University of Coimbra

Master in Informatics Engineering
2014-2015

SMITH - Smart MonITor Health System

DEI Advisor:
Bernardete Ribeiro

Author:
Daniel Frutuoso

EyeSee Advisors:
André Pimentel
João Redol

Final Internship Report

Coimbra, 06th July 2015

University of Coimbra

Master in Informatics Engineering
2014-2015

SMITH - Smart MonITor Health System

DEI Advisor:
Bernardete Ribeiro

Author:
Daniel Frutuoso

EyeSee Advisors:
André Pimentel
João Redol

Panel Membership

President and Main Examiner
Alberto Cardoso

Examiner
Tiago Cruz

Supervisor
Bernardete Ribeiro

Abstract

Diabetes is a huge health problem that is affecting more and more people over the time. When it comes to diagnosing such disease in people, doctors make their diagnoses based on some proper tests and may not take into consideration other factors that are related to the disease. The creation of tools that can analyse information about the current health status of patients can support doctors by providing more information for the diagnosis.

Since there is still no cure for this disease, a person who has been diagnosed with diabetes has to control his blood sugar level between some thresholds. This is extremely important as non-controlled level of glucose can lead the subject to severe health complications and compromise his lifestyle. Tools that forecast the subject's glucose level within a prediction horizon may let the individual take preemptive actions to avoid crossing the normal thresholds.

This thesis aims to investigate machine learning methods for such problems. These types of methods are already being used in the medicine and will allow us to come up with computational models that offer more relevant data to support the medical team when it comes to diagnosing diabetes in people and avoid thresholds overpassing when patients are controlling their glucose level.

Both these problems represent highly challenging tasks. For the diabetes diagnosis problem, we built several models, tuned and tested them using the PIMA dataset. A key contribution of this work is the diverse methods introduced, analysed and tested to handle missing values present in the dataset. The method of substituting the missing values by the mean of the features considering the class they belong to along with Random Forest yielded the best results with an accuracy of 87.66%. Regarding the glucose level predictions we also created various models based on real patient datasets provided by Associação Protectora dos Diabéticos de Portugal(APDP), which are an important asset. Besides, two prediction methods namely direct and iterative prediction methods were investigated and tested. From the computational experiments, the Linear Regression with direct prediction method is the most advantageous combination resulting on an RMSE average of 14.25 mg/dL and 23.46 mg/dL for 30 and 60 minutes ahead prediction.

Both these domains represent highly challenging tasks and our methods demonstrate that we can attain excellent performance on these tasks. From an application standpoint, there remains many challenging problems in both Diabetes Diagnosis and Prediction of Glucose. Indeed, in the advent of the Internet of Things by combining many sensors available with our methods, we will come to reach a Smart Monitor Health System able to prevent, diagnosis, treatment and after care for the society in general.

Keywords: Diabetes detection; Glucose level prediction; Machine learning

Resumo

A Diabetes é um grande problema de saúde que está a afectar cada vez mais gente ao longo do tempo. No que toca ao seu diagnóstico, os médicos fazem-no baseado em testes específicos e poderão não ter em conta outros fatores que estão relacionados com a doença. A criação de ferramentas que analisem a informação acerca do estado de saúde dos pacientes pode ajudar os médicos a elaborar o diagnóstico com base em mais informação.

Uma vez que, ainda não existe cura para esta doença, a pessoa a quem tenha sido diagnosticada a diabetes necessita de controlar o seu nível de glucose entre um certo intervalo. Isto é de uma importância extrema dado que se os níveis de glucose não estiverem controlados, o indivíduo pode agravar a sua saúde e comprometer o seu estilo de vida. Ferramentas que sejam capazes de prever o nível de glucose dos portadores desta doença a longo prazo poderão permitir que os pacientes tomem medidas preventivas afim de evitar ultrapassar os níveis normais.

Este trabalho tem por objectivo investigar métodos por aprendizagem por computador que lidem com os problemas anteriormente descritos. Este tipo de métodos já são usados na medicina e permitirão-nos criar modelos computacionais de apoio à decisão no que toca ao diagnóstico da diabetes. Para além disso, servirão para evitar que os limites normais da glucose sejam ultrapassados nas pessoas que sofrem de diabetes.

Ambos os problemas representam tarefas bastante desafiadoras. No caso do problema do diagnóstico da diabetes, vários modelos foram construídos, ajustados (no que toca aos parâmetros) e testados usando o PIMA dataset. Uma contribuição chave deste trabalho são os vários métodos apresentados, analisados e testados para lidar com os dados em falta no dataset. O método de substituir os dados em falta pela média das características considerando a classe a que pertencem, juntamente com o algoritmo Random Forest produziu os melhores resultados, com uma precisão de 87.66%.

No que toca ao problema de prever os níveis de glucose, também aqui criámos vários modelos baseados em dados reais de pacientes diabéticos. Estes dados foram-nos fornecidos pela Associação Protectora dos Diabéticos de Portugal e representam uma mais valia para este trabalho. Para além disto, dois métodos de previsão intitulados previsão directa e previsão passo-a-passo foram investigados e testados. Das experiências realizadas, a regressão linear juntamente com o método de previsão directa formam a melhor combinação. Com ela, obtive-mos uma média de RMSE igual a 4.25 mg/dL e 23.46 mg/dL para previsão a 30 e 60 minutes.

Ambos os domínios dos problemas representam tarefas desafiadoras mas os nossos métodos demonstraram que conseguem atingir um bom desempenho nesses domínios. Do ponto de vista de aplicação, continua a haver muitos problemas desafiadores em ambos os problemas: Diagnóstico da Diabetes e Previsão da Glicose. De facto, no advento da Internet das Coisas, através da combinação de vários sensores disponíveis com nossos métodos, nós conseguiremos criar um sistema inteligente de monitorização da saúde dos utilizadores capaz de prevenir, diagnosticar, tratar e prestar cuidados de saúde para a sociedade em geral.

Palavras-chave: Aprendizagem por computador; Deteção de diabetes; Previsão dos níveis de glucose

Acknowledgements

First of all, I am thankful to God as without Him by my side, I would not be here; for the strength given to me to fight against all adversities. The following document summarizes a year's worth of effort, frustration and achievement. There are several people with whom I am thankful for their contribution in this work.

I would like to express my sincere gratitude to Professor Bernardete Ribeiro for accepting to be my advisor, for her almost infinite patience, guidance and all the knowledge transmitted even before in the Patter Recognition course. Her advice and friendship helped me in all the time of research and writing this document.

To EyeSee especially to Eng. André Pimentel, I am grateful for the endless support and ideas given to me since the beginning. In addition, I would like to thank Eng. João Redol, for the invitation to visit the EyeSee's facilities.

I am also grateful to Associação Protectora dos Diabéticos de Portugal (APDP), particularly to Dr. Rogério Ribeiro, for offering the opportunity to present this project and for the provision of datasets.

Finally but not for last, I am indebted to my parents for all the support and education given to me through my life, which made what I am today. Particularly, to my mother, I am forever grateful for being the buttress that I can always count with to help me to fight my problems.

Contents

List of Figures

List of Tables

Abbreviations

Acc	**Acc**uracy
AGE	**A**dvanced **G**lycation **E**nd
APDP	**A**ssociação **P**rotectora dos **D**iabéticos de **P**ortugal
BMI	**B**ody **M**ass **I**ndex
CGM	**C**ontinuous **G**lucose **M**onitoring
EU	**E**uropean **U**nion
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
FPG	**F**asting **P**lasma **G**lucose
Findrisk	**Fin**nish **D**iabetes **Risk** score
GDA	**G**eneralized **D**escriminant **A**nalysis
GRNN	**G**eneralized **R**egression **N**eural **N**etwork
K-NN	**K-Nearest** **N**eighbour
LapSVM	**Lap**lacian **S**upport **V**ector **M**achine
LDA	**L**inear **D**iscriminant **A**nalysis
LS-SVM	**L**east **S**quare **S**upport **V**ector **M**achine
MKS-SSVM	**M**ultiple **K**not - **S**mooth **S**upport **V**ector **M**achine
MSPE	**M**ean **S**quare **P**rediction **E**rror
NN-LPA	**N**eural **N**etwork plus **L**inear **P**rediction **A**lgorithm
OGTT	**O**ral **G**lucose **T**olerance **T**est
PCA	**P**rincipal **C**omponent **A**nalysis
PDF	**P**robability **D**ensity **F**unction
PH	**P**rediction **H**orizon
PHS	**P**ersonal **H**ealth **S**ystem
PID	**P**ima **I**ndian **D**iabetes dataset

RBF	**R**adial **B**asis **F**unction
RMSE	**R**oot **M**ean **S**quared **E**rror
Sens	**Sens**ibility
Spec	**Spec**ificity
SSVM	**S**mooth **S**upport **V**ector **M**achine
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector **R**egression
TG	**T**emporal **G**ain
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
T1DM	**T**ype **1** **D**iabetes **M**ellitus
T2DM	**T**ype **2** **D**iabetes **M**ellitus

Notation

$\mathbb{R}$	the set of reals
d	dimensionality of input space
$\mathbf{x_i}$	input pattern i
x	a variable (real number)
y	target values or classes (in pattern recognition)
n	number of training examples
$\mathbf{w}$	weight vector
b	constant offset
ϵ	parameter of the ϵ-intensive loss function
ξ	slack variable
D	dataset
ω	class
k	number of neighbours
Z	time series
$\mathbf{z}$	vector of predictors
z	real value on a time series
$\hat{z}$	estimated value
$\overline{z}$	mean of observed data
$\Vert\cdot\Vert$	norm
W	number of predictors
μ	mean
σ	standard deviation

Chapter 1

Introduction

"Good health is not something we can buy. However, it can be an extremely valuable savings account." by Anne Wilson Schaef.[1] Nowadays, with the current lifestyle that people are taking, most of them do not think properly in their health and about saving it. If in the past people migrated from the villages to the cities since there they would have easy access to shops, banks, etc., today's people mindset is starting to change as we see them moving from the cities back to the villages. This is happening because individuals want to be more close to the Nature and want to improve their health by doing so.

Despite all, there are people who cannot afford such a lifestyle change for several reasons. For all people but especially for those ones, there is the need of developing technologies and devices capable of understanding our eating habits, physical exercise habits, etc. in order to advise us to change them and possibly to prevent diseases or manage them properly. With the recent technological developments, this task is becoming easier and easier every day. Electronical devices manufacture are building tiny but accurate sensors to incorporate on smartphones, watches, clothes, etc., that are capable of recording several types of data that can be analyzed and can be used to improve our health.

The introduction of technology in the health area revolutionized everything from the way that patients' files are handled to telesurgery with which a surgeon can save a patient's life on the other side of the world. Since a few years ago, the medical staff and the investigators are trying to change the kind of health care that they provide. In other

[1] http://www.brainyquote.com/quotes/quotes/a/annewilson169948.html

words, instead of relying on reactive medicine, which aims to treat the symptoms once the individual is already ill, they are betting on preventive medicine that has the intent to prevent ones from getting sick in the first place. This allows the government to save money, which in turn can be invested in seeking for cures that does not exist for other diseases, on the creation of systems that given certain symptoms advises the patient to what he must or must not do bringing healthcare closer to the one and avoiding unnecessary visits to the hospitals (especially for the older people due to their usually associated conditions like walking difficulties) and other projects to improve healthcare.

European Union (EU) has funded several eHealth projects since 2007 which have "the interest of the citizens at their core such as projects that have the objective of diagnosing a patient more accurately and more quickly" and also projects that "make self-monitoring possible and/or help create better and safer care in general" [1]. Projects with the focus on Personal Health Systems (PHS), mobile health, telehealth, etc. but also projects on diabetes, cardiovascular disease and other areas are the ones in the sights of the EU. This is particularly important for elderly people because of the reason already mentioned earlier. By developing PHS, it is possible to provide continuous, quality controlled and personalized health services to the ones. If an intelligent processing unit is used along with the device, it becomes possible to analyze data and define proper actions to overcome problems either on the individual being monitored or to the health practice more generally concerning information provision and/or more active engagement in discovering new information that could lead medical staff and researchers to the diagnosis, treatment and rehabilitation.

In a connected world, the appearance of a system that could gather, in a easy and efficient way, data from the patient and use it to build a model that would allow the identification and prediction of several pathologies, would be extremely valuable for every person in the world. SMITH (Smart MonITor Health system) pretends to be that system that would help an individual to be more aware of his health condition and, at the same time, provide more and better information for the medical teams to analyse and diagnose pathologies.

The first pathology SMITH will focus, hence the origin of this work, is on Diabetes, given its growth and lack of diagnosis - as said, it is expected that half of the individuals with Diabetes are not diagnosed. Moreover, the main concern and focus of this thesis is on the ability to build a method or set of methods that would allow, with a good degree of confidence, the classification of Diabetes and the prediction of hypoglycaemia and hyperglycaemia.

1.1 Background

Diabetes mellitus, commonly known as diabetes is a metabolic and chronic disease characterized by the absence or low production of insulin associated or not with deficient action of it in the organism. Our body cells need energy to survive and keep working and the insulin is important for making that happen. Insulin is a hormone that allows the cells to use glucose (sugar) for energy from the food that we eat.

Diabetes has three different types. Type I diabetes mellitus (T1DM) is caused by an autoimmune reaction where the body's defence system attacks the cells that produce insulin. The reason for this is not fully understood. Pancreas, which is the organ responsible for the production of insulin, in people with type I diabetes does not produce any quantity making the person insulin dependent. T1DM develops most commonly among younger people. In type II diabetes mellitus (T2DM), there is a deficit in the production of insulin and a resistance to it, which means that the body requires a larger quantity for the same amount of glucose. This is the most common type of diabetes representing 90% of diabetic people worldwide. There is also a third type of diabetes named gestational diabetes [2]. During pregnancy women who develop a resistance to insulin and consequent high blood glucose are considered to have gestational diabetes. It is suspected that hormones produced by the placenta are responsible for the block of the action of the insulin. After the birth this type of diabetes disappear. Nonetheless, the baby has a higher lifetime risk of obesity and developing T2DM.

Nowadays, the diabetes diagnosis is made through blood tests, which involve drawing blood at a health care facility and then send the sample to a laboratory for analysis for better accuracy when compared with traditional glucose measuring devices like finger-stick devices. There are three tests that can be made to make a proper diagnosis which are the following: A1C test, Fasting Plasma Glucose (FPG) and Oral Glucose Tolerance Test (OGTT) [3]. The first one aims to measure the average glucose for the past 2 to 3 months, the second one check the glucose level after 8 hours without having eaten or drunk anything except water and the last one is a test that measures the blood glucose level before and 2 hours after drinking a special sweet drink.

There are symptoms associated with the glucose level on the blood. A person is on a hypoglycaemia state when the glucose level is bellow 70 mg/dl and if it is ignored it can lead to unconsciousness, permanent brain damage or death. If it is above 200 mg/dl two hours after eating or greater than 126 mg/dL when fasting, the subject is considered to be in a hyperglycaemia state, which can conduct to cardiovascular disease, blindness, kidney failure or damage, nerve damage, etc. Taking in consideration all problems associated with diabetes, it is considered a serious health problem that should be treated to prevent further body damage.

1.2 Motivation

Today's lifestyle is full of advantages, but also filled with disadvantages. Due to the improvement of the living life, changes in diet, less sport and other factors, the global incidence of diabetes is increasing rapidly. Regarding the costs of treating diabetes complications, it was proven that it costs more than 2 times of the cost of controlling the blood glucose level.

Many people already consider it the 21st century disease. World Health Organization[2] classified diabetes as one of the 10 top causes of death between the year 2000 and 2012. In 2013, 382 million people worldwide had diabetes. By the year 2035 it is expected that 592 million people will be living with this disease, which corresponds to an increase of 55% [2] and it is expected that in 2030 diabetes will be the 7th leading cause of death. Since that there is still no cure for diabetes and people who is or will be suffering from this pathology need to control it in order to prevent health complications, it is essential to create systems, devices, etc. that helps them in their daily life.

From the perspective of doctors it is important to have systems and tests that give useful information that simplifies the decision process. News[3] from April 2014 reported that one in twenty adult patients is misdiagnosed in outpatient clinics and doctors' offices. Regarding diabetes, 46% is the estimated percentage of undiagnosed people with this

[2] http://www.who.int/mediacentre/factsheets/fs310/en/

[3] http://www.cbsnews.com/news/12-million-americans-misdiagnosed-each-year-study-says/

pathology [2]. While in some cases people do not suffer serious injuries, in other cases their clinical status gets worse.

Furthermore, it is an advantage for people to do some kind of a fast test to check their probability of developing such disease without leaving their homes. By doing so we are no more talking about hospital medicine but about pre-emptive medicine. In another words, instead of dealing with already ill people, we try to prevent them from suffering diseases, which saves money to the public health system. Since that undiagnosed or misdiagnosed cases can lead to death even when it comes to diabetes, this is considered a serious problem that needs to be addressed.

1.3 Objectives

The main objective of this internship is to study and develop models capable of extracting valuable information among biomedical data related to diabetes.

In particular, the objectives of this work are the following:

1. Develop a model capable of testing the possibility of a person having diabetes
2. Build a model able to predict hyper and hypoglycaemia crisis

Each one of these goals involves several tasks which may be divided into several sub-objectives which are listed bellow

1. Definition of the pathology to be studied
2. Research on related work and existing commercial applications
3. Search for datasets or create them
4. Analysis of methods and algorithms
5. Analysis of results and conclusions

With this work, we investigate several algorithms capable of helping medical staff in the diagnosis of diabetes in their patients. Usually doctors rely on some tests and forget other factors that may be associated with this disease, which can indicate another clinical picture, ending eventually by making a diagnosis that is not as accurate as it could be. Also, the patients to assess their risk of developing diabetes can use it.

After a person has been diagnosed with diabetes, there is the need to control the level of sugar in the blood. Although there is the need to follow it frequently, the majority of patients measure it from 3 to 4 times a day, which might not prevent them from suffering the consequences of presenting low/high sugar levels. Having said that, an algorithm that predicts the glucose level with some time ahead would allow individuals to take the necessary actions to avoid trespassing the limits. This is the second part of this project.

1.4 Document Structure

The document is organized as follows:

Chapter 1 – Introduction: this current chapter aims to introduce the dissertation, i.e., gives an overview of the work, depict its importance, the motivation behind the project, the objectives and some medical background that helps the reader to understand some issues related to the disease.

Chapter 2 – State of Art: this part of the dissertation is divided into 3 sections. In the first two sections, we describe how machine learning and diabetes and how time series and glucose prediction are linked. Besides that, it is presented related work to both problems. In the third section, commercial solutions are investigated and described.

Chapter 3 - Diabetes diagnosis: in this chapter the dataset used in this part of the work is described. After that, we define the model development phases, which includes data-preprocessing, the algorithms and the evaluation technique. The last two sections present the computational experiments, the discussion of the results and the conclusions of this part of the work.

Chapter 4 – Glucose level prediction: as in the previous chapter, we describe the datasets, the different model development stages and the computational experiments performed. In the final, an analysis of the results is made and the conclusions are presented.

Chapter 5 – Final Conclusions and Future Work: in this final chapter, we give a review of the work and make a reflection about the achievements accomplished. Some ideas to improve the current work are given in the last section.

Appendix A - Mobile development frameworks: in this appendix, new frameworks developed by the two most famous electronic devices, which can help in the development of a future application are described.

Appendix B - Glucose level prediction full tests: this appendix contains all the results of the executed tests regarding the glucose level prediction problem.

Appendix C - Minutes of the meetings: in this annex, a summary of each meeting held over the year is given.

Chapter 2

State of the Art

In the first part of this chapter, the literature review related to diabetes diagnosis (Section 2.1) and glucose level prediction (Section 2.2) is offered. Also we analyse some commercial solutions regarding these two problems in Section 2.3.

2.1 Diabetes diagnosis

Diabetes diagnosis is made based on some specific tests described in the first chapter. With the evolution of technology, new methods of data analysis were discovered that allow researchers extracting more information and in a faster way when compared to doing so manually. Since part of this work is about diabetes detection, machine learning is introduced and also some related work made on the field is reviewed in the following subsections.

2.1.1 Machine Learning and Diabetes

Since the combination of informatics with health and medicine it became possible to exploit the capabilities of the computers to process patient's information and discover relationships between data. At the present time, computers are used to acquire x-ray images with the proper equipment, to plan radiation therapy or surgery and to create 3D body part's model to have a better insight of a problem. It can also be used to give a second opinion when it comes to, for example, diabetes diagnosis.

Several studies have been made to find methods to accurately diagnose diabetes. Pattern recognition is a branch of machine learning focused on the recognition of patterns and on the classification of those patterns into categories or classes. A pattern is defined as a combination of features, i.e. a group of distinct aspects or characteristics that may be symbolic (e.g. colour) or numeric (e.g. height).

Considering the fact that we want a method to identify if a person has diabetes or not, we are facing a binary classification problem in which machine learning and artificial neural networks are used to do so.

For this kind of problems we want to estimate a function $f : \mathbb{R}^d \to Y, Y = \{0, 1\}$ using input-output training data

$$X = (x_1, y_1), ..., (x_n, y_n) \in \mathbb{R}^d \times Y \tag{2.1}$$

such that f can correctly classify unobserved data $(\mathbf{x}, y)$. In another words, we want to have $f(\mathbf{x}) = y$ for examples ($\mathbf{x}$,y) that follow the same probability distribution $P(\mathbf{x}, y)$ as the training data. $\mathbf{x}_i$ is a feature vector of length d ($\mathbf{x} \in \mathbb{R}^d$) and y_i ($y \in Y$) is the correct label/class of $\mathbf{x}_i$

In the rest of this section we analyse several strategies used by other investigators to address the diabetes diagnosis problem.

2.1.2 Related Work

Yu et al. [4] used Support Vector Machines (SVM) to see how robust it can be in disease diagnosis. The approach consisted in creating two classification schemes: a) people with diagnosed or undiagnosed diabetes vs. pre-diabetes or no diabetes and b) undiagnosed diabetes or pre-diabetes vs. no diabetes. Data between 1999 and 2004 from the National Health and Nutrition Examination Survey was used in this work. Results showed that Radial Basis Function (RBF) kernel function performed better in the first classification scheme with accuracy of 83.47% and for the second one, the linear kernel function was the best with an accuracy of 73.18%. They also compared these values with the ones obtained by a logistic regression analysis and concluded that there is no statistical difference in their discriminative power, i.e., the performance of both methods are equal.

Kumari and Chitra [5] also achieved an accuracy of 78% with the SVM algorithm and the same kernel function (RBF) using the PIMA[1] database obtained from the UCI Repository of Machine Learning Repository[2].

A modified version of SVM algorithm named Laplacian Support Vector Machine (LapSVM) was tested by Wu et al. [6]. In this study, a LapSVM was trained as a fully supervised learning classifier and another one as a semi-supervised learning classifier. The accuracy results achieved were 79.17% and 82.29%, respectively.

A feature selection technique based on genetic algorithm and the SVM algorithm were used by Kumar et al. [7] on several medical datasets including PIMA. Genetic algorithms allow to efficiently look up for the best combination of features over a big search space. Based on the idea of the natural selection, a certain number of individuals are created forming a population where each one represents the features to take into consideration. After that, the individuals are evaluated by the accuracy of the SVM with the features that they encode and then a mutation and/or a cross over may occur. This process is repeated until there is convergence or a maximum iteration limit is reached. The proposed method accomplished 77% of accuracy.

Polat and Güneş [8] suggested a method to attain better results in this problem. In their study, they used Principle Component Analysis (PCA) to reduce the number of features present in the PID dataset. After that, diabetes' diagnosis is performed by a neuro-fuzzy inference classification system. This approach reached an accuracy of 89.47%.

Polat and Arslan [9] compare the results obtained with Least Square Support Vector Machine (LS-SVM) and their method that is a combination of Generalized Discriminant Analysis (GDA) and LS-SVM. GDA is used as a pre-processing data method while LS-SVM classifies the data. While they got an accuracy of 78.21% with LS-SVM, they accomplished 79.16% with GDA-LS-SVM.

2.2 Glucose level prediction

After a person has been diagnosed with diabetes, the subject has to control the glucose level in order to avoid a state of hypoglycaemia or hyperglycaemia. It is much more

[1] https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

[2] https://archive.ics.uci.edu/ml/index.html

desirable to prevent these episodes rather than deal with them when people are already suffering from their effects. Sometimes it is not possible for a subject to see for example, an hypoglycaemia event coming and when he/she realizes what is happening it can be too late making this aspect a really important one.

2.2.1 Time Series and Glucose level prediction

Throughout life, a diabetic patient has to measure his blood glucose. Since those measurements have a temporal order, we approach the task of predicting the future blood glucose levels as a time series forecasting problem. In time series prediction, the task is to estimate the future value of a target function based on the current and past data samples [10] and can be generalized as shown in Equation 2.2.

$$\hat{z}_{t+PH} = f(z_t, z_{t-1}, z_{t-2}, ..., z_{t-n+1}) \tag{2.2}$$

For an observed time series Z with n points, where t is most recent observation, $t - n$ is the oldest one and PH is the prediction horizon, i.e. the number of steps ahead of the actual point, a future value at $t + PH$ can be estimated with a function f. This function is a model that can be used to get an estimated value to $\hat{z}_{t+PH}$.

Several studies have been conducted to create methods to forecast the sugar level with the objective of warning the subject in advance. Throughout the rest of this text we are going to present some studies done on the field.

2.2.2 Related Work

Sparacino et al. [11] studied the possibility to predict future glucose levels. To this purpose data from 28 type 1 diabetic volunteers for 48 hours was analysed which was collected through a continuous glucose monitoring device with sampling intervals of 3 minutes. The behaviour of a first-order polynomial and a first-order autoregressive model were compared. At each step a new set of model parameters is dynamically computed using the weighted least square technique. In order to remove high frequency noise from the data, a low pass first order Butterworth filter was used. Tests with different forgetting factors and prediction horizons (PH) were performed. The forgetting factor stipulates the length of the "memory" i.e., it is a weight given to each sample that takes part in the calculation of the new prediction. The assessment of both approaches was assessed by resorting to the mean square prediction error (MSPE) and, the delay between the predicted and original glucose curve through positive and negative trends. They conclude that it is possible to predict glucose levels and predict hypo/hyperglycemia events 20-25 min ahead in time with a PH of 30 min. Also, the auto-regressive model yielded more accurate results when compared to the polynomial one.

Baghdadi and Nasrabadi [12], to try to accurately predict the glucose level in the future, have used a Radial Basis Function (RBF) neural network. They considered 4 intervals during the day (morning, afternoon, evening and night) and for each one they created a separated neural network. The data used for training and testing covers a continuous period of 77 days of a single patient. They chose the best set of features for each neural network by a pruning method. With this method, they achieved a root mean square error (RMSE) of 0.5202 mg/dl, 0.6804 mg/dl, 0.4392 mg/dl and 0.1134 mg/dl for the morning, afternoon, evening and night neural network, respectively.

Zecchin et al. [13] proposed an algorithm which combines a fully connected and feedforward neural network model and a first-order polynomial extrapolation algorithm named Neural Network plus Linear Prediction Algorithm (NN-LPA) for glucose level prediction with PH value equal to 30 minutes. In addition to the past CGM readings, the algorithm also takes advantage from information about carbohydrate intakes quantified through a physiological model. The neural network consists of one hidden layer with eight neurons and one neuron as the output layer. Each one of the neurons in the hidden layer uses a tangent sigmoid activation function and the output neuron uses a linear transfer function. Network parameters were randomly initialized and they were tuned by backpropagation Levenberg-Marquardt training algorithm. For training and validation purposes, 20 simulated type I diabetic profiles and 15 type I diabetic real patients' information were used. In order to evaluate the performance, 4 indices were calculated: RMSE, temporal gain (TG), normalized energy of the second-order differences ($\text{ESOD}_{\text{norm}}$) and the index J. $\text{ESOD}_{\text{norm}}$ measures the risk of generating false hypo and hyperglycaemia alerts (the closer the value is to 1, the better) and the index J measures the effective "usability" of the predicted profile (the lower values, the better the prediction is). To assess the solution, nine daily profiles containing hypo and hyperglycaemia events were used. The average results of RMSE, TG, $\text{ESOD}_{\text{norm}}$ and

J for these nine real CGM test series were 14.0±4.1 md/dl, 16.2±3.7 min, 2.7±1.6 and 10.8±7.4, respectively.

A preliminary study, in which the authors used Support Vector Regression (SVR) algorithm to create a model capable of predicting the blood glucose level of a T1DM patient was made by Marling et al. [14]. They selected a pivot point about 1 month into the patient's study to divide data for training and testing purposes. The data from 7 days before the pivot points was used to train the model while the data from the 3 days after the pivot point (including that point) was used to test. To predict the blood glucose level 30 and 60 minutes ahead, two SVR models were created and the results were compared to a baseline, which uses the present measurement to predict any future blood glucose level based on RMSE. Information about the present time blood glucose level, a simple moving average over the last 4 points (including the actual one), an exponentially smoothed rate of change in blood glucose level from the past 4 points counting with the actual point, bolus dosage totals, basal rate averages and meal carbohydrate amount starting 30 mins before prediction time and exercise duration and intensity was used as features. Results showed that SVM models outperformed the baseline. For 30 min ahead predictions, a RMSE of 18.0 mg/dL was achieved while for 60 min that value was 30.9 mg/dL.

The authors in [15] used the SVR algorithm to came up with a model capable of accurately predict the subcutaneous glucose level. Information about the level of plasma insulin, the rate of appearance of glucose in plasma after a meal, current and past subcutaneous glucose level as well as some variables related to exercise was used in such task. The model was trained and tested with data from two type I diabetes individuals. Results showed that for patient 1 there was an average RMSE of 18.5 and 27.23 mg/dL when predicting 30 and 60 minutes ahead, respectively. For patient 2 the results were better: 15.33 and 22.8 mg/dL correspondingly for the same prediction length as patient 1.

2.3 Commercial Solutions

From the point of the view of a company or startup it is important to analyse what already exists in order to introduce something new or improve the current solutions.

The tests that currently exists are invasive as it is necessary a small amount of blood to measure the sugar level on the body. It is important to have fast, low-priced, precise and non-invasive or at least the minimum invasive as possible tests to increase people's comfort.

In the next sections we will describe some commercial solutions that exists on the market for diabetes diagnosis and blood glucose level prediction.

2.3.1 Diab-spot

Diad-spot [16] is a non-invasive machine (see Figure 2.1) created by a company named Diagnoptics to identify people at risk of having diabetes or pre-diabetes. It is a machine that measures the tissue accumulation of Advanced Glycation End products' level (AGEs), which are sugar-derived substances [17]. AGEs normally accumulate during a person's lifetime but in the case of people with pre(diabetes) this process is faster than normal people.

To make the diagnosis, a person has to put his forearm on the top of the machine. After that, the skin is illuminated by a light that excites fluorescent moieties present in the tissue that will emit light with another wavelength as a response. Using a spectrometer to capture the light and other techniques, a more selective discrimination of specific AGEs can be obtained. This information combined with some person's characteristics like person's height and weight, makes it possible to Diab-spot calculate the test result that is shown on the machine's screen.

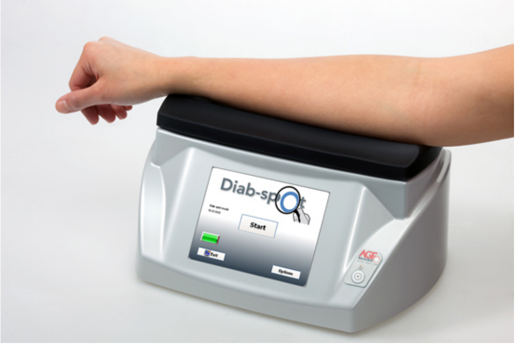

FIGURE 2.1: Diab-spot [source: http://goo.gl/pWjUEc]

2.3.2 Findrisk

Finnish Diabetes Risk score (Findrisk) is a free questionnaire originally developed and validated in Finland, which assesses an individual's risk of developing T2DM [18]. Composed of 8 simple questions that people can easily answer, it detects diabetes risk in a 10-year period. Each answer is assigned with a different score that was computed using a logistic regression.

At the time of the creation of the questionnaire, there was a concern about considering only factors that were easy to assess without any laboratory tests or other clinical measurements that could require special skills. The final list of parameters that are related to diabetes and used to calculate a patient's risk is the following one:

- Age
- Body Mass Index (BMI)
- Waist circumference
- Physical activity level
- Vegetable and/or fruit eating habits
- High blood pressure history
- Previous high blood glucose
- Family history

There are 5 risk categories that range from low, which means that 1 person in 100 develops diabetes, through to very high where 1 in 2 will develop this condition in the period of 10 years. Given the individual's answers for the questions, the score is calculated and the risk assessed. By doing this, it is possible to create and/or adapt strategies for each risk group to try to reduce the risk of developing diabetes, as the variables that are being taken into account are fully understandable by people.

2.3.3 Bayer

Bayer has a vast variety of products [19] to measure blood glucose level that can fit each patient's diabetes situation such as simple measurement systems for basic testing needs or more complete ones that help people to better manage their diabetes through a smarter meter.

An example of such a device is the Contour next USB (Figure 2.2). It allows the patient define the target ranges for fasting, before/after meals blood sugar and also a total maximum and minimum values (the values should be discussed with a doctor). With the AutoLog feature enabled, the subject can choose to which category the test belongs to (fasting, before meal, after meal or no mark) whenever a test is made. If the result is out of the range between the pre-defined min and max values, a screen with large-sized orange numbers will alert the subject that his blood sugar is low or high. It is also possible to configure a test reminder. When the reminder time is reached, 20 beeps will sound to warn the person. Information about carbohydrates, insulin and notes can be logged into the system. Trends can also be shown with which it is possible to see how many tests' results are above, within or bellow the targets' range.

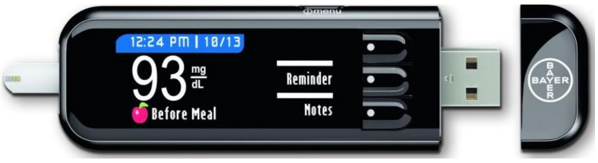

Figure 2.2: Contour Next Blood Glucose Monitoring System [source: http://goo.gl/UqAVRC]

Besides the USB model features, Contour Next Link can send through wireless the results to the Medtronic MiniMed$^{\text{TM}}$pump's bolus wizard to adjust the insulin dose.

Glucofacts$^{\text{TM}}$ Deluxe is management software that can be used with some devices which analyses data and generates reports. It can be very helpful since the patient can see results for combined days or weeks, trend reports and patterns through graphics.

2.3.4 LifeScan

LifeScan created the OneTouch family products [20] to address the diabetes-monitoring problem. Like Bayer, LifeScan has simple and more advanced devices.

OneTouch Verio Sync (see Figure 2.3) is a device that normally measures the glucose level. The special feature in this device is that it can communicate with Apple's devices to send the results back through Bluetooth. Using the mobile application – OneTouch Reveal – it is possible to see a summary screen with information about 14-day glucose tests (how many tests are below the normal range, within it or above it), the average value, carbohydrates, activity and medication information. Tests' results can be tagged as before meal (or after meal) to enrich the semantics of the data. Incorrect tagging causes inaccurate and/or misinterpreted values and messages. Information about carbohydrates (in grams), activity (minutes of activity and the intensity) and medication (units of insulin taken) can be added. Logbook feature allows the patient to check blood sugar results over the last 90 days. If there are results above and/or bellow the normal ranges, they will be displayed in blue and/or red, respectively. There is the possibility to see the results in graphics with different icons according to them. High and low patterns are also detected based on the time that the test was made. For a high pattern to be identified, at least 3 results have to be over the before meal high limit and within the last 5 days. For the low pattern, only 2 results are needed. These limits can be configured with medical help. Reminders are also available.

Like one of the Bayer's devices, OneTouch UltraLink meter communicates with Medtronics' insulin pump to adjust the insulin dose.

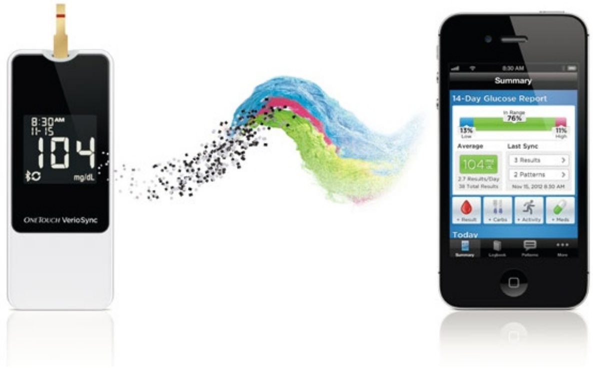

FIGURE 2.3: OneTouch® Verio®Sync [source: http://goo.gl/hjkjhc]

2.3.5 Medtronic

Guardian REAL-Time CGM (Figure 2.4) is a system created by Medtronic [21] to keep track of patient's glucose level and it is composed of a glucose sensor and a monitor. MiniLink REAL-Time transmitter and glucose sensor (Enlite sensor) is small, discrete and waterproof object that is inserted on the patient's belly with the help of Enlite Serter.

Instead of reading the glucose on the blood, it measures the glucose on the interstitial fluid, which is a fluid that surrounds the cells of the tissue bellow the skin. This is possible because the glucose moves from the blood vessels into the interstitial fluid. The monitor, to where the glucose information is sent, warns the patient for oncoming low and high glucose levels, presents the latest reading, a graph made of readings from a period of time (3, 6, 12, or 24 hours) and trend arrows that show the direction and rate of change of the glucose level. It is also possible to configure 8 thresholds to alert the patient when a threshold is surpassed. To calibrate the sensor, fingersticks are required 3 to 4 times per-day for optimal accuracy.

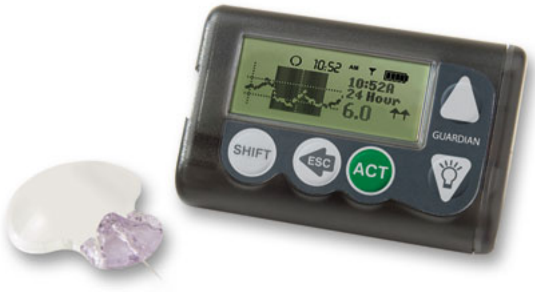

FIGURE 2.4: Medtronic - Monitor and Transmitter+Sensor [source: http://goo.gl/bcvEyd]

Enlite sensor can be used to transmit the glucose level information directly via wireless to MiniMed Revel Insulin pump to adjust insulin delivery.

2.3.6 Dexcom

Dexcom[22] has also a CGM system named Dexcom G4 Platinum (Figure 2.5), which is a combination of a wireless transmitter, a glucose sensor and a receiver (monitor). In the monitor, it is possible to see continuous sensor glucose readings every 5 minutes for up to 7 days, the actual trend and velocity and the most recent measurement. The readings on the graphic are marked with different colours depending whether the reading is above, bellow or on the normal range for better perception. The monitor vibrates and plays different sounds to warn the patient when the glucose level is high or low. As Medtronic's Enlite sensor, Dexcom sensor needs to be inserted on the patient's belly and requires calibration up to every 12 hours.

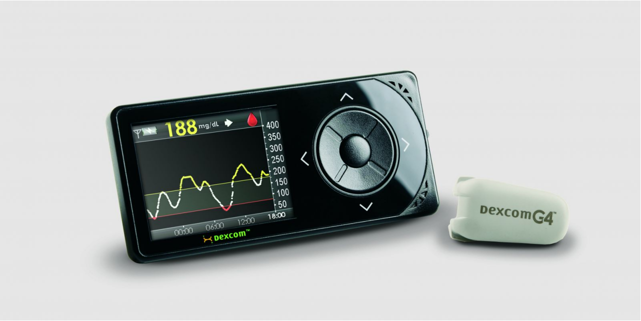

FIGURE 2.5: Dexcom G4 Platinum - Monitor and Transmitter+Sensor [source: https://goo.gl/iLyEVB]

2.3.7 Google[x] Labs

Google is currently developing contact lenses that are able to track the glucose through the tears. The contact lens consist in a tiny sensor and wireless antenna implanted in the middle of two layers of soft contact lens material (see Figure 2.6). Devices that are capable of making measurements every second are already being tested.

Google is also investigating the possibility of integrating LED lights that would turn on if the glucose level goes out of the normal ranges.

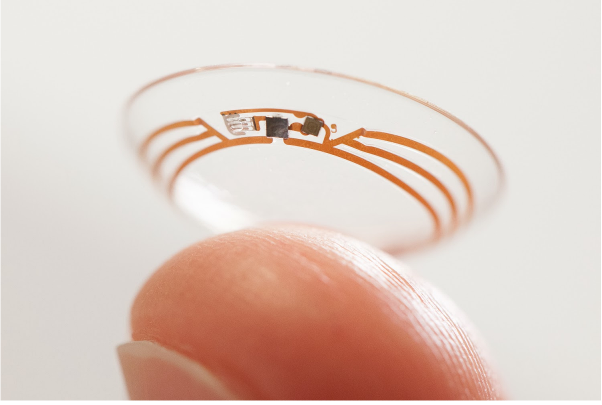

FIGURE 2.6: Google's Contact Lens [source: http://goo.gl/YLocLW]

The advantage of such device is that it is almost non-invasive, a much less painful and disruptive away of checking the sugar level compared to the tradition finger pricks. However, the idea of putting LEDs on the lens may not be a good idea since it can blind the user momentarily, depending on how much light it will emit, and cause long-term complications.

Chapter 3

Diabetes diagnosis

Nowadays, the massification of the Internet and other forms of content distribution and the relatively inexpensive means to store data, have provided access to people to analyse, for example, medical data and extract helpful information from it. To do this, machine learning provides tools for intelligent data analysis. It is possible to create models that accurately make a diagnosis of a person with a certain error. In order for a model to be somewhat intelligent, it needs to learn just like Humans, as without learning there is no intelligence. In general, it is only necessary to give a set of examples of well-classified diagnosis and the algorithms automatically develop, for instance, a classifier that can be used to support specialists diagnosing a patient's problem.

In Section 3.1, it is given a description of the dataset that is used in this work. Secondly, in Section 3.2 we discuss the model development phases which includes the description of some classification algorithms and performance metrics. Finally, we present the obtained results in section 3.3 along with their analysis.

3.1 Dataset

Pima Indians Diabetes (PID) dataset can be found in [23] and was donated by Vincent Sigillito who was a member of the Applied Physics Laboratory in the Johns Hopkins University in 1990. It is a collection of medical diagnostic reports from 768 patients. In particular, all patients are female who are at least 21 years old of Pima Indian heritage and live near Phoenix, Arizona, USA.

There are 268 cases in class 1 (positive test for diabetes) and 500 cases in class 0 (negative test for diabetes), which corresponds to 34.9% and 65.1% of all dataset, respectively. There are 8 attributes (plus class) that can be described as follows:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)$\hat{2}$)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

In Table 3.1 a brief statistical analysis of the data is given.

Attribute Number	Mean	Standard Deviation
1.	3.8	3.4
2.	120.9	32.0
3.	69.1	19.4
4.	20.5	16.0
5.	79.8	115.2
6.	32.0	7.9
7.	0.5	0.3
8.	33.2	11.8

TABLE 3.1: Pima Indian dataset statistical analysis

Instead of relying only in blood analysis features, this dataset has also information about physical characteristics that might be correlated with the diabetes diagnosis. The main drawbacks are that it only has records of female individuals, there is a low number of records and it is not recent.

3.2 Model Development

In order to achieve a model capable of distinct diabetic from non-diabetic people given certain information about them, several steps have to be taken.

Figure 3.1 shows those required steps to build the final model, which starts by the data-gathering phase where it is necessary to find data related to the problem. The following step is pre-processing the data since it can have inconsistencies, which the model may be unable to handle. In the coding step, the chosen algorithms selected in the previous stage (tool selection) are implemented. After that, to check the quality of the model several tests must be made. Eventually it can be necessary to tune parameters in the code and test those changes with regard to evaluate their effectiveness. At the point when there are no more enhancements that can be made and/or the results are satisfactory, the model is accepted.

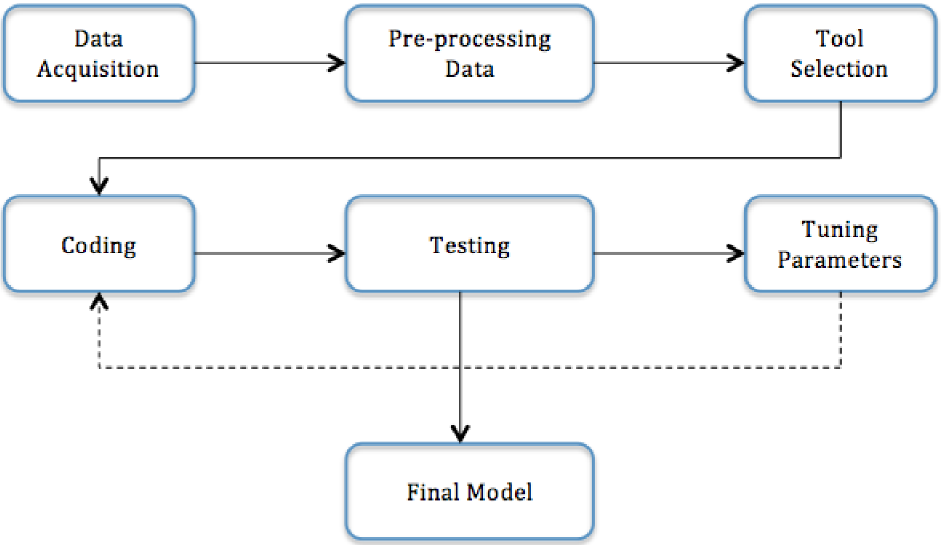

FIGURE 3.1: Design and Model Implementation flow diagram

In the next subsections, we will go through the model's implementation phases starting by the pre-processing step.

3.2.1 Data pre-processing

The pre-processing step plays a very important role. It is the phase where we remove or fix records that are not fully acceptable, since they have missing values, calculate new features, as several algorithms do not accept non-numerical data, normalize data, etc.

A dataset is a set of values of observed variables manually or automatically collected. For this reason, there can be noise and/or missing values, which make the pre-processing a necessary step in order to deal with this problem.

Although, in the analysed papers it is not mentioned the existence of missing values the fact is, when comparing the features names with its own values we conclude that there are missing values (the value of those features for some examples is zero) which was also stated by Breault [24]. For example, it is not possible to have a blood diastolic blood pressure equal to zero on a living being.

Considering the importance of this phase towards the creation of a good computational model, it is described some techniques to deal with this problem in subsection 3.2.1.1. Another problem is the fact of the dataset may be unbalanced i.e., there exists more examples of one class than the other. This problem is detailed in subsection 3.2.1.2.

3.2.1.1 Missing values

When dealing with missing values, there are at least two options to solve this problem, which are the following: 1) Data deletion [25] or 2) Imputation [26]. If a dataset has a considerable size, it is possible to consider the deletion of records with missing data. However, since health information is very hard to find and usually datasets have a small number of records, it is not advised to resort on the first option. Furthermore, without the deleted instances, it is possible that the generalization capability of the model is impaired as those records might be dissimilar to the remaining ones.

The second option aims to replace missing data with an estimation of their values. Since we are working with health data, we rely on this option. Such substitution can be made for example using one of the following techniques: Random value, Average of the k-nearest neighbours, Feature's class median or Feature's class mean.

The Random value method consists in calculating each feature mean (μ) and standard deviation (σ) and replace each missing value with a value taken randomly from a normal distribution with the same mean and standard deviation as the feature. Let us consider that the value x_{ij}, where i and j are the i-th example and the j-th feature, respectively, of the m-th class, C_m, is missing then it will be replaced by

$$\widehat{x}_{ij} = N(\mu, \sigma) \tag{3.1}$$

Substituting the missing values with this technique has the advantage of introducing a value that is similar to the others but outliers affect the parameters of the normal distribution.

For the average of the k-nearest neighbours as the name says, the k-nearest neighbours of the example that has the missing value are found. To find those neighbours just the complete examples (records without any missing value) are considered. After that, the new value is the feature's average value of those neighbours. Using the same notation as before, the new value will be

$$\widehat{x}_{ij} = \sum_{i=1}^{k} \frac{x_{ij}}{k} \tag{3.2}$$

With this method, the similarity of the examples is captured which is given by the distance between them. There are some distance functions like Euclidean, Manhattan, Mahalanobis, etc. yet we used the first one. Furthermore, the choice of the correct number of neighbours (k) is a problem since that a small k puts more emphasis on some records. However, a bigger value of k may include instances that are very different from the instance that has the missing value and it will be time consuming if the dataset is big.

Feature's class median and feature's class mean are almost the same. While in the first one we substitute the missing value by the median of values of a feature considering the examples classes, in the second one we consider the mean. For example, for a feature F, the mean of that feature is determined for each of the existing classes. Once again, considering the previous notation, the missing value will be replaced by

$$\widehat{x}_{ij} = median_{\{x_{ij} \in C_m\}}(x_ij) \tag{3.3}$$

if the median is used and

$$\widehat{x}_{ij} = mean_{\{x_{ij} \in C_m\}}(x_ij) \tag{3.4}$$

if the mean is adopted.

Since the mean is computed considering the class of each instance, this means that the number of instances of the class where the example with the missing value belong is taken into account. Also, with this method every example of each class contributes in the same way to the new estimated value. Both techniques have the disadvantage that the same value is used to substitute every missing values within each feature. Regarding the median, the calculated value will not contemplate the influence of the other instances as it occurs for example with the mean.

Due to the fact that every imputation method has its own advantages and disadvantages, tests were made to compare their effectiveness. The best method was chosen in regard to the final model accuracy and the results can be seen later on section 3.3.

3.2.1.2 Balanced vs Unbalanced dataset

In the classification problems' area, unbalanced datasets appear frequently since the majority of real world applications lead to that like fraud detection or in the manufacturing industry. The problem resides in the fact that the instances of one class significantly outnumber the examples of the other one. Also, the minority class is usually the most important one; the one that represents the most important concept to be learned.

What usually happens when this problem is ignored is that the generated computational models have a good accuracy among the examples of the class with more instances, whereas the minority class is misclassified. It also affects the analysis of the results since some performance metrics take into equal consideration both classes like the accuracy.

This problem can be addressed in 2 different levels: data or algorithm. Since most algorithms do not work well with this problem, methods at data level are chosen. Within that category, it is possible to divide into 2 types: oversampling[27] and undersampling[28].

Oversampling is a simple method that consists on randomly replicate examples of the minority class to increase its size. This possibly leads to overfitting which is a disadvantage of this technique.

On the other hand, undersampling the dataset ignores potentially useful information provided by the ignored instances. However, it speeds up the training phase and toward to reduce its problem, the instances to delete are randomly selected. For these reasons, undersampling is applied on this work.

3.2.1.3 Normalization

Data normalization is an important issue in many classification algorithms, since a lot of them only work on normalized or scaled data when dealing with features of different units and scales. Because some algorithms may use, for example the Euclidean distance, all the features should have the same scale for a fair comparison between them. There are at least 2 methods for data normalization [29]: min-max normalization and z-score normalization.

Min-max normalization performs a linear transformation on the original data to map a value x of the feature F to x' in the range [0,1]. This is done, by using the following equation

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.5}$$

where x_{min} and x_{max} are the minimum and maximum value of the feature F, correspondingly.

Z-score method normalizes the values of feature F based on its mean and standard deviation. A value x is normalized to x' by computing

$$x' = \frac{x - \mu}{\sigma} \tag{3.6}$$

where μ if the mean of feature F and σ is its standard deviation.

Although min-max normalization method preserves the relationships among the original data value, one needs to previously know the minimum and maximum value of each feature. It also may lead us to “out-of-bounds” problem if a future value falls out the range of the original feature’s data. For these reasons, we cannot rely on this technique.

3.2.2 Algorithms for Diabetes Detection

Machine learning models can be used to distinguish diabetic from non-diabetic people making classification techniques essential for this problem. Different classifiers were used on related works and some of them are introduced in the following subsection.

3.2.2.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a method to find a linear combination of features, which characterizes two or more classes of, for example, objects. This combination can be used as a linear classifier or as dimensionality reduction for later classification.

The decision boundary can be described as a linear discriminant function given by the following formula

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b \tag{3.7}$$

where $\mathbf{w}$ is the weight vector and b is the bias.

To find such function, a training set is used in which we are interested in minimizing a criterion function, for instance, the training error using Equation 3.8.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x})) \tag{3.8}$$

This algorithm will serve as a baseline from which we will try to have better results.

3.2.2.2 K-Nearest Neighbour

The K-Nearest Neighbour (KNN) is a simple algorithm, which is based on the samples i.e., each sample is classified considering the class of the k-nearest points [30]. During the training phase an internal representation of the train set is built along with the corresponding classes.

This algorithm is considered to be in the lazy learning algorithms category since there is no actual training. To classify a new sample, a certain distance metric between the sample being classified and all the others is calculated. After that, the k nearest labelled samples are selected and the classification is grounded on these points. In other words, the resulting label is given by the majority among the k neighbours (see Figure 3.2).

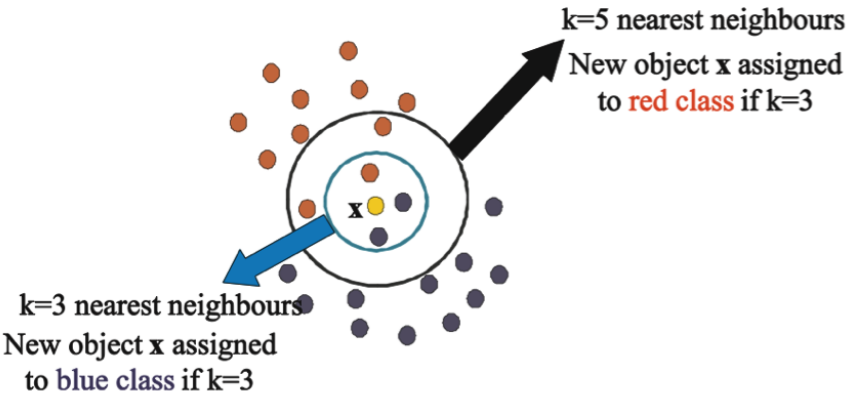

Figure 3.2: K-Nearest Neighbour: effect of different values for k [30]

3.2.2.3 Support Vector Machines

Support Vector Machines (SVM), originally purposed by Cortes and Vapnik [31], is one of the most widely used algorithm for classification problems. For training purposes, it uses a set of records properly labelled to come up with a model in which a hyperplane that correctly separates the classes is defined (see Figure 3.3) and can be used for future classification of new samples.

This problem can be mathematically formulated as

$$f(\mathbf{x}) = sgn(\mathbf{w} \cdot \mathbf{x} + b) \tag{3.9}$$

To find the optimal hyperplane, the following optimization problem [32] has to be solved

$$\begin{aligned} \text{minimizes} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 \\ & i = 1, ..., n \end{aligned} \tag{3.10}$$

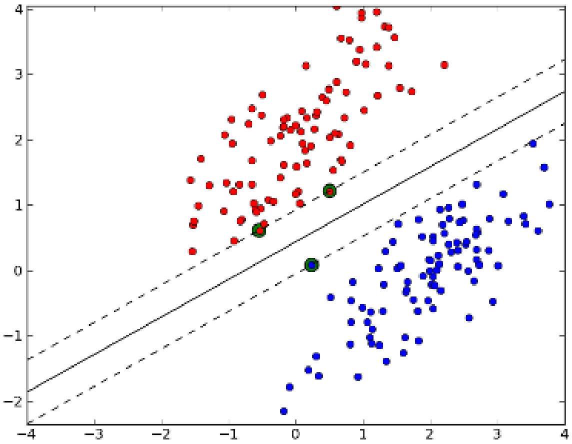

Figure 3.3: SVM linear decision boundary [source: http://goo.gl/lNDkyw]

If the data is non-linearly separable this problem cannot be directly solvable. However, the introduction of the slack variable ε_i helps to solve the problem allowing some instances to fall within the margin but penalizes them. Now, the optimization problem becomes

$$\begin{aligned} \text{minimizes} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}\varepsilon_i \\ \text{subject to} \quad & y_i \cdot ((\mathbf{w}\cdot\mathbf{x}_i) + b) \geq 1 - \varepsilon_i \\ & \varepsilon_i \geq 0 \\ & i = 1, ..., n \end{aligned} \tag{3.11}$$

Despite the introduction of ε, sometimes it is not easy to have a good linear decision hyperplane. When classes cannot be divided by a linear boundary, it is still possible to create a model using a technique called kernel trick, which maps the samples to a much higher dimension where it is possible to define a linear decision boundary. An example of such kernel is the Radial Basis Function (RBF) kernel, which is the one that we are going to use.

If $\Phi(\mathbf{x})$ represents a function to map the data into a different space, the Wolfe dual of this problem [32] is the following

$$
\begin{aligned}
\text{minimizes} \quad & \frac{1}{2}\sum_{i,j}(\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)) - \sum_i \alpha_i \\
\text{subject to} \quad & C_i \geq \alpha_i \geq 0 \\
& \sum_i \alpha_i y_i = 0 \\
& i = 1, ..., n
\end{aligned}
\tag{3.12}
$$

where

$$k(\mathbf{x}_i,\mathbf{x}_j) = (\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)) \tag{3.13}$$

which means map the data into the new space and calculate the inner product of the new vectors.

Considering Equation 3.12 and 3.13, the decision function is:

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{n} y_i\alpha_i \cdot k(\mathbf{x},\mathbf{x}_i) + b\right) \tag{3.14}$$

Given the fact that Equation 3.12 represents a convex quadratic programming problem, it means the solution exists and is unique. A more detailed explanation of SVMs can be found in [33].

Since SVM are widely used in classification problems for their capacity of generalization, we decided to use this algorithm.

3.2.2.4 Random Forest

Random Forest, originally proposed by Breiman [34], is a different kind of algorithms when compared to the ones described earlier as it belongs to a different class of machine learning algorithms named ensemble methods.

Based on the idea that a group of “weak learners” can come together to form a “strong learner”, it combines several models to solve a single classification problem. Combining multiple models that in this case are decision trees, it associates all the individual predictions to form the final one (see Figure 3.4). This prediction usually is as good as or better than the prediction made by any of the classifiers since it follows a divide and conquer methodology which is used to improve performance.

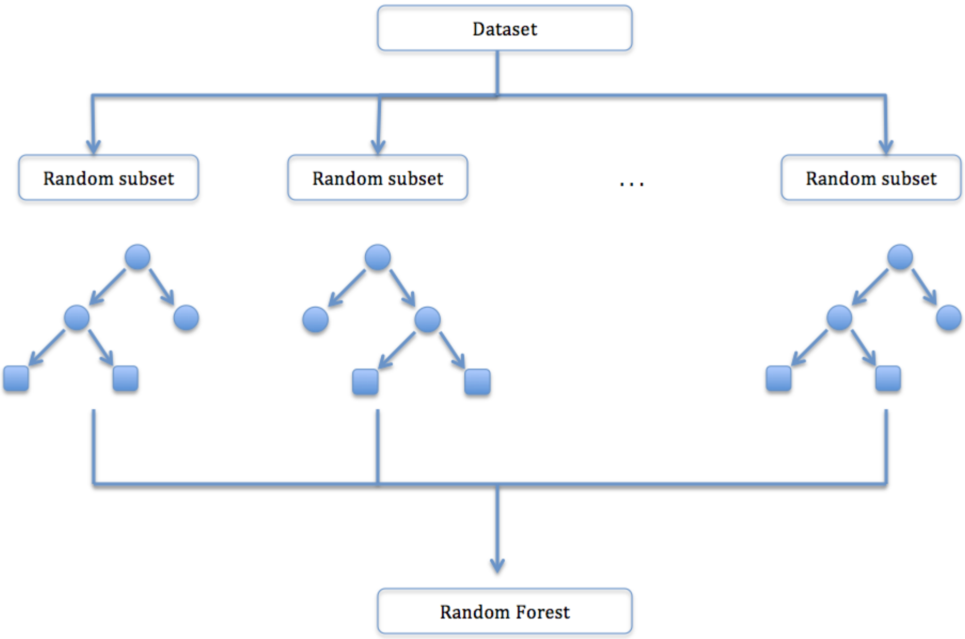

FIGURE 3.4: Random Forest classifier scheme

To build an ensemble of B trees, C cases at random with replacement are sampled to create a subset of data. Then at each node, m features are selected randomly from all the features and the one that provides the best split, according to a certain function, is used to do a binary split on that node. At the next node, m features are chosen again at random and the same action as before is performed. Usually, the value of m is $\sqrt{d}$ being d the total number of features. The final result (class) is given by the majority vote of the trees.

Due to its nature and the fact that the parameter that needs to be selected is the number of trees, random forest algorithm is suitable to use in this work.

3.2.3 Evaluation Technique

To identify which model is the better, K-fold cross validation is performed. With this method, the dataset is divided into K folds of equal size whenever it is possible. Each block is used to train and test in different stages (see Figure 3.5. In other words, the model is trained K times with $K-1$ folds and in each iteration the one that was held out is used to test. A more extended description is given in [35].

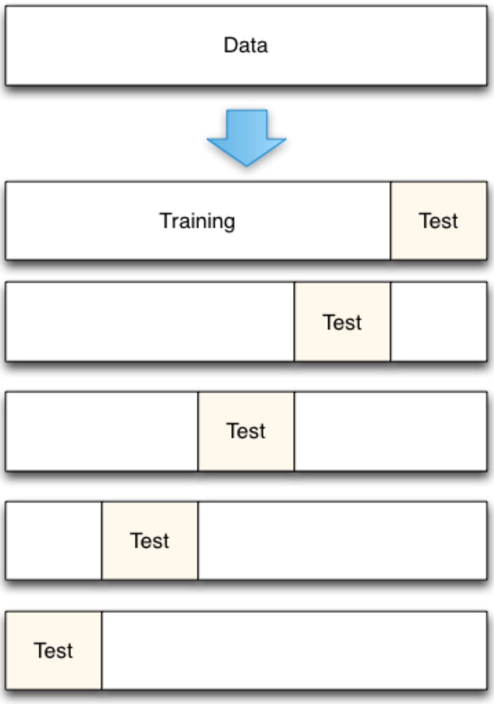

FIGURE 3.5: K-fold cross validation scheme [source: http://goo.gl/Ww8dAg]

The error can be calculated using the following equation, which gives the mean error of the K tests

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \tag{3.15}$$

where E_i is the error obtained when using fold i to test the model.

3.2.4 Performance Metrics

To help us comparing the results of an algorithm against another one some metrics were calculated. These metrics are Accuracy, Sensitivity, Specificity and F1 Score and they are described bellow. Classified examples are categorized as True Positive (TP), True Negative (TN), False Positive (FP) or False Negative (FN) depending on the classification label and the true label.

3.2.4.1 Accuracy

The Accuracy (Acc) is a metric that gives the proportion of corrected labelled samples by the model that it is being tested and is calculated with Equation 3.16.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.16}$$

3.2.4.2 Sensitivity / Recall (r)

Sensitivity (Sens) measures the percentage of actual positives which are classified as such. The equation bellow is used to determine its value.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.17}$$

3.2.4.3 Specificity

Specificity (Spec) is the volume of negatives that are identified as being negatives.

$$Specificity = \frac{TN}{TN + FP} \tag{3.18}$$

3.2.4.4 Precision

Precision (p) is the number of positive predictions divided by the total number of positive class values predicted (Equation 3.19).

$$p = \frac{TP}{TP + FP} \tag{3.19}$$

3.2.4.5 F1 Score

F1 Score can be seen as a balance between precision and recall (Equation 3.20).

$$F1 = 2 \times \frac{p \times r}{p + r} \tag{3.20}$$

when $\beta = 1$ on the general formula:

$$F_\beta = (1 + \beta^2) \times \frac{p \times r}{(\beta^2 \times p) + r} \tag{3.21}$$

If $\beta > 1$ we put more emphasis on recall than on precision. However, if $\beta < 1$ it weights precision higher then recall.

3.3 Computational Experiments

In the present work, as mentioned earlier, several algorithms were implemented to develop models capable of distinguishing non-diabetic people from people with the disease. This section aims to present the results obtained with k-fold cross validation ($k = 10$). In order to have statistically reliable results, 30 runs were performed on every test.

3.3.1 Missing values methods' results

Previously, several methods to deal with missing values were presented. Due to the fact that each method has its own advantages and disadvantages, several tests were made to check which one is the best. The results obtained are given in Table 3.2. The parameters used in the algorithms were selected based on preliminary experiments and may not be the best ones.

	LDA	KNN	SVM	Random Forest
Random Value	75.95 ± 1.19	76.53 ± 1.27	78.13 ± 1.15	77.26 ± 1.55
Average of the KNN	74.47 ± 1.11	75.27 ± 0.98	76.00 ± 1.01	75.65 ± 1.23
Feature's class median	74.30 ± 1.15	75.71 ± 1.04	76.06 ± 1.32	75.53 ± 1.38
Feature's class mean	76.58 ± 0.96	79.98 ± 1.02	80.85 ± 1.05	87.38 ± 0.94

TABLE 3.2: Missing values methods' results in terms of accuracy (%)

Each value represents the accuracy obtained considering the algorithm and the method to fix missing values. As it is possible to see, the method where the missing values are substituted by the feature's mean considering the class of the example with the missing value, has better results in all algorithms when compared to the others. Given this fact, the feature's class mean was the selected method to deal with the missing values.

3.3.2 Algorithm parameter selection

When dealing with parametric algorithms, it is important to choose the best ones to maximize the model's performance and consequently improve the results. In the next sections the tests made to discover those parameters are explained.

3.3.2.1 K-NN

The K-NN classifier considers the k closer neighbours calculated by a certain function to classify a new example of being of one class or another. Since that parameter is important, in Figure 3.6 the results with different possible values are shown. The selected range of values chosen was [1,25].

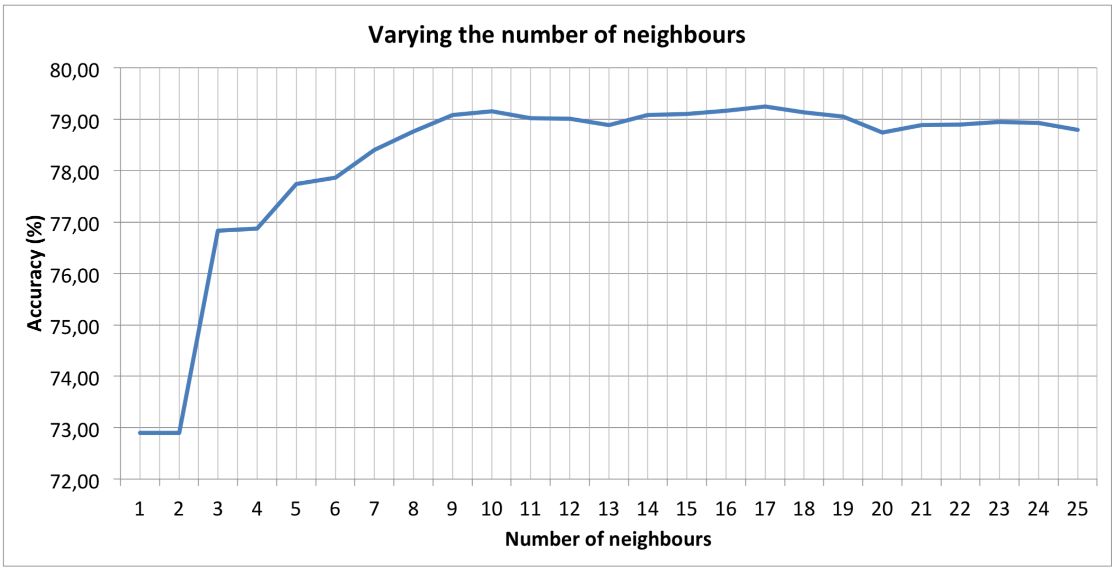

Figure 3.6: Selection of the number of neighbours

The number of neighbours that yielded the best result was 10 with an accuracy of 79.16%.

3.3.2.2 SVM

Regarding the SVM, a grid search was performed to test each pair (C, γ) where C controls the trade-off between margin maximization and error minimization and γ defines the width of the Gaussian. The results are given in a heatmap image (see Figure 3.7).

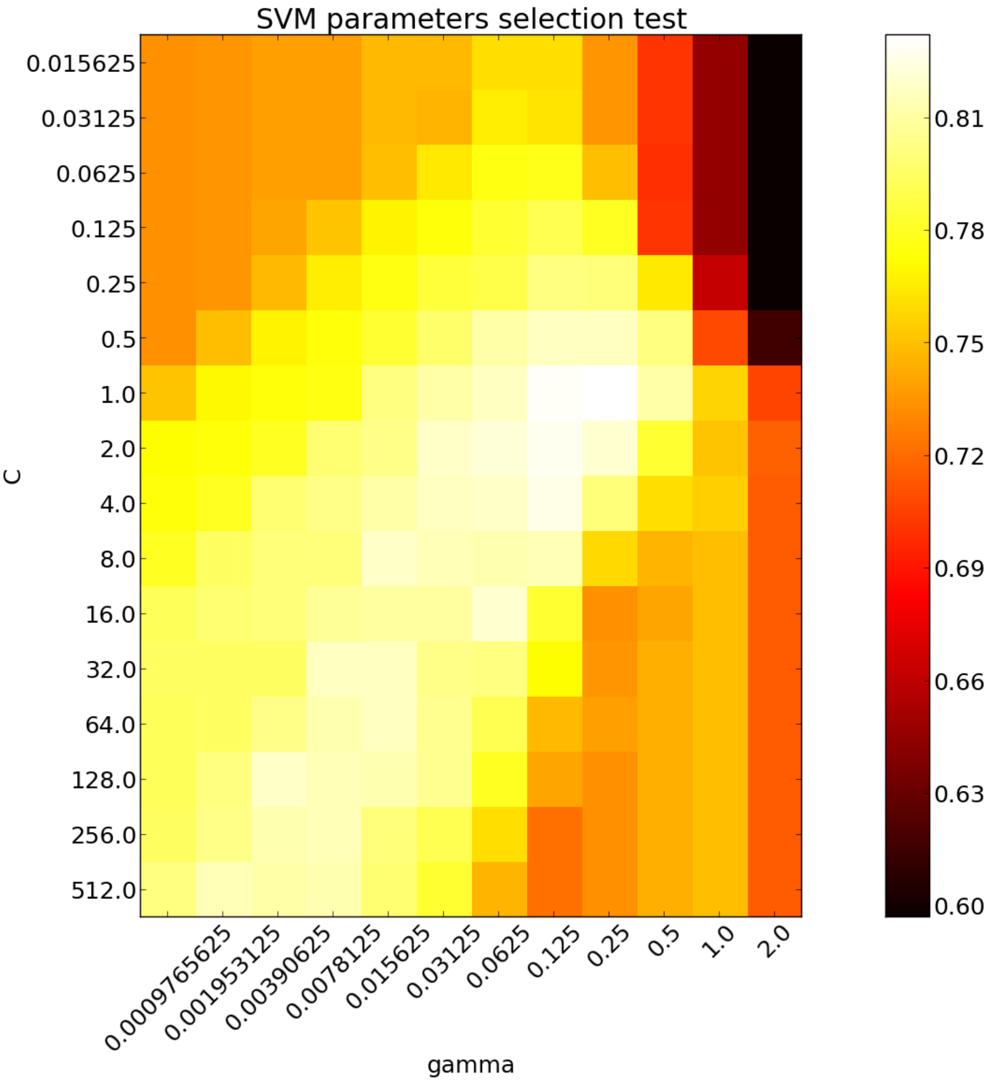

FIGURE 3.7: Selection of the SVM parameters

The best parameters are the ones where the square is whiter which corresponds to $C = 1.0$ and $\gamma = 0.25$.

3.3.2.3 Random Forest

Since that Random Forest is an ensemble of trees, there is the need to choose the proper number of trees for a certain problem. Like it was done for the other two algorithms, several numbers of trees were tested to check the best quantity. In Figure 3.8 is it possible to see the variation of the accuracy as the number of trees grows.

From the chart, it is possible to see that 800 trees gave the best result. For this reason, we will consider this value as the appropriate one for this case.

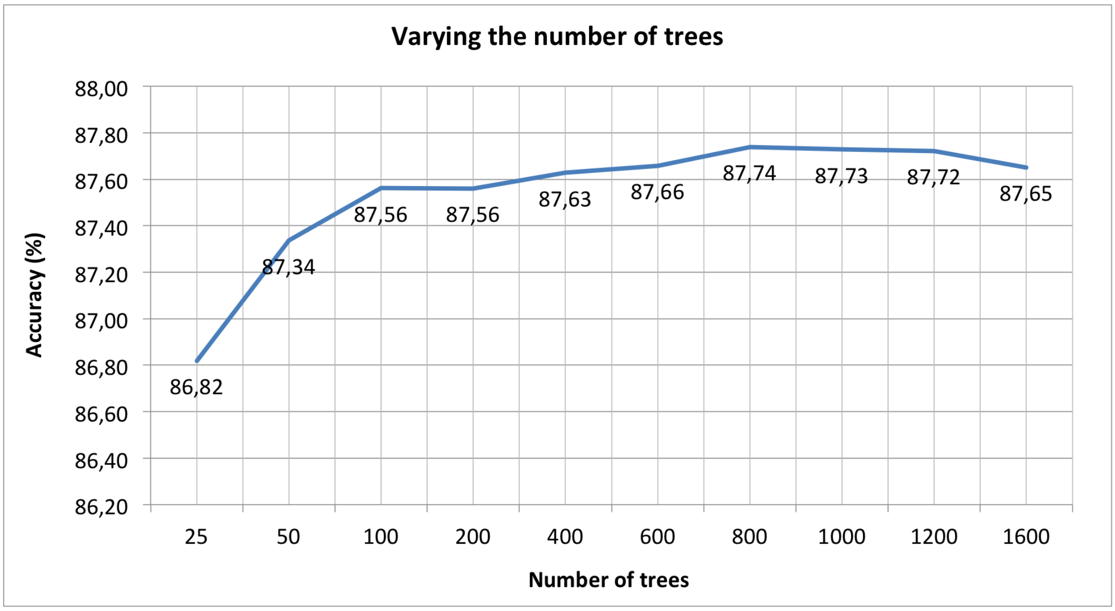

FIGURE 3.8: Selection of the number of trees

3.3.3 Results

After the selection of the best method to treat missing values and the most suitable parameters for the algorithms we performed a final experiment taking those results into consideration. The outcome is shown in Table 3.3.

	Accuracy	Specificity	Sensitivity	F1 Score
LDA	76.86 ± 0.84	81.70 ± 1.53	72.02 ± 0.82	75.55 ± 0.79
KNN	79.24 ± 1.09	82.19 ± 1.91	76.30 ± 1.36	78.53 ± 1.08
SVM	80.85 ± 0.95	79.34 ± 1.60	81.67 ± 1.32	80.68 ± 0.92
Random Forest	87.66 ± 1.16	88.05 ± 1.50	87.28 ± 1.16	87.59 ± 1.16

TABLE 3.3: Diabetes diagnosis final results (%)

In the overall, all the methods have exceeded the baseline method (LDA) in more than 2% for accuracy and F1 score. Considering the Random Forest algorithm with which we built the model that yielded the best results, the accuracy was 87.66%. By inspecting the specificity result, we may say that, for example, in a sample of 100 people, 88 of them would be correctly identified has not having diabetes. In respect to the sensibility i.e., the number of people who had been properly classified as having diabetes for the same sample we would have 87.

When comparing the algorithms, the obtained results match the expectations. The LDA, which tries to find a linear relationship between features and the labels, got the worse outcomes. In relation to KNN and SVM, although SVM performed better, the KNN results are similar. This may lead us to conclude that the dataset examples can not be completely divided correctly. Otherwise, the SVM algorithm would have found a barrier that would separate both classes. For this reason, K-NN has an identical performance due to the way it works. By considering the k nearest neighbours, the algorithm works relatively well on this kind of datasets since it may find close neighbours of the correct class.

Random Forest with its many trees adapted to the dataset. In other words, since every tree is built based on a randomly with replacement sample of examples, each one of them can precisely characterize part of the dataset. Also, since their output is given by the majority (each tree contributes with one vote) they usually achieve good results, which is the case.

3.4 Discussion and Conclusions

Our best results were achieved using the Random Forest algorithm, which proved to be quite efficient distinguishing diabetic people from non-diabetic people when compared to the other ones. In medicine, the sensitivity of the models that test the possibility of a patient having or not a certain disease, the presence of a substance, etc. is extremely important. The reason behind that fact is that it is not acceptable that a patient is misdiagnosed which can lead him to a more problematic clinical status or even death. Presenting a sensitivity of 87% and an accuracy of almost 88% we may say that our model is among the best results for this dataset.

In terms of the results listed on the state of the art, the model built on this work has a better performance except for the one presented by Polat and Güneş [8]. They used PCA along with adaptive neuro-fuzzy inference system (ANFIS) and achieved an accuracy of 89.47%. The PCA was used to reduce the number of features to 4 while the ANFIS learned the relationship between features and labels. The fact that they got better results does not reside on the application of the PCA, since that using it along with the Random Forest algorithm, it decreased our accuracy to 70.39±1.78% which also happened to the other ones. Thus, we consider that the ANFIS is responsible for the increase of accuracy. ANFIS is a combination of a fuzzy inference system and artificial neural network, which mimics the human ability to make decisions in an environment of uncertainty. In this fact, it may relay the reason of why they perform better than our model.

For the other results presented on the SOTA, our model performs better. Wu et al. [6] and Polat and Arslan [9] used modified versions of the SVM algorithm accomplishing 82.29% and 79.16% of accuracy, respectively. To select the best set of features, Kumar et al. [7] resorted on a genetic algorithm and on SVM to test each set. They reported an accuracy of 77%.

Despite the accuracy of the models, an important fact is that none of the previous analysed works mention that the dataset contains missing values. Besides, there is no information about how the dataset was handled in regard to the problem of being unbalanced. In this work we introduced some ways to deal with this problem, which lead us to build a model with good performance.

Chapter 4

Glucose level prediction

Patients with diabetes should constantly check their glucose levels to prevent from entering in hypoglycaemia and hyperglycaemia states. These states are associated with short and long-term problems that should obviously be avoided. Due to the natural evolution of technology, it is now possible to use small devices to record every 5 minutes the glucose level.

In this chapter, we describe the development of a model that predicts the future glucose levels with different intervals. Such a model can prevent patients from getting abnormal glucose levels since if we predict that the glucose level will drop beneath a certain level, we can warn the individual and he can take pre-emptive actions like eating a sandwich to avoid such situation.

4.1 Datasets

In this section, the datasets used in this part of the work are described. As in the diabetes diagnosis problem, we need a dataset to create the model described earlier. However, since this is a time series problem the datasets used here have a special characteristic that is the records have a temporal relation between them.

4.1.1 CGM dataset

A dataset covering approximately 6 and a half months can be obtained from [36]. It contains information about a 14 years old patient's subcutaneous glucose level.

The advantage of this dataset is that it records data in a quasi-continuous way, during some periods of time, which allows having a better insight of the variation of the glucose level. Thus, it will for sure make better predictions in the future. On some other periods of time, there are no samples, which may be due to the lack of battery.

The lack of information about if, when and how much insulin was taken by the patient and also the kind of diet he followed (e.g. carbohydrates) might prevent of developing a system that makes good predictions.

4.1.2 APDP datasets

After the presentation of this project along with some experimental results, on a meeting with a member of Associação Protectora dos Diabéticos de Portugal (APDP), it was given to us 2 datasets to check how our model would behave with those datasets.

Both datasets keeps records of the subcutaneous glucose level variation for almost 7 days with a sampling time of 5 minutes. Each dataset is identified by code in order to maintain the subjects' privacy, which is AR and AML1.

4.1.3 Recent datasets

Besides the 2 datasets described in the last section, there are 2 more. These datasets were also provided by Associação Protectora dos Diabéticos de Portugal since they were enthusiastic about the results that we were giving and discussing that the datasets that are described next were provided recently.

Within each one of the datasets, the glucose variation for approximately 7 days of 6 people was recorded. Given that there is the necessity for the data to anonymous to maintain people's privacy, the name of the files were codified as AC1 and RR.

4.2 Model Development

In order to create a model capable of predicting the glucose level ahead in time, there is the need to follow certain steps that begins with data pre-processing. After that, one must provide the data to the algorithm with the aim to build a model that has its performance evaluated by some metrics.

In the following sections, the steps listed before are better explained.

4.2.1 Data pre-processing

Data pre-processing is not just important in classification problems. When dealing with time series datasets, some underlying issues need to be checked.

Regarding the CGM dataset, we verified that it was not complete, i.e., there were gaps (missing values), which does not happen with the APDP datasets. To solve this problem, we analysed all data and found the biggest interval of contiguous values. It has 862 readings that correspond to almost 3 days.

For every dataset, data was normalized between zero and one before it is used with any of the algorithms. This was done with min-max normalization technique already described in section 3.2.1.3. In this case, it is possible to use this method since glucose level may oscillate between 40 and 600. However, to make sure that no new value would appear, that range was enlarged setting the minimum possible value to 20 and the maximum to 800.

After normalizing data and to use it with algorithms, there is the need to organize data in the proper format. Considering that we only have information about the glucose level, the model will predict future values based on past glucose measurements. Considering $z = f(t)$, which represents the glucose level at instance t, the value is mapped to the glucose level at target time $y = f(t + PH)$, as explained earlier in Section 2.2.1 making our dataset $D = \{(z_1, y_1), (z_2, y_2), ..., (z_N, y_N)\}$. Although it is possible to train a model using pairs of $(\mathbf{z}_i, y_i)$, it is expected that if we include more information, the model will yield better results. In other words, instead of relying just on the present glucose level one might give more past values. Given that we extended each pattern with W past measurements, $\mathbf{z}$ is now $\{f(t), f(t-1), f(t-2), ..., f(t-W)\}$. For this reason, we transformed the original dataset into another datasets taking into account the different values of W and PH.

Finally, data is divided into a training set and a testing set. The first one corresponds to 70% of all data while the second one corresponds to the remaining 30%.

4.2.2 Algorithms for Glucose Level Prediction

When developing a model, it is important to test several algorithms for the purpose of checking which one builds the better one. Given this, in the next sections some algorithms that are used on this work are explained.

4.2.2.1 Naïve method

This method predicts a future value based on the last known value. Considering a time series with T observations $\{z_1, z_2, ..., z_T\}$ and would like to predict a value for instant $T + 1$, with this method $\widehat{z}_{T+1} = z_T$.

Naïve method is used to establish a baseline performance. All the developed models will be compared with this one to check if the more complex methods do better than this approach.

4.2.2.2 Linear Regression

Linear Regression is an algorithm that tries to find a straight line that best fits the data. If there is just an independent variable to use in the regression, it is considered a simple regression. However, if there are two or more it is called a multiple regression.

In order to find the line that best fits the data, it is uses a procedure named least-squared method in which the interest is in reducing the sum of the squared differences between each data point and the line. By doing so, the best line is the one that has the least error.

In terms of its equation, it has the standard one, $y = mx + b$, where y is the predicted value, b is the bias, x is the input value and m is the slope.

4.2.2.3 Support Vector Regression

Support Vector Regression (SVR) introduced in [37] is a specific case of the SVM as it keeps the same principles with some minor differences. Usually in a regression problem, the output is a scalar ($y \in \mathbb{R}$), which makes harder the task of forecasting since there can be lots of possible solutions.

In ε-SVR, the goal is to create a function f that has at most ε deviation between the actual value and the obtained one for all samples in the training data and is as flat as possible (see Figure 4.1). In another words, it is created a margin of tolerance where if errors are less than ε, they are not taken into consideration. This cost function can be formalized as shown in Equation 4.1.

$$\mid y - f(\mathbf{x}) \mid_\varepsilon = \begin{cases} 0 & if \mid y - f(\mathbf{x}) \mid < \varepsilon \\ \mid y - f(\mathbf{x}) \mid - \varepsilon & otherwise \end{cases} \tag{4.1}$$

Given that we want to find

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b \tag{4.2}$$

Written as a constrained optimization problem, this reads

$$\begin{aligned} \text{minimizes} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & ((\mathbf{w}\cdot\mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i \\ & y_i - ((\mathbf{w}\cdot\mathbf{x}_i) + b) \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \\ & i = 1, ..., N \end{aligned} \tag{4.3}$$

where ξ_i and ξ_i^* are slack variables,$\|\cdot\|$ denotes the norm and C is a regularization constant. If $C > 0$, we will have a model that puts more emphasis on minimizing the slack variables, i.e. a model that tries to best fit the training data losing generalization ability.

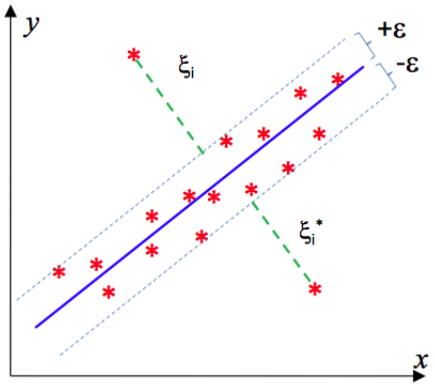

FIGURE 4.1: SVR margin [source: http://goo.gl/TLbv6N]

In order to perform non-linear regression, kernel functions are used to map the features into a higher dimensional space. Using Lagrange multipliers, the optimization problem is now the following one

$$\begin{aligned}
\text{maximize} \quad & W(\alpha, \alpha^*) = -\epsilon \sum_{i=1}^{n} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) y_i \\
& - \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i^* - \alpha_i)(\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}_j) \\
\text{subject to} \quad & 0 \leq \alpha_i, \alpha_i^* \leq C \\
& \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0 \\
& i = 1, ..., n
\end{aligned} \tag{4.4}$$

The regression estimate takes the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) k(\mathbf{x}_i, \mathbf{x}_j) + b \tag{4.5}$$

4.2.2.4 Generalized Regression Neural Network

Generalized Regression Neural Network (GRNN) was proposed by Specht [38] and is often used to approximate any arbitrary function between input and output vectors, drawing the function estimate directly from the training data. This is a one-pass learning algorithm, which means that it does not have a training phase like the feedforward back-propagation method. The GRNN can be viewed as the normalized Radial Basis Function

(RBF) network in which there is a unit centred at every training case. At the structure level, it consists on 4 layers: Input layer, Pattern layer, Summation Layer and Output layer. In Figure 4.2 it is shown its structure.

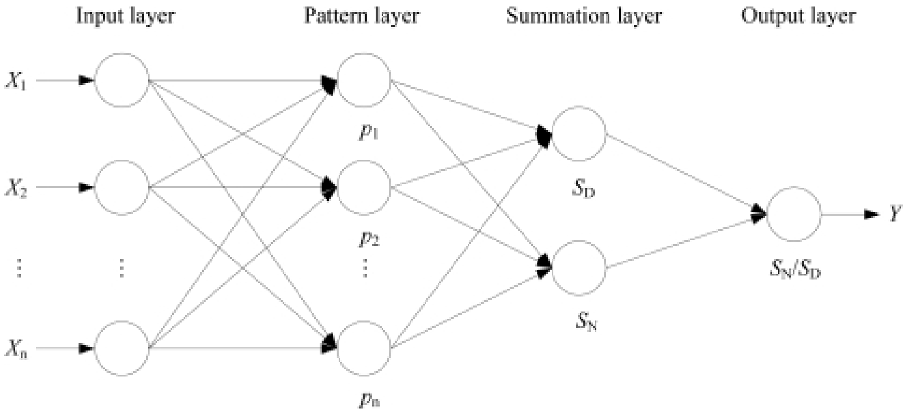

FIGURE 4.2: Structure of Generalized Regression Neural Network [source: http://goo.gl/fCY06I]

The input layer is responsible for the reception of all variables in the input vector x meaning that there is a unique neuron for each predictor variable. These neurons just pass the data into the pattern layer without processing it. In the second layer, a RBF neuron is used to process data in a way that the relationship between an input and the output is "memorized". This means that there is the same number of neurons in the pattern layer as the number of input vectors. In each unit a Gaussian probability density function (pdf) is applied to the network input such that

$$\theta = e^{-\frac{(\mathbf{x}-U_i)'(\mathbf{x}-U_i)}{2\sigma^2}} \tag{4.6}$$

where $\mathbf{x}$ is the input vector of predictor variables, U_i is the specific training vector represented by pattern neuron i and σ is the smoothing parameter or spread. In other words, θ represents the distance of the input vector to the stored pattern. Using the Gaussian kernel function, the smoothing parameter is the only parameter that needs to be manually inserted as Equation 4.6 shows. Its value determines the shape of the pdf; that is, how quickly the function declines as the distance increase from the point. The larger the value is, the smoother the function is and greater influence the distant points have.

After the computation of the output of each neuron, these results are forwarded to the summation layer where they are combined by two summation neurons. These processing units computes the simple arithmetic summation and the weighted summation as it can be seen in Equation 4.7 and Equation 4.8, respectively.

$$S_D = \sum_{i=1} \theta_i \tag{4.7}$$

$$S_N = \sum_{i=1} w_i \theta_i \tag{4.8}$$

Finally, the outputs of the summation neurons are sent to the output layer, where the output neuron performs the following division to yield the predicted value of the output variable

$$Y = \frac{S_N}{S_D} \tag{4.9}$$

4.2.2.5 Random Forest for Regression

The Random Forest algorithm for regression also presented by the author of the same algorithm for classification (Breiman [34]), is formed of B trees each of which is created based on a randomly sampled examples with replacement as described in 3.2.2.4.

The main difference is that the output values are numerical ($\widehat{z} \in \mathbb{R}$) instead of class labels. The prediction given by this algorithm is calculated by taking the average over the B trees. Mathematically, this can be represented by

$$\widehat{z} = \frac{1}{B} \sum_{i=1}^{B} f_i(\mathbf{z}) \tag{4.10}$$

where f_i represents the i-th tree and $\mathbf{z}$ is the input vector.

4.2.3 Prediction Methods

To predict the glucose levels on a individual there are some multi-step-ahead prediction techniques, that given a time series $\{z_1, z_2, ..., z_N\}$ composed of N observations consist on predicting $\{z_{N+1}, z_{N+2}, ..., z_{N+PH}\}$, where PH is the prediction horizon ($PH \geq 1$).

In this section, the two prediction strategies [39–41] used in this work are described: iterative prediction method and direct prediction method.

4.2.3.1 Iterative Prediction method

The most intuitive and simple method to predict values several steps ahead is the iterative prediction one. In this method, once a one-step-ahead prediction is computed, the value is given again as an input to the next step following a recursive strategy.

The one-step ahead prediction model depending on W past points has the form

$$\widehat{z}_{t+1} = f(z_t, z_{t-1}, ..., z_{t-W+1}) \tag{4.11}$$

After the value of $\widehat{z}_{t+1}$ has been computed it is used to predict $\widehat{z}_{t+2}$ and so on until the desired PH. In general, the estimation of the PH next values is returned by

$$\begin{cases} \widehat{z}_{t+1} = f(z_t, z_{t-1}, ..., z_{t-W+1}) \\ \widehat{z}_{t+2} = f(\widehat{z}_{t+1}, z_t, ..., z_{t-W+2}) \\ ... \\ \widehat{z}_{t+PH} = f(\widehat{z}_{t+PH-1}, \widehat{z}_{t+PH-2}, ..., \widehat{z}_{t+PH-W}) \end{cases} \tag{4.12}$$

From Equations 4.12 it is possible to conclude that to predict the value for $\widehat{z}_{t+PH}$ it requires PH one-step ahead predictions.

4.2.3.2 Direct Prediction method

An alternative to the iterative method is the direct prediction method which for PH-steps ahead the value is given by

$$\widehat{z}_{t+PH} = f_{ph}(z_t, z_{t-1}, ..., z_{t-W+1}) \tag{4.13}$$

where $1 \leq ph \leq PH$ and f_{ph} is a model that predicts directly the value at instant $t+ph$ without any predicted values. In general, the prediction of the PH next values is returned by

$$\begin{cases} \widehat{z}_{t+1} = f_1(z_t, z_{t-1}, ..., z_{t-W+1}) \\ \widehat{z}_{t+2} = f_2(z_t, z_{t-1}, ..., z_{t-W+1}) \\ ... \\ \widehat{z}_{t+PH} = f_{PH}(z_t, z_{t-1}, ..., z_{t-W+1}) \end{cases} \tag{4.14}$$

To predict all the values from $\widehat{z}_{t+1}$ to $\widehat{z}_{t+PH}$, PH models need to be built. However, it is possible to use just one of those models accordingly to the chosen PH, if the model gets new data with the same interval as it needs to predict. For example, considering that we are on instant t and $PH = 2$ with a sample time of 5 minutes, we would compute $\widehat{z}_{t+2}$ with $f_2(z_t, z_{t-1}, ..., z_{t-W+1})$. After 5 minutes that corresponds to instant $t+1$ the value of z_{t+1} on instant t is available which now becomes the value of z_t and can be used to predict $\widehat{z}_{t+3}$ with f_2. This strategy was chosen as it implies to build less models and consequently it demands less computational power and takes less time to train.

4.2.4 Evaluation Technique

When developing a model, there is the necessity of choosing the best algorithms' parameters and evaluate its performance. Since data is related to each other by the time factor, we have to have a proper method to do it. Firstly, the data is divided into a train and test set. The earlier one is used to build the model while the other one is used to evaluate it.

Walk-forward test is an evaluation technique that divides data into chunks and then trains and tests the model with them. For finalcial This method was employed by Cao and Tay [42] for financial time series forecasting. In Figure 4.3 it is possible to see a diagram of how the method works.

The train data is divided into equal chunks and then the first 2 are used to train the model. The following one is used to test the model's performance based on the Root Mean Squared Error (RMSE), which is described in the next section. Once the test is

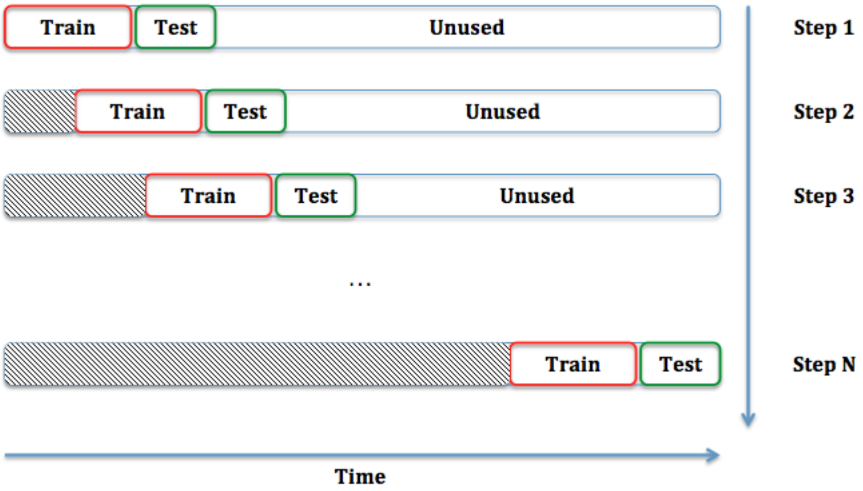

FIGURE 4.3: Scheme of the walking-forward test

complete for that step, we move forward the training and testing window and repeat the test again with the new data.

The final RMSE is given by the following equation

$$Overall\_RMSE = \frac{1}{N}\sum_{i=1}^{N} RMSE_{step\_i} \tag{4.15}$$

When choosing the number of chunks into which we want to divide the data and also the number of blocks to train and test the model, it is essential to keep in mind the dataset size. If the data is divided into a great number of chunks, there will be few records in every one. However, if the number is small, there is going to be a small number of blocks, but several records will compose each one.

An alternative to this method is not moving the training window and including in that set, the testing one. In other words, after we complete the first step we would include in the training set, the set that we used to test the model on that stage. This way, we would increment the training set at each step. However, this approach would give more emphasis on the older data than in the new one since we would train with the first chunk N times, the second one $N-1$ times and so on. For this reason we took the first option.

4.2.5 Performance Metric

In order to check and compare the performance of the generated models using the above algorithms, we have to calculate a metric to do so.

Root Mean Squared Error represents the standard deviation of the differences between the real values and the observed ones. It can me measured by the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}{N}} \tag{4.16}$$

This metric has the advantage that it is an easy interpretable statistic since it has the same units as the data plotted on the vertical axis. RMSE is a good measure of how accurately the model predicts the response and the smaller the value is, the better the model.

4.3 Computational Experiments

In order to build a good model capable of predicting the glucose level of a patient ahead in time, several algorithms were implemented. This section aims to present the tests performed taking into account different prediction horizons, the different algorithms, the two prediction methods and also the number of predictors. The results were obtained from the test set that was created upon the division the dataset into train and test sets. Given that, there are a considerable number of tests for each algorithm, we only show here a compilation of the best results. To see them all, please check Appendix B.

The number of predictors used in each test is presented in W's column. The values represent the number of past points ($\mathbf{z} = \{z_1, z_2, ..., z_W\}$) considered to map the output (z_{t+PH}). For instance, if $W = 4$ then we are considering $\mathbf{z} = \{z_t, z_{t-1}, z_{t-2}, z_{t-3}\}$. To check if the RMSE drops with the addition of more information, its value is incremented by 3 which means that the information about 15 more minutes (3 readings) is being added.

In the following subsections the results for each dataset are presented and discussed.

4.3.1 CGM file

With the purpose of comparing the results of more complex algorithm, we used a naïve method to establish a baseline. The results of all algorithms are presented in Table 4.1.

The first conclusion that it is possible to take is that Linear Regression yielded better results for long-term predictions ($PH >= 30$ min) while E-SVR with linear kernel got better outcomes for lower values of PH. Although, there are some differences in the outcomes of both algorithms they are not significant as those differences are at most of 0.28 mg/dL.

There is a linear relationship between the predictor and the output since both Linear Regression and E-SVR with linear kernel provide good results. Supposedly, E-SVR with RBF kernel ought to perform better than the one with linear kernel yet it does not happen, which reinforces the theory of a linear relationship. In terms of the other two algorithms - GRNN and Random Forest - their models were are not as good as the former ones being even worse than the naïve method.

Regarding the number of predictors, it does not grow as the value of PH increases as it would be expectable. However, for $PH = 45$ min and $PH = 60$ min, its values are greater than for the others since the algorithms need to capture the glucose fluctuation over more time to have a higher accuracy when making the prediction. In other words, if more past glucose readings are added to the vector to map a future outcome then, when introducing a new vector that has a similar variation of the glucose, its output will be identical.

In terms of the prediction method, as it is possible to see, the iterative one was better than the direct. Despite that, the differences in the RMSE values are very small being at most approximately 1 mg/dL, which has a residual effect on the final predicted value and does not affect the patient. For this reason, we opt for the direct prediction method as with it, less computational calculations are needed and consequently it requires less time to train and test.

Taking in consideration, for example, the Linear Regression with the direct prediction method, it is possible to see in Figure 4.4 the differences between the predicted values and the real ones. Ideally, both signals should be overlapping, which means that the RMSE would be zero and would be no error predicting future glucose levels. In this case,

Algorithm	Prediction Method	PH									
		10 min		20 min		30 min		45 min		60 min	
		W	RMSE	W	RMSE	W	RMSE	W	RMSE	W	RMSE
Naive	—	49	7.43	49	13.40	49	17.82	49	21.53	49	22.75
Linear Regression	Direct	34	5.10	19	11.03	19	16.08	16	20.87	40	21.92
	Iterative	34	5.10	19	11.03	19	**16.00**	37	**20.71**	43	**21.58**
E-SVR (Linear kernel)	Direct	37	**4.95**	28	11.08	28	16.63	19	21.68	34	22.02
	Iterative	37	4.98	28	**10.97**	31	16.11	40	20.99	40	21.65
E-SVR (RBF kernel)	Direct	19	5.05	19	11.22	28	16.55	19	21.28	34	21.97
	Iterative	22	5.08	31	11.12	28	16.44	34	21.19	46	21.70
GRNN	Direct	4	11.15	4	16.43	4	20.28	49	23.02	49	23.16
	Iterative	4	9.28	4	16.37	7	20.99	25	23.03	25	24.00
Random Forest	Direct	4	7.92	4	15.11	4	20.82	49	27.41	16	29.57
	Iterative	4	7.91	4	15.15	4	20.74	40	25.34	40	25.23

TABLE 4.1: Best results on each algorithm for the CGM file - RMSE in mg/dL

it is possible to observe that the predicted values are lagged when considering the real ones. Although, several numbers of predictors were tested to check if it was possible to eliminate that lag (see Appendix B), this is the best outcome that we had. Despite that, the model is capable of following the variation of the glucose level with an acceptable RMSE.

4.3.2 AML1 and RA file

The results for the AML1 file exposed in Table 4.2 show that the Linear Regression model with the iterative prediction method was the best among the others for every value of PH. The same happens with the RA file, as it is possible to check in Table 4.3.

In both cases, the more complex algorithms outperformed the naïve method, as it was expected. Once again, like in the case of the CGM file, the existence of a linear relationship, between past glucose readings and future values, was proved since the results of the linear regression model and E-SVR with linear kernel are similar. However, in these two datasets, the E-SVR with RBF kernel achieved better results than using the linear kernel. When it comes to the GRNN and Random Forest, the models built with these two algorithms are not capable of modelling the referred relationship.

Referring to the number of predictors, it does not grow when considering bigger values of PH like what happened with the CGM file. For the AML1 file, W has the value of 25 except for $PH = 10$ min where it is equal to 10 and, for the RA file, the best value is 13 for $PH = 10$ min and 7 for the rest. This fact leads us to conclude that the RA patient has its glucose levels more controlled than the AML1 one, i.e., there are not so many glucose variations and the ones that exist are among a smaller range.

Comparing the results of the prediction methods on AML1 file, despite the fact that the iterative method yielded better results, the differences between that method and the direct one are at most 0.51 mg/dL, which is insignificant. This fact leads us to choose the linear regression model with the direct prediction method as the best option for this file.

In Figure 4.5 and Figure 4.6, it is possible to observe the real variation of glucose level and the predicted on for the AML1 and RA file, respectively. These graphics were obtained using the linear regression model with the direct prediction method, considering a 30

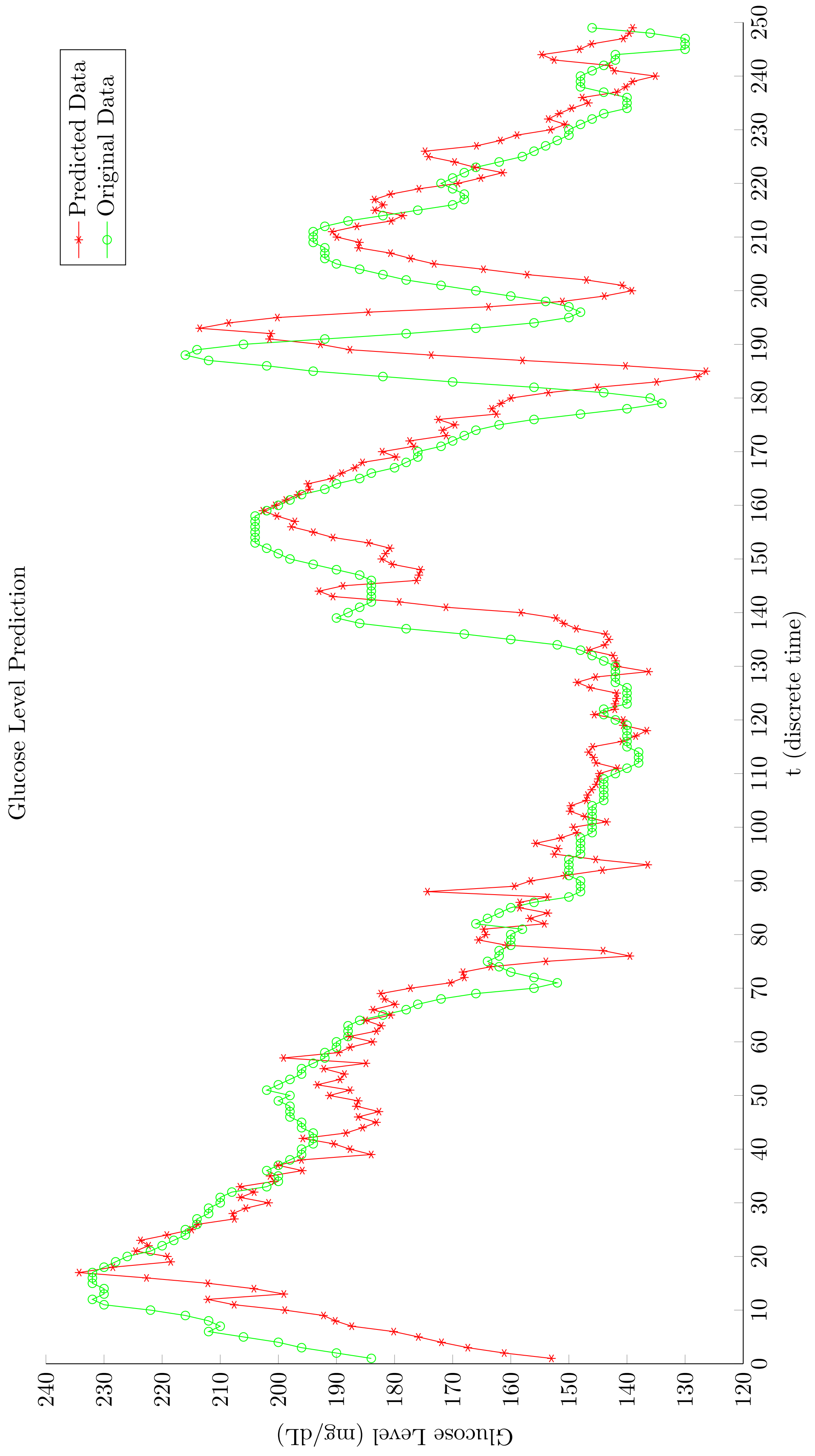

FIGURE 4.4: Prediction of glucose levels 30 minutes ahead using Linear Regression with direct prediction method - CGM file

Algorithm	Prediction Method	PH									
		10 min		20 min		30 min		45 min		60 min	
		W	RMSE	W	RMSE	W	RMSE	W	RMSE	W	RMSE
Naive	—	4	10.86	4	20.93	4	30.28	4	42.61	4	53.06
Linear Regression	Direct	10	**5.11**	22	11.32	22	19.06	19	30.45	19	40.03
	Iterative	10	**5.11**	25	**11.31**	25	**18.96**	25	**30.20**	25	**39.52**
E-SVR (Linear kernel)	Direct	22	5.53	22	12.02	19	19.86	4	31.88	16	41.85
	Iterative	19	5.62	31	11.94	46	19.72	22	31.30	22	41.40
E-SVR (RBF kernel)	Direct	25	5.35	19	11.61	10	19.67	7	31.60	7	41.43
	Iterative	49	5.51	22	11.89	19	19.82	19	32.18	22	43.15
GRNN	Direct	4	8.95	4	16.98	4	25.06	4	36.47	4	46.10
	Iterative	4	9.69	4	19.64	4	29.57	4	42.63	4	53.10
Random Forest	Direct	4	7.36	4	15.72	4	24.12	4	35.98	4	46.57
	Iterative	4	7.60	4	16.10	4	25.39	7	38.84	10	50.41

TABLE 4.2: Best results on each algorithm for the AML1 file - RMSE in mg/dL

Algorithm	Prediction Method	PH									
		10 min		20 min		30 min		45 min		60 min	
		W	RMSE	W	RMSE	W	RMSE	W	RMSE	W	RMSE
Naive	—	4	4.35	4	7.61	4	9.80	4	10.95	4	10.47
Linear Regression	Direct	13	**3.06**	13	5.63	13	7.61	10	8.55	7	8.43
	Iterative	13	**3.06**	7	**5.62**	7	**7.57**	7	**8.49**	7	**8.40**
E-SVR (Linear kernel)	Direct	22	3.28	13	6.00	13	7.94	10	8.92	10	8.72
	Iterative	22	3.27	13	5.95	16	7.85	34	8.84	7	8.94
E-SVR (RBF kernel)	Direct	10	3.25	7	5.99	13	7.83	10	8.76	10	8.64
	Iterative	49	3.27	31	5.93	16	8.11	46	8.98	46	8.96
GRNN	Direct	4	3.87	4	6.92	16	8.49	13	8.64	10	8.59
	Iterative	4	4.19	4	7.34	22	8.81	25	9.04	22	9.18
Random Forest	Direct	4	3.55	7	6.60	7	8.74	43	9.57	40	9.63
	Iterative	4	3.58	10	6.79	13	9.23	13	10.32	13	10.33

TABLE 4.3: Best results on each algorithm for the RA file - RMSE in mg/dL

minutes ahead prediction. When looking at the result of the AML1 file, it seems that the model behaves better than with the RA file. Even though both signals are almost overlapped, it is necessary to pay attention to the glucose level axis as in the first file the glucose fluctuation varies between a wider range of values than in the second one. This is the reason why the RMSE is higher in the AML1 file despite the closeness of the signals.

4.3.3 Recent datasets

Due to the impressive results achieved with the AML1 and RA dataset, two more were provided recently as detailed in 4.1.3. Despite that, we performed some tests using the linear regression algorithm with the direct prediction method that was the best combination as it is explained in the next section.

In Table 4.4, the results obtained are presented. As it is possible to observe, the model has an excellent performance even when doing long-term predictions.

Algorithm	PH									
	10 min		20 min		30 min		45 min		60 min	
	W	RMSE	W	RMSE	W	RMSE	W	RMSE	W	RMSE
AC1	31	5.99	31	11.17	31	16.64	31	23.49	31	28.42
RR	7	3.16	7	5.97	7	8.39	7	10.85	7	12.20

Table 4.4: Best results on each file with Linear Regression method with direct prediction - RMSE in mg/dL

4.4 Discussion and Conclusions

From the performed computational experiments, we conclude that the best model capable of predicting future glucose levels is the one built using linear regression. A model with similar performance was obtained with E-SVR.

In terms of the prediction method as it was already observed, the direct one has more advantages over the iterative one. Besides, it requires less computational calculation, which by turn reduces the time needed to train and test and in theory is less subject to error. When predicting in an iterative way, since each prediction has an associated

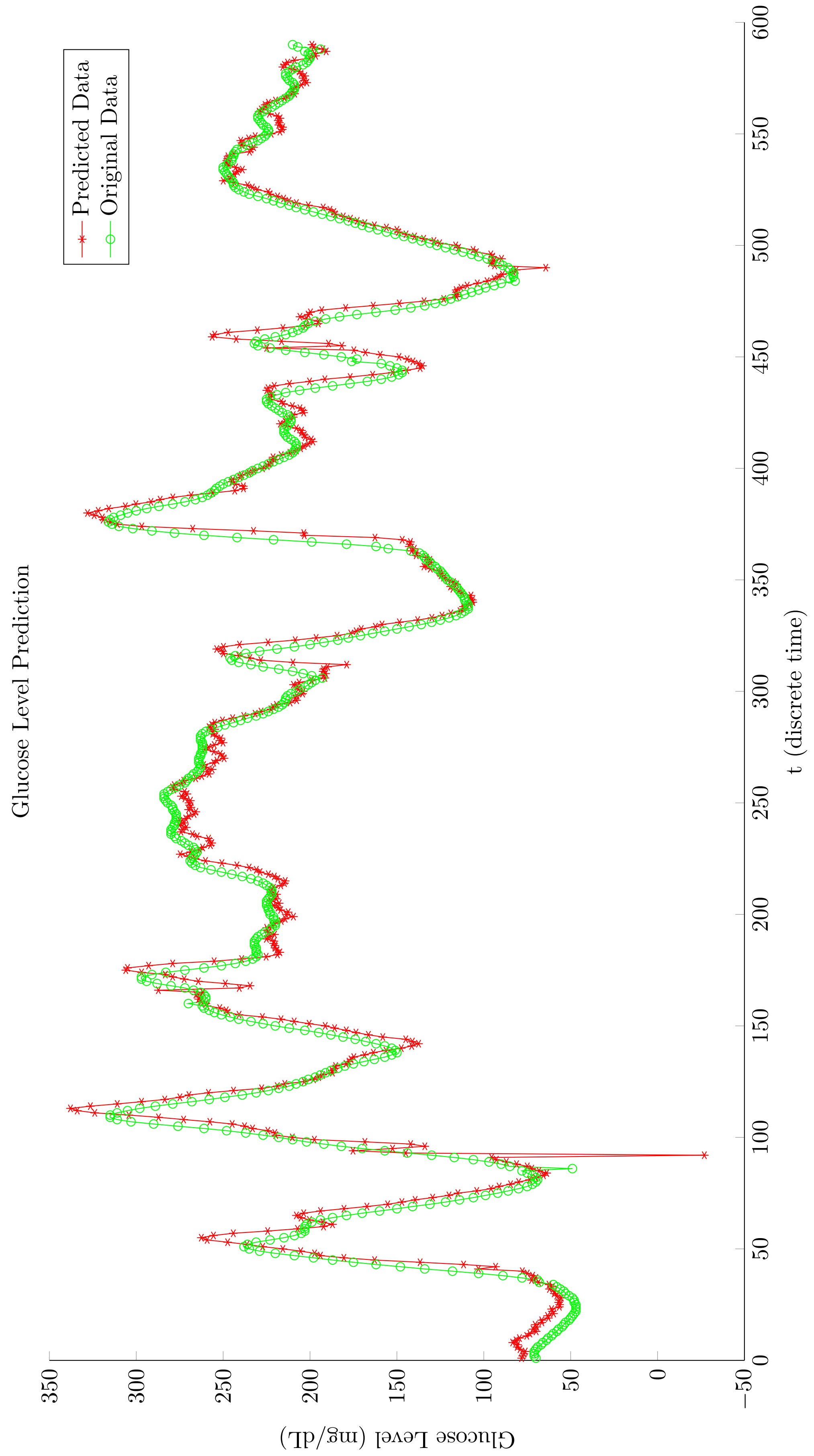

Figure 4.5: Prediction of glucose levels 30 minutes ahead using Linear Regression with direct prediction method - AML1 file

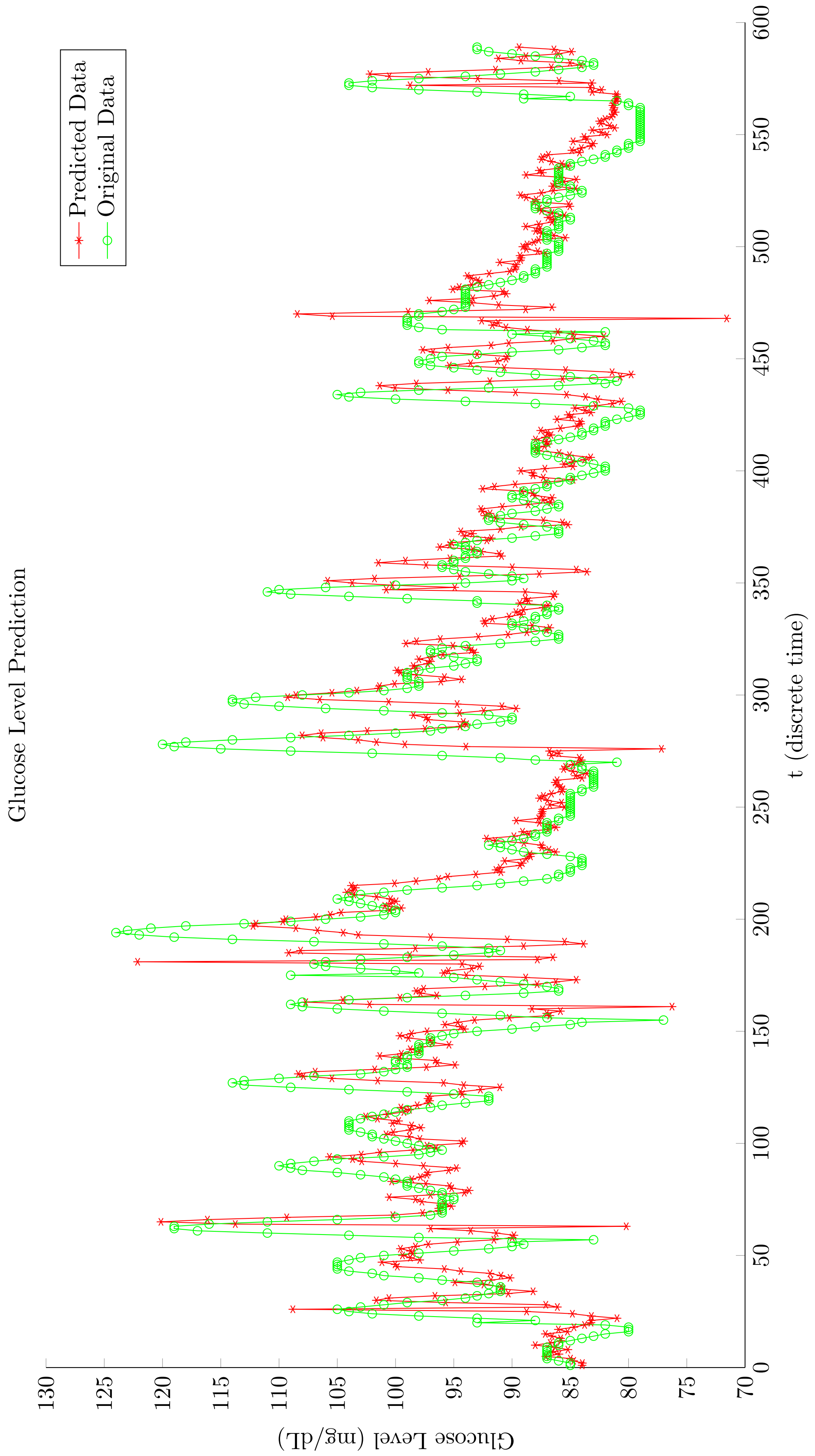

FIGURE 4.6: Prediction of glucose levels 30 minutes ahead using Linear Regression with direct prediction method - RA file

error, this error propagates in every iteration up to the final prediction. Using the direct technique, the relationship between some predictors and a time-ahead response variable is mapped. Thus, it is only necessary one prediction instead of several. Among the three datasets (CGM, RA and AML1), the maximum addition of error that we get from opting by a direct model is of 0.51 mg/dL, which will not influence significantly the prediction and the patients' health.

When analysing the values of W it is not possible to capture a pattern i.e., we cannot conclude that when doing short-term prediction the model uses less past measurements and more of them when doing the opposite. Since that each person has its own biological system and it does not work in the same way for everyone, the glucose dynamics is also not the same for all people. For this reason, the model, in order to capture as better as possible the glucose variation, needs to find the best number of past measurements which varies between individuals.

At the time we introduced this project to Dr. Rogério Ribeiro, who is project manager and also a researcher in the Associação Protectora dos Diabéticos de Portugal, we presented the results obtained with the CGM file which he found as extremely interesting and among the bests that he has seen. Since the outcomes of the other datasets are better or at least approximately the same to a value of $PH <= 30$ min, we may conclude that these models have a good performance. For greater values of PH, the RMSE is not as good as for the lower ones with an observed maximum difference of around 18 mg/dL, for predicting 60 min ahead.

Comparing the results of the state of the art, our results outperformed those presented in Section 2.2.2 with a RMSE average of 13.56 mg/dL and 22.20 mg/dL when predicting 30 and 60 minutes ahead. Zecchin et al. [13] used a fully connected and feedforward neural network model and a first-order polynomial extrapolation algorithm for glucose level prediction with $PH = 30$ min. Georga et al. [15] used the SVR algorithm achieving a RMSE average of 16.92 mg/dL and 25.02 mg/dL predicting 30 and 60 minutes ahead. Marling et al. [14] also tried to predict the glucose levels with the SVR algorithm accomplishing a RMSE of 18.00 mg/dL and 30.9 mg/dL for $PH = 30$ and $PH = 60$ min, respectively. Since that, the lower the RMSE is, the better the model, our model is superior even for $PH = 60$ minutes.

For all the mentioned reasons, we consider the linear regression with direct prediction method, the best model to predict glucose levels ahead in time offering a lower RMSE when compared to the models presented in the state of the art.

Chapter 5

Final Conclusions and Future Work

The Smart Monitor Health System, as described in the first chapter, aims to oversee the patient's health and give alerts when something is going to get outside of the normal parameters. With this work, we focused on how such system could help people regarding diabetes and came up with the idea of having computational models that in a first stage differentiates diabetic people from health ones and in a second stage helps diabetics controlling their glucose levels. These models can be introduced in the main system creating a bigger and more intelligent system to help people managing their health.

The proposed objectives were all accomplished. In the following sections, we present the conclusions of this work and the future work that can be made to improve this work.

5.1 Conclusions

Diabetes is a serious health problem that affects millions of individuals and has several consequences if it is not controlled. Statistics made by International Diabetes Federation[1] shows that 1 in 12 people has diabetes and 1 in 2 people with diabetes does not know they have it. Even worse, every 7 seconds 1 person dies from diabetes. IDF estimates that there were 387 million people in 2013 around the word with diabetes and the number of people with the disease is set to rise beyond 592 million in less than 25

[1] http://www.idf.org/worlddiabetesday/toolkit/gp/facts-figures

years [2]. Since it is affecting more and more people every day, regarding the gender, age, race, etc. and given the fact that there is still no cure, tools to test the possibility of a person having diabetes and mechanisms to help controlling the glucose levels for the ones that already suffer from the disease are important.

Doctors rely their diagnosis on some specific tests and may not take into consideration other factors that might be related to the disease such as the diastolic blood pressure. Furthermore, since a doctor is a human and humans make errors, it is important to have systems that may be capable to give a second opinion when it comes to the diagnoses.

This work has proposed and tested several algorithms to created a computation model to test the possibility of a person having diabetes or not. To build the model, hypotheses had to be formulated and for each learner hypothesis a given number of parameters need to be adjusted. For such task we used the PIMA dataset that contains several examples each one with 8 characteristics (features) plus a label (diabetic/non-diabetic). These characteristics include general information about the patient and his/her blood test. Although, on the analysed related work there is no mention to the presence of missing values, a close examination of the values revealed, for example, a few cases with the diastolic blood pressure equal to zero which is impossible for a living being. This fact leads us to assert that there is in fact missing values.

A key contribution of this work has been the introduction of diverse methods to deal with the missing values in order to fix the dataset. The computational experiments show that the best performance model consists of substituting the missing values by the mean of the features considering the class they belong to. This method along with Random Forest algorithm yielded an accuracy of 87.66±1.16%, which is 10% higher than the baseline method. Comparing our results with the ones presented in the state of the art, we found that ours are superior to any of them except for one. The difference observed between the presented result by Polat and Güneş [8] and ours is of 1.81% that is not statistically significant as some statistical tests could prove it.

In regard to predicting the individual's glucose level, models that could precisely forecast the referred level with a considerable prediction horizon would help diabetic people controlling their sugar level. For this problem, we explored several algorithms with different PH values to check how precisely those values could be predicted. We also tested various values of predictors (W) i.e., we considered different numbers of past measurements in order to choose the best one when predicting the glucose PH-steps ahead. Finally, two prediction methods were investigated and tested: direct prediction and iterative prediction. The tests were performed on 3 distinct datasets: one of them was found by us and the other four were provided by Dr. Rogério Ribeiro from APDP.

The results showed us that the best model to predict future glucose levels was the one built using the linear regression algorithm. Just on the CGM dataset and for $PH = 10$ and $PH = 20$ minutes, the E-SVR performed better than the linear regression. However, the results were obtained using a linear kernel. These facts show that there is a linear relationship between past glucose measurements and future ones. In terms of the value for W it is not possible to establish the best one even for each of the distinct values of PH as it varies between datasets. Finally, in regard to the prediction method, the iterative one with the linear regression gave us the best RMSE for every prediction horizon. Nonetheless, we opted for the direct method since the maximum observed error between methods was of 0.51 mg/dL, which is not significant in the final prediction and would not affect the patient. Despite that, with that method fewer computation calculations are needed which consequently reduces the time required to train and test.

Given what was said, our final results regarding the prediction of future glucose levels were obtained with linear regression with direct prediction method and were an average RMSE of 13.56 mg/dL and 22.20 mg/dL for 30 and 60 minutes-ahead prediction, respectively. These are excellent results, which mean that if a diabetic individual uses this model, it could prevent him from entering in hyper and hypoglycaemia states and may even save his life.

5.2 Future Work

In regard to the diabetes diagnosis problem, our suggestions go in the direction of improving the Findrisk as it is a simple and easily answered questionnaire that gives the risk of a person develops diabetes in 10 years. Perhaps, by discriminating even more the intervals, for example, about the age and body mass index, considering other information like the smoking status and blood pressure, which are factors related to the disease, it may be possible to measure even better the risk of a person developing diabetes or even testing the possibility of having it.

Several key challenges emerge in the matter of improving the models to predict even better future glucose levels. If on one hand it is desirable to have more types of information to improve the predictions, on the other hand, this information must be obtained without or at least with the minimum user interference. So, we suggest taking into account easily recordable and non-invasive information such as the time of the day, which is registered along with glucose reading within each patient's file. Since that our biological system does not work equally during the day and night as during the night we normally are resting and during the day working which burns more sugar, information about the time can help the model making more accurate predictions. Also, a mobile application that could correctly identify the food that the user is going to eat and measures the calories, sugar and other parameters would give information that the model can use to better predict future glucose levels. At the algorithms level, we suggest to explore other neural networks like Focused Time-Delay Neural Network as they are fast on training and also widely used in time series prediction with good results.

With the introduction of the Internet of Things, there are various new ideas to help diabetic people controlling their glucose levels. Since the concept of that paradigm is to have all things connected to the Internet, we may take advantage of it. For example, one mobile application that suggests what to eat to rise the glucose level preventing the user from entering on a hypoglycaemia state, based on what exists in the fridge. The overall Smart Monitor Health System can also take advantage of all the existing sensors to provide valuable information, comfort, etc. to the user. Imagine that a patient body temperature is being monitored and there is information about the room temperature. Combining these data, the system could forecast the body's temperature and turn on the heat, if the value was lower than the usual.

Considering all the information of several existing sensors along with an intelligent processing unit, that would allow the identification and prediction of several pathologies, we are aware the Smart Monitor Health System will became a reality.

Appendix A

Mobile development frameworks

Nowadays, almost every person has a smartphone with which can make phone calls, send text messages, play some games, etc. A diabetic person, as said before, has to control his glucose level in order to avoid health problems. For this type of people an application that keeps track of patient's blood glucose level would be an advantage.

The next sections describes new frameworks developed by the two most famous electronic devices sellers which could help in the development of a future application like the one described above.

A.0.1 Apple HealthKit

HealthKit is a new tool created by Apple to help users to keep track of their fitness data and personal health. There are a lot of applications that monitors health indicators like the weight or the high blood pressure. However that information is not shared among other applications. HealthKit centralizes everything on its store as long as the users allow it by specifying which data can be shared. For example, we might have an application that measures the user's heart beat rate and another that records the running routes. If they work together they are more useful as the person can see his/her heart beat rate across the running route. The information is also shown in their new app – Health - where the user can have a whole picture of his health.

Since this is a sensitive type of information the HealthKit data is not saved to iCloud neither synced across multiple devices. Instead it is maintained locally on the user's device and encrypted when the device is locked with a passcode or Touch ID. The user can choose which type of data an app can access. For example, a user may let an app to read your body temperature but prevent it from having knowledge of your blood glucose level. From the app's point of view, it does not know if the user denied access to the second one or it does not have any data. This is a good feature because it prevents people from knowing that the user may have diabetes.

When developing an application developers have two methods to access data: direct calls to the HealthKit store, which can just be used to retrieve user's gender, blood type and date of birth, and query methods, with which the programmer can get a single value once or whenever there is a change on the parameter being monitored. It is also possible to use queries to obtain statistics like how many calories the user burned during the day or a collection of statistics to have data for example about a whole week until now.

To create an application that uses the HealthKit framework, developers must provide a privacy policy and follow Apple's guidelines[1] like the following advertisement:

> *You cannot sell information gained through HealthKit to advertising platforms, data brokers or information resellers.*

HealthKit and the Health app are not available on iPad, which may be an obstacle to reach a market segment that just has this device. However it seems to me that it is a worthy framework since it lets users to have their favourite fitness and/or health apps, removes the need of having APIs to share data between apps and deals automatically with unit's conversion since it supports multiple metrics. Another advantage is that developers do not need to program methods to capture data from the sensors and/or wearable products as long as there are another apps that provide the same data.

A.0.2 Google Fit

Google Fit [43] provides a set of APIs (Android API and a REST API) for apps and devices manufacturers to store and access activity data from fitness applications and sensors on Android and other devices including all Android wearable devices. In order

[1] https://developer.apple.com/library/prerelease/ios/documentation/HealthKit/Reference/HealthKit_Framework/index.html#//apple_ref/doc/uid/TP40014707

to mobile and client devices (browser) get access to the data, a central repository on the cloud (Cloud Datastore) is used with the user's permission.

Google Fit APIs for Android are available for Android devices with Android 2.3 and above and it consists on 3 principal APIs: Sensors, Recording and History API. The first one provides access to raw data from the Android's device sensors and also from wearables. The second one provides automated storage of fitness data using subscriptions in the background syncing it with the Cloud Datastore while the third one allows to retrieve data that has been stored and perform read, insert and delete operations by the applications.

Instead of making data only available on the mobile devices, Google created a REST API to allow developers to come up with web applications that uses the fitness data stored in the cloud infrastructure.

In terms of security, an application needs permissions from the user to have access to 3 major data categories: Activity, Body and Location. Unlike Apple, Google Fit is more oriented to fitness and less to health as it does not include, for example, blood info (blood glucose, type and alcohol level) and sleep patterns in their default data types.

Appendix B

Glucose level prediction full tests

In this appendix there are all the tests made in regard to the glucose level prediction problem. The bold values show the best results within each prediction horizon. When there is more than one bold result for the same PH we prefer the one that has the smallest value of W as it uses less data.

Each table represents the results obtained in regard to the different values of W and PH for the algorithms and prediction methods.

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	7.77	14.23	19.23	24.10	26.61
7	7.79	14.26	19.22	23.99	26.35
10	7.80	14.25	19.15	23.82	26.01
13	7.80	14.20	19.02	23.56	25.62
16	7.77	14.10	18.86	23.26	25.17
19	7.72	13.98	18.69	22.95	24.70
22	7.66	13.88	18.52	22.68	24.35
25	7.61	13.79	18.36	22.42	24.06
28	7.61	13.79	18.36	22.42	24.06
31	7.58	13.70	18.22	22.18	23.80
34	7.55	13.63	18.11	22.03	23.56
37	7.51	13.56	18.00	21.91	23.29
40	7.48	13.50	17.89	21.81	23.06
43	7.45	13.45	17.85	21.71	22.92
46	7.44	13.40	17.83	21.61	22.82
49	**7.43**	**13.40**	**17.82**	**21.53**	**22.75**

Table B.1: Results using Naive method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.27	11.26	16.48	21.72	23.92
7	5.21	11.22	16.49	21.74	23.80
10	5.28	11.28	16.46	21.54	23.41
13	5.28	11.23	16.32	21.21	22.89
16	5.26	11.11	16.10	**20.87**	22.60
19	5.21	**11.03**	**16.08**	20.96	22.68
22	5.23	11.16	16.24	21.08	22.69
25	5.24	11.14	16.20	21.00	22.59
28	5.24	11.16	16.22	21.09	22.71
31	5.25	11.20	16.28	21.07	22.51
34	**5.10**	11.08	16.16	20.94	22.26
37	5.11	11.12	16.20	20.91	21.98
40	5.20	11.18	16.17	20.88	**21.92**
43	5.19	11.15	16.26	21.19	22.39
46	5.20	11.25	16.50	21.47	22.60
49	5.25	11.35	16.63	21.49	22.44

Table B.2: Results using Linear Regression with direct prediction method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.28	11.33	16.56	21.79	23.91
7	5.20	11.13	16.28	21.47	23.61
10	5.28	11.27	16.45	21.55	23.46
13	5.28	11.22	16.35	21.38	23.23
16	5.27	11.20	16.32	21.28	23.02
19	5.21	**11.03**	**16.00**	20.75	22.35
22	5.21	11.10	16.14	20.92	22.49
25	5.25	11.15	16.20	20.90	22.35
28	5.24	11.15	16.20	20.96	22.51
31	5.26	11.23	16.36	21.22	22.89
34	**5.10**	11.04	16.07	20.82	22.21
37	**5.10**	**11.03**	16.03	**20.71**	21.85
40	5.23	11.36	16.52	21.27	22.04
43	5.18	11.13	16.14	20.79	**21.58**
46	5.19	11.14	16.21	20.80	21.60
49	5.24	11.29	16.44	21.07	21.88

Table B.3: Results using Linear Regression with iterative prediction method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.15	11.46	16.92	22.48	24.43
7	5.15	11.75	17.03	22.83	24.36
10	5.22	11.70	17.45	22.80	24.01
13	5.05	11.68	17.19	22.41	23.42
16	5.01	11.39	16.93	21.80	22.84
19	4.96	11.28	16.77	**21.68**	22.63
22	5.01	11.21	16.85	21.96	22.65
25	5.01	11.17	16.80	21.81	22.51
28	4.98	**11.08**	**16.63**	21.75	22.60
31	4.98	11.22	16.74	22.13	22.71
34	4.97	11.43	**16.63**	21.86	**22.02**
37	**4.95**	11.22	16.71	22.01	22.74
40	5.03	11.40	17.06	22.11	22.56
43	5.03	11.40	17.24	22.10	23.15
46	5.04	11.57	17.78	22.60	23.03
49	5.09	11.56	17.94	22.51	23.51

Table B.4: Results using E-SVR with direct prediction method and linear kernel on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.16	11.55	17.20	23.61	25.25
7	5.16	11.68	17.31	23.10	24.52
10	5.25	11.72	17.51	22.25	24.15
13	5.17	11.66	17.09	21.81	23.49
16	5.07	11.64	17.03	21.81	22.80
19	**5.05**	**11.22**	16.67	**21.28**	22.96
22	5.13	11.26	16.78	21.62	22.50
25	5.11	11.25	16.76	21.58	22.14
28	5.10	11.34	**16.55**	21.64	22.25
31	5.10	11.35	16.70	21.75	22.07
34	5.19	11.51	16.81	21.73	**21.97**
37	5.23	11.66	16.93	22.06	22.10
40	5.30	11.87	17.45	22.11	22.22
43	5.61	11.93	17.41	22.09	22.89
46	5.35	12.14	17.72	22.41	24.12
49	5.64	12.38	18.04	22.62	23.83

TABLE B.5: Results using E-SVR with direct prediction method and RBF kernel on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.09	11.11	16.34	21.94	24.54
7	5.03	11.18	16.59	22.14	24.49
10	5.20	11.42	16.79	22.19	24.39
13	5.08	11.06	16.34	22.06	24.27
16	5.17	11.26	16.56	21.95	24.24
19	5.01	11.04	16.67	22.40	25.01
22	5.16	11.21	16.47	21.49	23.32
25	5.16	11.29	16.22	21.28	23.08
28	5.02	**10.97**	16.23	21.52	23.49
31	5.02	11.01	**16.11**	21.02	22.64
34	5.06	11.27	16.40	22.62	25.54
37	**4.98**	11.20	16.61	22.32	23.14
40	5.01	11.03	16.47	**20.99**	**21.65**
43	5.04	11.21	16.61	21.88	23.19
46	5.10	11.37	17.30	21.52	21.86
49	5.08	11.97	17.35	21.78	22.23

Table B.6: Results using E-SVR with iterative prediction method and linear kernel on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.15	11.66	17.60	24.16	27.58
7	5.22	11.73	17.94	25.28	28.90
10	5.29	11.69	17.39	24.30	27.29
13	5.24	12.01	17.19	23.13	25.79
16	5.11	11.83	17.69	24.13	25.13
19	5.13	11.41	17.03	22.69	25.29
22	**5.08**	11.27	16.67	22.32	24.68
25	**5.08**	11.53	16.81	22.13	24.32
28	5.10	11.21	**16.44**	22.38	24.71
31	5.14	**11.12**	16.74	21.86	23.76
34	5.25	11.59	16.60	**21.19**	22.54
37	5.33	11.30	16.45	21.41	22.72
40	5.39	11.57	16.70	21.50	22.24
43	5.32	11.74	16.98	21.28	22.19
46	5.46	12.19	17.20	21.29	**21.70**
49	5.33	12.44	17.57	21.76	22.18

Table B.7: Results using E-SVR with iterative prediction method and RBF kernel on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**11.15**	**16.43**	**20.28**	23.90	25.08
7	12.86	17.33	20.83	24.45	24.77
10	13.91	18.33	21.59	23.77	25.12
13	15.30	18.89	21.21	23.57	25.08
16	15.91	18.98	21.25	23.73	24.96
19	16.36	19.34	21.67	23.70	24.82
22	17.13	19.89	21.94	23.76	24.76
25	17.86	20.33	22.19	23.86	24.80
28	18.28	20.70	22.48	24.09	25.20
31	18.84	21.03	22.69	24.21	25.27
34	19.15	21.27	22.90	24.36	25.22
37	19.43	21.52	23.01	24.35	24.95
40	19.63	21.66	23.04	24.11	24.57
43	19.73	21.49	22.70	23.72	24.02
46	19.61	21.16	22.44	23.35	23.59
49	19.42	21.08	22.29	**23.02**	**23.16**

Table B.8: Results using GRNN with direct prediction method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**9.28**	**16.37**	21.16	26.10	28.31
7	13.48	17.73	**20.99**	24.49	26.15
10	14.32	18.60	21.78	24.98	26.62
13	15.43	19.76	22.12	25.62	26.75
16	17.01	19.94	22.13	24.17	25.72
19	17.03	19.85	21.88	23.67	24.59
22	16.97	19.57	21.56	23.22	24.05
25	17.00	19.53	21.36	**23.03**	**24.00**
28	17.33	19.80	21.57	23.12	24.20
31	17.72	20.10	21.76	23.17	24.26
34	18.22	20.48	22.04	23.56	24.62
37	18.76	21.02	22.60	24.22	25.32
40	19.09	21.43	23.01	24.77	26.04
43	19.14	21.58	23.36	25.35	26.97
46	19.11	21.61	23.69	26.16	28.67
49	19.53	22.25	24.45	27.11	29.92

Table B.9: Results using GRNN with iterative prediction method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**7.92**	**15.11**	**20.82**	27.64	29.63
7	8.80	15.95	22.05	28.06	30.41
10	10.30	17.74	22.91	28.91	30.48
13	11.26	18.01	24.11	28.80	29.77
16	11.55	18.57	23.64	27.60	**29.57**
19	11.42	18.46	24.09	28.96	30.76
22	11.17	19.12	24.98	30.07	30.85
25	11.07	18.85	25.46	29.40	30.17
28	11.71	19.28	24.97	29.32	30.42
31	11.37	20.07	25.03	29.12	30.17
34	12.90	20.77	24.83	28.49	30.67
37	13.57	21.15	25.59	28.30	31.09
40	13.92	20.85	25.20	28.80	30.87
43	13.75	21.11	25.36	29.96	30.71
46	14.13	20.86	25.28	28.06	29.99
49	12.18	20.94	24.79	**27.41**	30.81

Table B.10: Results using Random Forest with direct prediction method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**7.91**	**15.15**	**20.74**	26.48	30.11
7	8.43	15.49	21.47	27.82	30.08
10	9.50	16.69	22.63	27.61	29.52
13	10.79	17.53	24.07	28.30	31.02
16	11.16	18.63	24.90	31.27	30.10
19	11.07	18.39	24.84	27.65	29.79
22	11.06	18.59	22.65	26.57	28.04
25	10.72	17.39	23.69	27.66	28.18
28	11.07	18.46	23.01	26.92	26.93
31	10.83	18.41	23.14	26.85	27.17
34	11.69	18.75	23.10	26.88	27.63
37	12.42	18.94	22.76	26.27	26.83
40	13.90	18.84	22.73	**25.34**	**25.23**
43	12.19	18.96	22.21	26.92	27.57
46	11.39	18.79	22.60	26.33	27.62
49	11.50	18.36	22.45	26.74	27.59

Table B.11: Results using Random Forest with iterative prediction method on CGM file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**10.86**	**20.93**	**30.28**	**42.61**	**53.06**
7	10.87	20.95	30.30	42.65	53.11
10	10.88	20.97	30.33	42.69	53.15
13	10.89	20.99	30.35	42.72	53.20
16	10.90	21.00	30.38	42.76	53.24
19	10.91	21.02	30.41	42.79	53.29
22	10.92	21.04	30.43	42.83	53.33
25	10.93	21.06	30.46	42.87	53.38
28	10.94	21.07	30.48	42.90	53.42
31	10.94	21.09	30.51	42.94	53.46
34	10.94	21.09	30.51	42.94	53.46
37	10.95	21.11	30.53	42.97	53.51
40	10.96	21.13	30.56	43.01	53.55
43	10.97	21.15	30.58	43.04	53.59
46	10.98	21.16	30.61	43.08	53.63
49	10.99	21.18	30.63	43.11	53.67

TABLE B.12: Results using Naive method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	5.38	11.82	19.56	30.86	40.38
7	5.17	11.49	19.27	30.74	40.39
10	**5.11**	11.36	19.15	30.72	40.45
13	5.12	11.39	19.20	30.74	40.37
16	5.13	11.40	19.18	30.64	40.17
19	5.12	11.35	19.08	**30.45**	**40.03**
22	**5.11**	**11.32**	**19.06**	30.53	40.32
25	**5.11**	11.35	19.15	30.82	40.86
28	5.16	11.47	19.38	31.27	41.49
31	5.20	11.60	19.61	31.62	41.80
34	5.23	11.69	19.75	31.75	41.93
37	5.24	11.70	19.77	31.80	41.96
40	5.25	11.73	19.81	31.80	41.88
43	5.25	11.71	19.75	31.69	41.72
46	5.24	11.70	19.72	31.62	41.55
49	5.25	11.70	19.72	31.57	41.35

Table B.13: Results using Linear Regression with direct prediction method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
7	5.15	11.42	19.18	30.82	40.79
10	**5.11**	11.35	19.13	30.67	40.37
13	**5.11**	11.36	19.15	30.69	40.39
16	5.13	11.42	19.25	30.89	40.72
19	5.12	11.37	19.12	30.57	40.16
22	**5.11**	11.32	19.01	30.33	39.76
25	**5.11**	**11.31**	**18.96**	**30.20**	**39.52**
28	5.16	11.45	19.26	30.83	40.55
31	5.19	11.53	19.43	31.15	41.02
34	5.23	11.67	19.70	31.67	41.79
37	5.24	11.70	19.76	31.78	41.95
40	5.25	11.74	19.83	31.87	42.03
43	5.25	11.71	19.78	31.78	41.90
46	5.24	11.70	19.74	31.66	41.67
49	5.25	11.70	19.73	31.66	41.65

Table B.14: Results using Linear Regression with iterative prediction method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	5.89	12.71	20.08	**31.88**	41.87
7	5.70	12.56	20.12	32.16	41.96
10	5.54	12.22	19.92	31.92	41.97
13	5.55	12.18	19.95	31.98	41.87
16	5.56	12.20	20.21	32.07	**41.85**
19	5.54	12.09	**19.86**	31.89	**41.85**
22	**5.53**	**12.02**	20.26	32.00	42.47
25	5.57	12.09	19.97	32.39	42.76
28	5.61	12.31	20.26	32.56	43.02
31	5.65	12.58	20.48	32.92	43.13
34	5.69	12.62	20.49	32.85	43.22
37	5.67	12.57	20.34	32.83	43.44
40	5.68	12.57	20.61	32.73	43.33
43	5.68	12.52	20.54	32.72	43.18
46	5.68	12.02	20.19	32.76	43.28
49	5.69	12.47	20.47	32.83	43.35

Table B.15: Results using E-SVR with direct prediction method and linear kernel on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	5.92	12.62	20.20	31.72	41.87
7	5.72	12.47	19.72	**31.60**	**41.43**
10	5.55	11.85	**19.67**	31.74	41.75
13	5.57	11.86	19.81	31.73	41.75
16	5.60	11.86	19.77	31.85	42.45
19	5.62	**11.61**	19.76	31.69	42.38
22	5.58	11.76	19.84	32.27	43.27
25	**5.35**	11.88	20.09	32.85	43.67
28	5.62	11.95	20.16	33.63	43.99
31	5.71	12.17	20.66	34.00	44.37
34	5.69	11.98	20.59	34.07	44.84
37	5.48	12.04	20.83	34.31	45.27
40	5.47	12.13	20.83	34.39	45.42
43	5.44	12.10	20.94	34.63	45.44
46	5.47	12.14	20.85	34.92	45.64
49	5.50	12.07	20.94	35.01	46.18

Table B.16: Results using E-SVR with direct prediction method and RBF kernel on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	5.93	12.49	20.59	33.03	44.29
7	5.70	12.26	20.25	32.45	44.39
10	5.68	12.37	20.49	32.84	43.82
13	5.70	12.38	20.48	31.81	42.38
16	5.69	12.38	20.50	32.86	43.80
19	**5.62**	12.19	20.06	31.94	42.58
22	5.63	12.19	20.23	**31.30**	**41.40**
25	5.64	12.21	20.11	32.13	42.89
28	5.68	12.34	20.35	32.32	42.76
31	5.76	**11.94**	20.04	32.36	43.23
34	5.76	12.46	20.61	32.21	42.90
37	5.75	12.47	20.23	32.37	43.24
40	5.76	12.55	20.37	33.00	42.91
43	5.74	12.47	20.25	32.68	43.68
46	5.74	12.46	**19.72**	31.49	41.58
49	5.69	12.33	20.02	32.19	43.85

Table B.17: Results using E-SVR with iterative prediction method and linear kernel on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	6.04	12.87	20.98	33.13	44.51
7	5.76	12.43	20.62	32.83	44.07
10	5.72	12.49	20.91	33.74	43.76
13	5.63	12.28	20.22	32.56	43.33
16	5.68	11.97	20.23	32.88	44.12
19	5.69	12.37	**19.82**	**32.18**	44.14
22	5.72	**11.89**	20.03	32.46	**43.15**
25	5.68	11.93	20.00	32.84	43.71
28	5.73	12.12	20.48	32.82	43.56
31	5.74	12.15	20.54	33.11	45.66
34	5.86	12.21	20.58	33.62	44.22
37	5.87	12.88	20.18	33.77	44.38
40	5.79	12.19	20.59	34.24	45.06
43	5.52	12.22	20.75	34.65	45.65
46	5.53	12.35	21.08	34.27	45.17
49	**5.51**	12.25	20.92	34.25	45.12

Table B.18: Results using E-SVR with iterative prediction method and RBF kernel on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	**8.95**	**16.98**	**25.06**	**36.47**	**46.10**
7	11.67	19.70	27.80	39.14	48.25
10	13.90	22.44	30.69	41.69	49.94
13	15.93	24.78	33.25	43.59	51.02
16	18.50	27.31	35.31	44.78	51.70
19	20.80	29.14	36.50	45.23	52.27
22	23.44	30.98	37.63	46.11	53.27
25	25.52	32.43	39.05	47.59	54.92
28	26.85	34.02	40.62	49.31	56.59
31	28.40	35.61	42.33	51.24	58.54
34	29.83	37.20	44.16	53.39	60.72
37	31.55	39.15	46.30	55.67	62.61
40	33.73	41.58	48.96	57.99	64.08
43	36.62	44.66	51.82	60.05	65.13
46	39.77	47.76	54.42	61.58	65.92
49	42.99	50.49	56.44	62.76	66.58

Table B.19: Results using GRNN with direct prediction method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	**9.69**	**19.64**	**29.57**	**42.63**	**53.10**
7	12.18	21.62	31.56	45.50	56.41
10	16.52	26.94	36.21	48.20	57.12
13	17.89	27.40	36.27	47.84	56.97
16	18.59	28.28	37.68	50.65	58.70
19	20.22	32.74	40.49	50.51	58.50
22	25.37	33.26	40.77	50.64	58.55
25	26.42	34.14	41.29	50.67	58.38
28	27.43	35.36	42.63	51.88	59.40
31	28.74	37.28	45.10	54.63	62.46
34	30.21	39.20	47.30	57.33	65.30
37	31.88	41.13	49.74	60.92	65.40
40	33.94	43.20	51.99	63.60	65.72
43	36.45	45.72	54.56	66.42	66.19
46	39.32	48.21	56.80	63.22	66.84
49	42.33	50.75	59.43	63.98	67.65

Table B.20: Results using GRNN with iterative prediction method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**7.36**	**15.72**	**24.12**	**35.98**	**46.57**
7	8.16	16.99	25.53	39.09	49.00
10	8.64	17.45	27.27	40.57	51.14
13	8.73	18.35	28.32	41.69	51.62
16	9.54	18.71	28.98	41.12	50.90
19	9.32	18.81	28.66	40.32	50.23
22	9.40	18.63	28.19	40.18	50.66
25	9.62	19.07	28.88	41.20	50.88
28	9.62	19.48	29.05	41.97	52.58
31	9.87	19.41	30.06	43.01	54.88
34	9.72	19.72	30.53	43.94	58.04
37	10.04	20.40	31.52	44.20	59.23
40	10.40	20.89	32.55	47.61	61.97
43	10.81	21.14	32.55	49.40	63.53
46	10.94	21.72	34.88	50.46	64.16
49	10.94	21.65	33.73	50.12	63.95

Table B.21: Results using Random Forest with direct prediction method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**7.60**	**16.10**	**25.39**	39.22	50.71
7	8.33	16.56	25.48	**38.84**	50.58
10	8.80	17.37	27.09	40.02	**50.41**
13	9.03	17.94	27.35	40.87	52.01
16	9.16	18.33	28.26	41.38	51.89
19	9.45	19.18	28.17	41.25	51.35
22	9.32	18.57	28.18	41.05	51.60
25	9.62	18.52	28.58	43.12	51.92
28	9.58	18.39	28.85	41.87	52.93
31	9.68	18.54	28.53	42.63	53.50
34	9.79	19.08	28.50	41.82	54.56
37	9.90	18.98	28.71	43.24	54.35
40	10.67	20.74	29.11	43.38	53.00
43	10.72	20.03	30.31	43.78	58.13
46	10.95	20.33	30.75	46.07	60.04
49	10.72	20.30	31.89	46.86	61.08

Table B.22: Results using Random Forest with iterative prediction method on AML1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	**4.35**	**7.61**	**9.80**	**10.95**	**10.47**
7	**4.35**	7.62	9.81	10.96	10.48
10	**4.35**	7.62	9.82	10.97	10.48
13	4.36	7.63	9.83	10.97	10.49
16	4.36	7.63	9.84	10.98	10.50
19	4.36	7.64	9.84	10.99	10.51
22	4.37	7.65	9.85	11.00	10.52
25	4.37	7.65	9.86	11.01	10.52
28	4.38	7.66	9.87	11.02	10.53
31	4.38	7.67	9.87	11.02	10.54
34	4.38	7.67	9.87	11.02	10.54
37	4.38	7.67	9.88	11.03	10.55
40	4.39	7.68	9.89	11.04	10.56
43	4.39	7.68	9.90	11.05	10.57
46	4.39	7.69	9.91	11.06	10.57
49	4.40	7.70	9.92	11.07	10.58

Table B.23: Results using Naive method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	3.36	6.19	8.13	8.93	8.63
7	3.07	5.67	7.64	8.57	**8.43**
10	3.09	5.70	7.65	**8.55**	8.47
13	**3.06**	**5.63**	**7.61**	8.60	8.58
16	**3.06**	5.64	7.63	8.65	8.60
19	**3.06**	5.64	7.65	8.67	8.60
22	**3.06**	5.65	7.66	8.67	8.60
25	3.08	5.69	7.69	8.68	8.60
28	3.08	5.69	7.70	8.69	8.63
31	3.09	5.70	7.71	8.72	8.69
34	3.09	5.70	7.72	8.75	8.69
37	3.10	5.72	7.75	8.76	8.71
40	3.09	5.72	7.74	8.76	8.71
43	3.10	5.73	7.75	8.76	8.72
46	3.10	5.74	7.76	8.79	8.75
49	3.10	5.74	7.78	8.81	8.77

Table B.24: Results using Linear Regression with direct prediction method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	3.38	6.48	8.96	10.41	10.16
7	3.07	**5.62**	**7.57**	**8.49**	**8.40**
10	3.09	5.72	7.77	8.81	8.72
13	**3.06**	5.64	7.61	8.60	8.51
16	**3.06**	5.65	7.64	8.64	8.56
19	**3.06**	5.64	7.63	8.61	8.54
22	**3.06**	5.65	7.66	8.66	8.60
25	3.07	5.68	7.69	8.69	8.62
28	3.08	5.69	7.70	8.70	8.62
31	3.09	5.70	7.71	8.70	8.62
34	3.09	5.70	7.71	8.70	8.62
37	3.10	5.71	7.73	8.72	8.64
40	3.09	5.72	7.74	8.75	8.69
43	3.10	5.72	7.75	8.77	8.71
46	3.10	5.73	7.76	8.77	8.71
49	3.10	5.74	7.76	8.78	8.71

Table B.25: Results using Linear Regression with iterative prediction method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	3.58	6.80	8.67	9.40	9.00
7	3.33	6.11	7.97	8.93	8.83
10	3.33	6.16	7.99	**8.92**	**8.72**
13	3.29	**6.00**	**7.94**	9.12	**8.72**
16	3.31	6.11	7.98	9.02	8.79
19	3.30	6.13	8.06	9.02	8.85
22	**3.28**	6.13	7.99	9.11	8.82
25	**3.28**	6.05	8.04	9.05	8.88
28	3.30	6.06	8.07	9.18	8.84
31	3.29	6.11	8.13	9.23	8.83
34	3.31	6.08	8.17	9.23	8.86
37	3.32	6.12	8.07	9.17	9.03
40	3.31	6.08	8.10	9.18	8.89
43	3.32	6.06	8.07	9.08	8.90
46	3.32	6.20	8.02	9.08	8.91
49	3.31	6.09	8.15	9.01	8.91

Table B.26: Results using E-SVR with direct prediction method and linear kernel on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	3.58	6.69	8.47	9.32	9.08
7	3.33	**5.99**	7.87	8.94	8.67
10	**3.25**	6.04	7.88	**8.76**	**8.64**
13	3.30	6.03	**7.83**	9.04	8.76
16	3.33	6.10	8.03	8.93	8.76
19	3.29	6.16	8.00	8.97	8.76
22	3.29	6.05	7.96	8.96	8.84
25	3.29	6.15	7.98	9.03	8.76
28	3.30	6.16	7.99	9.09	8.82
31	3.31	6.04	7.96	9.09	8.86
34	3.29	6.13	8.00	9.12	8.87
37	3.34	6.12	7.95	9.06	8.84
40	3.34	6.12	7.92	9.03	8.94
43	3.33	6.14	7.99	9.15	8.92
46	3.35	6.13	8.08	9.23	8.92
49	3.34	6.14	8.08	9.19	8.91

Table B.27: Results using E-SVR with direct prediction method and RBF kernel on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	3.61	6.78	9.63	10.08	9.96
7	3.41	6.33	8.38	9.16	**8.94**
10	3.39	6.31	8.39	9.68	9.22
13	3.28	**5.95**	7.92	9.66	9.71
16	3.29	6.44	**7.85**	9.52	9.47
19	3.41	6.24	8.31	8.99	9.01
22	**3.27**	5.98	7.94	9.40	9.52
25	**3.27**	6.42	8.41	9.30	9.29
28	3.28	6.07	7.91	9.07	9.37
31	3.41	6.21	8.28	9.24	9.30
34	3.32	6.08	8.16	**8.84**	9.15
37	3.32	6.11	8.40	9.02	9.13
40	3.31	6.12	8.27	9.41	9.44
43	3.32	6.64	8.88	9.02	8.99
46	3.31	6.09	8.58	9.31	9.10
49	3.32	6.07	7.94	9.04	9.04

Table B.28: Results using E-SVR with iterative prediction method and linear kernel on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	3.70	6.93	9.73	10.96	12.01
7	3.38	6.31	8.53	10.77	11.27
10	3.40	6.29	8.59	9.17	9.25
13	3.34	6.17	8.39	9.51	11.29
16	3.29	6.04	**8.11**	9.62	9.98
19	3.36	6.30	8.73	9.11	9.04
22	3.32	6.10	8.29	9.64	9.46
25	3.32	6.07	8.46	10.12	10.48
28	3.35	5.97	8.22	10.44	9.26
31	3.34	**5.93**	8.22	9.32	9.02
34	3.40	6.37	8.76	10.33	10.67
37	3.36	6.33	8.32	9.71	9.95
40	3.34	6.20	8.30	9.57	9.84
43	3.45	6.45	8.44	9.26	9.14
46	3.37	6.24	8.53	**8.98**	**8.96**
49	**3.27**	6.11	8.47	9.22	9.26

Table B.29: Results using E-SVR with iterative prediction method and RBF kernel on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	**3.87**	**6.92**	8.70	9.01	8.70
7	4.29	7.11	8.73	8.78	8.61
10	4.92	7.73	8.59	8.65	**8.59**
13	5.58	7.89	8.51	**8.64**	8.66
16	6.43	8.03	**8.49**	8.67	8.73
19	6.86	8.09	8.53	8.70	8.77
22	7.05	8.15	8.60	8.75	8.80
25	7.11	8.18	8.65	8.80	8.87
28	7.17	8.26	8.70	8.86	8.95
31	7.25	8.32	8.72	8.90	9.05
34	7.41	8.38	8.76	8.97	9.14
37	7.55	8.43	8.81	9.04	9.21
40	7.68	8.52	8.91	9.09	9.25
43	7.81	8.57	8.94	9.12	9.31
46	7.87	8.58	8.95	9.20	9.38
49	7.92	8.63	8.97	9.23	9.38

Table B.30: Results using GRNN with direct prediction method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	**4.19**	**7.34**	9.02	9.43	9.34
7	4.56	7.78	9.18	9.61	9.48
10	5.28	7.92	9.34	9.52	9.76
13	5.48	7.94	9.06	9.31	9.46
16	5.99	8.48	8.92	9.13	9.30
19	7.60	8.38	8.83	9.06	9.19
22	7.60	8.42	**8.81**	9.07	**9.18**
25	7.63	8.46	8.84	**9.04**	**9.18**
28	7.66	8.46	8.88	9.05	9.19
31	7.71	8.49	8.93	9.13	9.23
34	7.74	8.52	8.93	9.15	9.28
37	7.77	8.56	8.95	9.16	9.29
40	7.83	8.59	8.99	9.17	9.31
43	7.88	8.63	9.03	9.22	9.42
46	7.97	8.67	9.07	9.25	9.44
49	7.98	8.70	9.09	9.28	9.49

TABLE B.31: Results using GRNN with iterative prediction method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	5.38	11.84	19.51	30.69	40.14
4	**3.55**	6.74	8.87	10.00	9.95
7	3.68	**6.60**	**8.74**	10.02	10.36
10	3.69	6.64	8.77	10.09	10.84
13	3.75	6.82	9.00	10.42	11.05
16	3.84	7.03	9.29	10.83	11.41
19	4.02	7.50	9.68	10.81	10.99
22	4.20	7.46	9.63	10.44	10.33
25	4.15	7.55	9.52	10.23	10.08
28	4.11	7.39	9.50	10.13	9.89
31	4.13	7.53	9.59	10.00	9.96
34	4.16	7.42	9.43	9.85	9.93
37	4.14	7.46	9.46	9.83	9.73
40	4.26	7.44	9.40	9.66	**9.63**
43	4.18	7.39	9.23	**9.57**	9.74
46	4.24	7.40	9.24	**9.57**	9.78
49	4.28	7.32	9.30	9.75	9.86

Table B.32: Results using Random Forest with direct prediction method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**3.58**	6.96	9.67	11.39	11.12
7	3.70	6.87	9.33	10.64	10.41
10	3.72	**6.79**	9.29	10.40	10.53
13	3.85	6.88	**9.23**	**10.32**	10.40
16	4.03	7.02	9.36	10.73	10.56
19	3.97	7.26	9.63	10.89	10.81
22	4.05	7.50	9.82	11.14	10.97
25	4.18	7.46	9.89	11.19	10.96
28	4.22	7.51	9.83	11.05	11.26
31	4.25	7.57	9.87	11.17	10.86
34	4.09	7.45	9.81	11.10	**10.33**
37	4.21	7.39	9.72	10.88	10.77
40	4.20	7.52	9.73	11.12	10.51
43	4.22	7.44	9.69	11.50	10.53
46	4.22	7.24	9.57	11.06	10.79
49	4.18	7.61	9.65	11.30	10.58

Table B.33: Results using Random Forest with iterative prediction method on RA file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	6.08	11.43	17.08	24.27	29.58
7	6.02	11.30	16.96	24.22	29.56
10	6.01	11.29	16.96	24.22	29.53
13	6.03	11.33	17.00	24.22	29.52
16	6.03	11.35	17.05	24.28	29.56
19	6.04	11.38	17.08	24.32	29.52
22	6.04	11.38	17.06	24.20	29.21
25	6.03	11.31	16.90	23.84	28.76
28	6.00	11.22	16.71	23.58	28.50
31	**5.99**	**11.17**	**16.64**	**23.49**	**28.42**
34	5.99	11.20	16.71	23.56	28.49
37	6.02	11.24	16.74	23.58	28.50
40	6.03	11.26	16.75	23.60	28.52
43	6.03	11.26	16.76	23.61	28.51
46	6.03	11.27	16.77	23.62	28.53
49	6.04	11.28	16.77	23.63	28.55

Table B.34: Results using Linear Regression with direct prediction method on AC1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	10.28	18.19	26.82	38.02	46.72
7	10.19	18.08	**26.75**	37.95	46.62
10	**10.18**	**18.07**	**26.75**	37.96	46.71
13	**10.18**	18.09	26.81	38.19	47.05
16	10.20	18.17	27.01	38.49	47.17
19	10.25	18.28	27.11	38.42	46.92
22	10.26	18.24	27.02	38.27	46.68
25	10.27	18.28	27.06	38.25	46.58
28	10.29	18.30	27.06	38.21	46.46
31	10.29	18.30	27.04	38.10	46.27
34	10.29	18.27	26.97	37.96	46.12
37	10.30	18.25	26.90	**37.90**	**46.10**
40	10.30	18.27	26.95	37.95	46.15
43	10.31	18.26	26.94	38.05	46.22
46	10.32	18.28	27.06	38.11	46.35
49	10.32	18.32	27.10	38.18	46.41

Table B.35: Results using Linear Regression with direct prediction method on AF1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	9.99	22.85	36.98	56.41	71.32
7	9.44	**21.89**	**36.04**	**55.73**	**70.87**
10	9.50	21.98	36.10	**55.73**	71.00
13	9.53	22.05	36.17	55.81	71.17
16	9.57	22.12	36.35	56.25	71.97
19	9.65	22.34	36.74	56.93	72.68
22	9.65	22.38	36.93	57.16	72.75
25	9.64	22.40	36.88	56.94	72.49
28	9.66	22.38	36.82	56.80	72.39
31	**9.38**	22.01	36.42	56.53	72.33
34	9.40	22.07	36.53	56.72	72.46
37	9.43	22.09	36.56	56.71	72.44
40	9.43	22.07	36.53	56.72	72.36
43	9.46	22.17	36.70	56.87	72.39
46	9.47	22.19	36.75	57.02	72.76
49	9.46	22.19	36.82	57.31	73.15

Table B.36: Results using Linear Regression with direct prediction method on GC1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	8.90	**16.69**	**24.75**	**35.16**	**43.07**
7	**8.87**	16.77	24.89	35.28	43.19
10	8.88	16.77	24.89	35.36	43.41
13	8.90	16.85	25.03	35.66	43.88
16	8.90	16.86	25.11	35.92	44.14
19	8.96	17.02	25.38	36.15	44.22
22	8.98	17.06	25.39	36.13	44.22
25	9.00	17.14	25.42	36.19	44.26
28	9.03	17.09	25.37	36.19	44.25
31	9.10	17.14	25.42	36.19	44.24
34	9.16	17.21	25.46	36.21	44.20
37	9.06	17.11	25.41	36.17	44.10
40	9.09	17.14	25.44	36.27	44.25
43	9.10	17.10	25.50	36.45	44.50
46	9.12	17.27	25.76	36.71	44.72
49	9.07	17.14	25.63	36.61	44.68

Table B.37: Results using Linear Regression with direct prediction method on GN1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	**9.76**	20.46	31.58	46.55	59.05
7	9.77	20.67	31.96	46.90	59.36
10	**9.76**	20.46	31.55	**46.34**	**58.91**
13	**9.76**	**20.43**	**31.50**	46.35	58.94
16	9.81	20.64	31.91	46.81	59.45
19	9.91	20.81	32.05	46.99	59.81
22	9.91	20.79	32.03	47.08	59.95
25	9.94	20.96	32.29	47.38	60.50
28	10.01	21.01	32.39	47.75	60.98
31	10.04	21.06	32.55	48.02	61.25
34	10.13	21.28	32.81	48.19	61.40
37	10.10	21.21	32.79	48.31	61.64
40	10.19	21.46	33.13	48.76	62.24
43	10.25	21.50	33.18	48.98	62.56
46	10.22	21.49	33.28	49.20	62.76
49	10.27	21.61	33.48	49.39	62.84

Table B.38: Results using Linear Regression with direct prediction method on LM1 file - RMSE in mg/dL

W	PH				
	10 min	20 min	30 min	45 min	60 min
4	3.28	6.12	8.57	11.05	12.37
7	**3.16**	**5.97**	**8.39**	**10.85**	**12.20**
10	3.18	6.04	8.49	10.90	**12.20**
13	**3.16**	5.98	8.43	10.89	12.24
16	3.17	6.00	8.45	10.90	12.25
19	3.18	6.02	8.49	10.96	12.27
22	3.17	6.01	8.46	10.93	12.25
25	3.17	6.00	8.46	10.93	12.26
28	3.17	6.01	8.49	10.97	12.29
31	3.17	6.00	8.48	10.97	12.30
34	3.17	6.00	8.48	10.97	12.31
37	3.17	6.01	8.49	10.99	12.34
40	3.17	6.02	8.51	11.02	12.35
43	3.18	6.03	8.51	11.02	12.33
46	**3.16**	6.02	8.49	11.00	12.28
49	3.19	6.05	8.53	11.00	12.31

Table B.39: Results using Linear Regression with direct prediction method on RR file - RMSE in mg/dL

Appendix C

Minutes of the meetings

In this appendix we show the minutes of all the meetings held over the year. Each one of the following sections represent a meeting. The fist one is the oldest while the last one is the most recent.

11th September 2014

Participantes

Daniel Frutuoso (Estagiário)
Eng^{o} André Pimentel (Orientador da EyeSee)
Eng^{o} Joao Redol (Orientador da EyeSee)
Bernardete Ribeiro (Orientadora do DEI)

Resumo

A reunião decorreu com a presença dos responsáveis da EyeSee, Eng. João Redol e Eng. André Pimentel, no Laboratório LARN, Torre E, Piso 6, Sala, das 14:30h-17:30h, e teve a seguinte ordem de trabalhos:

1. Apresentação da empresa
2. Objectivos dos estágio
3. Tecnologias
4. Datasets

5. Planificação do estágio

6. Templates dos reports do 1^o semestre

7. Local do estágio: LARN/DEI

Foram discutidos vários aspetos do Estágio tendo o Estagiário ficado de entregar até ao final de Setembro o estado da arte sobre aplicações e plataformas de monitorização de Diabetes. Foram estabelecidos reuniões regulares com os responsáveis da EyeSee por skype. Foi criada uma partilha de endereços de skype para se poder dar continuidade às reuniões quinzenais e mensais. Foram entregues chaves do LARN e arranjada a logística para que possa trabalhar no laboratório. Foi ainda o Estagiário convidado a visitar a empresa ainda durante o primeiro semestre.

29th September 2014

Participantes
Daniel Frutuoso (Estagiário)
Bernardete Ribeiro (Orientadora do DEI)

Resumo
A reunião de hoje foi via Skype iniciando-se por volta das 15h e terminando, sensivelmente, 1h depois. Nela foram discutidos os seguintes pontos:

1. Estrutura do relatório com principal foco no State of the Art

2. Qualidade dos artigos encontrados e a serem encontrados pelos estagiários

3. Dicas sobre a elaboração do State of the Art

4. Data de entrega do State of the Art

5. Formatação e cuidados a ter com a bibliografia

6. Questões de logística nomeadamente a possibilidade de serem colocados monitores externos no laboratório para os estagiários o que não é possível devido à falta dos mesmos para empréstimo

7. Obtenção de permissão para usar a impressora localizada na área do CISUC

8. Marcação de nova reunião para a próxima sexta-feira no mesmo horário

Foi também fornecido aos estudantes, por parte da coordenadora, um website onde será possível encontrar mais artigos e toolboxes que poderão ser usadas no futuro.

16th October 2014

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Resumo

Nesta reunão foram esclarecidas dúvidas acerca do classificador a construir.

22nd October 2014

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Resumo

Nesta reunião foram discutos os seguintes assuntos:

1. Esclarecimento do que é pretendido com Model Definition for Anomalies Characterization.

2. Troca de ideias para arranjar casos negativos para o classificador por mim proposto.

14th November 2014

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Eng^{o} André Pimentel (Orientador da EyeSee)

Resumo

Esta reunião teve a seguinte ordem de trabalhos:

1. Ponto de situação do projecto
2. Apresentação e justificação do uso dos datasets
3. Apresentação das abordagens já tomadas
4. Pequena discussão dos resultados
5. Foi apontado a mais valia da construção do dataset e a necessidade de haver persistência por parte do estagiário junto de vários locais para o conseguir fazer
6. Agendamento de uma reunião presencial em Lisboa no dia 15 de Dezembro
7. Agendamento de uma hora para a explicação de sugestões por parte da Professora a incorporar no relatório

15th December 2014

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Eng^{o} André Pimentel (Orientador da EyeSee)

Eng^{o} Joao Redol (Orientador da EyeSee)

Resumo

Esta reunião teve a seguinte ordem de trabalhos:

1. Apresentação da situação do projecto
2. Apresentação dos resultados preliminares
3. Discussão de alguns pontos do trabalho

02nd February 2015

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Eng^{o} André Pimentel (Orientador da EyeSee)

Resumo

Esta reunião serviu para realizar um dry run da apresentação para a defesa intermédia.

26th February 2015

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Eng^{o} André Pimentel (Orientador da EyeSee)

Resumo

Nesta reunião foram esclarecidas dúvidas gerais que o estagiário tinha.

25th March 2015

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Eng^{o} André Pimentel (Orientador da EyeSee)

Resumo

No dia 25 de Março de 2015 pelas 14:30h, no Laboratório de Redes Neuronais, teve lugar uma das reuniões intermédias. Feita uma pequena apresentação, por parte do estagiário Daniel Frutuoso, sobre o actual estado e os resultados obtidos foram discutidos os seguintes pontos:

1. Tratamento de missing values: procurar mais métodos, escolher um conjunto deles e apresentar o melhor justificando a escolha

2. Discussão geral dos algoritmos e abordagens exploradas.
3. Corrigir o formulário criado para a recolha de dados: colocar o termo mais conhecido para cada tipo das pressões sanguíneas
4. Explorar a possibilidade de prever a glucose com um modelo polinomial
5. Investigar sobre as Generalized Regression Neural Networks para o problema da monitorização
6. Reunião com um profissional dos CHUC para a obtenção de dados (a agendar)
7. Terminada a fase de discussão deu-se por encerrada a reunião às 18h.

13th May 2015

Participantes

Daniel Frutuoso (Estagiário)

Eng^{o} André Pimentel (Orientador da EyeSee)

Resumo

No dia 13 de Maio de 2015 pelas 15:00h, na sede da Associação Protectora dos Diabéticos de Portugal (APDP) em Lisboa, teve lugar uma reunião com o objectivo de apresentar o projecto SMITH.

Foi feita uma apresentação do projeto em que o estagiário está a trabalhar por parte do mesmo e ainda foram apresentados alguns resultados obtidos. Dado isto, os resultados foram discutidos com o Dr. Rogério que os classificou como bons. Relativamente ao problema do diagnóstico da diabetes, o Dr. Rogério explicou-nos que existe já uma ferramenta simples e aprovada a nível nacional e também noutros países (Findrisk) para o calculo do risco de um indivíduo vir a sofrer de diabetes. Ainda referente a esse problema, foi discutida a lista de parâmetros que constitui o dataset em processo de criação e uma comparação entre o Findrisk e o nosso projecto. No que toca ao problema da monitorização da glucose, e como já foi referido, o Dr. Rogério achou os resultados bastante interessantes. Daí surgiu a possibilidade de criar um protocolo de colaboração de forma a que fossem facultados mais dados e assim verificar o desempenho dos modelos com estes novos dados.

Terminada a fase de discussão deu-se por encerrada a reunião por volta das 16.30h.

26th May 2015

Participantes

Daniel Frutuoso (Estagiário)

Bernardete Ribeiro (Orientadora do DEI)

Eng^o André Pimentel (Orientador da EyeSee)

Resumo

No dia 26 de Maio de 2015 pelas 11:00h, no Laboratório de Redes Neuronais, teve lugar uma das reuniões intermédias. Feita uma pequena apresentação, por parte do estagiário Daniel Frutuoso, sobre o actual estado e os resultados obtidos foram discutidos os seguintes pontos:

1. Verificar a nomenclatura de certos métodos de previsão
2. Analisar e incorporar o Findrisk no relatório
3. Importância da discussão de resultados e comparação com os do SOTA
4. Realizar experiências com os algoritmos já implementados com os dados fornecidos pela APDP
5. Data de entrega final

Terminada a fase de discussão deu-se por encerrada a reunião às 12.30h.

Bibliography

[1] European Union. Research and innovation — digital agenda for europe — european comission, 2015. URL https://ec.europa.eu/digital-agenda/en/research-and-innovation-ehealth. [Accessed Jun. 19, 2015].

[2] International Diabetes Federation. Idf diabetes atlas. Technical report, 2013. URL http://www.idf.org/sites/default/files/EN_6E_Atlas_Full_0.pdf. [Accessed Jul. 3, 2015].

[3] American Diabetes Association®. Diagnosing diabetes and learning about prediabetes, 2014. URL http://www.diabetes.org/diabetes-basics/diagnosis/. [Accessed Jul. 3, 2015].

[4] W. Yu, T. Liu, R. Valdez, M. Gwinn, and M. Khoury. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and prediabetes. *BMC Medical Informatics and Decision Making*, 10(1):16, 2010. ISSN 1472-6947. doi: 10.1186/1472-6947-10-16. URL http://www.biomedcentral.com/1472-6947/10/16.

[5] V. Kumari and R. Chitra. Classification of diabetes disease using support vector machine. *International Journal of Engineering Research and Applications*, 3(2):1797–1801, 2013.

[6] J. Wu, Y. Diao, M. Li, Y. Fang, and D. Ma. A semi-supervised learning based method: Laplacian support vector machine used in diabetes disease diagnosis. *Interdisciplinary Sciences: Computational Life Sciences*, 1(2):151–155, 2009. ISSN 1913-2751. doi: 10.1007/s12539-009-0016-2. URL http://dx.doi.org/10.1007/s12539-009-0016-2.

[7] G. Kumar, G. Ramachandra, and K. Nagamani. An efficient feature selection system to integrating svm with genetic algorithm for large medical datasets. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(4):272–277, 2014. ISSN 2277 128X. URL http://www.ijarcsse.com/docs/papers/Volume_4/2_February2014/V4I2-0164.pdf.

[8] K. Polat and S. Güneş. An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. *Digital Signal Processing*, 17(4):702 – 710, 2007. ISSN 1051-2004. doi: http://dx.doi.org/10.1016/j.dsp.2006.09.005. URL http://www.sciencedirect.com/science/article/pii/S1051200406001370.

[9] S. Polat, K. Güneş and A. Arslan. A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine. *Expert Systems with Applications*, 34(1):482 – 487, 2008. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2006.09.012. URL http://www.sciencedirect.com/science/article/pii/S0957417406002995.

[10] C. Marling, M. Wiley, T. Cooper, R. Bunescu, J. Shubrook, and F. Schwartz. *The 4 Diabetes Support System: A Case Study in CBR Research and Development*, volume 6880 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23290-9. doi: 10.1007/978-3-642-23291-6_12. URL http://dx.doi.org/10.1007/978-3-642-23291-6_12.

[11] G. Sparacino, F. Zanderigo, S. Corazza, A Maran, A Facchinetti, and C. Cobelli. Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series. *Biomedical Engineering, IEEE Transactions on*, 54(5): 931–937, May 2007. ISSN 0018-9294. doi: 10.1109/TBME.2006.889774.

[12] G. Baghdadi and AM. Nasrabadi. Controlling blood glucose levels in diabetics by neural network predictor. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 3216–3219, Aug 2007. doi: 10.1109/IEMBS.2007.4353014.

[13] C. Zecchin, A Facchinetti, G. Sparacino, G. De Nicolao, and C. Cobelli. Neural network incorporating meal information improves accuracy of short-time prediction

of glucose concentration. *Biomedical Engineering, IEEE Transactions on*, 59(6): 1550–1560, June 2012. ISSN 0018-9294. doi: 10.1109/TBME.2012.2188893.

[14] C. Marling, M. Wiley, R. Bunescu, J. Shubrook, and F. Schwartz. Emerging applications for intelligent diabetes management. *AI Magazine*, 33(2):67, 2012.

[15] E. Georga, V. Protopappas, D. Polyzos, and D. Fotiadis. Predictive modeling of glucose metabolism using free-living data of type 1 diabetic patients. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 589–592, Aug 2010. doi: 10.1109/IEMBS.2010.5626374.

[16] Diagnoptics. Diab-spot: Diabetes screening, 2011. URL http://www.diagnoptics.com/en/diab-spot/. [Accessed Jul. 3, 2015].

[17] M. Peppa, J. Uribarri, and H. Vlassara. Glucose, advanced glycation end products, and diabetes complications: What is new and what works. *Clinical Diabetes*, 21(4):186–187, 2003. doi: 10.2337/diaclin.21.4.186. URL http://clinical.diabetesjournals.org/content/21/4/186.short.

[18] J. Lindström and J. Tuomilehto. The diabetes risk score a practical tool to predict type 2 diabetes risk. *Diabetes care*, 26(3):725–731, 2003.

[19] Bayer. Diabetes product portfolio from Bayer, 2014. URL http://diabetes.bayer.com/products/. [Accessed Jul. 3, 2015].

[20] LifeScan. Diabetes supplies & information - blood glucose meters, test strips, lancets & more - life first, 2014. URL http://www.onetouch.com. [Accessed Jul. 3, 2015].

[21] Medtronic, 2014. URL http://www.medtronicdiabetes.com/home. [Accessed Jul. 3, 2015].

[22] Dexcom, 2014. URL http://www.dexcom.com. [Accessed Jul. 3, 2015].

[23] K. Bache and M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml. [Accessed Jul. 3, 2015].

[24] J. Breault. Data mining diabetic databases: Are rough sets a useful addition. In *In Proc. 33rd Symposium on the Interface, Computing Science and Statistics*, 2001.

[25] P. Laencina, J. Gómez, and A. Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010. ISSN 0941-0643. doi: 10.1007/s00521-009-0295-6. URL http://dx.doi.org/10.1007/s00521-009-0295-6.

[26] H. Kang. The prevention and handling of the missing data. *Korean J Anesthesiol*, 64(5):402–406, 2013. doi: 10.4097/2Fkjae.2013.64.5.402. URL http://synapse.koreamed.org/DOIx.php?id=10.4097%2Fkjae.2013.64.5.402.

[27] B. Yap, K. Rani, H. Rahman, S. Fong, Z. Khairudin, and N. Abdullah. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In Tutut Herawan, Mustafa Mat Deris, and Jemal Abawajy, editors, *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, volume 285 of *Lecture Notes in Electrical Engineering*, pages 13–22. Springer Singapore, 2014. ISBN 978-981-4585-17-0. doi: 10.1007/978-981-4585-18-7_2. URL http://dx.doi.org/10.1007/978-981-4585-18-7_2.

[28] M. de Souto, V. Bittencourt, and J. Costa. An empirical analysis of undersampling techniques to balance a protein structural class dataset. In Irwin King, Jun Wang, Lai-Wan Chan, and DeLiang Wang, editors, *Neural Information Processing*, volume 4234 of *Lecture Notes in Computer Science*, pages 21–29. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-46484-6. doi: 10.1007/11893295_3. URL http://dx.doi.org/10.1007/11893295_3.

[29] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 3rd edition, 2011.

[30] B. Ribeiro. Pattern recognition techiques' course slides, 2013.

[31] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 0885-6125. doi: 10.1007/BF00994018. URL http://dx.doi.org/10.1007/BF00994018.

[32] B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Learning*. Philomel Books, 1999. ISBN 9780262194167. URL http://books.google.pt/books?id=_NYamXKkNM8C.

[33] J. Sa. *Pattern Recognition: Concepts, Methods, and Applications*. Springer, 2001. ISBN 9783540422976. URL http://books.google.pt/books?id=O5vwppJQQwIC.

[34] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A%3A1010933404324.

[35] R Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[36] B. Zaitlen, L. Desborough, and A. Ahmadia. Continuos Glucose Monitor Data. 07 2013. URL http://dx.doi.org/10.6084/m9.figshare.741296. [Accessed Jul. 3, 2015].

[37] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines, 1996.

[38] D. Specht. A general regression neural network. *Neural Networks, IEEE Transactions on*, 2(6):568–576, Nov 1991. ISSN 1045-9227. doi: 10.1109/72.97934.

[39] L. Zhang, W. Zhou, P. Chang, J. Yang, and F. Li. Iterated time series prediction with multiple support vector regression models. *Neurocomputing*, 99(0):411 – 422, 2013. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2012.06.030. URL http://www.sciencedirect.com/science/article/pii/S0925231212005863.

[40] S. Taieb, A. Sorjamaa, and G. Bontempi. Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing*, 73(10–12):1950 – 1957, 2010. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2009.11.030. URL http://www.sciencedirect.com/science/article/pii/S0925231210001013. Subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

[41] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse. Methodology for long-term prediction of time series. *Neurocomputing*, 70(16–18):2861 – 2869, 2007. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2006.06.015. URL http://www.sciencedirect.com/science/article/pii/S0925231207001610. Neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN 2005).

[42] L. Cao and F. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *Neural Networks, IEEE Transactions on*, 14(6):1506–1518, Nov 2003. ISSN 1045-9227. doi: 10.1109/TNN.2003.820556.

[43] Google. Google Developers - Google Fit, 2014. URL https://developers.google.com/fit/. [Accessed Jul. 3, 2015].