoglycemia events (sensitivity), however, predicting non-hypoglycemia events (specificity) was 67%. Sensitivity for second method (B) was 87.5%, but the specificity was better (74%). The third method (C) had a sensitivity of 89% and a specificity of 78%.

2.3.3 Machine learning based hypoglycemia prediction

The study presented in [60] used a classification learning technique to predict the occurrence of hypoglycemia within a prediction horizon of one hour. Their used dataset is MIMIC II [61] (Multiparameter Intelligent Monitoring in Intensive Care database II), which is an Intensive Care Unit database where blood glucose was measured every hour for patients under intravenous (IV) insulin infusion. Data was collected from 3116 patients in the Intensive Care Unit (ICU). A classification tree was built using the See5 algorithm (which is the Window's implementation of C5.0 [62]). C5.0 is an improved commercial version of C4.5 (or what is referred to J4.8 in WEKA). The input features were the glucose measurements along with insulin infusion rate organized the well-known tree-based classification scheme in a time series. The output class was “1” for experiencing hypoglycemia in the next hour, “0” if not. According to the generated tree, the most important feature is the last two blood glucose values, whereas, the second most important feature is the slope of

change in blood glucose levels with respect to change in insulin infusion rate. 70% of the dataset records were used for training the developed model, the rest (30%) were used for evaluation. The classification tree accuracy for predicting acute hypoglycemia (patient glucose <= 60 mg/dL) was 86% (and specificity = 89.87%). On the other hand, predicting severe hypoglycemia (patient glucose <= 40 mg/dL) accuracy was 78.76% and specificity = 80.53%.

Our TBTC approach to predict hypoglycemia, presented in Chapter 4 of this thesis, has some similarity with the one developed in [60] in the sense that both models are using tree based classifiers with time series attributes. However, [60] failed to describe how its model has overcome the limitation of its tree classifier; in addition to, it does not take order of instances into consideration.

2.4 Research Direction

Following the literature review presented, one can conclude that plenty of research went into the area of diabetes diagnosis using machine-learning technique. The goal of researchers in such a domain has been the achievement of high diagnosis accuracy, and they were able to achieve such a goal. Glucose prediction on the other hand, also has had significant research put into it with the target of such efforts resulting in glucose predicted values coming as close to the actual glucose measurements as possible. They were able to develop models that are accurate in the euglycemic range (70 mg/dL -120 mg/dL). However, their models are not well suited for out-of-range glucose values, i.e., hyperglycemia (> 180 mg/dL) or hypoglycemia (<= 60 mg/dL). Moreover, these same researchers predicted glucose at a specific point in time only (t+k), and not at each step within a given prediction horizon (t+k), i.e., (at k steps away from the current time t). Hypoglycemia prediction using machine learning techniques has to date not received similar attention. In this work, our goal is to develop prediction models that accurately predict hypoglycemia. To achieve such a goal, we followed two approaches in hypoglycemia prediction, namely, the Time Sensitive Artificial Neural Network (TS-ANN) and the Tree Based Temporal Classification (TBTC). In TS-ANN, we utilized the artificial neural network to predict subcutaneous glucose measurements within a prediction horizon of thirty minutes, based on the recent history of subcutaneous glucose measurements, the rate of appearance of glucose following meal intake, and plasma insulin resulting from insulin infusion. The predicted subcutaneous glucose measurements are used to

decide on the occurrence of the hypoglycemia events. In the temporal classification approach, global and temporal features of the glucose signal that best discriminate hypoglycemia events from non-hypoglycemia ones are extracted and then used to predict the hypoglycemia events. In this research, whether TS-ANN or TBTC, we created a model for each patient separately, to reflect the diverse lifestyles and insulin regimens of the different patients.

The prediction tools used in this study are those found in WEKA [63] (Waikato Environment for Knowledge Analysis) and MATLAB [64]. WEKA is an open source machine learning software that has multiple algorithms for the different aspects of machine learning, namely, data preprocessing, classification, association, clustering, and forecasting. It is capable of visualizing, analyzing, and comparing the performance of the different algorithms. WEKA accepts files in both ARFF [65] and CSV [66] forms. We have also used a Time Series Neural Network in MATLAB [64]. Such a toolbox supports different neural network architectures, various activation functions, and training algorithms. It is well suited for time-dependent attributes; in addition to this, it deals with nonlinear multivariate problems.

In order to assess the performance of the two approaches proposed in this study, we collected data from the American Hospital of Dubai located in the United Arab Emirates (UAE). The details of the collected dataset are described in the next section.

2.5 Data Acquisition Phase

The dataset used in this study is obtained from the American Hospital (AH) in Dubai, the first hospital in the Middle East to be awarded the Joint Commission International Accreditation (JCIA). The dataset contained 10 patients who attended the AH and used the Medtronic CGM sensor [14] for monitoring their subcutaneous glucose levels and insulin doses for treatment. Collecting the data has been approved by the AH ethical committee, since the confidentiality of the patients has been maintained throughout this study.

The records of the different patients are separated into different excel files; a summary data about these files are presented in Table 6. Each file contains a set of readings, each with its own date and time (mm/dd/yyyy hh:mm:ss), thus forming time series attributes. The main measurement is the subcutaneous glucose, which is measured, in mg/dL, every five minutes using a Medtronic CGM and was calibrated twice per day with a standard glucometer [67]. The amount of carbohydrate in each

patient's meal, in grams, is also recorded. Finally, the infused insulin through the patient-attached pump is recorded. Two types of insulin are recorded, namely, basal and bolus [68]. Basal is the background insulin that is infused every hour with a constant amount that can be changed by the patient. This type of insulin guarantees a constant supply to keep the amount of blood glucose in balance. Bolus, on the other hand, is the extra amount of insulin that is infused in response to food intake. The amount of bolus depends on the quantity of carbohydrate in a given meal.

The hypoglycemia event is not logged into the patients' excel files. As a result, a binary attribute for hypoglycemia (hypo attribute) is added to the collected data. Its value is one where the glucose is less than or equal to sixty; otherwise it is otherwise. The hypoglycemia threshold (60 mg/dL) was not chosen arbitrary; it was the suggestion of the AH physician advising this study. Six patients out of the ten, as shown in Table 6, have experienced hypoglycemia during their monitoring period, whereas, patients 2, 4, 6, and 7 did not.

Table 6. Meta-Data describing the participating diabetic patients

Patient ID	Gender	Age	Number of instances	Monitoring duration (day:hh)	Prevalence of hypoglycemia
1	Female	21	2261	7 days: 20 hrs	15%
2	Male	20	236	19 hrs	0%
3	Female	19	3402	11 days: 19 hrs	5.39%
4	Female	18	1491	5 days: 4 hrs	0%
5	Female	19	12860	44 days: 15 hrs	2.46%
6	Female	11	1591	5 days: 12 hrs	0%
7	Female	15	3810	13 days: 5 hrs	0%
8	Female	15	4268	14 days: 19 hrs	1.06%
9	Male	42	2272	7 days: 21 hrs	1.56%
10	Female	70	17787	61 days: 18 hrs	0.46%

Chapter 3

Predicting Hypoglycemia Events using the Time Sensitive Artificial Neural Network (TS-ANN) Technique

This chapter presents the details of the first approach, i.e., TS-ANN, developed to predict hypoglycemia events. In this approach, and as illustrated in Figure 6, a time sensitive artificial neural network is utilized to predict the future subcutaneous glucose measurements. These values are then used, to predict the occurrence of hypoglycemia events within a given prediction horizon. Regarding the input to ANN, different scenarios were considered: firstly, using current and past subcutaneous glucose measurements as inputs to ANN; secondly, the inputs are both subcutaneous glucose measurements and insulin; thirdly, using the carbohydrate amount as well as subcutaneous glucose measurements; and finally, glucose predictions are made using subcutaneous glucose measurements, insulin, and carbohydrates. The goal here is to investigate the effects of these input variables on the accuracy of the developed hypoglycemia prediction method.

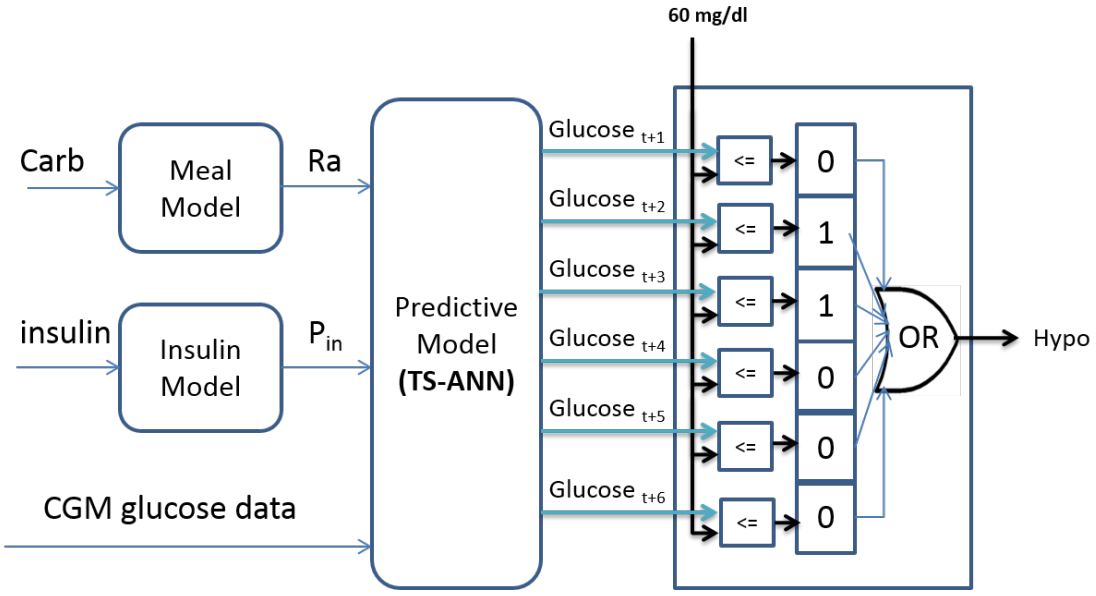

Figure 6. Prediction model of the TS-ANN approach

Figure 6 illustrates the final scenario. Two mathematical models were used, namely, meal and insulin models in order to compute "the rate of appearance of glucose in plasma after a meal (R_a)" and "plasma insulin (P_{in})", respectively. Thus, the input to the predictive model is R_a, P_{in}, as well as the current and past subcutaneous glucose measurements. The output generated from the time sensitive ANN predictive model are also the future 6 subcutaneous glucose measurements, generated once each five minutes. The if-else block is used to compare the predicted subcutaneous glucose measurements with the hypoglycemia threshold (60 mg/dl). If the value is below the hypoglycemia threshold, one is stored in the corresponding cell of the output array; otherwise zero is stored. The values of the array cells are ORed using the binary OR function, therefore at least one occurrence of hypoglycemia within the prediction horizon will result if the Hypo output equals to one. Figure 7 illustrates the preprocessing stage for the subcutaneous glucose input, as well as the Hypo output. In this figure, the horizontal line crossing the figure represents the hypoglycemia threshold, where any value below the line is considered a hypoglycemic event, and 1 represents this value, while contrary 0 values represent non-hypoglycemic events for glucose values above the line.

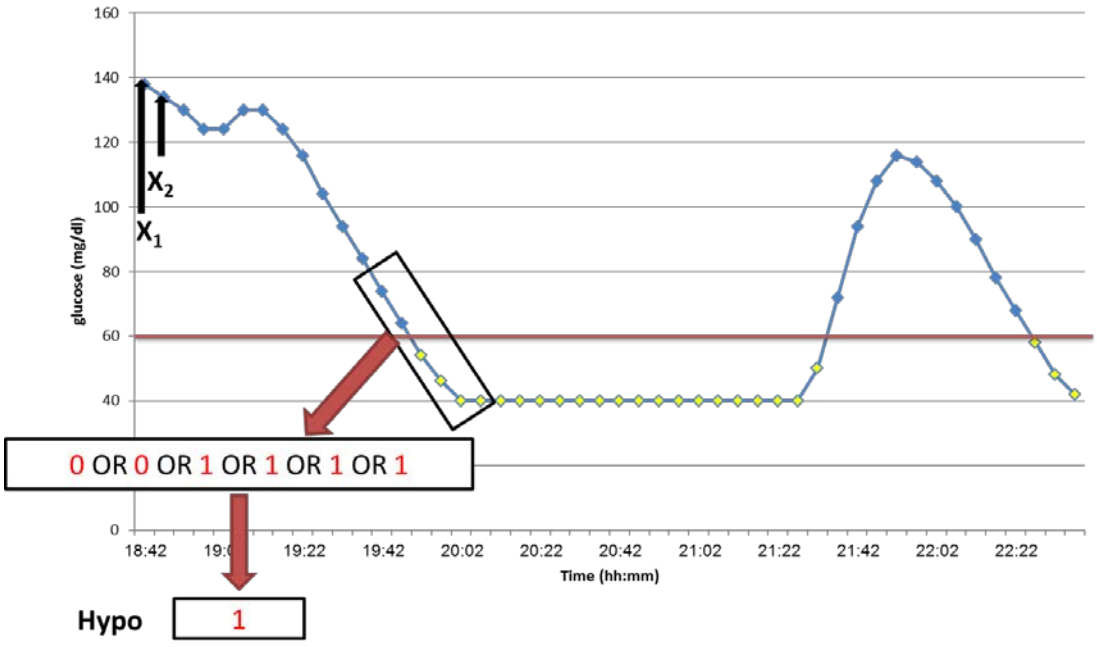

Figure 7. Illustration of the preprocessing of s.c. glucose input, and Hypo output for Patient 1

3.1 The Meal Mathematical Model

The amount and time of carbohydrates (Carb) in a meal is logged in the file of any given patient; mainly there are three meals a day, but snacks between the meals are also recorded. A meal model is used to describe the ingestion and absorption of carbohydrates intake. The model by Lehmann and Deutch [69] is used for this purpose. Specifically, the model calculates the "rate of appearance of glucose in plasma" G_{in} assuming that the rate of gastric emptying G_{empt} (mmol/hr) is a trapezoidal function. In this research, we reproduced the equations of [69]; next the equations and their description are presented, and finally these are used to calculate the input R_a of the prediction model of Figure 6.

The $G_{empt}(t)$ function for Carb millimoles of carbohydrate in a meal greater than $Carb_{crit}$ is defined as:

$$G_{emp}(t) = \begin{cases} \frac{Vmax}{Tasc}, & t < Tasc \\ Vmax, & Tasc < t \leq Tasc + Tmax \\ Vmax - (\frac{Vmax}{Tasc})(t - Tasc - Tmax), & T_{asc}+T_{max}<t\leq T_{max} + T_{asc} + T_{desc} \\ 0, & otherwise \end{cases} \quad \textbf{(1)}$$

$$Tmax = \frac{Carb - \left(\frac{1}{2} * Vmax * 2 * (Tasc + Tdesc)\right)}{Vmax} \quad \textbf{(2)}$$

Where T_{max} is the duration of time in which G_{emp} is constant and maximum (V_{max}). V_{max} is the maximum rate of gastric emptying function. T_{asc}, T_{desc} is the duration time of the ascending and descending of G_{emp} curve where the default value for T_{asc}, and T_{desc} is 0.5 hr (30 min) for each of them. However, for Carb value below critical level $Carb_{crit}$, T_{asc}, and T_{desc} are defined as:

$$\mathbf{Tasc} = \mathbf{Tdesc} = \mathbf{2} * \frac{\mathbf{Carb}}{\mathbf{Vmax}} \quad \textbf{(3)}$$

Where $Carb_{crit}$ is defined as:

$$\mathrm{Carb_{crit}} = \frac{[(\mathrm{Tasc} + \mathrm{Tdesc}) * \mathrm{Vmax}]}{2} \quad \textbf{(4)}$$

The amount of glucose in the gut, q_{gut}, following the ingestion of a meal containing Carb millimoles of glucose equivalent carbohydrates is defined as:

$$\frac{\mathrm{d(Ggut)}}{\mathrm{dt}} = \mathrm{Gemp} - \mathrm{kgabs} * \mathrm{Ggut} \quad \textbf{(5)}$$

Where $kgabs$ is the rate constant of glucose absorption from the gut into the systemic circulation, and $Gemp$ is the gastric emptying function defined in Equation 2.
Finally, the rate of appearance of glucose in plasma (mmol/hr) is given as:

$$\mathrm{Gin(t)} = \mathrm{Kgabs} * \mathrm{Ggut(t)} \quad \textbf{(6)}$$

Meal Model Implementation:

Our carbohydrate amount is measured in grams as opposed to the meal model of [69], which deals with millimoles of carbohydrate. Therefore, we converted the carbohydrate amount of a meal from grams to millimoles as follows:

1. Determine the molar mass of the carbohydrate compound, using the molecular formula of carbohydrate ($C_6h_{12}O_6$) and the periodic table [70] to find the atomic masses of each atom in the compound.
2. Calculate the total molar mass (gram/mole) of the carbohydrate compound by summing up the atomic masses of each atom in the compound. The molar mass of carbohydrate, therefore, is 6*(12.01) + 12*(1.008) + 6*(16) which computes to 180.156 gram/mole.
3. Using (7) and substituting molar mass with 180.156 converts the amount of carb from grams to moles. Then multiply the (Carb in moles) value, obtained from (7), by 1000 in order to convert it to millimoles.

$$\text{Carb in moles} = (\#\text{ of grams in Carb}) \div (\text{molar mass of Carb}) \quad \textbf{(7)}$$

After converting the carbohydrate amount into millimoles, we then synchronized the carbohydrate data with the CGM sensor glucose. The next step is to implement the meal model using our dataset. Applying equations 1 through 6 and using the parameters given in [69] we have calculated the rate of glucose appearance in plasma G_{in}. We have solved the ordinary differential equation (Equation 5) numerically using the Euler first order differential equation technique [71]. The meal model is represented diagrammatically in the flowchart of Figure 8. The flow chart is implemented using java code. For each patient, the corresponding data file is opened for reading; then date, time, and subcutaneous glucose values are extracted and sorted in ascending order according to date and time. The subcutaneous glucose and the Carbohydrate amount (Carb) for each patient are stored in one excel file but on separate sheets. Thus, the extraction and sorting is repeated for the Carb. The Carb sheet contains many NULLs as the meals are taken only a few times a day; accordingly, the NULL is removed from Carb. Then the Carb amount is converted from gram to mmol. As mentioned earlier, the Carb and subcutaneous glucose are separated into two sheets, thus the value for same timestamp is different. Any couple of events, one in the Carb and the other in glucose, is synchronized when the condition mentioned in the following lines of code is satisfied, namely:

```
diffInMin = Time2 – Time1;
if diffInMin <= 2 minutes

Write to file Time1 and Carb and glucose
```

Where the difference between the events is less than or equals to two minutes. The meal model developed earlier is used to simulate the Gin presented above. Finally, the subcutaneous glucose and the simulated Gin, along with the corresponding date, are stored in a file.

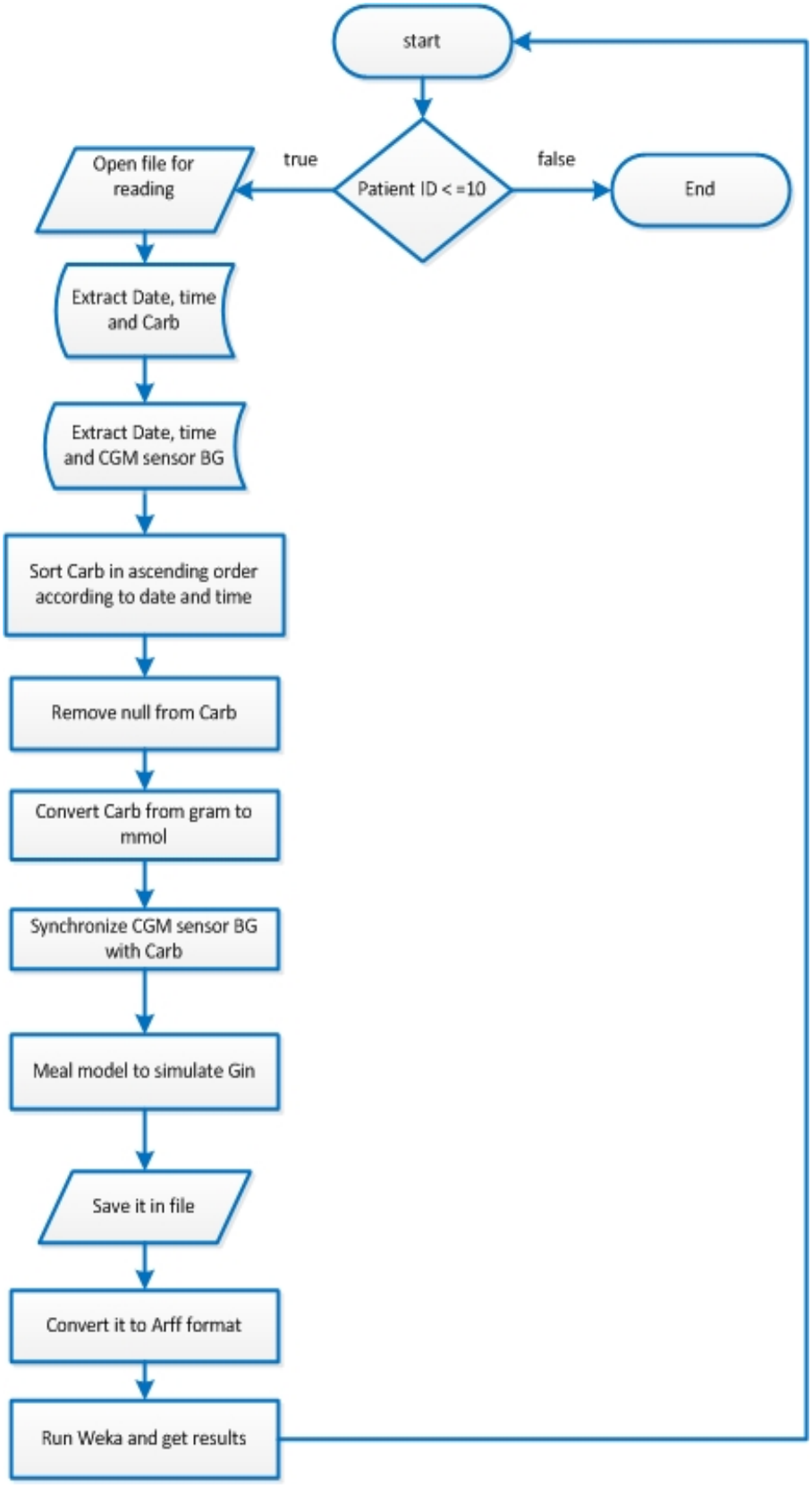

Figure 8. The flow chart of the meal model simulation

Euler Method:

The Euler method [71] is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. In our case, the step size h is five minutes (1/12 hr). G_{in} (t) is simulated for each carbohydrate intake. G_{in} (t) , as shown in Figure 9, will eventually reach zero after a certain time; this time depends on the amount of Carb. Thus in order to keep track of the different G_{in} (t) that corresponds to the different carbohydrate intakes, we used a stack; and for each time step, we have summed up the different G_{in} (t) available in stack. Gin (t) is popped out from the stack when it reaches zero.

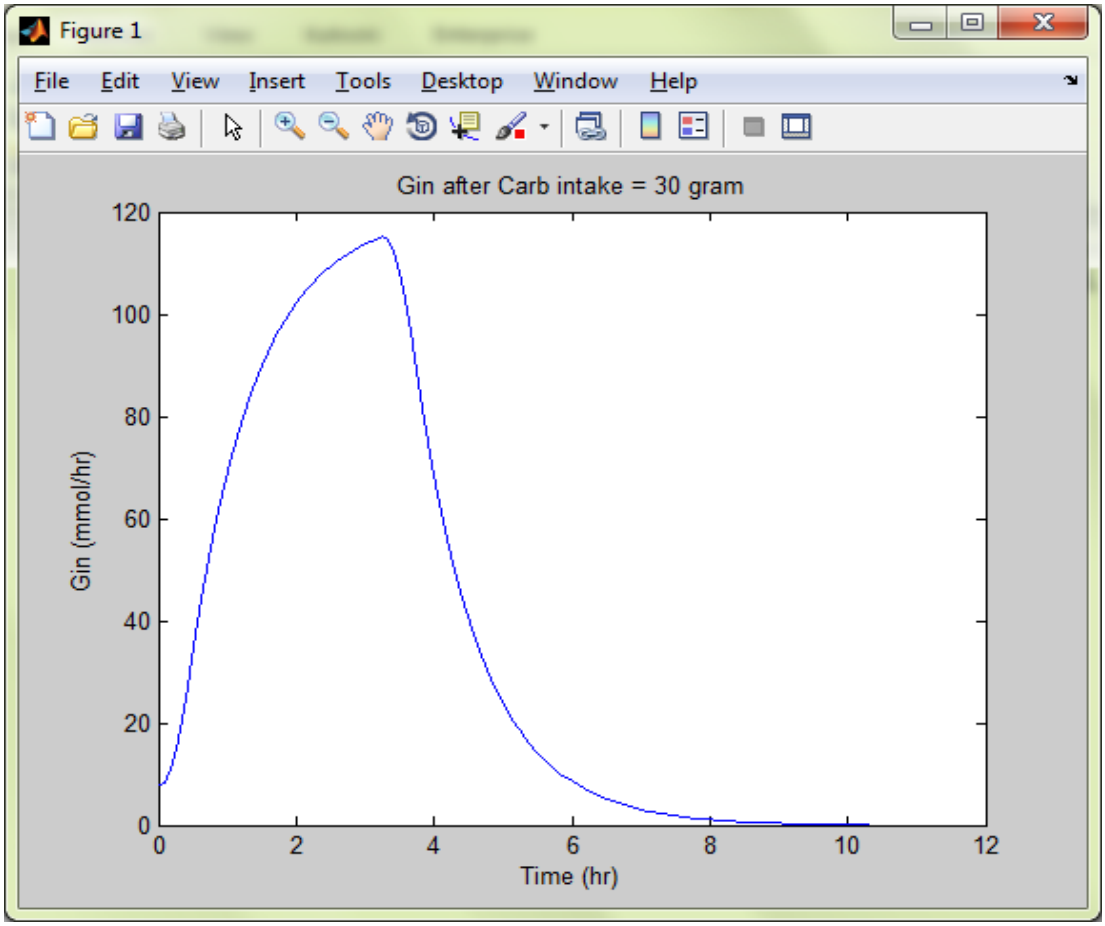

Figure 9. The rate of appearance of glucose in plasma (Gin) following ingestion of 30 gram of carbohydrate

3.2 The Insulin Mathematical Model

Insulin is a peptide hormone, produced by beta cells of the pancreas. It is crucial for controlling the metabolism of carbohydrate and fat in the body. The glucose included in the carbohydrate will be absorbed into the bloodstream and blood glucose levels will rise. Insulin causes cells in the liver, skeletal muscles, and fat tissue to remove glucose from the blood [72]. The insulin model is used to simulate the subcutaneous absorption and the pharmacokinetics/pharmacodynamics of insulin following subcutaneous injections; moreover, it describes how insulin enters plasma after absorption. In literature, the insulin models mainly cover two types of insulin:

- Soluble (regular, hexameric) insulin
- Rapid-acting insulin analogs (monomeric insulin)

Most research efforts went toward soluble insulin; however, few researchers analyzed the novel monomeric insulin administered through the pump; as reflected in our focus in this study. The developed models are used to quantify two processes: absorption of insulin and insulin transportation to plasma after subcutaneous injection. The different available models vary mainly in the first process, where plasma insulin is assumed to be a single compartment in all developed models. As mentioned earlier, most of the developed models are suited for soluble insulin, contrariwise, two researches mainly analyzed monomeric insulin [73], [74]. We have used the proposed model of [74] to simulate the plasma insulin after subcutaneous insulin injections. [74] simplified the proposed model of [75] for subcutaneous insulin absorption and estimated the models' parameter values; furthermore, the model was extended to analyze monomeric insulin absorption.

- **Subcutaneous insulin absorption:**

According to the model proposed by [74], the subcutaneous absorption of insulin analogues in spherical coordinates can be quantified using the partial differential equation (8).

$$\frac{\partial m(r,t)}{\partial t} = D\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial m}{\partial r}\right) - Bm \qquad \textbf{(8)}$$

Where m is monomeric insulin, D is diffusion constant, and B is absorption rate constant.

Equation (8) is solved numerically by spatial discretization of subcutaneous region into fifteen spherical shells, as shown in Figure 10, where the center is the injection point. Constant insulin fluxes between two consecutive shells are assumed.

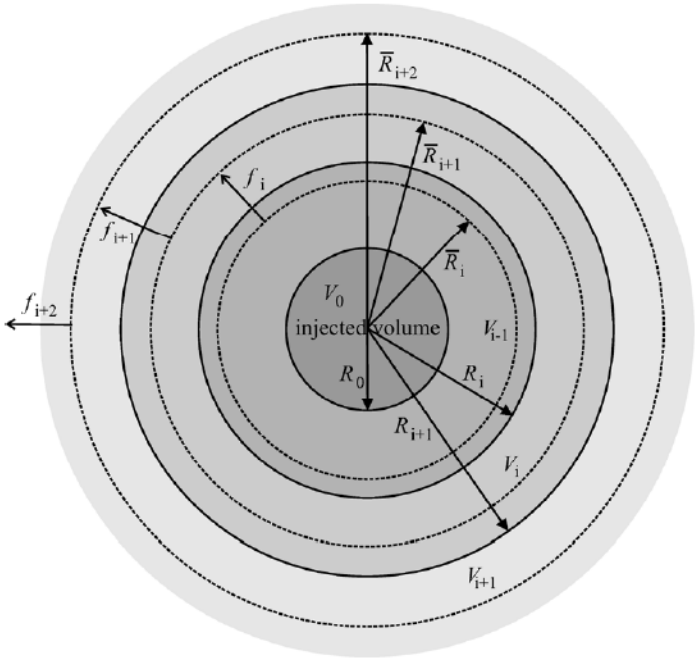

Figure 10. Spheric grid for the spatial discretization [74]

The pdepe function in MATLAB software [64] was used to solve the PDE of equation (8). The approximate solution was obtained by integrating the Ordinary Differential Equation (ODE) resulting from the discretization in space.

sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan) solves the initial-boundary value problems for systems of parabolic and elliptic PDEs in the one space variable x and time t. pdefun, icfun, and bcfun are function handles that defines the components of partial differential equation, initial conditions, and boundary conditions, respectively.

pdepe solves PDE of the following form:

$$\boldsymbol{c}\left(\boldsymbol{x},\boldsymbol{t},\boldsymbol{u},\frac{\boldsymbol{\partial y}}{\boldsymbol{\partial x}}\right)\frac{\boldsymbol{\partial u}}{\boldsymbol{\partial t}} = \boldsymbol{x}^{-\boldsymbol{m}}\frac{\boldsymbol{\partial u}}{\boldsymbol{\partial x}}\left(\boldsymbol{x}^{\boldsymbol{m}}\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{t},\boldsymbol{u},\frac{\boldsymbol{\partial u}}{\boldsymbol{\partial x}}\right)\right) + \boldsymbol{s}(\boldsymbol{x},\boldsymbol{t},\boldsymbol{u},\frac{\boldsymbol{\partial u}}{\boldsymbol{\partial x}}) \quad \textbf{(9)}$$

Where the PDEs hold for a time interval of $t_0 \leq t \leq t_f$ and a space of $\leq x \leq b$. m can be 0, 1, or 2, corresponding to slab, cylindrical, or spherical symmetry, respectively. f (x,t,u,∂u/∂x) is the flux term and s (x,t,u,∂u/∂x) is the source term.

For $t = t_0$ and all x, the solution components satisfy initial conditions of the form presented in Equation 10.

$$u(x,t_0) = u_0(x) \quad \textbf{(10)}$$

For all t and either x = a or x = b, the solution components satisfy a boundary condition of the form.

$$p(x,t,u) + q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right) = 0 \quad \textbf{(11)}$$

In order to solve (8), we need two boundary conditions and one initial condition.

The boundary conditions for (8) are:

1. $\frac{\partial m}{\partial r}\Big|_{x=0} = 0$ **(12)**

2. $m\Big|_{x=0} = 0$ **(13)**

And the initial condition is:

$$m\Big|_{t=0} = \begin{cases} 40 & 0 < x < R_n \\ \\ 0 & x > R_n \end{cases} \quad \textbf{(14)}$$

$$\text{Where } R_n = \sqrt[3]{\frac{3*V0}{4\pi}} \quad \textbf{(15)}$$

Where $V0$ is the injection volume (ml)

$$\text{And injection volume (ml)} = \frac{\text{Dose (U)}}{\text{insulin concentration (U/ml)}} \quad \textbf{(16)}$$

It is worth mentioning here, Equation 15 of [74] has a mistake where it uses the square root instead of cubic one, according to Archimedes formula for spherical volume. The results obtained by the numerical approximation of the monomeric insulin equation (8), with the predefined boundary and initial conditions are given in Figure 11.

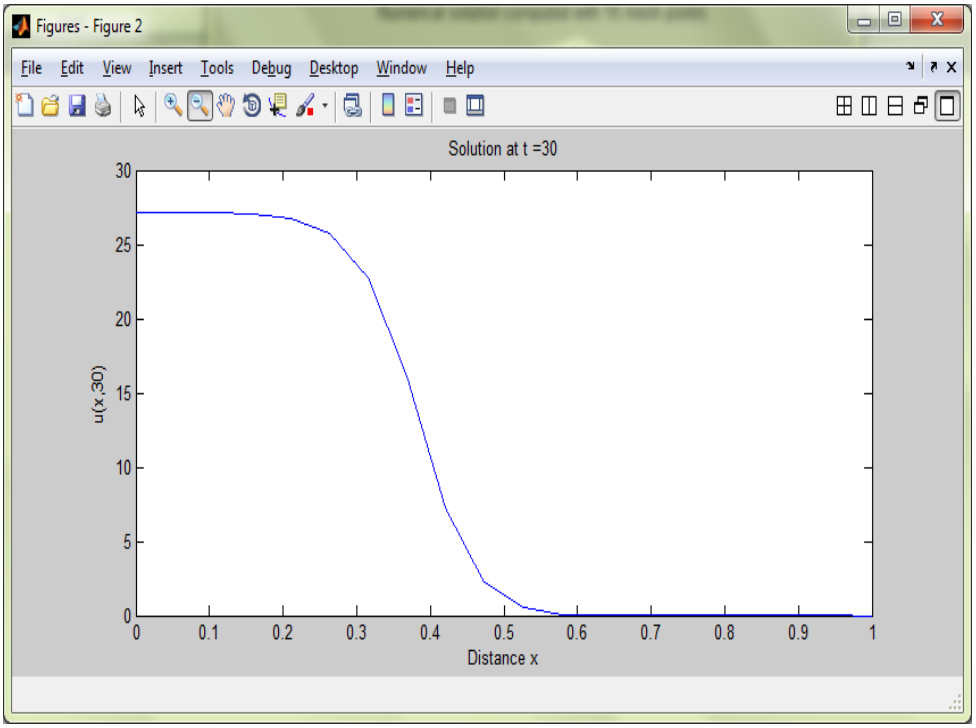

Figure 11. Concentration of monomeric insulin following injection of 10 U, 40 U/ml after 30 min obtained by numerical solution

- **Plasma insulin:**

Subcutaneous insulin, after its absorption, is transported to plasma. According to [74], plasma insulin can be calculated using the following Ordinary Differential Equation (ODE).

$$\frac{di}{dt} = \frac{1}{V_p}\int_{V_{sc}} (Bd)dV - K_e i \qquad \textbf{(17)}$$

Where Vp, B, and Ke are the distribution volumes for insulin, the absorption rate constant, and the insulin elimination rate constant consequently. The values for these constants are defined in [74]. i is the concentration of plasma insulin and d is the concentration of monomeric insulin. The integration part of Equation 17 was computed numerically using the trapezoidal method (with unit spacing). Precisely Z = trapz(X, Y) function of matlab was applied, where Y= Bd. Now we have the following ODE:

$$\frac{di}{dt} = \frac{Z}{Vp} - ke * i \qquad \textbf{(18)}$$

Equation 18 was been solved numerically using variable the step Runge-Kutta method to resolve first order differential equations. The results obtained by solving Equation 18 numerically are provided in Figure 12.

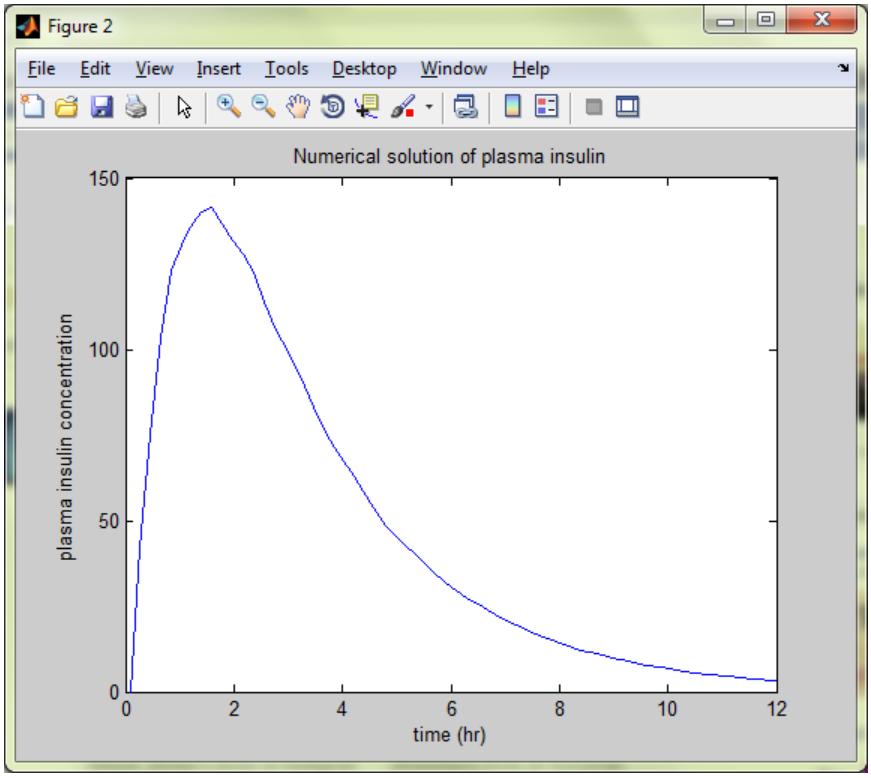

Figure 12. plasma insulin concentration after subcutaneous injection of 10 U monomeric analogue

3.3 Neural Network based glucose prediction

Artificial neural networks (ANN), or simply neural networks (NN), represent a type of computing model inspired by the way the human brain performs computations. Neural networks are good at fitting non-linear functions and recognizing patterns [76]. A neural network consists of interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation [78]. Figure 13 illustrates the structure of a single artificial neuron that is the constitutive unit in ANN.

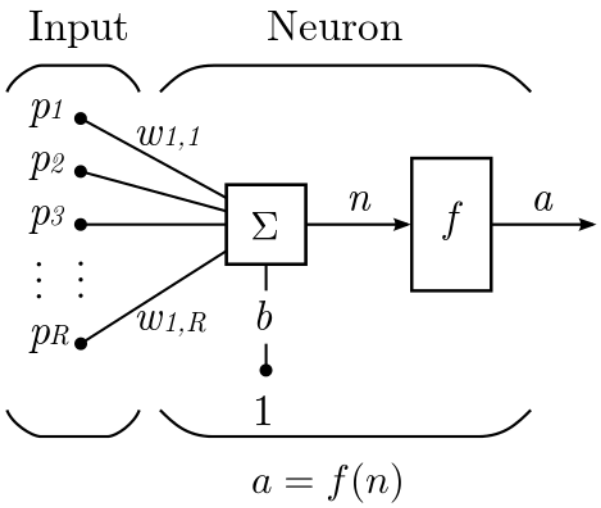

Figure 13. Artificial neuron [79]

The individual inputs $p1, p2, \ldots, p_R$ are each weighted by the corresponding elements $w_{1,1}, w_{1,2}, \ldots, w_{1,R}$ of the weight matrix W, the net input n can be calculated by:

$$n = w1,1\, p1 + w1,2\, p2 + \ldots + w1,R\, p_R + b \quad \textbf{(19)}$$

Where b is the bias of the neuron. We can rewrite Equation 19 in the matrix form as:

$$n = Wp + b \quad \textbf{(20)}$$

Accordingly, the output of the neuron (a) can be written as $a = f(Wp + b)$ where f is the transfer function. The transfer function can be a linear or non-linear function of the net input n. Table 7 lists the three transfer functions f that we used in our prediction problem. The structure of the neuron is defined by the number of inputs

and the transfer function used. During the training process, weights and biases of the neurons are adjusted to give the most accurate output.

Table 7. The utilized Transfer functions

Name	Functionality	Icon	MATLAB Name
Linear	$a = n$		Purelin
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		Logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		Tansig

Neural networks are composed of single or multiple layers of interconnected neurons. The architecture of the neural network is defined by the number of layers, number of neurons per layer, and the type of connections between these neurons.

Recently, neural networks have been used widely to solve various problems in many fields. Depending on the type of the problem, suitable neural network architectures are used. Our problem is subcutaneous glucose prediction, specifically a time series prediction. Neural networks can be categorized into two broad groups: static and dynamic neural networks. Static networks have no feedback elements and contain no delays; the output is calculated directly from the input through feedforward connections. In dynamic networks, the output depends not only on the current input, but also on the previous inputs, outputs, or states of the network. For time series predictions, dynamic neural networks are used. In the following section, wedescribe the most widely used neural network architectures that can be used to simulate the time series prediction.

3.3.1 Neural network architectures

In this thesis, the MATLAB Neural Network Toolbox, which contains several neural networks architectures, is used. The following subsections describe briefly the various dynamic neural network architectures suitable for our problem, i.e., time series analysis and dynamic systems. The following subsections are taken from MATLAB documentations [80].

- **Time delay neural network (TDNN)**

TDNN is a straightforward dynamic neural network organized at the input, and as shown in Figure 14, in a feed forward fashion with a tapped delay line (TDL). The TDL breaks the input up in time and these delayed values are fed into the network. TDNN is considered part of a general class of dynamic networks, called focused networks. FTDNN is essentially a static multilayer feed-forward network; however, its input structure introduces the dynamism into the system. Figure 14 illustrates a two-layer FTDNN.

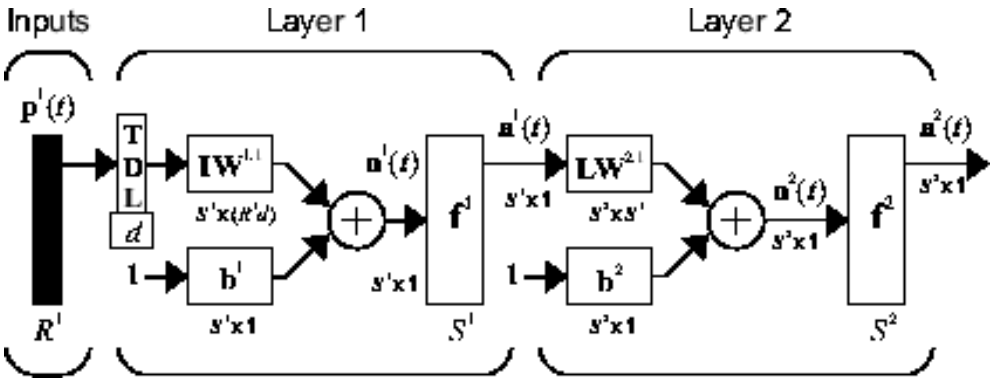

Figure 14. The organization of the Time delay neural network (TDNN) [80]

- **The nonlinear autoregressive network with exogenous inputs (NARX)**

The previously mentioned neural network is a focused network, with the dynamism introduced only the input layer. However, the nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosing several layers of the network. Figure 15 illustrates the architecture of a two-layer NARX network.

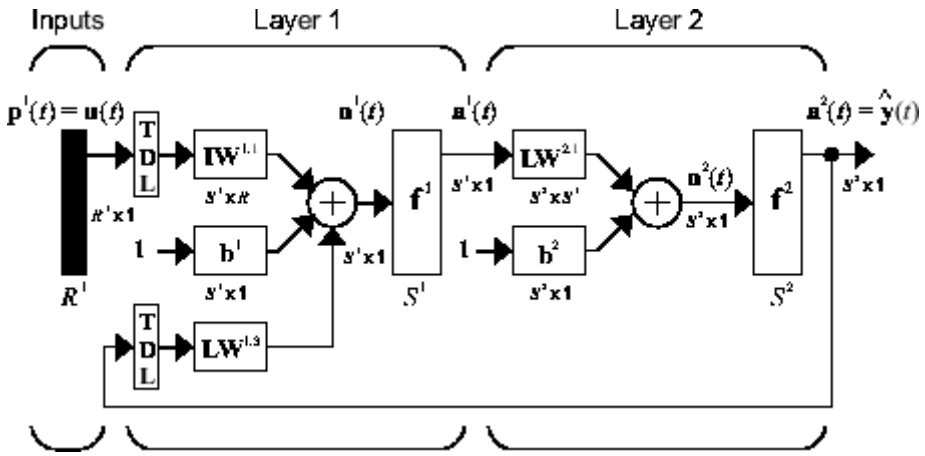

Figure 15. Two layer nonlinear autoregressive network with exogenous input [80]

The defining equation for the NARX model is:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n), u(t-1), u(t-2), \dots, u(t-m)) \quad \textbf{(15)}$$

According to Equation 15, the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal (y(t-i) where i= 1, 2, .., n), and previous values of an independent (exogenous) input signal (u(t-s), where s=1, 2, .., m).

- **The distributed time delay neural network (DTDNN)**

The FTDNN, as described above, has the tapped delay line memory only at the input to the first layer of the static feed-forward network. Contrary, the tapped delay lines of the DTDNN are distributed throughout the network. The architecture of the distributed TDNN is illustrated in Figure 16. The distributed TDNN was first introduced in [81] for phoneme recognition. The original architecture was highly specialized to this particular problem.

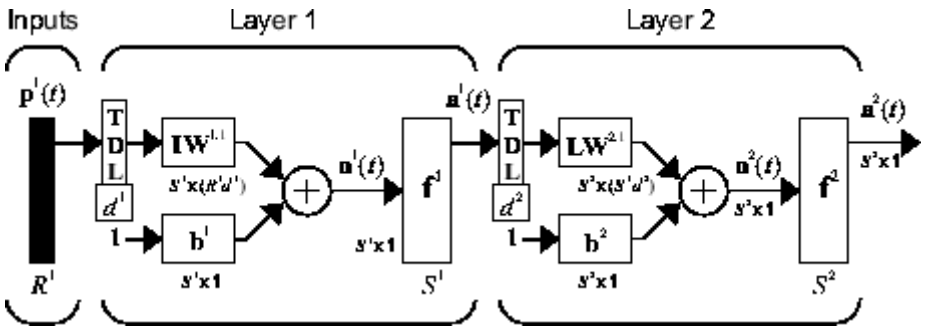

Figure 16. Two layer distributed time delay neural network [80]

3.3.2 Neural network training algorithms

During the training process of the neural network, the training function adapts the weights and biases of the neurons, in order to produce the most accurate output. The MATLAB Neural Network Toolbox contains various training functions to train a neural network. Table 8 lists the training algorithms available in the neural network toolbox. We use these training algorithms as they are; the details of these algorithms are out of the scope of this study.

Table 8. Training algorithms

Function name in MATLAB	Training algorithm
trainlm	Levenberg-Marquardt back-propagation
trainbr	Bayesian Regularization
trainbfg	BFGS Quasi-Newton back-propagation
trainrp	Resilient Backpropagation back-propagation
trainscg	Scaled Conjugate Gradient back-propagation
traincgb	Conjugate Gradient with Powell/Beale Restarts back-propagation
traincgf	Fletcher-Powell Conjugate Gradient back-propagation
traincgp	Polak-Ribiére Conjugate Gradient back-propagation
trainoss	One Step Secant back-propagation
traingdx	Variable Learning Rate Gradient Descent back-propagation
traingdm	Gradient Descent with Momentum back-propagation
traingd	Gradient Descent back-propagation

3.3.3 Inputs and outputs of the neural network

Here we are predicting the subcutaneous glucose measurements using neural networks. Regarding predicting the glucose concentration, we used several approaches as mentioned earlier. Firstly, glucose prediction uses an univariate glucose time series. Secondly, glucose prediction uses a multivariate time series of glucose and insulin. Thirdly, glucose prediction uses a multivariate time series of glucose and carbohydrate. Finally, glucose prediction uses a multivariate time series of glucose,

insulin, and carbohydrate. The inputs to the NNs for the first approach are seen in the recent history values of subcutaneous glucose measurements $[y(t-1), y(t-2), \dots, y(t-n)]$; for the second approach, in addition to the recent history values of subcutaneous glucose measurements, recent history of plasma insulin measurements are used $[u(t-1), u(t-2), \dots, u(t-m)]$. In the third approach, the recent history of rate of appearance of glucose (Gin) $[z(t-1), z(t-2), \dots, z(t-m)]$ instead of plasma insulin is used. The final approach uses all these attributes: recent history values of subcutaneous glucose measurements, recent history of plasma insulin measurements, and recent history of rate of appearance of glucose as inputs to the ANN. The output in all the aforementioned cases is the next step subcutaneous glucose measurement $y(t)$. The dataset collected from diabetic patients of the American Hospital of Dubai is divided into three parts. The first block is 70% of the dataset used for training, the second block is 15% of dataset used for validation, and finally, the remaining 15% of the dataset is used for testing.

In the following sections all the results were obtained from a representative subject (patient 4), and it can be noted that the results obtained from the other patients are quite similar. The output in all the following sections is the next step subcutaneous glucose measurement. The input to TDNN, NAR, and distributed TDNN, is the recent history of subcutaneous glucose measurement; however, for NARX network Gin and plasma insulin attributes are used in addition to the subcutaneous glucose measurements. The prediction window (PW) used, in all the following sections, is a function of the prediction horizon, namely the PW is twice the prediction horizon.

3.3.4 Selecting best neural network architectures and training algorithms

In an effort to find the most appropriate architecture for our problem, we experimented with all types of the neural network structures, previously mentioned, using the training algorithms listed in Table 8, all while measuring the RMSE of the different combinations. It is obvious from Table 9, that both the trainlm and trainbr algorithms have outperformed all other training algorithms. Moreover, NAR and NARX networks architectures performed better than TDNN and DTDNN networks. In the following section, we will will look at the different combinations of the two best network architectures and training algorithms while changing the number of neurons in the hidden layer and trying different transfer functions.

Table 9. RMSE of the different neural network architectures trained using various training algorithms

Training algorithm	Neural Network Structure				
	TDNN	NAR	NARX	DTDNN	Average (across all network structures)
trainlm	5.03	4.85	2.99	11.48	5.34
trainbr	5.08	4.06	1.99	3.82	2.99
trainbfg	7.12	9.42	5.48	21.88	10.98
trainrp	7.73	22.63	8.40	51.38	22.54
trainscg	16.20	7.89	5.15	26.00	13.81
traincgb	20.27	7.16	5.26	12.87	11.39
traincgf	20.49	8.68	5.63	10.40	11.30
traincgp	23.02	10.33	5.52	23.01	15.47
trainoss	32.69	8.13	6.13	43.66	22.65
traingdx	88.76	38.97	33.08	71.33	58.04
traingdm	191.53	144.46	142.16	155.69	158.46
traingd	191.53	144.46	142.13	155.69	158.45
Average (across all training algorithms)	50.79	34.25	30.33	48.93	

3.3.5 Selecting the transfer function and number of neurons

In the previous section, we concluded the NAR and NARX networks, when trained using trainlm or trainbr training algorithms, gave the best prediction performance. Here, we attempt the different combinations of the best network architectures and training algorithms presented above while changing number of neurons in the hidden layer and altering the transfer function of the hidden layer. We studied two different transfer functions in the hidden layer, "logsig" and "tansig", but the transfer function of the output layer is kept as "purelin". Moreover, different numbers of neurons in the hidden layer are implemented namely, 10, 20, 30, …, and

100 neurons are used. It is observable from Table 10, Figure 17, and Figure 18 that "trainbr" is more stable than "trainlm" where increasing number of neurons in the hidden layer results in a lower RMSE error rate. "trainlm" tends to have some random nature where its performance did not improve when using more neurons. However, "trainlm" needs less training time as compared to "trainbr". In addition, the use of the "logsig" activation function results in a slightly better performance than using the "tansig" activation function. To sum up, for our prediction model we used the NAR or NARX networks trained using "trainbr" with 20 neurons in the hidden layer and using "logsig" as activation function. It is also noticeable from Table 10 that NARX outperformed NAR in all the different combination of number of neurons and activation functions. Although a higher increase in performance was expected, it was the limitation of the used mathematical model and the intra- and inter-individual differences that limited the increase.

Table 10. Comparison between the RMSE of NAR and NARX networks when using different number of neurons and different activation functions in a single hidden layer

Number of neurons	Neural Network Structure							
	NAR				NARX			
	trainlm		trainbr		trainlm		trainbr	
	Tansig	Logsig	Tansig	logsig	Tansig	Logsig	Tansig	Logsig
10	5.85	5.28	4.27	4.03	2.59	2.17	2.43	2.17
20	5.67	6.96	4.63	4.05	2.99	2.03	1.99	1.99
30	6.85	5.93	4.05	4.05	2.59	2.39	2.00	2.01
40	10.17	7.37	4.05	4.03	3.45	4.07	2.04	2.01
50	7.66	7.11	4.03	4.03	2.61	2.40	2.01	1.99
60	10.98	9.18	4.03	4.03	5.22	3.27	2.01	2.01
70	11.43	8.82	4.05	4.05	5.29	4.66	2.01	1.99
80	10.15	12.17	4.03	4.05	5.97	4.90	2.32	1.99
90	13.98	10.26	4.10	3.83	6.22	5.63	1.99	2.01
100	13.04	11.10	3.87	3.53	6.44	4.80	1.99	1.97

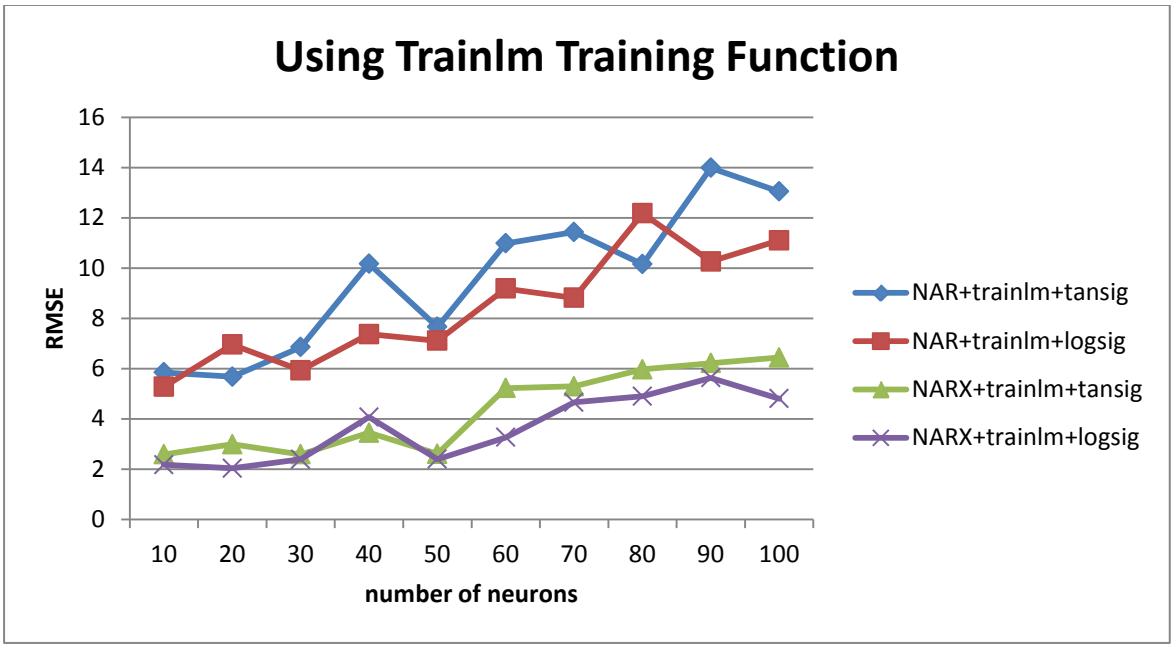

Figure 17. RMSE of various neural networks that were trained using trainlm training function

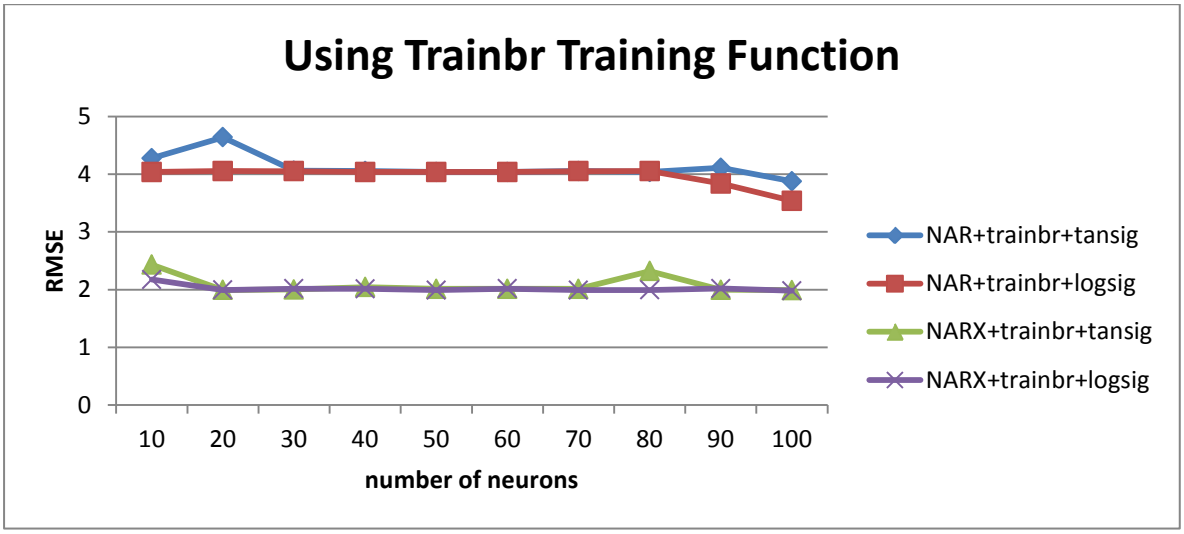

Figure 18. RMSE of various neural networks that were trained using trainbr training function

3.3.6 Selecting the appropriate inputs to the neural network

As mentioned previously in Section 3.3.3 regarding the input attributes to the neural network, we are following different approaches: firstly, the glucose prediction using univariate glucose time series; secondly, the glucose prediction using

multivariate time series of glucose and plasma insulin; thirdly, the glucose prediction using multivariate time series of glucose and G_{in} (the rate of appearance of glucose in plasma); finally, glucose prediction using multivariate time series of glucose, plasma insulin, and G_{in}. Figure 19 illustrates the performance of the different scenarios.

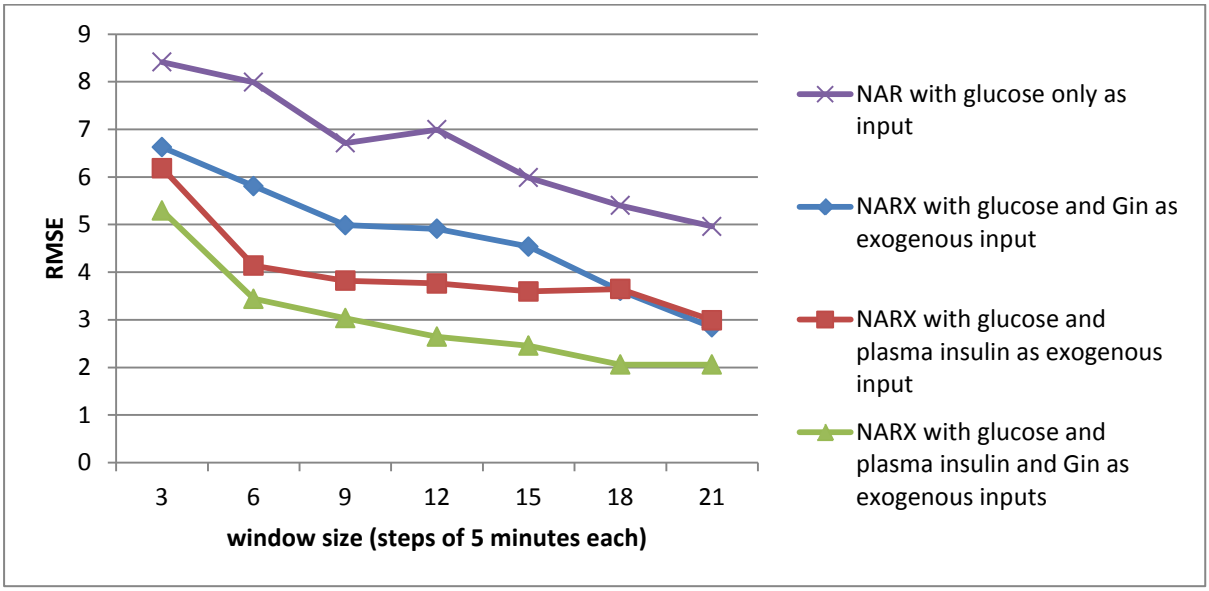

Figure 19. RMSE of neural networks when using different exogenous inputs

It is noticeable from Figure 19 that the worst performance was obtained when using NAR network, i.e., no exogenous inputs are used (only the glucose is used). The use of both exogenous inputs, G_{in} and plasma insulin, resulted in the lowest error rate

3.3.7 Nonlinear autoregressive with exogenous input neural network multistep predictor

What we have discussed so far regarding the configuration of the neural networks, has been for the single step prediction. In this section, the best NN configurations identified will be used to simulate the multistep subcutaneous glucose prediction. The neural network used is NARX with plasma insulin and Gin as exogenous inputs, trained using 'trainbr' training algorithm, and has a single hidden layer with 20 neurons and 'logsig' as its activation function. The prediction horizon (PH) used is thirty minutes, six steps where each step is five minutes. Thus, six different subcutaneous glucose measurements are predicted - one for each step. The neural networks were trained for each patient separately in order to reflect the diverse lifestyles and insulin regimens of the different patients. The results are discussed in Chapter 5.

3.3.8 Predicting hypoglycemia events using time sensitive ANN

The neural networks discussed earlier are designed to predict future subcutaneous glucose measurements; however, the goal of this study is to calculate the future occurrence of hypoglycemia events. The best neural network mentioned in section 3.3.7 is used to predict the six steps ahead of the subcutaneous glucose measurements, as shown in Figure 6: glucose $_{t+1}$, glucose $_{t+2}$, …, and glucose $_{t+6}$. These predicted measurements are compared to the hypoglycemia threshold; a threshold of 60 mg/dl is used. If the subcutaneous glucose value is below the threshold, then we store in the corresponding vector of cells the value 1, 0. This vector will be fed into a binary OR function, the output becoming Hypo. Hypo ends up being; on the other hand, the Hypo is 1 if at least one of the vector cells falls below 60, otherwise, it is zero. From there, this value of Hypo is stored into the corresponding cell of the Predicted_Hypo array. This process is iterated until all the instances in the dataset are covered. The actual_Hypo contains the actual occurrences of hypoglycemia events in the dataset. We created a matlab script for comparing the actual versus the predicted hypoglycemia; and to help us calculate the performance metrics of the prediction models. This process is carried out for each patient separately.

Chapter 4

Predicting Hypoglycemia Events using Tree Based Temporal Classification (TBTC) Technique

In the temporal classification, we predicted the future occurrence of hypoglycemia events using a set of features extracted from a patient's subcutaneous glucose signal. The target attribute (Hypo) is a binary one, while the input is the glucose time series; therefore the hypoglycemia problem becomes a temporal binary classification one. The objective of the prediction model in the TBTC approach is not to predict future values of the time series like the TS-ANN, but rather to classify hypoglycemia events based on past observations of time series. Three approaches for temporal classification are possible [82]:

1. Develop learning algorithms to deal specifically with temporal classifications. This is an approach taken, for example, with dynamic time warping and hidden Markov models [83].

2. Represent the problem in such a way as to allow the application of relational learning techniques (i.e., ILP) [84].

3. Represent the problem in such a way that allows the application of the off the shelf machine learning algorithms [82].

In this study, we followed the third approach. The main issue here is how to represent the temporal attributes in a manner that makes it amenable to traditional machine learning techniques. [82] proposed a novel feature construction technique of metafeatures. A metafeature employs event primitives to analyze the training data and extract events. Metafeatures parameterize events in a way as to capture their properties, as well as their temporal characteristics. As with traditional supervised classification, the input comprises a set of training instances and the related class labels. Nevertheless, the instances here are streams. Every stream is composed of a series of frames. Each frame represents an instant of time of the stream, and consists of a set of measurements from different sources, called channels. Figure 20 illustrates

the relationship between the aforementioned terms. In our case, as shown in Figure 21, we have one channel (sourcerepresented as subcutaneous glucose measurements. Moreover, the stream (instance) is composed of 13 frames, i.e. ,we used the last one hour to predict the hypoglycemia.

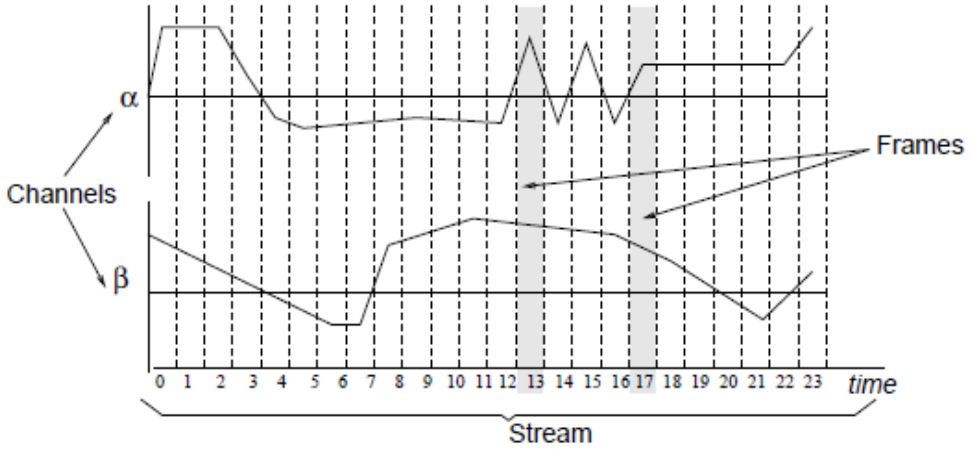

Figure 20. The relationship between channels, frames, and streams [82]

The data is processed in two ways: global feature calculation and event extraction. A global feature calculation computes various features for the stream as a whole. Typical examples for global features are global maxima and global minima. Event extraction is applied by using a set of parameterized event primitives (PEPs). PEPs are domain specific, thus they represent types of events expected to occur in a particular domain. Every PEP is defined as follows:

- A tuple of parameters $p = (p1, \dots, pk)$ describes a particular event. Let $P1$ be the range of $p1$ parameter. Let P be the parameter space $P = P1 \times P2 \times \dots \times Pk$, i.e., the space of all possible parameter values.

- A finding function f takes a stream s and returns a set of parameterized events e.

It is worth mentioning here that the temporal characteristics of the events must be explicitly represented as parameters in any given PEP. This means it is using the time to aid learning. A simple example of PEP is local minimum. Two parameters can define this PEP; the time at which the local minimum occurred and its height. The

PEP is represented with tuples of (t, h) where t is the time at which the local minimum occurred and h is the height of the minimum. The finding function would be given a stream, thereby returning all pairs of (t, h). Since each PEP is described using a set of parameters, clustering can be performed in the parameter space of each PEP. The clustering algorithm will divide the parameter space into regions (clusters). Each cluster forms an event attribute. For each stream (training instance), the value for the event attribute is true only if the stream has an event belonging to that cluster. The membership be a binary simple true/false value or a relative membership may be used. In relative membership, the membership value denotes the confidence the event belongs to a particular region. Intuitively, the higher the membership value, the more confident the event belongs to a particular region. Relative membership gives results that are more accurate, therefore we will use this in our prediction model. [82] implemented the key functionality of their proposed temporal feature extraction model in architecture called TClass. It is open source written in java, thus it can use the WEKA APIs.

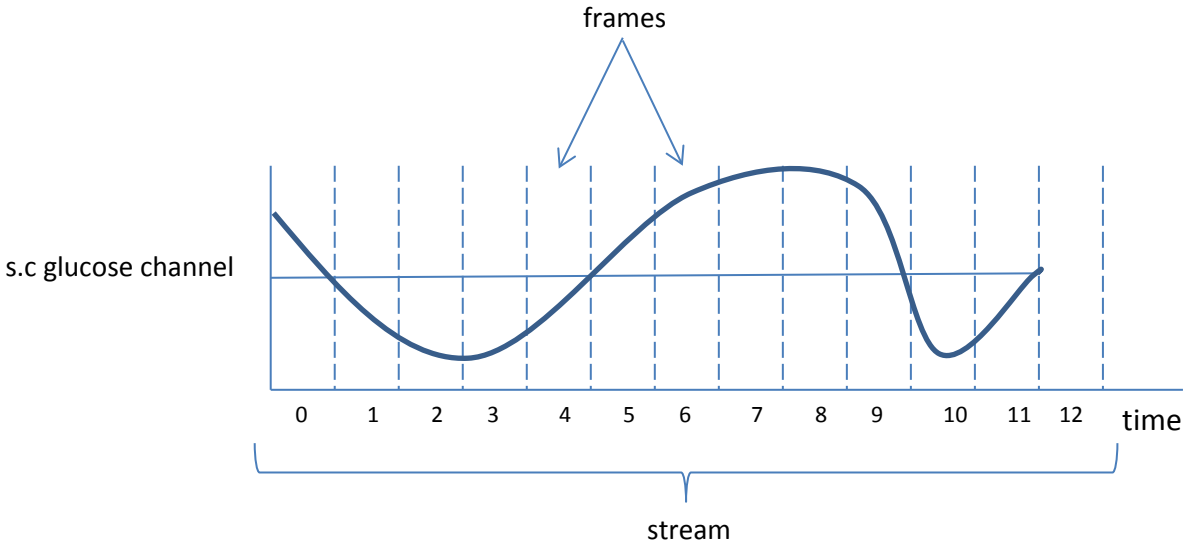

Figure 21. Data Representation of our temporal classification

4.1 Preprocessing phase

Let $\mathbf{X = \{x_t, t=1, \ldots, N\}}$ be the time series of subcutaneous glucose concentrations, and $\mathbf{Y = \{Y_t, t=1, \ldots, N\}}$ be the binary {1 for an event, 0 for a nonevent} hypoglycemic time series attribute. Given that $\mathbf{x_t}$ is the current glucose measurement, our goal is to predict the occurrence of hypoglycemia within the prediction horizon of m steps $\{\mathbf{Y_{t+1}, Y_{t+2} \ldots, Y_{t+m}}\}$. The previous subcutaneous glucose data points (lags)

$\{\mathbf{x_{t-1}}, \mathbf{x_{t-2}}, \ldots, \mathbf{x_{t-p}}\}$ are used for extracting features that are fed into the prediction model. Values of a variable occurring prior to the current observation are called lag values. Each input pattern is formed from a window of fixed length moving along the subcutaneous glucose time series. The event characterization function **E(t)** that addresses our goal is, E(t) = $\mathbf{Y_{t+1}}$ OR $\mathbf{Y_{t+2}}$ **…**OR $\mathbf{Y_{t+m}}$, such that if hypoglycemia event occurred (i.e. $Y_{t+i} = 1$) at any time step, then g(t) will end up being equal to one. Our prediction model will be of the form:

$$\mathbf{g(t)} = \mathbf{F}\left(\mathbf{x_{t-1}}, \mathbf{x_{t-2}}, \ldots, \mathbf{x_{t-p}}\right) \qquad \textbf{(21)}$$

Where $\mathbf{x_{t-1}}, \mathbf{x_{t-2}}, \ldots, \mathbf{x_{t-p}}$ are the lags; p is the window size, and F is the prediction algorithm.

The inputs to the prediction model are the recent history values of subcutaneous glucose measurements: $\mathbf{x_{t-1}}, \mathbf{x_{t-2}}, \ldots, \mathbf{x_{t-p}}$**.** For creating the p lags we used TimeSeriesTranslate, which is a predefined function on WEKA Forecasting package used for creating lag (past values of a variable) or lead variable (future values of a variable). By creating p copies of BG after renaming them as BG_lag-1, BG_lag-2,…BG_lag-p. We have applied TimeSeriesTranslate on each BG_lag-i with an instanceRange = -i, where i = {1, 2, …,p}. In other words, BG_lag-i will have the values of the BG attribute shifted to the past with i step(s). The target attribute is the occurrence or non-occurrence of hypoglycemia within a prediction horizon of thirty minutes. We have created six lead variables of hypoglycemia with {1, 2, …, 6} steps into the future (i.e., duration of 30 minutes). After renaming them as hypo_lead+1, hypo_lead+2, …, hypo_lead+6, we reapplied the TimeSeriesTranslate of WEKA to BG, only this time with +1, +2, +3, +4, +5, and +6 instanceRanges. Regarding the target attribute (Hypo attribute), it is the result of "Oring", i.e., applying "logical OR operator" to the events hypo_lead+1, hypo_lead+2, …, hypo_lead+6. The Hypo attribute ends up having the value "1" if the hypoglycemia event occurred at least once within the prediction horizon. "0" will be the value in case of no hypoglycemia event occurring during the prediction horizon.

Metafeature Implementation:

The last one hour of subcutaneous glucose measurements is used (window size = 1 hr) to predict the hypoglycemia events within a thirty minutes prediction horizon.

We choose the window size (one hour) to be double the prediction horizon (thirty minutes). Thus each instance (stream) is composed of 13 frames (12 lags + 1 the current subcutaneous glucose value), and associated with binary class (occurrence or non-occurrence of hypoglycemia), as shown in Table 11. This Table demonstrates a sample of the preprocessed patient1 file. Channel, in our case, is the subcutaneous glucose measurements. Regarding the features, we have used two aggregate global features and two PEPs. The global features measure some property of each training instance as a whole, rather than looking at the temporal structure. Despite simplicity, practice shows such features can be surprisingly useful [74]. The aggregate global features used in this research are:

Table 11. Sample of patient 1 preprocessed dataset

Stream	Channel (Subcutaneous glucose time series)	Hypo Nominal
1	94, 92, 90, 90, 90, 92, 92, 92, 92, 90, 88, 86, 82	0
2	92, 90, 90, 90, 92, 92, 92, 92, 90, 88, 86, 82, 76	0
3	90, 90, 90, 92, 92, 92, 92, 90, 88, 86, 82, 76, 74	0
4	90, 90, 92, 92, 92, 92, 90, 88, 86, 82, 76, 74, 72	0
5	90, 92, 92, 92, 92, 90, 88, 86, 82, 76, 74, 72, 70	0
6	92, 92, 92, 92, 90, 88, 86, 82, 76, 74, 72, 70, 68	0
7	92, 92, 92, 90, 88, 86, 82, 76, 74, 72, 70, 68, 68	0
8	92, 92, 90, 88, 86, 82, 76, 74, 72, 70, 68, 68, 68	1
9	92, 90, 88, 86, 82, 76, 74, 72, 70, 68, 68, 68, 68	1
10	90, 88, 86, 82, 76, 74, 72, 70, 68, 68, 68, 68, 68	1
11	88, 86, 82, 76, 74, 72, 70, 68, 68, 68, 68, 68, 66	1
12	86, 82, 76, 74, 72, 70, 68, 68, 68, 68, 68, 66, 64	1
13	82, 76, 74, 72, 70, 68, 68, 68, 68, 68, 66, 64, 62	1

Mean

This global feature extractor calculates the mean value of subcutaneous glucose measurement over a given stream. The assumption is the channel is continuous. It

takes one parameter, which is the channel, to calculate its mean. This operation is iterated until all the streams are covered.

Minimum (Min)

This global feature extractor calculates the minimum of subcutaneous glucose measurements for each instance/stream. Just like Mean feature extractor, it takes one parameter, the channel values of a stream to calculate the corresponding minimum. Table 12 demonstrates the global features used for Patient1.

Table 12. Global features of patient 1 for the streams mentioned in Table 11

Stream	Aggregate global features		Hypo
	Min	Mean	
1	82	90.00	0
2	76	88.62	0
3	74	87.23	0
4	72	85.85	0
5	70	84.31	0
6	68	82.62	0
7	68	80.77	0
8	68	78.92	1
9	68	77.08	1
10	68	75.23	1
11	66	73.38	1
12	64	71.54	1
13	62	69.69	1

The parameterized event primitives (PEPs) used in this work are:

1. **Decreasing**

The decreasing Metafeature detects when the subcutaneous glucose stream is decreasing. This PEP requires a continuous channel. The shortest duration signal considered in the decreasing interval is 3 frames. More interestingly, it copes with noise in a signal. Thus given two decreasing intervals separated by one noisy sample, this noisy sample is considered an exception and the interval is considered a decreasing one. It gives a list of tuples as output; each tuple consists of the four parameters mentioned below.

- **midTime:** This is the point that temporally lies in the middle of the subcutaneous glucose decreasing event. It is expressed using a frame number.

- **Average**: It is the average value of the subcutaneous glucose decreasing event. This value will have the same data type as the channel.

- **Gradient:** It represents the rate of change of the stream during the subcutaneous glucose decreasing event. More precisely, the best fit line is tailored to the decreasing interval; then the line's gradient is measured.

- **Duration:** This is the length of subcutaneous glucose decreasing interval. It is expressed in the number of frames.

2. **LocalMin**

The localMin Metafeature detects when the subcutaneous glucose stream has a local minimum, i.e., a point in time when just before and after that point the signal is higher. It produces instantiated feature tuples consisting of the following parameters:

- **Time:** This represents the frame number at which the subcutaneous glucose LocalMin event occurred.

- **Value:** The height of the local minimum subcutaneous glucose measurement.

After computing the aggregate global features and PEPs, they are then integrated together and fed into a machine learning algorithm. One of the crucial advantages of TClass, over the already available ones within temporal algorithms literature, is its capability of integrating the aggregate global features, with the temporal ones. Since all of these processes produce attribute-values vectors, they can be concatenated into a single long attribute value tuple ready for passing to a machine-learning algorithm.

4.2 Model Development

Classification trees [17] are predictive models that map observations about an item to conclusions about the item's target value. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees gained popularity because of their simplicity and easy interpretability, especially in medical machine learning studies where physicians need to understand the model in order to trust it. In this study, two different decision trees, namely J4.8 and REPTree algorithms, are used. In addition, an ensemble learning method, the bagging algorithm, has been implemented.

4.2.1 Predicting hypoglycemia using the two aggregate global features and the two PEPs as input to prediction model

The decision trees were trained for each patient separately. The input features to the prediction model are the two aggregate global features, mean, and min, as well as the two PEPs decreasing and localMin features described in Section 4.1. The target attribute, on the other hand, is Hypo as is also described in Section 4.1. The prediction models used are J4.8, REPTree and Bagging. The performance results and the comparison of the different algorithms are reported in section 5.2.

- **J4.8**

  J48 [17] is the implementation of the C4.5 decision tree in WEKA. The algorithm uses information gain to build the tree, and at each level it calculates the information gained from the different attributes, and finally, it picks the one with the highest value to split at. The process continues recursively until the training data is completely classified [17].

- **REPTree**

  REPTree [17], on the other hand, is a fast decision tree learner that builds a decision or regression tree using information gain or variance reduction and prunes it using reduced-error pruning technique.

- **Bagging**

  Bagging [17] is an ensemble learning method combining multiple models in order to improve classification outcomes in terms of stability and accuracy. It works by dividing the training set into subsets, and for each subset it creates a separate model. The outputs of these models are then combined using voting or averaging techniques. Bagging is generally applied to decision tree models; however, it can be used with any type of models. In this study, we applied it to J4.8 model.

4.2.2 Predicting hypoglycemia using the difference between current BG and BG_lag-12 values in addition to the previously mentioned features

While going through the data of the different patients, we have found the differences between the current subcutaneous glucose measurement and the value before an hour is related to the hypoglycemia event. As a result, we added a new aggregate global feature called difference. Difference is calculated by subtracting the last value of a stream and its first one. Table 13 shows the "difference" global attribute for the streams, defined in Table 11 for patient 1.

Table 13. The addition of difference global feature for patient 1 for the streams of Table 11

Stream	Aggregate global features			Hypo
	Min	Mean	Difference	
1	82	90.00	-12	0
2	76	88.62	-16	0
3	74	87.23	-16	0
4	72	85.85	-18	0
5	70	84.31	-20	0
6	68	82.62	-24	0
7	68	80.77	-24	0
8	68	78.92	-24	1
9	68	77.08	-24	1
10	68	75.23	-22	1
11	66	73.38	-22	1
12	64	71.54	-22	1
13	62	69.69	-20	1

Chapter 5

Evaluation and Results

In this chapter, the results obtained are discussed using the different predictive approaches and then these results are compared between one another. The quality of the prediction models is assessed by mathematical metrics that quantify the error between predicted hypoglycemia events and the actual ones. Specifically, predictions were evaluated with respect to accuracy, specificity, and sensitivity. Accuracy is the proportion of testing set examples that is correctly classified by the model. Sensitivity refers to the proportion of instances with hypoglycemia and is classified as such. On the other hand, specificity refers to the ability of the model to correctly diagnose non-hypoglycemic events as such. These metrics, presented above, are defined next using equations 22 through 24 as follows:

- Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad \textbf{(22)}$$

- Specificity

$$specificity = \frac{TN}{TN + FP} \quad \textbf{(23)}$$

- Sensitivity

$$sensitivity = \frac{TP}{TP + FN} \quad \textbf{(24)}$$

Where, the true positives (TP) and true negatives (TN) are correct classifications. Specifically, TP refers to the number of instances that were predicted as hypoglycemia, when they are as such. Likewise, TN represents number of instances that were predicted as non-hypoglycemia, when they are as such. A false positive (FP) is when the outcome is incorrectly predicted as positive (hypoglycemia), but when it is actually negative (non-hypoglycemia). Contrary, a false negative (FN) is when the

outcome is incorrectly predicted as negative (non-hypoglycemia), but when it is actually positive (hypoglycemia), thus decreasing FN is the focal goal.

Obviously, we aim to achieve the highest sensitivity, as we do not want to miss many hypoglycemic events. However, a higher sensitivity will generally result in a lower specificity, and vice versa. So it is a tradeoff between sensitivity and specificity, where very low specificity will end up in many false alarms (1 – specificity); therefore, this will reduce the reliability of the model to patients.

5.1 Results of the TS-ANN approach

The neural network used, as detailed in Section 3.3, is NARX with subcutaneous glucose, plasma insulin, and Gin as exogenous inputs, trained using 'trainbr' training algorithm, and has a single hidden layer with 20 neurons and 'logsig' as its activation function. The prediction horizon (PH) used is thirty minutes, six steps where each step is five minutes. Table 14 summarizes the Root Mean Squared Error (RMSE) of the different prediction models for the different patients. The prediction window (PW) used is a function of the prediction horizon; namely, PW is twice the prediction horizon.

Table 14. Prediction model performance for the different patients using root mean squared error

	Patient ID					
	1	**3**	**5**	**8**	**9**	**Average**
RMSE	15.27	14.88	9.03	13.58	13.71	13.29

It is noticeable from Table 14, that the performances of predicting future BG values on a 30 minutes prediction horizon is quite accurate. The average RMSE error rate is 13.29. Figure 22 shows the predicted subcutaneous glucose along with the actual one for patient 9 on a prediction horizon of 30 min. The two values are fairly close. This figure illustrates the subcutaneous glucose measurements in two and half days. Figure 23 shows the results of the other patients.

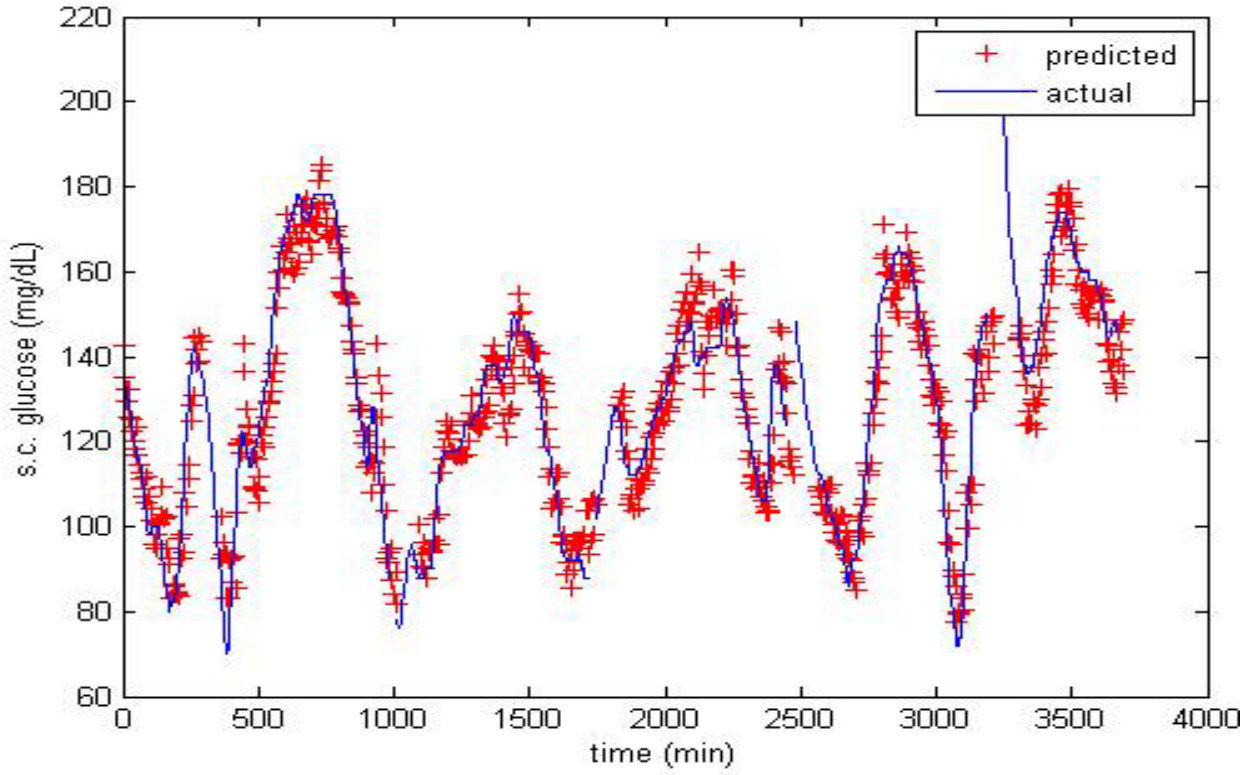

Figure 22. Model evaluation for patient 9, actual subcutaneous glucose measurements vs. predicted on prediction horizon of 30 minutes

5.1.1 Predicting hypoglycemia events

As mentioned in Section 3.3.9, at the end of TS-ANN prediction process two arrays for each patient, Predicted_Hypo and Actual_Hypo, are generated. A simple matlab code was written to calculate accuracy, specificity, and sensitivity for the Predicted_Hypo versus Actual_Hypo using Equations 22 through 24. The resulting performance is shown in Table 15. The performance of the neural network based on the subcutaneous glucose prediction for 30 minutes of the prediction horizon is quite accurate; see Table 14; however, the predictors are not accurate in forecasting hypoglycemia as it is observable from the Sensitivity column of Table 15. In this study, our focus is not to develop a model that is accurate in the euglycemic range, instead we need to develop a predictor that is precise in forecasting hypoglycemia. For that purpose, we used a cost function to penalize out-of-range glucose deviations. By default, in all predictors the cost of all errors is equal to one. In the case of cost sensitive classifier, a different cost is used for different errors; tthus for a predictor striving for lower error rate, avoiding the higher cost error is prioritized.

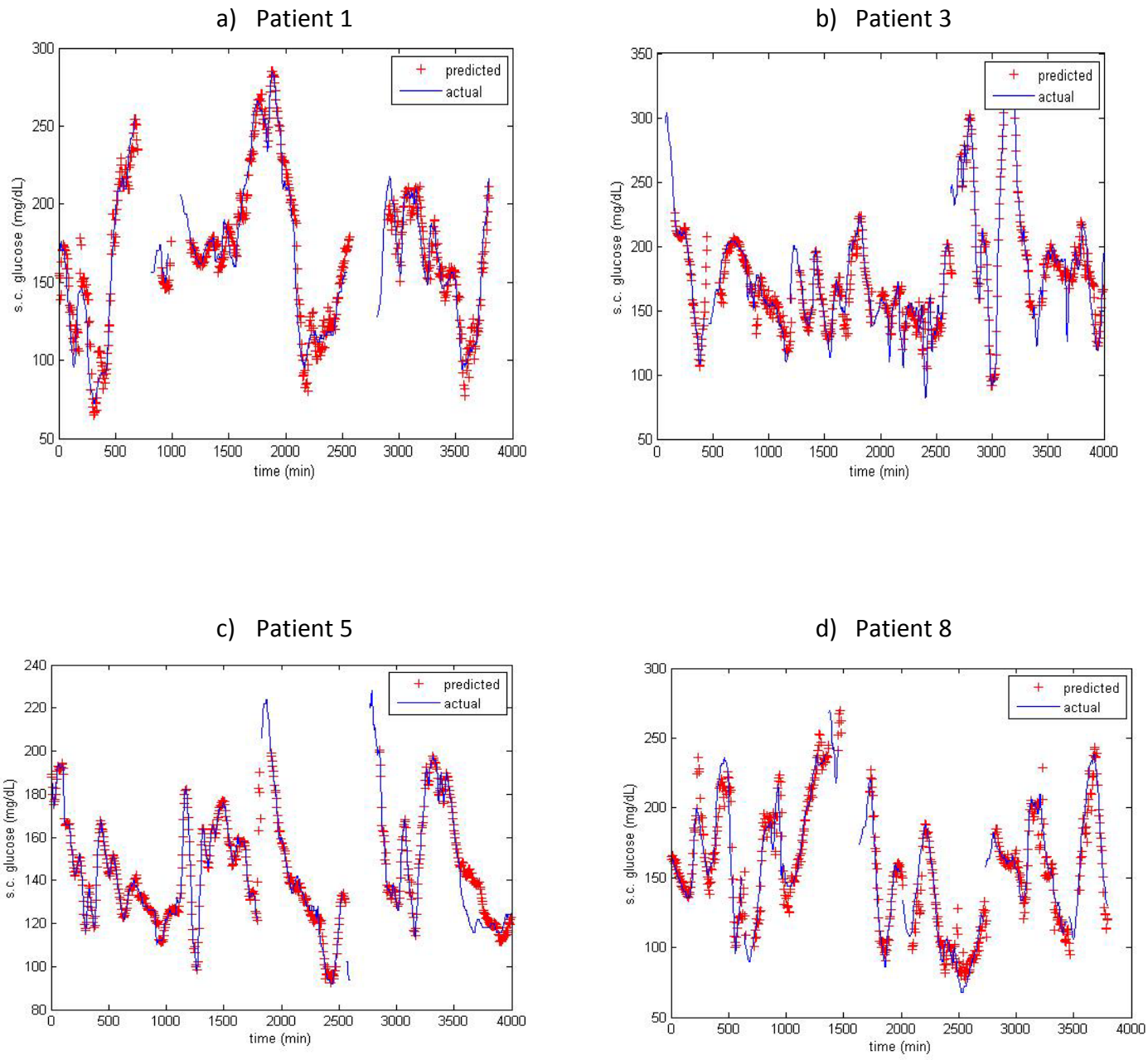

Figure 23 (a-d) illustrates the predicted vs actual subcutaneous glucose measurements, resulting from TS-ANN for the different patients

The following study utilizes the cost function of neural network, where we create error weights that prioritize avoiding mispredicting of the subcutaneous glucose measurements that is less than 60 mg/dl more than other misclassifications. Table 16

summarizes the performance results of the cost sensitive neural networks. It is noticeable from the table that the sensitivity for all the patients has increased in comparison to Table 15.

Table 15. Performance of the prediction models for the different patients using TS-ANN approach

Patient ID	Accuracy	Specificity	Sensitivity
1	97.86	0.99	0.85
3	95.96	0.98	0.59
5	98.91	0.99	0.63
8	99.64	0.99	0.66
9	97.19	0.99	0.37
Average	**97.91**	**0.99**	**0.62**

Table 16. Performance of the cost sensitive prediction model for the different patients using TS-ANN approach

Patient ID	Accuracy	Specificity	Sensitivity
1	97.59	0.98	0.93
3	95.87	0.97	0.81
5	98.78	0.99	0.80
8	99.37	0.99	0.76
9	96.24	0.97	0.69
Average	**97.57**	**0.98**	**0.80**

5.2 Results of Tree Based Temporal Classifications (TBTC) approach

We evaluated the models using 10-fold cross validation in order to obtain the sensitivity, specificity, and accuracy of each model. 10-fold cross validation divides data randomly into 10 parts by which the class is represented in approximately the same proportions as in the full dataset. Each part is then held out in turn and the learning scheme is trained on the remaining nine-tenths; from there the error rate is calculated on the holdout set. Thus, the learning procedure is executed for a total of 10 times on different training sets. Finally, the 10 error estimates are averaged to yield an overall error estimate.

Table 17 summarizes the results of the TBTC approach. As described in Chapter 4, the proposed model uses two aggregate global features and two PEPs as input features to the prediction models. These features are extracted from 12 lags. Three different prediction algorithms were implemented: J4.8, REPTree, and Bagging. Predictions were evaluated using a 10-fold cross validation with respect to accuracy, specificity, and sensitivity. It is observable from Table 17 that the performance of the different classifiers in terms of sensitivity, which is our main focus, is relatively poor. The reason for poor performance is the rarity of hypoglycemia events as shown in Table 6 where the prevalence of hypoglycemia events range between 0.46% and 15%. Moreover, one notices the higher the prevalence of hypoglycemia in the patient's dataset, the higher the sensitivity is. The lowest sensitivity values were obtained from patient 10 since she has the minimum prevalence of hypoglycemia. Contrary, patient 1’s classification models have the highest sensitivity values, as this patient has the highest prevalence of hypoglycemia. The explanation of this phenomenon is that the predictive model is better trained in the presence of many hypoglycemic events, thus it tends to be more sensitive to classifying them. Another conclusion that could be drawn from Table 17 is that the Bagging algorithm is more sensitive in detecting hypoglycemia, in the participating patients, in comparison to the REPTree and J4.8 algorithms, where the average sensitivity values for the different patients are 0.717, 0.677, and 0.644 for Bagging, REPTree, and J4.8 respectively.

5.2.1 Subsampling the patients' datasets

As mentioned earlier the prevalence of hypoglycemia in the patients' datasets is low. In order to resolve this issue, we did subsampling for the datasets. We noted the best performance was obtained from patient 1 dataset; it had a 15% prevalence of

hypoglycemia. Thus for the remaining patients, we randomly selected some non-hypoglycemic instances and deleted them, such that the resulting prevalence of hypoglycemia for each patient's dataset is 15%. In this instance, we utilized the Bagging algorithm since it outperformed other participating algorithms. In regard to metafeatures, the two aggregate global features, mean and min, and the two PEPS decrease and the localMin was used. Table 18 summarizes these performance results.

5.2.2 Adding aggregate global feature; difference to the subsampled datasets

The difference feature is discussed in the previous section 4.2.2. Here we show the performance results obtained from the Bagging algorithm after adding the "difference" global feature to the subsampled patients' datasets. It is noticeable from Table 19 that the sensitivity for all the patients increased when adding the difference feature, except for patient 10 where the addition of the new attribute resulted in a slight drop of sensitivity. Figure 24 shows the resulting decision tree of patient 3 using the Bagging algorithm as the prediction model, regarding the inputs to the model, three aggregate global features; difference, mean and min, and the two PEPS; decrease and localMin are used.

Table 17. Performance of the prediction models for the different patients

Patient ID (Prevalence of hypoglycemia)	Algorithm	Sensitivity	Specificity	Accuracy
1 (15%)	J4.8	0.954	0.995	99.00 %
	REPTree	0.930	0.996	98.8 %
	Bagging	0.958	0.996	99.11 %
3 (5.39)	J4.8	0.697	0.997	98.20 %
	REPTree	0.819	0.996	98.70 %
	Bagging	0.861	0.99	97.80 %
5 (2.46%)	J4.8	0.60	0.99	99.11 %
	REPTree	0.535	0.999	98.82 %
	Bagging	0.584	0.998	98.87 %
8 (1.06%)	J4.8	0.729	0.999	99.69 %
	REPTree	0.730	0.998	99.55 %
	Bagging	0.702	0.999	99.69 %
9 (1.56%)	J4.8	0.403	0.999	98.18 %
	REPTree	0.582	0.998	98.60 %
	Bagging	0.716	0.999	99.11 %
10 (0.46%)	J4.8	0.481	1	99.70 %
	REPTree	0.468	0.999	99.74 %
	Bagging	0.481	0.999	99.74 %

Table 18. Performance of the bagging prediction models for the different patients after subsampling

Patient ID (Prevalence of hypoglycemia)	Sensitivity	Specificity	Accuracy
1 (15%)	0.958	0.996	99.11%
3 (15%)	0.936	0.981	97.49%
5 (15%)	0.858	0.987	97.03%
8 (15%)	0.973	0.996	99.29%
9 (15%)	0.836	0.98	96.10%
10 (15%)	0.9	0.989	97.68%
Average	**0.910**	**0.988**	**97.78%**

Table 19. Comparison of the performance of bagging classifiers for the subsampled dataset after adding "Difference" global feature

Patient ID	Sensitivity	Specificity	Accuracy
1	0.951	0.987	98.26%
3	0.959	0.983	98.03%
5	0.937	0.988	98.10%
8	0.973	0.996	99.29%
9	0.955	0.987	98.25%
10	0.861	0.987	97.02%
Average	**0.939**	**0.988**	**98.16%**

V-mean <= 66.14286

| V-mean <= 62: Hypo (126.0/12.0)

| V-mean > 62

| | V-diff <= -14

| | | V-dec_4 <= 0.069351

| | | | V-dec_1 <= 0.018202: Hypo (7.0/1.0)

| | | | V-dec_1 > 0.018202: Non-Hypo (2.0)

| | | V-dec_4 > 0.069351: Hypo (8.0)

| | V-diff > -14: Non-Hypo (15.0)

V-mean > 66.14286

| V-mean <= 74.714287

| | V-diff <= -20

| | | V-dec_5 <= 0.007492: Hypo (22.0/4.0)

| | | V-dec_5 > 0.007492: Non-Hypo (2.0)

| | V-diff > -20

| | | V-lmin_1 <= 0: Non-Hypo (51.0)

| | | V-lmin_1 > 0

| | | | V-mean <= 72.714287: Non-Hypo (11.0/1.0)

| | | | V-mean > 72.714287: Hypo (6.0/1.0)

| V-mean > 74.714287

| | V-dec_3 <= 0.278888: Non-Hypo (961.0/6.0)

| | V-dec_3 > 0.278888

| | | V-diff <= -76

| | | | V-mean <= 110.714287: Hypo (7.0)

| | | | V-mean > 110.714287

| | | | | V-diff <= -104: Hypo (3.0)

| | | | | V-diff > -104: Non-Hypo (4.0)

| | | V-diff > -76

| | | | V-dec_3 <= 0.305605: Hypo (5.0/1.0)

| | | | V-dec_3 > 0.305605: Non-Hypo (88.0)

Figure 24. Patient 3 prediction model

5.3 Comparison of TS-ANN and TBTC approach results

In this section, we compare the results obtained using the two approaches; namely, the time sensitive artificial neural network and the tree based temporal classification approach. The comparison criteria are sensitivity, specificity, and accuracy. The specificity and accuracy for both approaches were above 95% and the differences negligible. This is, as mentioned earlier, due to the small number of hypoglycemia events, i.e., any average classifier for our dataset will result in accurate values of accuracy and specificity.

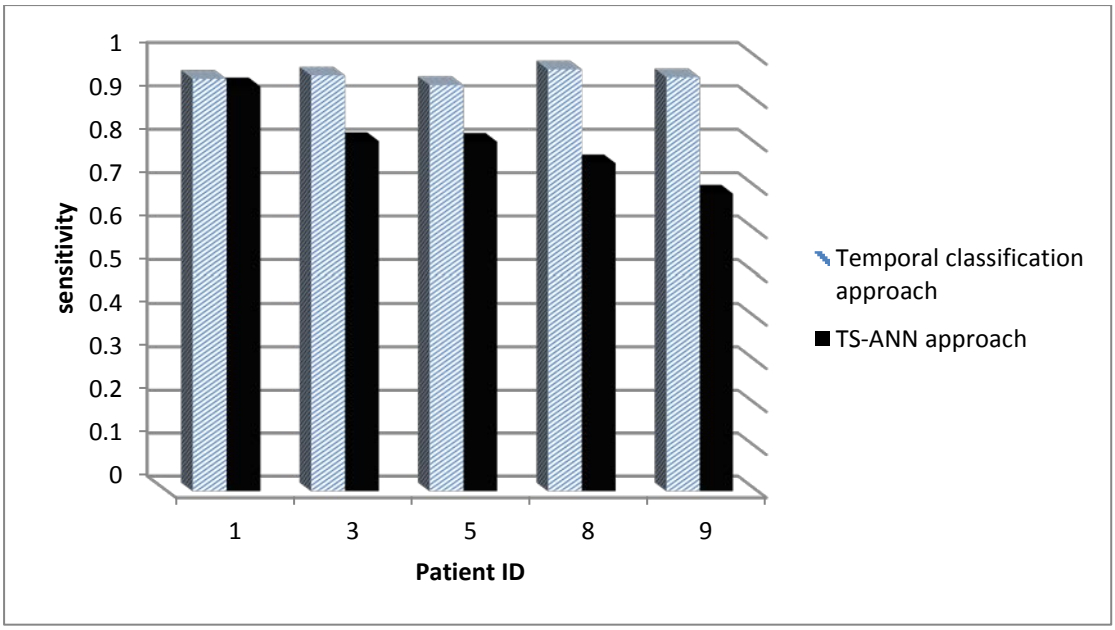

Figure 25. comparison of TS-ANN and TBTC approaches in terms of sensitivity

Figure 25 compares the two approaches in terms of sensitivity. It is observable from this graph that the temporal classification approach provided better sensitivity values than the TS-ANN one, across all participating patients. It is interesting to note the research trends in the literature went to the TS-ANN approach, i.e., using time sensitive models to predict glucose measurements, yet we identified a simpler and more accurate approach - the temporal classification approach.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Lately, medical machine learning has gained in interest by the scientific and research communities. Diabetes is considered as the world's fastest-growing chronic disease. It needs continuous self-management and control to maintain blood glucose level within the normal range, in order to prevent complications and prevent hypoglycemia events. Hypoglycemia is a condition that occurs when blood glucose is too low. The occurrence of hypoglycemia may result in seizures, unconsciousness, and possibly permanent brain damage or death.

This thesis aimed at predicting hypoglycemia within a short-term interval. Specifically, machine learning techniques were investigated and evaluated for predicting type I diabetes mellitus events using a continuous glucose monitoring sensor for measuring the glucose levels belonging to a population of 10 diabetic patients. An individual model was designed for each patient using the two approaches, namely, Time-Sensitive Artificial Neural Network (TS-ANN) and temporal classification (TBTC). The inputs to the TS-ANN based model are:

- Recent history of subcutaneous glucose measurements.
- Plasma insulin after subcutaneous injection of insulin.
- Plasma glucose rate of appearance after carbohydrate absorption.

The prediction model uses artificial neural network, and the output is the subcutaneous glucose measurements in a prediction horizon of 30 minutes. The final step of this approach uses the predicted subcutaneous glucose measurements in determining the occurrence of hypoglycemic and non-hypoglycemic events. The model predicts hypoglycemia events within a prediction horizon of thirty minutes and with an average sensitivity of 80.19%, an average specificity of 98.2, and an average accuracy of 97.56%.

The inputs to the temporal classification approach are:

- Three aggregate global features; mean, min, and difference.
- Two parameterized event primitives; decreasing, and local minimum.

These features are extracted from the recent history of subcutaneous glucose measurement and the output is hypoglycemia events within thirty minutes prediction horizon. The model was then able to predict hypoglycemia events within a prediction horizon of thirty minutes accurately (average sensitivity= 93.93%, average specificity= 98.8, and average accuracy= 98.15%).

The tree based temporal classification approach is a simpler approach, yet it outperformed the Time-Sensitive Artificial Neural Network one.

6.2 Future work

For future work, more input features can be used, e.g. exercise, heart rate, and metabolism rate. In addition, drawn blood samples of plasma insulin are needed to compare with the simulated values. As far as the meal modeling is concerned, we suggest using the patient's diary and nutrition tables rather than relying on the patient's estimation of the carbohydrate quantity in a meal.

Moreover, we recommend the proposed models to be tested on a larger dataset.

References

[1] WHO. "Diabetes Programme", Internet: http://www.who.int/diabetes/en/> [Sep. 27, 2014].

[2] BOUNDLESS. "Insulin and Glucagon." Internet: https://www.boundless.com/biology/hormones/hormone-pathways-and-regulation/insulin-and-glucagon/ , [Jan. 22, 2013]

[3] Gary W. Gibbons and Palma M. Shaw, 'Diabetic vascular disease: characteristics of vascular disease unique to the diabetic patient.' *Seminars in vascular surgery*, 25.2 (2012), 89-92.

[4] American Diabetes Association. "Diabetes Statistics." Internet: http://www.diabetes.org/diabetes-basics/diabetes-statistics/, March 6, 2013 [Apr. 19, 2014]**.**

[5] Parvez Hossain, Bisher Kawar, and Meguid El Nahas. "Obesity and diabetes in the developing world—a growing challenge." *New England Journal of Medicine*, vol. 356, no. 3, pp. 213-215, 2007.

[6] Mohammed El-Sadig and Enyioma Obineche Fatma Al-Maskari. "Prevalence and determinants of microalbuminuria among diabetic patients in the United Arab Emirates." Internet: http://www.emaratalyoum.com/local-section/health/2011-12-08-1.443589 [Apr. 19, 2014]

[7] American Diabetes Association. "Gestational Diabetes." Internet: http://www.diabetes.org/diabetes-basics/gestational/ [Apr. 19, 2014]

[8] Ronald Aubert, William Herman, Janice Waters, William Moore, David Sutton, Bercedis Peterson, Cathy Bailey, and Jeffrey P. Koplan, "Nurse Case Management To Improve Glycemic Control in Diabetic Patients in a Health Maintenance Organization," in *Annals of Internal Medicine*, vol. 129, no. 8, pp. 605-612, 1998.

[9] Ying Zhang. ''Predicting occurrences of acute hypoglycemia during insulin therapy in the intensive care unit', *Engineering in Medicine and Biology Society* (city: publisher, 2008), 3297-3300.

[10] DCCT Research Group, "The Effect of Intensive Treatment of Diabetes on the Development and Progression of Long-Term Complications in Insulin-Dependent Diabetes Mellitus: Diabetes Control and Complications Trial." *Journal of Pediatrics*, vol. 125, no. 2, 1993.

[11] Eyal Dassau, Fraser Cameron, Hyunjin Lee, B. Wayne Bequette, Howard Zisser, Lois Jovanovic, H. Peter Chase, Darrell Wilson, Bruce A. Buckingham, and Francis Doyle, "Real-Time Hypoglycemia Prediction Suite Using Continuous Glucos Monitoring," *Diabetes Care*, vol. 33, no. 6, pp. 1249-1254, 2010.

[12] Richard Weinstein, Sherwyn Schwartz, Ronald Brazg, Jolyon Bugler, Thomas Peyser, and Geoffrey McGarraugh. "Accuracy of the 5-day FreeStyle Navigator continuous glucose monitoring system comparison with frequent laboratory reference measurements." *Diabetes Care*, vol. 30, no. 5, pp. 1125-1130, 2007.

[13] James Brauke, Victoria Carr-Brendel, Laura Martinson, Paul Neale, and Mark Tapsak. "Optimized sensor geometry for an implantable glucose sensor." U.S. Patent no. 7,134,999. 14 Nov. 2006.

[14] John Mastrototaro, John Shin, Alan Marcus, and Giri Sulur. "The accuracy and efficacy of real-time continuous glucose monitoring sensor in patients with type 1 diabetes." *Diabetes technology & therapeutics*, vol. 10, no. 5, pp. 385-390, 2008.

[15] Eda Cengiz, and William Tamborlane. "A tale of two compartments: interstitial versus blood glucose monitoring." *Diabetes technology & therapeutics*, vol. 11, no. S1, 2009.

[16] Eleni Georga, Vasilios Protopappas, and Dimitrios Fotiadis, "Glucose Prediction in Type 1 and Type 2 Diabetic Patients Using Data Driven Techniques," in *Knowledge-Oriented Applications in Data Mining*, 2011.

[17] Ian Witten, and Eibe Frank, Data Mining Practical Machine Learning Tools and Techniques, Third Edition, Elsevier Inc, 2011.

[18] Nada Lavrac, "Selected techniques for data mining in medicine." *Artificial Intellegence in Medicine*, vol. 16, no. 1, pp. 3-23, 1999.

[19] Indranil Bose and Radha Mahapatra, "Business Data Mining - a machine learning perspective." *Information and management*, vol. 39, no. 3, pp. 211-225, 2001.

[20] Michael Shaw, Chandrasekar Subramaniam, Gek Woo Tan, and Michael Welge. "Knowledge management and data mining for marketing." *Decision Support Systems*, vol. 31, no. 1, pp. 127-137, 2001.

[21] Cristobal Romero, and Sebastian Ventura. "Educational data mining: A survey from 1995 to 2005." *Expert Systems with Applications*, vol. 33, no. 1, pp. 135-146, 2007.

[22] Ruben Canlas Jr., 'Data mining in healthcare: Current Applications and Issues' (MS thesis, Department of Information Technology, Carnegie Mellon University, Australia, 2009).

[23] Asma Al Jarullah, "Decision tree discovery for the diagnosis of type II diabetes," in *Innovations in Information Technology (IIT), 2011 International Conference on* IEEE, 2011, pp. 303-307.

[24] UCI Machine Learning Repository data sets. "pima-indians-diabetes." Internet: http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/?C=D;O=A [Apr. 19, 2014]

[25] Huy Pham, and Evangelos Triantaphyllou. "Prediction of diabetes by employing a new data mining approach which balances fitting and generalization." *Computer and Information Science*, pp. 11-26, 2008.

[26] Huy Pham, and Evangelos Triantaphyllou. "The impact of overfitting and overgeneralization on the classification accuracy in data mining." *Soft Computing for Knowledge Discovery and Data Mining*, pp. 391-431, 2008.

[27] Joachim Diederich. Rule Extraction from Support Vector Machines, Heidelberg: Springer, 2008.

[28] Nahla Barakat, Andrew Bradley, and Mohamed Barakat. "Intelligible support vector machines for diagnosis of diabetes mellitus." *Information Technology in Biomedicine, IEEE Transactions*, vol. 14, no. 4, pp. 1114-1120, 2010.

[29] Nahla Barakat, and Andrew Bradley. "Rule extraction from support vector machines: a sequential covering approach." *Knowledge and Data Engineering, IEEE Transactions,* vol. 19, no. 6, pp. 729-741, 2007.

[30] Nahla Barakat, and Joachim Diederich. "Eclectic rule-extraction from support vector machines." *International Journal of Computational Intelligence,* vol. 2, no.1, pp. 59-62, 2005.

[31] M. Asfour, et. al., "High prevalence of diabetes mellitus and impaired glucose tolerance in the Sultanate of Oman: Results of the 1991 national survey," *Diabetic Medicine*, vol. 12, pp. 1122–1125, 1995.

[32] B. Yegnanarayana, *Artificial Neural Networks*, Prentice-Hall of India Private limited, New Delhi, 2004.

[33] Manaswini Pradhan, and Dr. Ranjit Kumar Sahu, "Predict the onset of diabetes disease using Artificial Neural Network (ANN)," *International Journal of Computer Science & Emerging Technologies*, vol. 2, no. 2, pp. 303-311, 2011.

[34] Goldberg, *Genetic Algorithms*, Pearson Education, 2013.

[35] Hasan Temurtas, Nejat Yumusak, and Feyzullah Temurtas."A comparative study on diabetes disease diagnosis using neural networks." *Expert Systems with applications*, vol. 36, no. 4, pp. 8610-8615, 2009.

[36] Hao Yu, and B. Wilamowski. "Levenberg-marquardt training." The Industrial Electronics Handbook, 2011.

[37] Ron Kohavi. "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *international Joint Conferences on Artificial Intelligence (IJCAI)*, vol. 14, no. 2, 1995, pp. 1137-1145.

[38] Humar Kahramanli, and Novruz Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert Systems with Applications*, vol. 35, no.1, pp. 82-89, 2008.

[39] Leonid Reznic, Fuzzy Controllers Handbook: How to Design Them, How They Work, Oxford: Newnes, 1997.

[40] Carmen Pérez-Gandía, et al. "Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring." *Diabetes technology and therapeutics*, vol. 12, no.1, pp. 81-88, 2010.

[41] Riccardo Bellazzi, and Ameen Abu-Hanna., "Data Mining Technologies for Blood Glucose and Diabetes Management," *Journal of Diabetes Science and Technology*, vol. 3, no. 3, pp. 603-612, May 2009.

[42] Masaki Yamaguchi, C. Kaseda, Katsuya Yamazaki, and Masashi Kobayashi, "Prediction of blood glucose level of type 1 diabetics using response surface methodology and data mining." *Medical and Biological Engineering and Computing*, vol. 44, no. 6, pp. 451-457, June 2006.

[43] Eleni Georga, Vasilios Protopappas, and Dimitrios Fotiadis, "Glucose Prediction in Type 1 and Type 2 Diabetic Patients Using Data Driven Techniques." *Knowledge-Oriented Applications in Data Mining*, 2011.

[44] Wikipedia, "Mathematical Model". Internet: http://en.wikipedia.org/wiki/Mathematical_model [Apr. 19, 2014]

[45] Masaki Yamaguchi, S. Kanbe, Karin Wårdell, Katsuya Yamazaki, Masashi Kobayashi, Nobuaki Honda, Hiroaki Tsutsui, and Chosei Kaseda. "Trend estimation of blood glucose level fluctuations based on data mining," in *The 7th world multiconference on systemics, cybernetics and informatics*, 2003, pp. 86-91.

[46] Meri Raffetto ,"Glycemic Index Diet." Internet: http://www.dummies.com/how-to/content/how-to-measure-your-metabolic-rate.html [Apr. 19, 2014]

[47] E. D. Lehmann, and T. Deutsch. "A physiological model of glucose-insulin interaction in type 1 diabetes mellitus." Journal of biomedical engineering, vol. 14, no. 3, pp. 235-242, 1992.

[48] Cristina Tarin, et al. "Comprehensive pharmacokinetic model of insulin glargine and other insulin formulations." *Biomedical Engineering, IEEE Transactions*, vol. 52, no .12, pp. 1994‑2005, 2005.

[49] Alexandros Pantelopoulos, and Nikolaos Bourbakis. "A survey on wearable sensor-based systems for health monitoring and prognosis." *Systems, Man, and*

Cybernetics, Part C: Applications and Reviews, IEEE Transactions, vol. 40, no. 1, pp. 1-12, 2010.

[50] Carmen Pérez-Gandía, A. Facchinetti, G. Sparacino, C. Cobelli, E. Gómez, M. Rigla, A. De Leiva, and M. Hernando. "Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring." *Diabetes technology and therapeutics*, vol. 12, no.1, pp. 81-88, 2010.

[51] Hao Yu, and B. Wilamowski. "Levenberg-marquardt training." The Industrial Electronics Handbook, 2011.

[52] Pentaho. "Time series analysis and forecasting." Internet: http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka [Apr. 19, 2014]

[53] Marzia Cescon, "Linear Modeling and Prediction in Diabetes Physiology," M.A. thesis, Lund University, Sweden, 2011.

[54] Jae-Eung Lee, *Linear Multi-stage identification of ARMAX Processes with stochastic structural dynamics applications*. University of Michigan, 1991.

[55] Dalla Man, Michael Camilleri, and Claudio Cobelli. "A system model of oral glucose absorption: validation on gold standard data." *Biomedical Engineering, IEEE Transactions*, vol. 53, no. 12, pp. 2472-2478, 2006.

[56] Bruce Buckingham, H. Peter Chase, Eyal Dassau, Erin Cobry, Paula Clinton, Victoria Gage, Kimberly Caswell. "Prevention of nocturnal hypoglycemia using predictive alarm algorithms and insulin pump suspension." *Diabetes Care*, vol. 33, no.5, pp. 1013-1017, 2010.

[57] Diabetes Research in Children Network (DirecNet) Study Group. "Youth and parent satisfaction with clinical use of the GlucoWatch G2 Biographer in the management of pediatric type 1 diabetes." *Diabetes care*, vol. 28, no.8, pp. 1929-1935, 2005.

[58] Meriyan Eren-Oruklu, Ali Cinar, and Lauretta Quinn. "Alarms for Continuous Glucose Monitors: Hypoglycemia Prediction with Subject Specific Recursive Time

Series Models." *Journal of diabetes science and technology*, vol. 4, no.1, pp. 25-33, 2010.

[59] "Diabetes Research in Children Network (DirecNet)," Internet: http://direcnet.jaeb.org/Studies.aspx [Apr. 19, 2014]

[60] Ying Zhang. "Predicting occurrences of acute hypoglycemia during insulin therapy in the intensive care unit," *Engineering in Medicine and Biology Society, 30th Annual International Conference of the IEEE*, pp. 3297-3300, 2008.

[61] "MIMIC database II." Internet: http://mimic.physionet.org/database.html [Apr. 19, 2014]

[62] "Data Mining Tools See5." Internet: http://www.rulequest.com/see5-info.html [Apr. 19, 2014]

[63] "WEKA Data Mining Software in Java." Internet: http://www.cs.waikato.ac.nz/ml/weka/ [Apr. 19, 2014]

[64] MathWorks, "MATLAB the language of technical computing." Internet: "http://www.mathworks.com/products/matlab/ [Apr. 19, 2014]

[65] WEKA, "ARFF." Internet: http://weka.wikispaces.com/ARFF [Apr. 19, 2014]

[66] Wikipedia, http://en.wikipedia.org/wiki/Comma-separated_values [Apr. 19, 2014]

[67] "American Diabetes Association." Internet: http://www.diabetes.org/living-with-diabetes/treatment-and-care/blood-glucose-control/checking-your-blood-glucose.html [Apr. 19, 2014]

[68] "Basal and Bolus Insulin." Internet: http://type1diabetes.about.com/od/insulinandmedications/p/Basal-And-Bolus-Insulin.htm [Apr. 19, 2014]

[69] E. D. Lehmann, and T. Deutsch. "A physiological model of glucose-insulin interaction in type 1 diabetes mellitus." *Journal of biomedical engineering*, vol. 14, no. 3, pp. 235-242, 1992.

[70] "Periodic Table." Internet: http://www.chemicool.com/ [Apr. 19, 2014]

[71] Jaan Kiusalaas. "Numerical methods in engineering with MATLAB®", Cambridge university press, 2010.

[72] "Insulin." Internet: http://en.wikipedia.org/wiki/Insulin [Apr. 19, 2014]

[73] S. Shimoda, K Nishida, M Sakakida, Y Konno, K Ichinose, M Uehara, T Nowak, M Shichiri, "Closed-loop subcutaneous insulin infusion algorithm with a short acting insulin analog for long-term clinical application of a wearable artificial endocrine pancreas," Frontiers of medical and biological engineering: the international *journal of the Japan Society of Medical Electronics and Biological Engineering*, vol. 8, no. 3, pp. 197-211, 1997.

[74] P. Wach, Z. Trajanoski, P. Kotanko, and F. Skrabal, Numerical approximation of mathematical model for absorption of subcutaneously injected insulin, *Medical and Biological Engineering and Computing*, vol. 33, no. 1, pp. 18-23, 1995.

[75] Eric Mosekilde, Klaus Skovbo Jensen, Christian Binder, Stig Pramming, and Birger Thorsteinsson, "Modeling absorption kinetics of subcutaneous injected soluble insulin," *Journal of pharmacokinetics and biopharmaceutics*, vol. 17, no.1, pp. 67-87, 1989.

[76] Martin Hagan, Howard Demuth, and Mark Beale, *Neural network design*, Boston, London: Pws Pub., 1996.

[78] Patricia Melin, and Janusz Kacprzyk, Bio-inspired hybrid intelligent systems for image analysis and pattern recognition. vol. 256. Springer, 2009.

[79] Chris De Villiers, Biological Aspects that Inspired the Development of Artificial Neural Networks, 2012.

[80] Neural Network toolbox documentation
http://www.mathworks.com/help/nnet/modeling-and-prediction-with-narx-and-time-delay-networks.html [Apr. 19, 2014]

[81] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[82] Mohammad Waleed Kadous, "Learning Comprehensible Descriptions of Multivariate Time Series," in *International* Conference *on Machine Learning*, 1999, pp. 454-463.

[83] Müller, Meinard. "Dynamic time warping." *Information retrieval for music and motion*, pp. 69-84, 2007.

[84] Nada Lavrac, and Saso Dzeroski. "Inductive logic programming," in *Workshop on Logic Programming*, 1994, pp. 146-160.

Vita

Khouloud Abdel Aziz Safi Eljil was born on June 26, 1988, in Jeddah, Saudi Arabia. After completing her high school in 2006, she joined Computer Science program in Ajman University in the UAE from which she was graduated in 2010. Ms. Khouloud received a graduate teaching assistantship from American University of Sharjah to join the Computer Engineering master program.