Fall 2019

Non-invasive Hyperglycemia Detection using ECG and Deep Learning

Renato Silveira Cordeiro
San Jose State University

NON-INVASIVE HYPERGLYCEMIA DETECTION USING ECG AND DEEP LEARNING

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Renato Silveira Cordeiro

December 2019

The Designated Thesis Committee Approves the Thesis Titled

NON-INVASIVE HYPERGLYCEMIA DETECTION USING ECG AND DEEP LEARNING

by

Renato Silveira Cordeiro

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

December 2019

Nima Karimianbahnemiri, Ph.D.	Department of Computer Engineering
Young Park, Ph.D.	Department of Computer Engineering
Chandrasekar Vuppalapati, M.S.	Department of Computer Engineering

ABSTRACT

NON-INVASIVE HYPERGLYCEMIA DETECTION USING ECG AND DEEP LEARNING

by Renato Silveira Cordeiro

Hyperglycemia is characterized by an elevated level of glucose in the blood. It is normally asymptomatic, except for an extremely high level, and thus a person can live in that state for years before the negative - sometimes irreversible - health impacts appear. Unexpected hyperglycemia can also be an indication of diabetes, a chronic disease that, when not treated, can lead to serious consequences, including limb amputations and even death. Therefore, identifying hyperglycemic state is important. The most common and direct way to measure a person's glucose level is by directly assessing it from a blood sample by pricking a finger, which causes discomfort and even pain. The constant finger pricking can also lead to bruising and increases the possibility of infection. This thesis presents a non-invasive technique of detecting hyperglycemia by using a person's electrocardiogram (ECG) and deep learning. The ECG signal is preprocessed to remove noise, identify fiducial points, extract and adjust features, remove outliers and normalize the data. This thesis applied a novel approach to feature extraction in which, instead of just using fiducial amplitudes and intervals, a direct line was drawn between fiducial points and its length and slope were used as features. The labeled features were used in 10-layer deep neural network and resulted in an area under the curve (AUC) of 94.53%, sensitivity of 87.57% and specificity of 85.04%. Such strong performance indicates that ECG carry intrinsic information that can be used to identify hyperglycemic state, enabling the use of ECG-based hardware together with deep learning for non-invasive hyperglycemia detection.

ACKNOWLEDGMENTS

The development of this work was a substantial effort that would not be possible without the support of several people.

First, I would like to thank my advisor Prof. Nima Karimian, PhD who was fundamental in guiding me in the world of biometrics. The contribution to the field that this thesis aims for would not be possible without his orientation.

I would like also thank Dan Harkey, my academic program coordinator, and Susan Alston, my academic program specialist. They went far and beyond to make sure that I had the opportunity to learn and improve as much as possible during my time at SJSU.

In addition I thank Chun-Min Chang and the Research Center for Applied Sciences, Academia Sinica, Taiwan for providing the ECG dataset used on this thesis.

I would not be able to attend this university without the support of my family: father Claudio, mother Celina, sister Raquel, brother-in-law Andre, brother Rafael, sister-in-law Luana and mother-in-law Janete. Thank you all.

Last, but not least, I would like to thanks my wife Letícia Cordeiro for the unequivocal support during this whole time and for making double shift taking care our 3-year old son and 9-month twin daughters while I was preparing this material. Without doubt she was the one with the most challenging - and important - job. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN – Artificial neural networks
AV – Atrioventricular
BioSPPy – BioSignal Processing in Python
BBNN – Block-Based Neural Network
BGC – Blood Glucose Concentration
CNN – Convolutional Neural Networks
DNN – Deep Neural Networks
ECG – Electrocardiogram
HBV – Hepatitis B Virus
HCV – Hepatitis C Virus
HIV – Human Immunodeficiency Virus
IQR – Interquartile Range
LSTM – Long Short-Term Memory
PPG – Photoplethysmography
RNN – Recurrent Neural Networks
SA – Sinoatrial
SGD – Stochastic Gradient Descent
SVM – Support Vector Machine

1 INTRODUCTION

Hyperglycemia, also known as high blood sugar, is a physiological state in which there is a high concentration of glucose in the bloodstream. This state naturally occurs following meals but after a few hours, the body should return to a normoglycemic state mainly due to work of the hormone insulin [1]. Hyperglycemia is usually defined as a blood glucose concentration (BGC) of 100 mg/dl or higher in a fasting state. A person in a constant state of hyperglycemia, especially when fasting, can be in a situation in which the body is not able to process the glucose anymore. This is usually a consequence of insulin resistance, a condition in which the body becomes insensitive to insulin and the hormone is not able to decrease blood glucose concentration. Insulin resistance is a marker for type 2 diabetes mellitus [2], a disease that when not treated, can lead to serious health problems including blindness, limb amputation, heart diseases, and even death.

The traditional method for measuring blood glucose is via assessment of glucose concentration in a blood sample. The sample can be a few drops of blood acquired by pricking the finger or a larger amount obtained by health professionals. Both processes are invasive and generate pain and discomfort, creating a barrier for their widespread use as a screening mechanism. The exposure of blood also creates another under-recognized, but still critical problem which is the possibility of bloodborne pathogen transmissions [3] such as the hepatitis B/C virus (HBV / HCV), human immunodeficiency virus (HIV) or others due to the sharing of blood glucose meters or accessories among infected people. [4] states that 15 out of 18 HBV infections outbreaks since the year 1990 were attributed to the improper use of blood glucose monitoring systems. Lastly, the use of such invasive systems also creates a significant environmental impact due to the creation of medical waste [5]. Therefore, a non-invasive method with little waste to detect hyperglycemia would be useful for individuals and society in general.

Machine learning is the process of automatically identifying useful patterns in data [6]. It is a cross-industry tool that enables the clusterization or classification of data based on explicit or implicit characteristics of data called features. It has been largely used for problems such as customer segmentation, behavior prediction, and others. Recently it has been applied more often to healthcare to identify patterns of disease or biomarkers not directly visible to the human eye. Deep learning models, which are a specific type of machine learning models, have been achieving promising results in that area.

This thesis presents a deep learning model that is capable of identifying hyperglycemia using an electrocardiogram (ECG). In addition, this thesis presents a novel feature extraction mechanism from ECG signals focused on slopes and direct distances between fiducial points. This resulted in a feature size reduction of 97% when compared to a full ECG cardiac cycle.

2 LITERATURE REVIEW

Several studies investigating the impact of blood glucose concentration on different ECG metrics have been conducted. Amanipour et al. [7] analyzed the heart rate variability (HRV) frequency domain components of a diabetic female subject under normoglycemic and hyperglycemic conditions. They noticed a 6-fold decrease in the low frequency / high frequency ratio. Although the study was limited to just one person, it corroborated the results achieved by Fujimoto et al. [8] showing that the ratio was negatively correlated with blood glucose concentration. Perpiñan et al. [9] assessed the impact of taking a 75 g oral glucose test in subjects with metabolic syndrome. A control group was used and both groups had their HRV monitored after drinking the glucose solution. After 30 minutes, the metabolic syndrome group presented significantly higher HRV irregularity than the control group. After 60 minutes of the glucose intake, the HRV irregularity in subjects with metabolic syndrome decreased while that was not observed in the control group.

Another metric known to be impacted by blood glucose levels is the QT interval of the cardiac cycle. Suys et al. [10] monitored the ECG and blood glucose concentration of type 1 diabetic children by using a Holter and continuous glucose monitoring device. They identified a prolongation of QT and QTc intervals (QT interval adjusted by Bazett's formula) with lower blood glucose concentration. Christensen et al. [11] also studied the relationship between hypoglycemia and QTc interval in a group of type 1 diabetic adults and identified a moderate increase in the QTc interval. Marfela et al.'s [12] research on the impact of blood glucose in QT duration achieved a different result. More specifically, they found that acute hyperglycemia in healthy patients resulted in a significant increase in QTc interval, QTc dispersion, and PR interval. That apparent contradiction with existing literature that shows QTc prolongation associated with hypoglycemia could be explained by the fact that their study was composed only of healthy subjects instead of diabetic patients.

Nguyen et al. [13] analyzed the effect of hypoglycemia and hyperglycemia on several ECG parameters, including HR, QTc, PR, RTc, TpTec (T-peak to T end). They identified that low blood glucose was associated with prolonged RTc, QTc and TpTec. In contrast, high blood concentration was related not only to a decreased RTc, QTc, TpTec but also to an increased PR. In 2014, Naguyen et al. [14] went one step further by proposing a neural network model to detect hyperglycemia using 16 features extract from an ECG. This ECG features are summarized in Table 1.

Table 1
ECG Features List Used to Calculate Hyperglycemia

#	**Feature**	**Type**
1	HR	Intervals
2	PR	Intervals
3	QTc	Intervals
4	RTc	Intervals
5	TpTec	Intervals
6	Mean RR interval	Time-domain
7	Standard deviation of the RR Interval index (SDNN)	Time-domain
8	Root mean square of successive RR interval differences (RMSSD)	Time-domain
9	Percentage of consecutive RR intervals that differ by more than 50ms (pNN50)	Time-domain
10	HRV triangular index (HRVi)	Time-domain
11	Baseline width of the RR interval histogram evaluated through triangular interpolation (TINN)	Time-domain
12	Very low frequency (VLF)	Frequency-domain
13	Low frequency (LF)	Frequency-domain
14	High frequency (HF)	Frequency-domain
15	Total spectral power (TotalPw)	Frequency-domain
16	LF / HF ratio	Frequency-domain

Note: adapted from Nguyen et al. [14]

Their best model achieved moderate performance with a sensitivity of 70.59% and specificity of 65.38%. The dataset used was also very limited since it was restricted to only 10 subjects.

To overcome the limitations of the aforementioned study, this thesis introduces a new deep learning architecture along with a novel feature extraction technique and large dataset to improve performance.

3 THEORY BACKGROUND

3.1 Electrocadiogram

3.1.1 What is ECG

Electrocardiograms (ECGs) are recordings of cardiac electrical activity produced by depolarization and repolarization of atria and ventricles during each cardiac cycle [15]. Human heart signals were first captured by British physiologist Augustus Waller in 1887 by using a capillary electrometer and in 1903, Dutch physiologist Willem Einthoven created the first practical ECG machine based on a string galvanometer, a device that uses the electromagnetic field to detect very small currents. His contribution to clinical electrocardiography was recognized with the Nobel Prize of Physiology in 1924 [15]. ECG works by measuring the potential difference between electrodes placed in different locations of the human skin. The quantity and location of the electrodes can vary and are related to how many electrical views (also called ”leads”) are to be captured. Single-lead ECG readers use two or three electrodes to capture one view of heart electrical activity. This has been recently used on products for consumer use such as wristbands and smart watches. Although those devices provide just one electrical view (single lead), this is enough to provide HR and HRV calculation as well as detection of an arrhythmia. Another common configuration is the 12-lead ECG that uses 10 electrodes, which is the standard in clinical environments. More details about leads and the quantity and locations of electrodes are provided in section 3.1.3.

The human heart is often referred to as a pump that sends blood throughout the body. It is divided into 4 chambers as can be seen in Fig. 1. The top chambers are called the left and right atria and the bottom chambers left and right ventricles. The left atrium receives the oxygen-rich blood from the lungs which is passed to the left ventricle. From there the blood is pumped to the rest of the body, returning to the heart via the right atrium. This oxygen-depleted blood is then sent to the right ventricle and then to the lungs. A

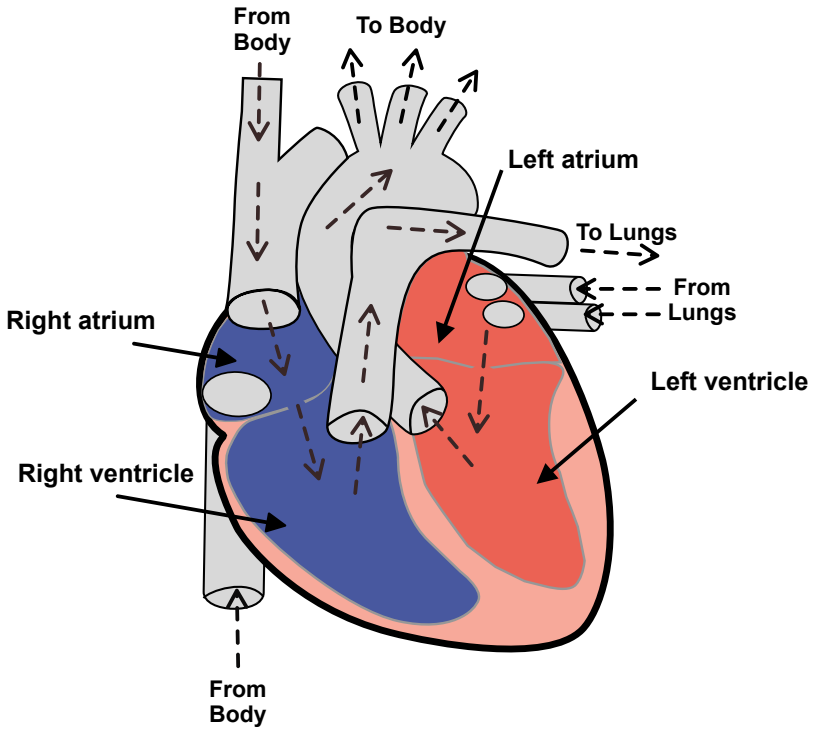

Fig. 1. Heart with a normal flow and its 4 chambers. Picture created by the author.

schematic of the blood flow inside the heart can also be seen in Fig. 1. Such movement is a consequence of cardiac muscle contraction triggered by an electrical impulse regularly generated by a group of cells located in the top part of the right atrium called the sinoatrial node (SA node or sinus node), also often referred as the heart's pacemaker [16]. The electrical impulse spreads through the heart cells making the atria and ventricles contract and relax sequentially, sending the blood from the heart to the rest of the body (organs and lungs). As mentioned before, it is possible to measure that electrical impulse as it moves through the heart with the use of electrodes placed on top of a person's skin.

3.1.2 ECG Signal and Main Components

The signal captured by ECG is a combination of waves that succeed each other in a period manner as the heart goes through its cardiac cycles. An illustration of an ECG signal can be seen in Fig. 2.

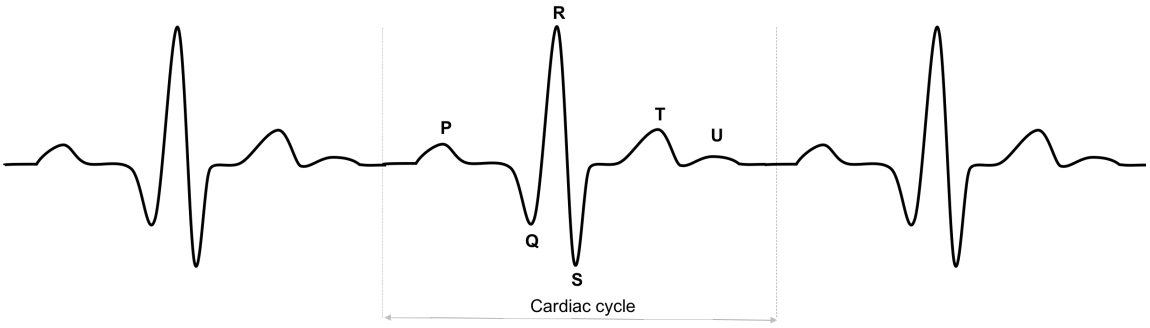

Fig. 2. Electrical impulse captured by ECG. Picture created by the author.

Einthoven named each of the waves inside a cardiac cycle with a letter: P, Q, R, S, T and U. That last one is not common and appears to be present in some people under specific situations. Each of the waves reflects a different time of the electrical impulse moving through the heart cells and abnormalities in the shape or duration of a wave can indicate the presence of a heart disease condition. Each wave is further detailed below.

3.1.2.1 P-wave: The first noticeable wave in a cardiac cycle is called P-wave and represents the contraction of the left and right atria due to a depolarization triggered by the SA node. This phenomenon is also called atrial systole. The normal duration for a P-wave is between 0.06 to 0.12 s and its amplitude falls between 0.02 and 0.03 mV.

3.1.2.2 Q-wave: After the electrical impulse propagates through the atria walls, it moves downwards and reaches the atrioventricular node (AV node) located at the interatrial septum. The Q-wave reflects the septal depolarization and has a duration and amplitude of less than 0.04 s and 0.02 mV, respectively.

3.1.2.3 R-wave: The R-wave reflects the electrical impulse flowing through the ventricular walls. It is the most prominent wave of the entire cardiac cycle and therefore used as a base point for several analyses, including heart rate calculation via the R-R interval.

3.1.2.4 S-wave: The Purkinje fibers are the last area to be depolarized and the S-wave reflects that. Different from other waves, the S-wave does not present great significance alone as an indicator of heart disease.

3.1.2.5 T-wave: Before the next heartbeat cycle can occur, the ventricles need to repolarized and that is what the T-wave represents. The repolarization of the other areas was already occurring concomitantly with the depolarization of the ventricles as can be seen in Fig. 3.

3.1.2.6 U-wave: The last wave in the cycle is the U-wave. This is a peculiar one since it is normally not seen and only has been reported to be present in some subjects under the influence of alcohol [17] or antiarrhythmic drugs, or as a consequence of electrolyte imbalances or drug toxicity [16]. Its origin is still not clear but it is believed that it is due to repolarization of the Purkinje fibers [16].

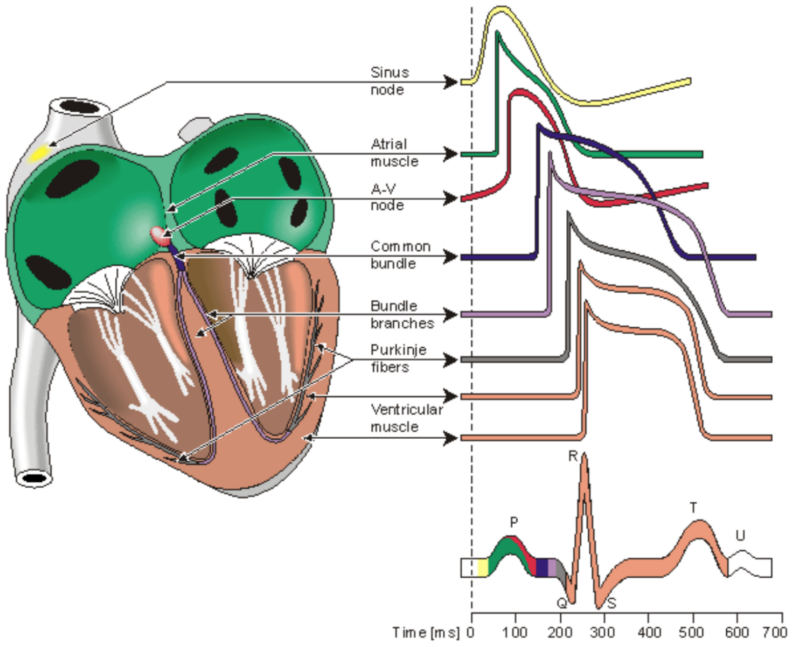

Fig. 3. Cardiac cycle composition. Reprinted with permission from [15].

3.1.2.7 QRS complex: The combination of the Q, R and S waves is known as the QRS complex and it reflects the left and right ventricles' depolarization. Its normal duration is between 0.06 and 0.12 s [16].

3.1.3 Capturing ECG Signal

As mentioned in section 3.1.1, ECG uses electrodes to capture the electrical activity of the heart. It works by measuring the potential difference between 2 electrodes, and thus, that is the minimum quantity required to have one ECG signal.

The signal captured by a pair of electrodes represents a specific view of the heart's electrical activity and is dependent on the electrodes' location on the body. That specific view of the heart is also referred to as lead or angle. Therefore, a single-lead ECG reader provides just one specific view of the heart. In the same way, a 12-lead ECG reader provides 12 views. It is important to not confuse that lead (view) term with the actual metal piece that is connected to the electrode, which is also called lead. Fig. 4 shows a single-lead ECG reader with 3 electrodes.

Fig. 4. Single-lead ECG reader and 3 electrodes. Picture taken by the author.

3.1.3.1 12-lead ECG: The most common setting in hospitals and clinical environments is the 12-lead ECG which is achieved by the use of 10 electrodes. Six of those leads are called limb leads since they are obtained by using 4 electrodes placed on

the person's limbs (left arm, right arm, left leg, right leg). The remaining 6 leads are called precordial leads and are obtained with 6 electrodes placed on the person's chest. Fig. 5 illustrate the location of all 10 electrodes.

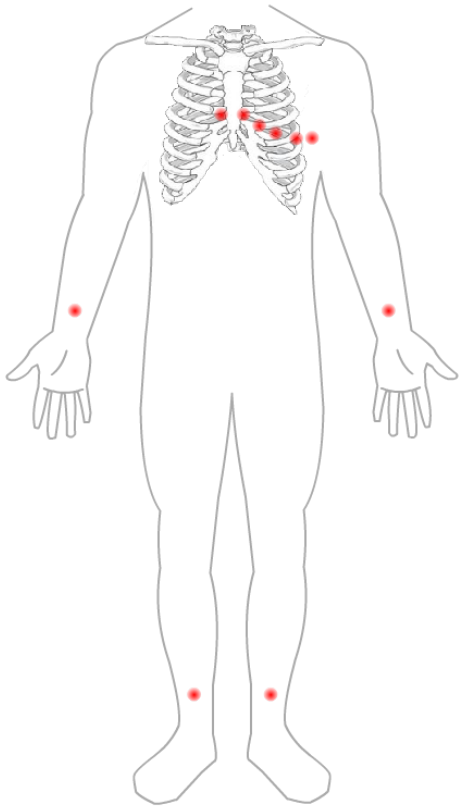

Fig. 5. Location of electrodes in a 12-Lead ECG. Picture created by the author.

The 6 limb leads are named lead I, lead II, lead III, lead aVR, lead aVL and lead aVF. The last 3 leads are actually virtual leads obtained from the combination of the former 3 leads (I, II, III) and starts with the prefix "a" to indicate "augmented". The 6 precordial leads are called V1, V2, V3, V4, V5, and V6. The ECG signals showed so far in this thesis is the one provided by lead II, in which we have a P-wave with positive amplitude, a complete QRS complex (small negative amplitude followed by a high positive one and then by a negative one) with a T-wave with positive amplitude at the end of the cardiac

cycle. The other leads would show some differences such as a P-wave with negative amplitude or different wave amplitudes.

3.1.3.2 Single-lead ECG: A simpler setting is the single-lead ECG which captures just one electrical signal with the use of 2 or 3 electrodes. That ECG view is equivalent to the lead II of a 12-lead ECG. Several personal ECG handheld devices available in the market, as well as other wearable and mobile devices with ECG reading capabilities, are single-lead. Although having just one lead means less information from the heart than the use of a 12-lead, it can still provide very useful information such as HR, HRV and arrhythmia detection. Fig. 6 shows an example of a single-lead handheld ECG reader manufactured by the company AliveCor.

Fig. 6. Single-lead handheld ECG reader. Picture taken by the author.

3.1.4 ECG Noise

ECG measures electrical activity and, as with any measurement of biological signals, is subject to noise. In particular, ECG suffers from several types of noise [16], [18] that are described below.

3.1.4.1 Respiration baseline wandering: This noise is generated by the respiratory activity of the person. It is reflected as a low frequency signal ranging between 0.05 Hz and 1 Hz [18]. The use of a high pass filter with a frequency of 1 Hz or higher is

usually enough to remove this type of noise. Fig. 7 shows the effect of respiration baseline wandering in an ECG signal.

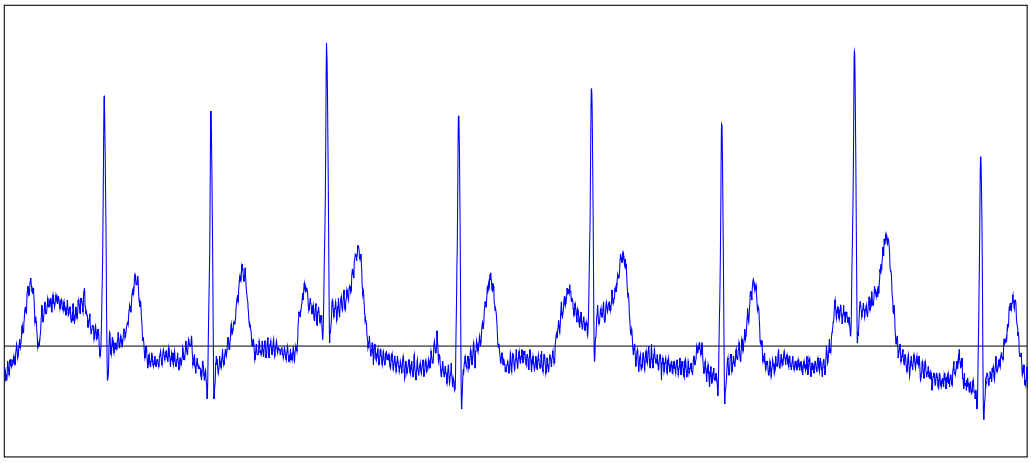

Fig. 7. ECG signal contaminated by baseline wandering noise.

3.1.4.2 Power line interference: ECG devices generate intrinsic noise due to their capacitive and inductive coupling. The noise due to capacitive coupling is usually centered at 50 Hz or 60 Hz [18] and therefore a low pass filter can be used to remove it. The inductive coupling produces low frequency noise that overlaps with the actual electrical signal being measured, making it difficult to remove without losing ECG data. [19] shows that a low frequency cut of 0.5 Hz already impacts the ST segment interpretation.

3.1.4.3 Muscle artifacts: Contractions of muscles near the heart can be captured by ECG, showing up as waves with small amplitudes. Other muscle movements can also be read and appear as normal noise. The former can be removed by preprocessing the signal and taking into consideration all the expected waves and respective amplitudes and duration inside a cardiac cycle. The latter can be difficult to remove [18].

3.1.4.4 Electrode motion artifacts: These artifacts are usually reflected as amplitudes spikes and sometimes as baseline drifts and are due to a person's movement during the ECG. It is very difficult to remove such noise.

3.1.4.5 Environment interference: The environment in which the person is located during the ECG can interfere in the measurements. Fluorescent light and other electronics such as cell phones can generate artifacts [16] and therefore it is recommended that appropriate actions are taken to remove, or at least decrease, such interference.

3.2 Machine Learning

3.2.1 What is Machine Learning

Machine learning is the process of automatically identifying useful patterns in data [6]. By identifying patterns, machine learning models can ”learn” from the data and apply that learning to data not yet seen. The data given to the model must be a collection of examples (also known as samples) that share one or more characteristics. A common characteristic across samples is called feature. Machine learning is usually segmented by its learning methods.

3.2.2 Learning Methods

Machine learning can be classified in 4 different methods: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. They mainly differ in the data structure given to the model and the objective. A brief description of each one is provided in the following subsections.

3.2.2.1 Supervised learning: In this method, the data used to train the model are labeled, so for each sample, the model receives the features (input) and also the label (output). The model treats that collection of inputs-outputs as conditions that it must try to fulfill by optimizing its internal parameters. The label could be a discrete or continuous value. If the former, it is said to be a classification problem, if the latter, a regression problem.

3.2.2.2 Unsupervised learning: The unsupervised learning method receives samples without labels, so the model works by finding similarities based on the features

provided. This method is commonly used for clustering and association problems and also for dimensionality reduction.

3.2.2.3 Semi-supervised learning: This method is very similar to unsupervised learning, but with the difference that some of the samples are labeled. Possible usage includes classification and clustering problems.

3.2.2.4 Reinforcement learning: Reinforcement learning works in a different way compared to the other methods. It receives a set of instructions or goals that it must achieve in a defined environment. The model tries to achieve that goal via trial and error. Examples of application of this method include control systems and games.

3.3 Artificial Neural Networks

Artificial neural networks (ANN) are machine learning models inspired by the human brain [6]. Neurons inside the brain are connected to each other in a structure that resembles a network and communicates with each other via electrical and chemical signals [16], [20]. Similarly, an artificial neuron is also connected to other artificial neurons and exchange information with each other. Fig. 8 illustrates a human neuron and an artificial neuron.

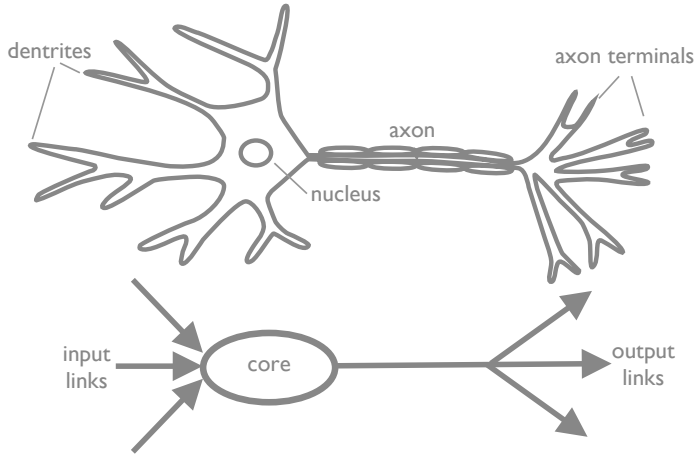

Fig. 8. Human neuron and artificial neuron. Picture created by the author.

The visual representation of an artificial neural network is composed of nodes and edges, with the former representing artificial neurons and the latter illustrating the existence of a connection between them.

3.3.1 Unit

The artificial neuron is also called node or unit. A unit receives inputs from other units and each input is multiplied by a weight. The unit sums all the weighted inputs with a bias and then passes the result into an activation function. An activation function has the purpose of adding non-linearity to the data, allowing the model to approximate complex functions. The result of the activation function is the output of the unit and is forwarded to the next unit. Fig. 9 illustrates a unit and its components.

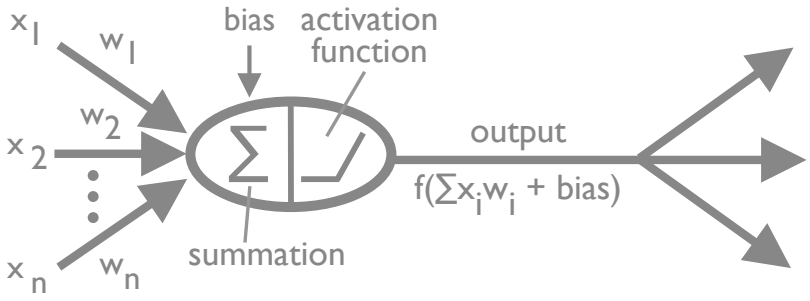

Fig. 9. Unit with its input, weights and activation function. Picture created by the author.

3.3.2 Feed-Forward Neural Networks

When the neural network has all connections going in the same direction it is called a feed-forward network [20]. Usually, the nodes are grouped together in layers and a node in one layer only receives inputs from nodes from the previous layers and sends its output to nodes in the next layer. An example of a 2-layer feed-forward network is shown in Fig. 10.

The first layer of a network is the input layer and the last one is the output layer. If the network has more than 2 layers, those extra layers are called hidden layers since they will be between the input and output layers. Neural networks are usually used in supervised

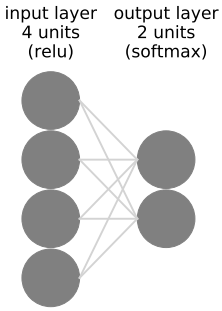

Fig. 10. Feed-forward network with 2 layers. Picture created by the author.

learning, although they can also be used in an unsupervised manner. In the former, the model receives a collection of samples that have features and labels. The model then starts learning from that data, a phase that is called training. During training, the model optimizes its internal parameters which in this case are the weight values used to multiply the inputs and also a bias which is a constant added to the weighted input. The model usually starts with arbitrary weights and update its weights based on a loss function that is calculated from the output of the model (predicted value) and the correct label (truth value). Based on the loss, the model goes back and updates its weight in a process known as back-propagation. This cycle repeats for a predefined number of loops (called epochs) or until the model reaches a specific predefined loss value.

3.3.3 Deep Neural Networks

A neural network with more than 2 layers is called a deep neural network and it is also classified as a deep learning architecture. The extra layers are denominated hidden layers since they are located between the input and output layers. Deep neural networks and deep learning, in general, have shown to be very effective in attacking complex problems and is a very active area of research [21].

4 HYPERGLYCEMIA DETECTION MODEL

4.1 Methodology

Extensive research and investigation of different fiducial features extraction methods, classification models, ANN architectures and hyperparameters were performed with the objective of identifying the best model to detect hyperglycemia from an ECG. A detailed list of models and configurations tested, as well as their performance, is found in section 4.5. The model with the best performance was compared against [14], the best hyperglycemia model in the literature. All of the work was done using the Python programming language, Biosignal Processing in Python (BioSPPy) library, NeuroKit framework for Python, Jupyter notebooks, Keras and Tensorflow.

4.2 Dataset

Several public ECG datasets are available for research purposes, specifically at the Physionet bank [22]. Unfortunately, in order to detect hyperglycemia, the dataset must include not only ECG data, but also glucose concentration measurements taken at the same time as the ECG. This thesis worked with a dataset kindly shared by fellow researcher Chun-Ming Chang from Academia Sinica, Taiwan. He had used this dataset in his research on using ECG and photoplethysmography (PPG) to predict blood glucose concentration [23].

The dataset was collected by the Research Center for Applied Sciences, Academia Sinica, Taiwan based on the following protocol:

- Each subject participated in two recording sessions.
- Each session consisted of the recording of a 60-second single-lead ECG and blood glucose concentration.
- ECG was acquired using Analog AD-8232 with a sampling rate of 1,000 Hz [24].
- Blood glucose concentration was measured using Accu-Chek Mobile blood glucose monitoring system [25].

A total of 1,119 subjects participated, composed of 386 females and 733 males with ages varying from 38 to 80 years old. The overall profile of the participants and the distribution of blood glucose concentration can be seen in Fig. 11. For this work, all 2,238 ECG recordings were analyzed and those informed with low quality were discarded, resulting in a dataset of 1,963 samples. The samples with glucose concentration higher than 100 mg/dl were labeled with hyperglycemia.

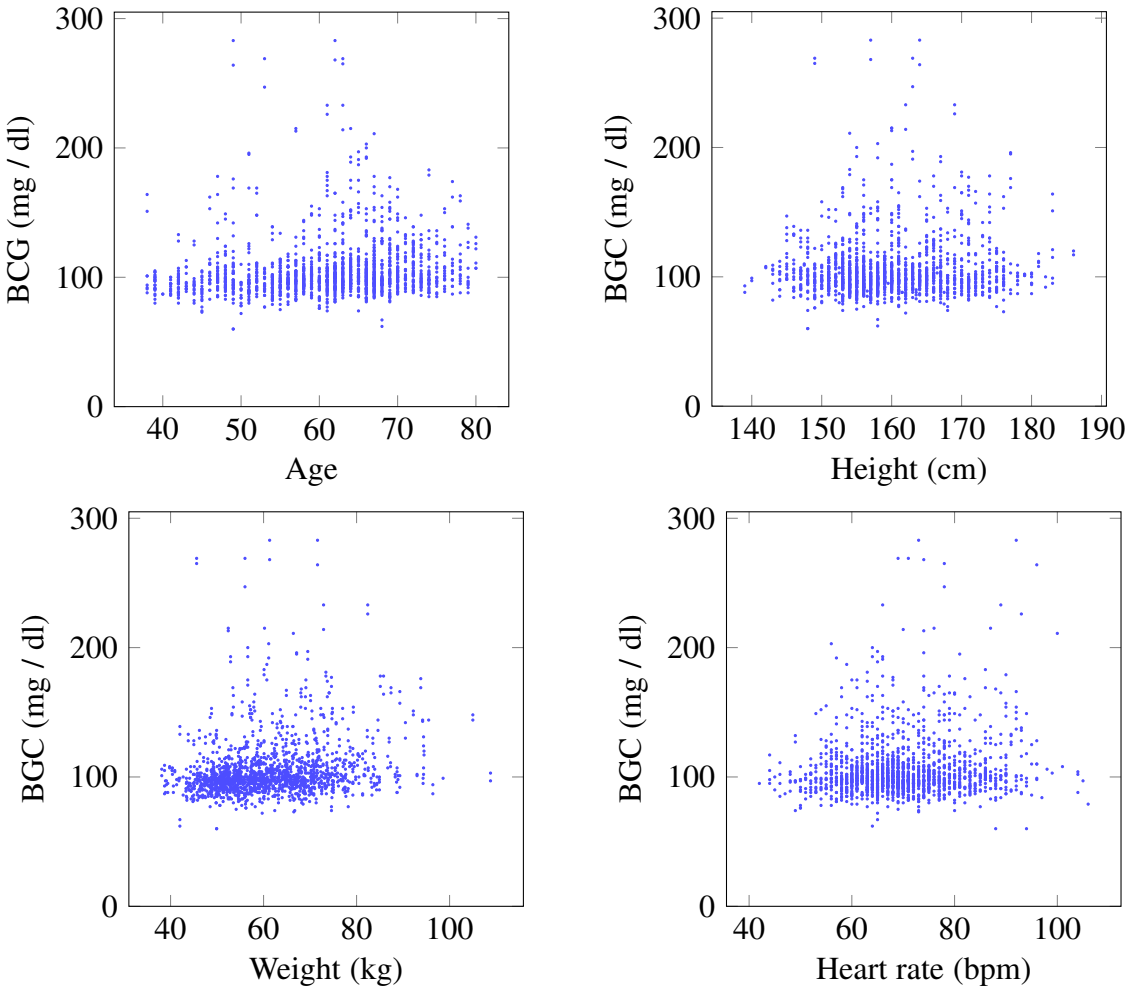

Fig. 11. Distribution of the blood glucose concentration with different profile information (age, height, weight, heart rate).

4.3 Preprocessing

A machine learning model is as good as the data it receives. Therefore, correctly understanding and preparing the data that will be provided to the model is often considered the most important part of a machine learning project. The preprocessing phase in this work consisted of 6 steps as illustrated in Fig. 12. Each of these steps is described in the following subsections.

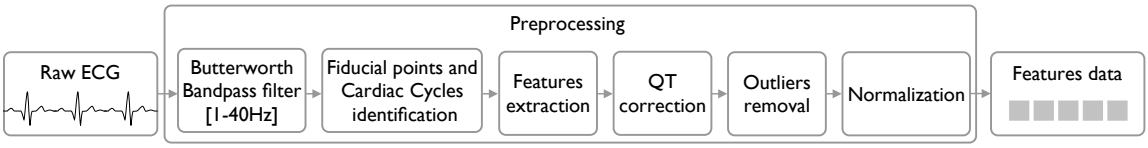

Fig. 12. Preprocessing steps. Picture created by the author.

4.3.1 Filtering

The filtering step removes (or at least attenuates) the noise mentioned in section 3.1.4. It is a fundamental task that improves the performance of the remaining preprocessing steps, reducing processing time and fiducial points misidentification. In order to remove eventual artifacts due to the setup and removal of the electrodes, the first and last 10 seconds of the raw ECG signal received were ignored. The remaining data were filtered using a Butterworth bandpass filter order 4 with a frequency range of 1 Hz to 40 Hz. BioSPPy library [26] was used to perform the filtering. An example of an ECG signal before and after filtering can be seen in Fig. 13.

4.3.2 Fiducial Points and Cardiac Cycles Identification

The next filtering step is to process the signal to identify the cardiac cycles via fiducial points. A modified version of the Hamilton-Tompkins algorithm [27] implemented in the BioSPPy library was used for R-peaks identification. The remaining waves - P, Q, S, T - were then identified with the help of the Neuro Kit library [28], which also computes the

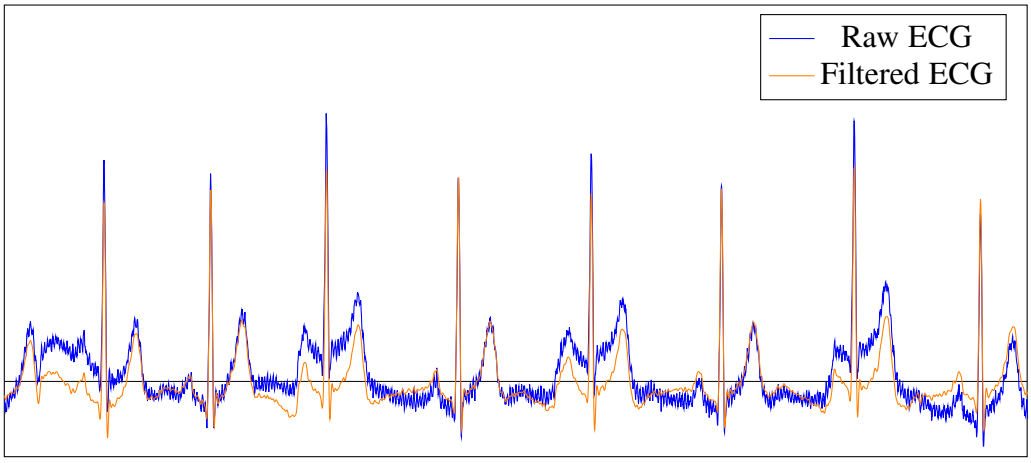

Fig. 13. Raw and filtered ECG signal.

cardiac cycles with the help of the BioSPPy library. A second fiducial point identification is done on each individual cardiac cycle in order to improve the peaks (and valleys) location for each wave. A cardiac cycle with the waves detected is shown in Fig. 14.

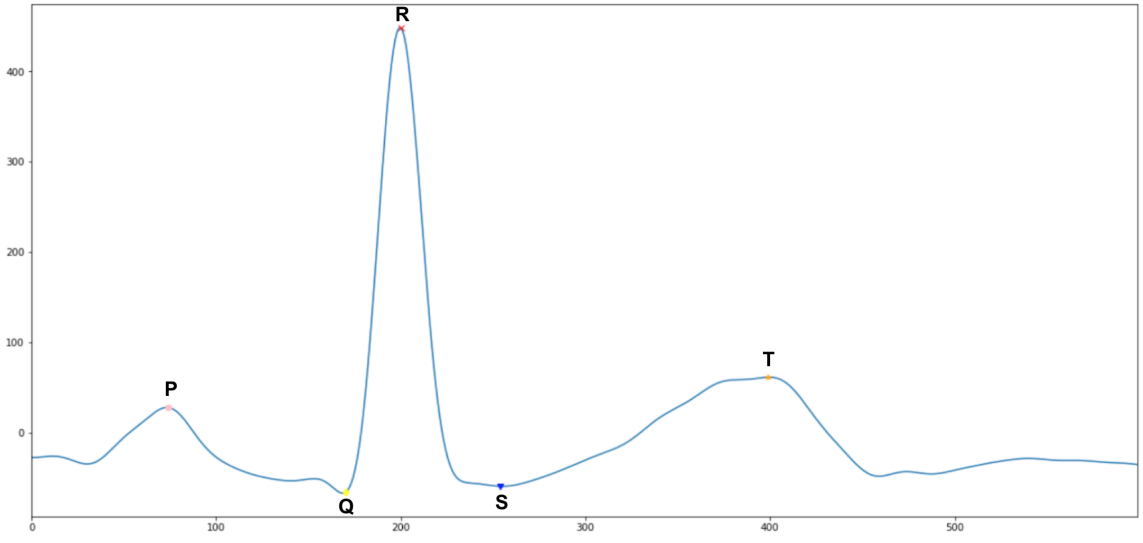

Fig. 14. Cardiac cycle with the fiducial points P, Q, R, S and T identified.

4.3.3 Feature Extraction

There are several different methods for feature extraction from ECG signals, ranging from simple and direct measurements based on fiducial points to more complex ones that are based on the entire wave morphology [29]. The former case includes amplitudes and distances between points and is known as a fiducial-based method. The latter includes wave frequencies and discrete wavelet transform (DWT) coefficients and is categorized as non-fiducial-based method. In this work, several experiments were performed testing a variety of fiducial-based features that could provide similar or better performance than using the whole cardiac cycle data (600 data points) as the model input. A collection of 18 features composed of 9 lines directly connecting different fiducial points and the respective slopes of such lines were found to accomplish that objective. Although slopes have been used before as a feature for biometric authentication systems, the collection of only 9 direct line lengths and 9 slopes used without any other fiducial data is a novel approach different from existing feature extraction available in the literature. The direct line length between two points, such as P and Q, was calculated using Euclidean distance as shown in Formula 1.

$$distance(P,Q) = \sqrt{PQx^2 + PQy^2} = \sqrt{(Qx - Px)^2 + (Qy - Py)^2} \tag{1}$$

And the slope was calculated using Formula 2:

$$slope(PQ) = \frac{PQy}{PQx} = \frac{Qy - Py}{Qx - Px} \tag{2}$$

The list of all 18 features used in this work is shown in Table 2 and a detailed calculation for each one of them is available in Appendix A.

Table 2
Features Extracted from ECG

#	Feature	#	Feature
1	PQ length	10	QR slope
2	PQ slope	11	QS length
3	PR length	12	QS slope
4	PR slope	13	QT length
5	PS length	14	QT slope
6	PS slope	15	RS length
7	PT length	16	RS slope
8	PT slope	17	RT length
9	QR length	18	RT slope

4.3.4 QT Correction

The QT interval is a metric that is known to have its value impacted by the subject's heart rate [30] but also from glucose concentration [31]. Therefore, to help the model better detect the glucose impact, it is important to have the heart rate interference reduced as much as possible. Several formulas have been developed to correct the QT interval for heart rate. A study performed by Vandenberk et al. [31] analyzed 5 different QT corrections techniques, including Bazett's formula. Their conclusion was that the Framingham formula provided the best correction and therefore that was used in this work. Bazett's formula, while still widely used, doesn't always work efficiently since it overcorrects when the heart rate is lower than 60 bpm and under-corrects when the heart rate is higher than 60. Based on the Framingham formula, the QT interval can be corrected by using Formula 3:

$$QTc = QT + 0.154 * (1 - RR) \tag{3}$$

where RR is can be calculated from the heart rate as shown in Formula 4:

$$RR = \frac{60}{heartrate} \tag{4}$$

4.3.5 Outliers Removal

The features extracted in the previous step can present some outliers. Removal of such non-consistent data contributes to faster training and better model performance. The outliers were identified using the interquartile range (IQR) method which defines a lower and upper bound based on the range between the first and third data quartiles. Formula 5 shows how IQR and the lower and upper bounds are calculated:

$$\begin{aligned} IQR &= Q3 - Q1 \\ Lowerbound &= Q1 - 1.5 * IQR \\ Upperbound &= Q3 + 1.5 * IQR \end{aligned} \tag{5}$$

where Q1 and Q3 are the first and third quartiles values respectively. Data points located below the lower bound or above the upper bound values were removed. A high number of outliers could be a consequence of a very noisy signal or also a poor fiducial points identification technique. Therefore, it is important to not only remove that data but also check the percentage of the whole data that were flagged as outliers. High percentages may indicate a need for revision of the fiducial points identification algorithm. The dataset used in this work had 68,274 samples and 16,756 were identified as having at least one outlier feature and thus were removed, leaving 51,518 samples to be used in the model. Removing the entire sample due to just one outlier feature is a very conservative approach and was used since the number of samples left was still significant and enough for training and testing. If a smaller dataset was used, different outlier removal approaches could be used such as replacing the outlier feature with the average value of that feature for that same subject.

4.3.6 Normalization

The features extracted in the previous step don't share the same unit, so it is very important to standardize them not only to remove their mean but also scale them to unit variance before feeding them to the machine learning model. Failing to do so can slow down training and even hinder the learning process. The normalization process can be done using Formula 6:

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

where μ is the mean and σ is the standard deviation of the samples. That calculation can be easily done with the help of the StandardScaler function available in Python Scikit-learn library [32].

4.4 Training and Testing

The 1,963 ECG readings were segmented in cardiac cycles and then an equal proportion of hyperglycemia and non-hyperglycemia were selected, resulting in dataset of 68,274. As mentioned in section 4.3.5, the outlier removal process purged 16,756 samples, leaving a net of 51,518 samples to be used in the model. A split 80/20 was used to create the training/testing dataset. Therefore, the training dataset contained 41,214 samples and the testing 10,304 samples, with almost equal representation of hyperglycemia and non-hyperglycemia samples.

4.5 Simulations & Results

Several machine learning models and configurations have been explored, including logistic regression, support vector machine and artificial neural networks (ANN). Extensive simulations of different ANN architectures were also performed in order to identify the optimal hyperparameters combination. The AUC of the receiver operating characteristic (ROC) curve was the performance metric used to compare the models since

in binary classification problems, the threshold used to distinguish between the 2 output labels have a direct impact in performance metrics such as accuracy, sensitivity and specificity. The ROC plots the model performance in terms of true positive rate (TPR) and false positive rate (FPR) across different thresholds. The area of that curve provides a combined performance measurement of all those thresholds. For non-ANN models, the simulations were configured with a maximum number of iterations of 10,000. For ANN simulations, the models were trained with 1,000 epochs and early stopping when no loss improvement has been achieved in the last 100 epochs. The training optimizer used was stochastic gradient descent (SGD) with learning rate 0.0001, which was reduced by half every time no improvement was obtained after 20 sequential epochs. Table 3 summarizes the simulations with the best performance for those different models. A comprehensive list of all models tested and their respective performance can be seen in Appendix B.

Table 3
Model Performance

Model	**AUC**
10-layer DNN	94.53%
Logistic regression (C=5)	62.44%
SVM linear (C=50)	58.99%
SVM polynomial (d=6)	56.36%
SVM Gaussian (C=2)	52.03%

The deep neural network with 10 layers and 500 units per layer, except for the output layer, provided the best performance among the models simulated. Its architecture and loss training can be seen in Fig. 15 and Fig. 16 respectively.

The model presented a training AUC of 98.44% and testing AUC of 94.53% as can be seen in the ROC curves in Fig. 17.

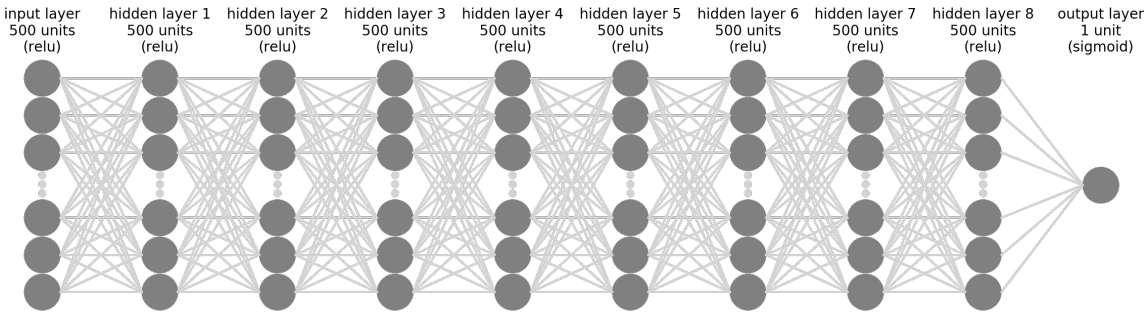

Fig. 15. Deep neural network architecture that provided the best performance. Picture created by the author.

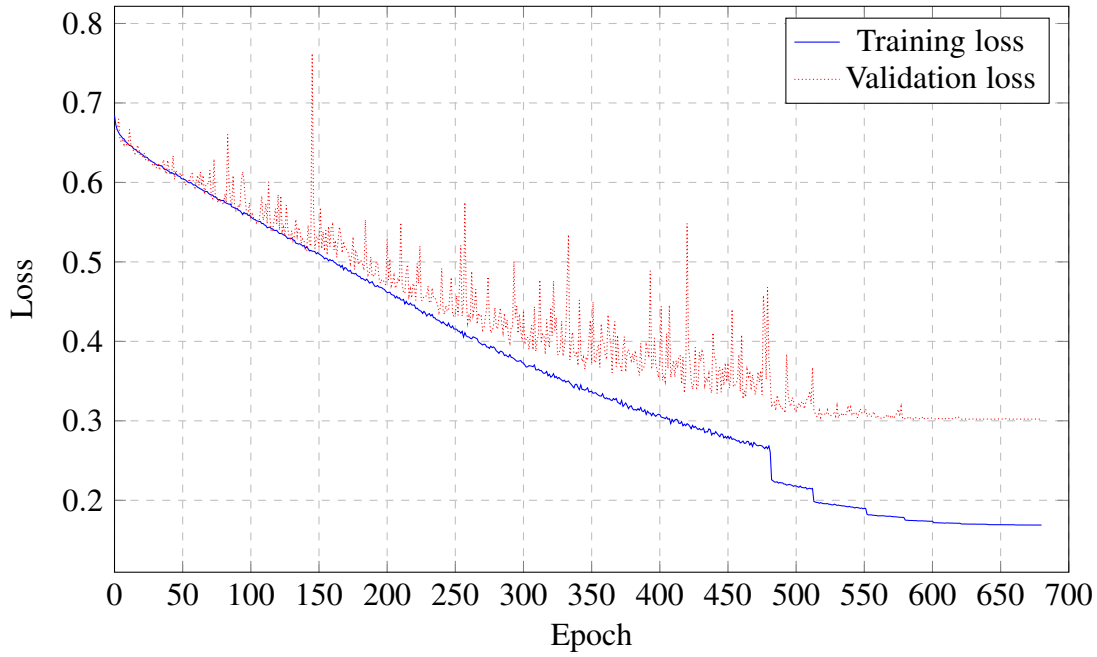

Fig. 16. 10-layer DNN training and validation loss.

A 10 k-fold cross validation, a technique that involves running the simulation with different combination of testing and training datasets, was also performed in order to verify the performance consistency of the model, resulting in average AUC of 93.65%. The AUC for each k-fold round can be seen in Table 4.

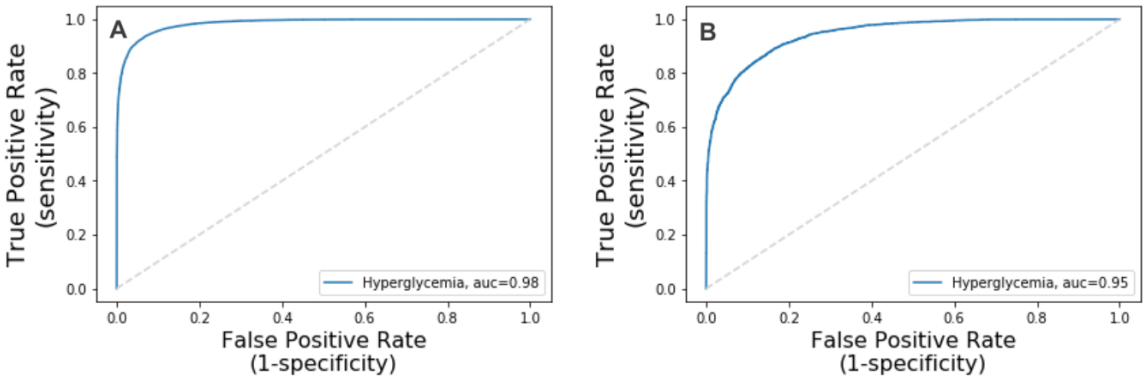

Fig. 17. 10-layer DNN ROC and AUC. A. ROC training dataset. B. ROC testing dataset

Table 4
10 K-fold Cross Validation (10-layer DNN)

k	AUC	k	AUC
1	96.98%	6	97.17%
2	97.23%	7	97.43%
3	96.40%	8	98.23%
4	97.34%	9	96.94%
5	96.03%	10	95.49%

The 10-layer DNN model was also compared to the best model identified in the literature. Since the work presented by [14] didn't have AUC values, we compared the model's performances using the geometric mean of the sensitivity and specificity, a metric available in their work. The binary classification threshold used in the DNN was 0.4426, which is the one that provided the maximum geometric meanwhile also having a sensitivity greater than the specificity, a characteristic often desired in screening devices.

The geometric mean of sensitivity and specificity of the model presented in this thesis is 86.30% versus the 67.94% from the 3-layer NN [14], showing a performance improvement of 27% as can be seen in Table 5.

Table 5
Model Comparison

	Sensitivity	**Specificity**	**Geometric Mean**
10-layer DNN	87.57%	85.04%	86.30%
3-layer ANN [14]	70.59%	65.38%	67.94%

5 CONCLUSIONS

This thesis presented a novel approach for ECG feature extraction and also a non-invasive hyperglycemia detection mechanism using ECG and deep learning. Its performance of 87.57% sensitivity and 85.04% specificity is an excellent indication that ECG signals possess intrinsic information that can indicate the level of blood glucose concentration. In addition, the model performance is, in aggregate, 27% better than the current best model in the literature to date [14]. The proliferation of consumer devices with ECG reading capabilities such as smart watches, wristbands, and even handheld ECG readers create an environment in which ECG acquisition becomes cheap and accessible to everyone. The knowledge created by this research provides a way to improve such devices to allow users to detect a hyperglycemic state in a quick, painless and easy way, contributing to the identification of health issues that don't manifest themselves until it is too late. Professional health environments in which patients have their ECG taken could also use this research to improve their devices to identify hyperglycemia without any extra hardware. Lastly, patients in need of close monitoring of their glucose levels such as pregnant women with gestational diabetes and diabetic individuals could benefit from a painless screening device powered by this research.

6 FUTURE WORK

The deep learning model presented in this thesis, a 10-layer deep neural network, was developed with 2 characteristics in mind: performance and simplicity. But in machine learning sometimes the best performance can only be achieved with complex models. Therefore, different and more complex architectures such as long short-term memory (LSTM), convolutional neural networks (CNN), recurrent neural networks (RNN), block-based neural network (BBNN) and others could be tested.

The thesis also focused in contributing to the field with a novel set of ECG features. Thus, experiments with established ECG fiducial features (intervals and amplitudes) and non-fiducial ones could also be developed and the performance compared.

Extension of this model to detect hypoglycemia and even the shift from a classification model to a glucose concentration prediction (regression) could also be sought after.

Literature Cited

[1] G. Taubes and M. Chamberlain, *Why we get fat*. Joosr Ltd, 2016.

[2] S. Kumar, "Insulin resistance, a surrogate marker, in type 2 diabetes mellitus," *International Journal of Scientific & Engineering Research*, vol. 5, 2014.

[3] U.S. Food and Drug Administration, "Self-monitoring blood glucose test systems for over-the-counter use - guidance for industry and Food and Drug Administration staff. U.S. Food and Drug Administration," 2016.

[4] N. D. Thompson and J. F. Perz, "Eliminating the blood: Ongoing outbreaks of hepatitis B virus infection and the need for innovative glucose monitoring technologies," *Journal of Diabetes Science and Technology*, vol. 3, pp. 283–288, Mar 2009.

[5] H. Ali, F. Bensaali, and F. Jaber, "Novel approach to non-invasive blood glucose monitoring based on transmittance and refraction of visible laser light," *IEEE Access*, vol. 5, pp. 9163–9174, 2017.

[6] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge University press, 2014.

[7] R. Amanipour, H. Nazeran, I. Reyes, M. Franco, and E. Haltiwanger, "The effects of blood glucose changes on frequency-domain measures of HRV signal in type 1 diabetes," in *Proc. nd Int CONIELECOMP 2012 Conf. Electrical Communications and Computers*, pp. 50–54, Feb. 2012.

[8] Y. Fujimoto, M. Fukuki, A. Hoshio, N. Sasaki, T. Hamada, Y. Tanaka, A. Yoshida, C. Shigemasa, and H. Mashiba, "Decreased heart rate variability in patients with diabetes mellitus and ischemic heart disease.," *Japanese Circulation Journal*, vol. 60, pp. 925–932, Dec. 1996.

[9] G. Perpiñan, E. Severeyn, S. Wong, and M. Altuve, "Nonlinear heart rate variability measures during the oral glucose tolerance test," in *Proc. Computing in Cardiology (CinC)*, pp. 1–4, Sept. 2017.

[10] B. Suys, S. Heuten, D. De Wolf, M. Verherstraeten, L. O. de Beeck, D. Matthys, C. Vrints, and R. Rooman, "Glycemia and corrected QT interval prolongation in young type 1 diabetic patients: what is the relation?," *Diabetes Care*, vol. 29, pp. 427–429, Feb. 2006.

[11] T. F. Christensen, L. Tarnow, J. Randløv, L. E. Kristensen, J. J. Struijk, E. Eldrup, and O. K. Hejlesen, "QT interval prolongation during spontaneous episodes of hypoglycaemia in type 1 diabetes: the impact of heart rate correction," *Diabetologia*, vol. 53, pp. 2036–2041, Sep 2010.

[12] R. Marfella, F. Nappo, L. De Angelis, M. Siniscalchi, F. Rossi, and D. Giugliano, "The effect of acute hyperglycaemia on QTc duration in healthy man.," *Diabetologia*, vol. 43, pp. 571–575, May 2000.

[13] L. L. Nguyen, S. Su, and H. T. Nguyen, "Identification of hypoglycemia and hyperglycemia in type 1 diabetic patients using ECG parameters," in *Proc. Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pp. 2716–2719, Aug. 2012.

[14] L. L. Nguyen, S. Su, and H. T. Nguyen, "Neural network approach for non-invasive detection of hyperglycemia using electrocardiographic signals," in *Proc. 36th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pp. 4475–4478, Aug. 2014.

[15] J. Malmivuo and R. Plonsey, *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press, 01 1995.

[16] C. Soto, *ECG: Essentials of Electrocardiography*. Cengage Learning, 2015.

[17] A. Lorsheyd, D. W. de Lange, M. L. Hijmering, M. J. M. Cramer, and A. van de Wiel, "PR and OTc interval prolongation on the electrocardiogram after binge drinking in healthy individuals.," *The Netherlands Journal of Medicine*, vol. 63, pp. 59–63, Feb. 2005.

[18] A. Su, "ECG noise filtering using online model-based bayesian filtering techniques," Master's thesis, University of Waterloo, 2013.

[19] F. Buendía-Fuentes, M. Arnau-Vives, A. Arnau-Vives, Y. Jiménez-Jiménez, J. Rueda-Soriano, E. Zorio-Grima, A. Osa-Sáez, L. Martínez-Dolz, L. Almenar-Bonet, and M. Palencia-Pérez, "High-bandpass filters in electrocardiography: source of error in the interpretation of the ST segment," *ISRN Cardiology*, vol. 2012, 2012.

[20] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 3 ed., 2009.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[22] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[23] C. Chang and K. Liu, "Non-invasive blood glucose measurement using PPG and ECG signals," 2018. Academia Sinica, Taiwan.

[24] "AD8232 Datasheet and Product info — Analog Devices."

[25] "The Accu-Chek Mobile system — Accu-Chek."

[26] C. Carreiras, A. P. Alves, A. Lourenço, F. Canento, H. Silva, A. Fred, *et al.*, "BioSPPy: Biosignal processing in Python," 2015.

[27] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database," *IEEE Transactions on Biomedical Engineering*, vol. BME-33, pp. 1157–1165, Dec. 1986.

[28] D. Makowski, "Neurokit: A python toolbox for statistics and neurophysiological signal processing (EEG, EDA, ECG, EMG...)," Nov. 2016.

[29] N. Karimian, Z. Guo, M. Tehranipoor, and D. Forte, "Highly reliable key generation from electrocardiogram (ECG)," *IEEE Transactions on Biomedical Engineering*, vol. 64, pp. 1400–1411, June 2017.

[30] M. Malik, "Problems of heart rate correction in assessment of drug-induced QT interval prolongation," *Journal of Cardiovascular Electrophysiology*, vol. 12, no. 4, pp. 411–420, 2001.

[31] B. Vandenberk, E. Vandael, T. Robyns, J. Vandenberghe, C. Garweg, V. Foulon, J. Ector, and R. Willems, "Which QT correction formulae to use for qt monitoring?," *Journal of the American Heart Association*, vol. 5, no. 6, p. e003264, 2016.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Appendix A

FIDUCIAL FEATURES

The detailed calculation of all 18 features used as input for the models simulated in this thesis are detailed below.

A.1 PQ Distance and Slope

$$distance(P,Q) = \sqrt{PQx^2 + PQy^2} = \sqrt{(Qx - Px)^2 + (Qy - Py)^2} \tag{7}$$

$$slope(PQ) = \frac{PQy}{PQx} = \frac{Qy - Py}{Qx - Px} \tag{8}$$

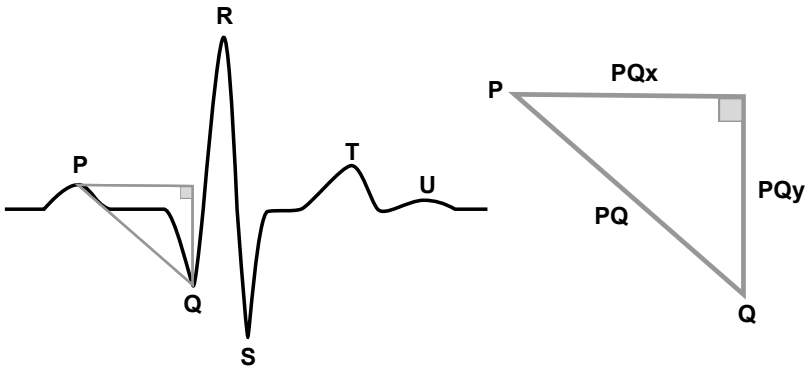

Fig. 18. PQ distance and slope

A.2 PR Distance and Slope

$$distance(P,R) = \sqrt{PRx^2 + PRy^2} = \sqrt{(Rx - Px)^2 + (Ry - Py)^2} \tag{9}$$

$$slope(PR) = \frac{PRy}{PRx} = \frac{Ry - Py}{Rx - Px} \tag{10}$$

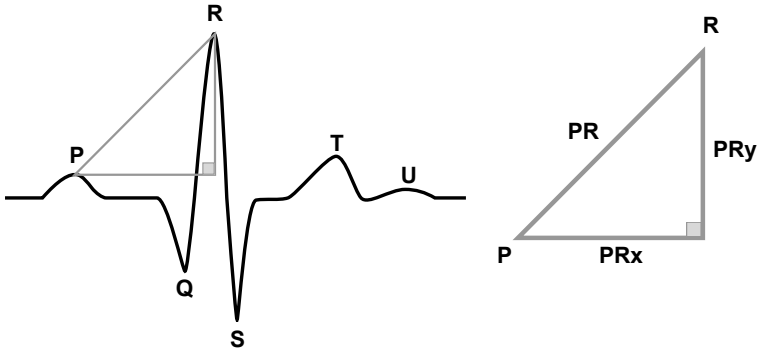

Fig. 19. PR distance and slope

A.3 PS Distance and Slope

$$distance(P,S) = \sqrt{PSx^2 + PSy^2} = \sqrt{(Sx - Px)^2 + (Sy - Py)^2} \tag{11}$$

$$slope(PS) = \frac{PSy}{PSx} = \frac{Sy - Py}{Sx - Px} \tag{12}$$

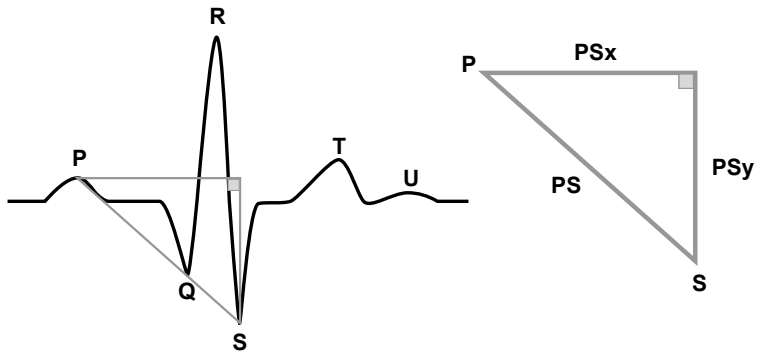

Fig. 20. PS distance and slope

A.4 PT Distance and Slope

$$distance(P,T) = \sqrt{PTx^2 + PTy^2} = \sqrt{(Tx - Px)^2 + (Ty - Py)^2} \tag{13}$$

$$slope(PT) = \frac{PTy}{PTx} = \frac{Ty - Py}{Tx - Px} \tag{14}$$

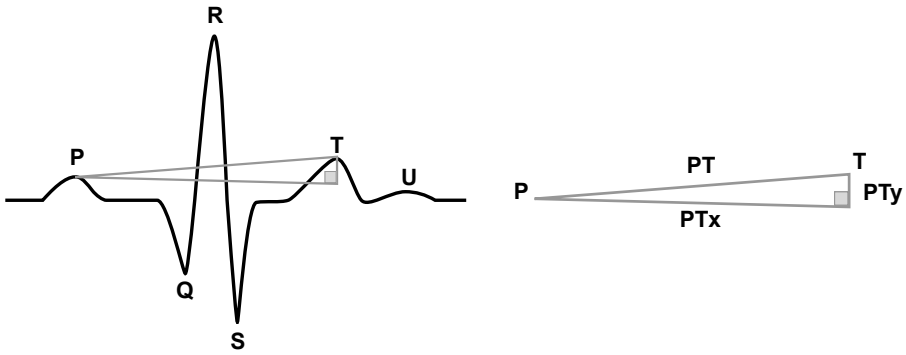

Fig. 21. PT distance and slope

A.5 QR Distance and Slope

$$distance(Q,R) = \sqrt{QRx^2 + QRy^2} = \sqrt{(Rx - Qx)^2 + (Ry - Qy)^2} \tag{15}$$

$$slope(QR) = \frac{QRy}{QRx} = \frac{Ry - Qy}{Rx - Qx} \tag{16}$$

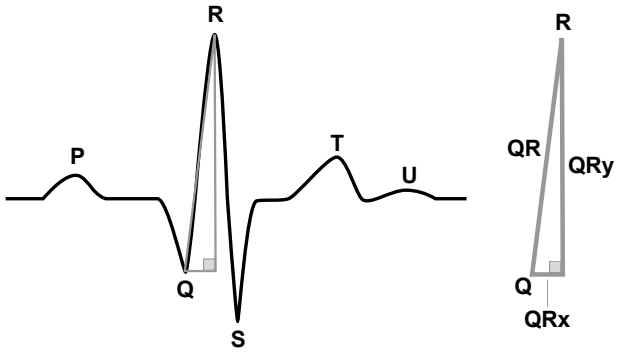

Fig. 22. QR distance and slope

A.6 QS Distance and Slope

$$distance(Q,S) = \sqrt{QSx^2 + QSy^2} = \sqrt{(Sx - Qx)^2 + (Sy - Qy)^2} \tag{17}$$

$$slope(QS) = \frac{QSy}{QSx} = \frac{Sy - Qy}{Sx - Qx} \tag{18}$$

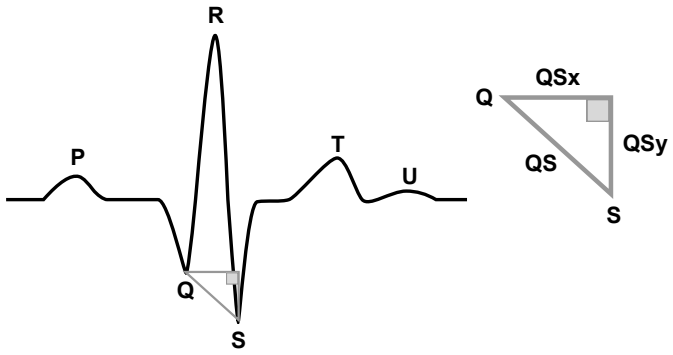

Fig. 23. QS distance and slope

A.7 QT Distance and Slope

$$distance(Q,T) = \sqrt{QTx^2 + QTy^2} = \sqrt{(Tx - Qx)^2 + (Ty - Qy)^2} \tag{19}$$

$$slope(QT) = \frac{QTy}{QTx} = \frac{Ty - Qy}{Tx - Qx} \tag{20}$$

A.8 RS Distance and Slope

$$distance(R,S) = \sqrt{RSx^2 + RSy^2} = \sqrt{(Sx - Rx)^2 + (Sy - Ry)^2} \tag{21}$$

$$slope(RS) = \frac{RSy}{RSx} = \frac{Sy - Ry}{Sx - Rx} \tag{22}$$

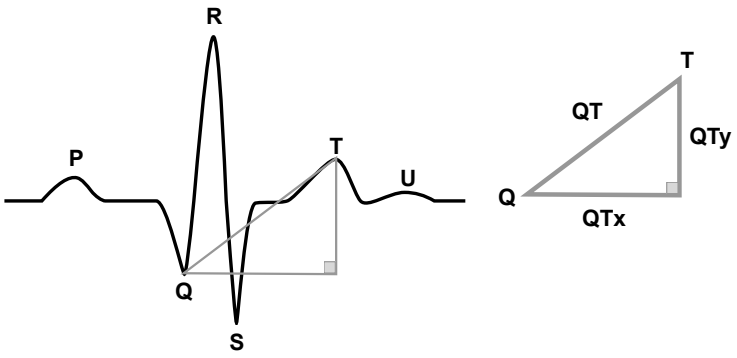

Fig. 24. QT distance and slope

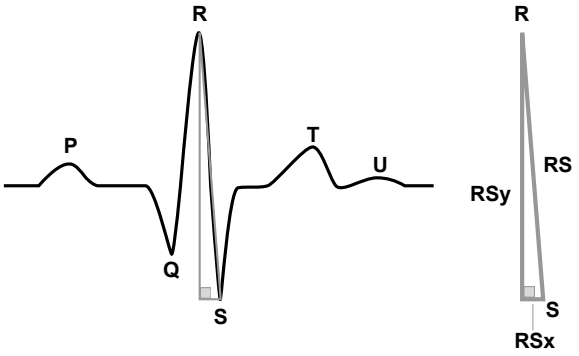

Fig. 25. RS distance and slope

A.9 RT Distance and Slope

$$distance(R,T) = \sqrt{RTx^2 + RTy^2} = \sqrt{(Tx - Rx)^2 + (Ty - Ry)^2} \tag{23}$$

$$slope(RT) = \frac{RTy}{RTx} = \frac{Ty - Ry}{Tx - Rx} \tag{24}$$

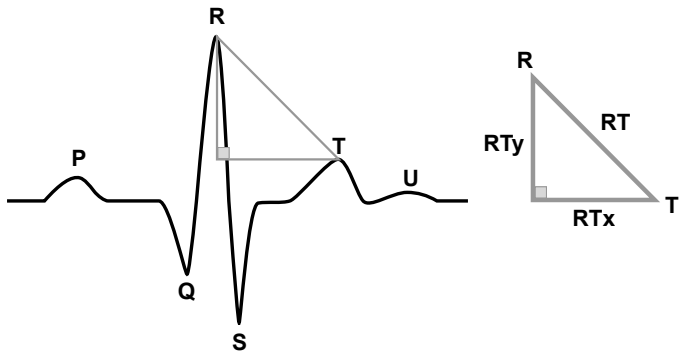

Fig. 26. RT distance and slope

Appendix B

MODELS SIMULATIONS RESULTS

The Table 6 below shows the performance, measured as the area under the curve of the testing dataset for different non-ANN models and C (or degree for SVN polynomial) parameters.

Table 6
AUC Values for Different Models and C / Degree Parameter

C or degree	Logistic Regression	SVM Linear	SVM Gaussian	SVM Polynomial
0.001	61.45%	58.02%	18.60%	-
0.01	62.09%	53.86%	18.16%	-
0.1	61.87%	42.17%	17.61%	-
1	62.37%	57.36%	47.90%	55.92%
2	61.87%	44.01%	52.03%	55.14%
3	61.87%	57.57%	52.03%	42.74%
4	61.92%	43.47%	52.03%	48.17%
5	62.44%	42.22%	52.03%	41.85%
6	61.95%	41.88%	52.03%	56.36%
7	61.91%	41.96%	52.03%	50.15%
8	62.43%	44.88%	52.03%	50.00%
9	61.85%	50.16%	52.03%	50.00%
10	62.39%	41.37%	52.03%	50.00%
20	61.86%	57.93%	52.03%	50.00%
30	61.84%	48.64%	52.03%	50.00%
40	61.90%	41.66%	52.03%	50.00%
50	62.44%	58.99%	52.03%	50.00%
60	61.86%	57.56%	52.03%	50.00%
70	61.87%	56.22%	52.03%	50.00%
80	61.93%	51.63%	52.03%	50.00%
90	61.92%	56.93%	52.03%	50.00%
100	62.37%	42.46%	52.03%	50.00%

For ANN models, Table 7 below shows the AUC for different number of layers and number of units per layer.

Table 7
AUC Values for Different Number of Layers and Units per Layer

# of units per layer (exc. output layer) **# of layers**	**100**	**200**	**300**	**400**	**500**
2	49.96%	50.00%	50.00%	79.64%	50.00%
3	79.34%	50.00%	50.00%	50.00%	50.00%
4	80.46%	49.99%	88.68%	50.00%	89.25%
5	83.64%	87.02%	88.57%	90.76%	50.00%
6	82.94%	89.85%	91.68%	91.06%	92.53%
7	85.69%	90.78%	91.94%	92.43%	93.20%
8	86.57%	89.49%	92.76%	92.26%	93.44%
9	88.81%	89.96%	92.80%	92.91%	93.59%
10	89.06%	91.88%	92.29%	94.34%	94.53%