University of Saskatchewan

Investigation of Machine Learning Techniques to Determine Informative Wavelengths for Noninvasive Glucose Monitoring

by

Khoa Nguyen

A thesis submitted in partial fulfillment for the

degree of Master of Science

in the

College of Graduate and Postdoctoral Studies

In the Division of Biomedical Engineering

Permission to Use and Disclaimer Statement

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, it is agreed that the Libraries of this University may make it freely available for inspection. Permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professors who supervised this thesis work or, in their absence, by the Head of the Division of Biomedical Engineering or the Dean of the College of Graduate Studies and Research at the University of Saskatchewan. Any copying, publication, or use of this thesis, or parts thereof, for financial gain without the written permission of the author is strictly prohibited. Proper recognition shall be given to the author and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

DISCLAIMER

The Raspberry Pi, Neospectra Si-ware System Company, and Dell Inc. were exclusively created to meet the thesis and/or exhibition requirements for the degree of Master of Science at the University of Saskatchewan. Reference [49-51] in this thesis/dissertation to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis/dissertation in whole or part should be addressed to:

Dean

College of Graduate and Postdoctoral Studies

University of Saskatchewan

116 Thorvaldson Building, 110 Science Place

Saskatoon, Saskatchewan S7N 5C9

Canada

OR

Head of the Division of Biomedical Engineering

College of Engineering

University of Saskatchewan

57 Campus Drive

Saskatoon, Saskatchewan S7N 5A9

Canada

Acknowledgements

I would like to express my sincere gratitude and deepest thanks to my supervisors at the University of Saskatchewan for your wonderful support and suggestions throughout my research. To Prof. Anh Dinh, I am extremely grateful for giving me your words of encouragement when I was about to lose my motivation. To Prof. Francis Bui, thank you very much for providing me extremely helpful knowledge and feedback. This thesis would not be possible without your extraordinary assistance. To my girlfriend, Nghi, for helping me survive all the stress from those years and for not letting me give up.

Abstract

The trend towards noninvasive blood glucose monitoring to reduce nerve damage and infection-related mortality rate of diabetic patients has led to the advent of near-infrared (IR) based devices. The overlaps between the absorption peaks of glucose and other molecules mean that many wavelengths are potentially correlated to the glucose concentration, and a suitable combination of spectral information across a range of wavelengths is necessary to determine the glucose concentration in an effective and robust manner. This work investigates the use of dimensional reduction and support vector machines (SVMs) as core algorithms to develop an automated and computationally efficient system to calibrate the relation between spectral wavelengths and glucose concentration, while facilitating feature selection of the informative wavelengths for accurate glucose monitoring.

Evaluations performed on two datasets, containing information regarding the absorbance of short-wave infrared (SWIR) by glucose solution with distilled water, demonstrated that wrapper methods of feature selection could be highly effective for glucose monitoring model using SVM. By utilizing the developed wrapper methods, training accuracy can be improved, achieving up to 91.53%, testing accuracy to 91%, f1 score to 90.97% for classification approach, and standard error of cross-validation (SECV) can be decreased to 45.12mg/dl, standard error of prediction (SEP) to 39.08mg/dl for regression approach. Furthermore, filter methods of feature selection were found to offer a trade-off between speed and performance for the proposed models when used in combination with wrapper methods. If time is an important constraint, then techniques of filter method should be added to the system, since this addition can increase the feature selection speed and training speed up to 17 and 9 times respectively with only a slight drop in

performance. Because wavelengths can be considered either discrete or continuous, different assumptions of continuity of wavelengths and their relative choice of evaluation metrics, whether following a classification or regression approach, were investigated to check for influences and consequentially found to impact information extraction ability of dimensionality reduction techniques.

The proposed system model consists of 3 phases, envisioned as three interacting modules: data acquisition, training pipeline and testing pipeline. The main training module is in turn composed of 4 major steps: preprocessing, dimensional reduction, hyperparameter tuning, and prediction (with SVMs). Using the proposed model, the obtained computational results suggest that the most informative wavelengths for noninvasive glucose monitoring, given the experimental datasets used in this investigation, should fall in the ranges of 1300-1600nm and 1800-2400nm or 2000-2600nm.

Table of Contents

List of Tables

List of Figures

Abbreviation

CEG	Clark Error Grid
KNN	K-nearest Neighbors
MI	Mutual Information
MIR	Mid-infrared
MLR	Multiple Linear Regression
NIR	Near-infrared
PCA	Principal Components Analysis
PCR	Principal Components Regression
PLSR	Partial least-squares regression
SECV	Standard error of cross-validation
SEP	Standard error of prediction
SFFS	Sequential Floating Forward Search
SKB	Select K Best
SV	Support Vector
SVM	Support Vector Machine
SVR	Support Vector Regression
SWIR	Short-wave Infrared

Chapter 1 Introduction

This chapter provides introductory background on diabetes and an overview of popular glucose monitoring techniques, with a particular focus on the optical technique of absorption spectroscopy. The chapter also reviews the relevant research literature on informative wavelengths for noninvasive glucose monitoring, on the strengths and weaknesses of available analysis models. In addition, the chapter also states the problem statement and objectives, challenges and assumptions, scopes and limitations as well as the summary of contributions.

1.1 Background on Diabetes and Glucose Monitoring

Diabetes mellitus (diabetes) is a chronic metabolic disease characterized by hyperglycemia, elevated levels of blood glucose, due to human bodies' inability to produce and/or effectively use insulin [1]. Diabetes is classified into three main groups: type 1 diabetes, type 2 diabetes, and gestational diabetes mellitus (GDM) [2]. The classification of these diabetes types is summarized in Table 1-1. The World Health Organization recognizes diabetes as a dangerous illness that exposes patients to increased risks of serious life-threatening health issues, results in millions of deaths every year [1, 3]. The prevalence of diabetes has risen fast with 108 million cases in 1980 to 422 million in 2014 [1]. It is estimated that by 2045, 639 million people worldwide will have been affected by diabetes [4]. Expenditure on diabetes accounted for 548 billion USD worldwide in 2013 and is expected to be 627 billion USD in 2035 [5]. These escalating figures explain a huge and growing global concern regarding diabetes.

Regardless of diabetes types, diabetic patients are advised to regularly monitor their blood glucose levels, as delayed management may lead to serious complications including, but not

limited to, long-term damage and failure of various organs; potential loss of vision; foot ulcers, amputations; an increased risk for gastrointestinal, genitourinary, cardiovascular disease [3]. The majority of commercially available instruments to provide support in diabetes diagnosis are invasive or minimally invasive. Invasive systems which require blood samples to be taken from patients are still standard techniques for home glucose monitoring reading through electrochemical, colorimetric or optical disposable strips for finger-pricking [6]. Such systems are painful, which often lead to non-compliance [7] and makes patients susceptible to infection and nerve damage [5]. Infection-related issues have been reported to have a high mortality rate of 4.7 per 1000 people [8]. On the other hand, minimally invasive devices utilize subcutaneous sensors to measure glucose concentration in interstitial fluid (ISF) of the skin and allow for repeated blood glucose monitoring [9-11]. However, this method still causes discomfort while exhibiting limited life span and stability [5, 12]. Therefore, the idea of accurate noninvasive glucose monitoring device to lessen the likelihood of infection and other disadvantages has attracted significant research.

Published research investigations on potential alternative noninvasive glucose monitoring, using various technologies that satisfied the requirements, can be categorized into two groups: non-optical techniques and optical techniques.

A type of non-optical technique is reverse iontophoresis which utilizes electrical current to transport glucose outward the skin [13]. The extracted glucose concentration is then collected by electrode containing the enzyme glucose oxidase. Another electrode detects H_2O_2 generated by glucose oxidase-catalyzed reaction [14]. Depending on the amount of H_2O_2, a current is generated and analyzed to predict glucose level. Some other techniques utilize ultrasonic energy

to the skin. Disadvantages of these technologies include high cost, time delay, skin irritation, inaccuracies, long calibration procedures and 2-3 hours warm-up period [15, 16].

Polarimetry is one of the common optical techniques. The principle of polarimetry is that the linear polarization vector of light can be rotated by the path characteristics such as thickness, temperature, and concentrations of the crossed sample. Therefore, many research groups have used polarimetry to measure the level of glucose. Because the skin has high scattering coefficient, which completely depolarizes the beam, the aqueous humor of the eye, which offers a clear optical media with reasonable path length and lag time in relation to blood glucose concentration, has been use as area of measurement [17]. Polarimetry methods might be affected by temperature and pH fluctuations. Other limitations include complicated safety regulations on light exposure to the eye, motion artifacts.

Raman spectroscopy is another popular optical technique. The technique is based on the Raman Effect in which a small fraction of scattered light displays wavelengths different from that of the exciting beam. The spectroscopy uses laser from in the range between visible and mid-infrared (MIR) light. Raman spectroscopy has the advantage of not being affected by interference from water due to its weak scattering indexes. Another advantage is that it allows easy signal separation, in contrast to absorption spectroscopy, due to its narrow resulting bands and distinct peaks [18]. However, there is a risk of photothermal damage in ocular measurement [19]. Raman spectroscopy also requires longer collection periods than other optical methods [20].

Table 1-1. Classification of diabetes [2]

Type	Definition	Etiology
Type 1 diabetes	• Pancreatic beta cell destruction	• An autoimmune process • and other unknown etiology for which pancreatic beta cell is destroyed
Type 2 diabetes	• Relative insulin deficiency • or insulin resistance	• Predominant insulin resistance • Predominant secretory defect
Gestational diabetes mellitus (GDM)	• Glucose intolerance with onset or first recognition during pregnancy	

From all the optical techniques, most attention has been given to absorption spectroscopy because it is fast, possesses no risk, and requires little to no prior sample preparation. The low cost and availability of equipment for the development of this technique also contribute to its potential. Absorption spectroscopy is based on the fact that light can be reflected, scattered and absorbed when contacts biological tissues. The amount of reflection, scattering, and absorption is proportional to the structure and chemical components of the sample. This possibility of molecular differentiation shows that by measuring and analyzing the light absorption, glucose concentration can be monitored using appropriate wavelengths. In fact, most of the efforts in monitoring glucose concentration noninvasively are focused on this type of spectroscopy. Although being considered the most promising technique, absorption spectroscopy has not yet led to a commercial product [5].

1.2 Background on Absorption Spectroscopy

Absorption spectroscopy has been investigated as a promising approach for noninvasive glucose determination. The principle of the approach is that light can penetrate the tissue and interacts with the chemical components of biological materials causing the light to be partially absorbed and scattered [21]. Due to differences in absorbance, different components of biological materials, such as water, glucose, hemoglobin, would transmit or reflect incoming light differently. The attenuation of light in tissue is described, according to light transport theory, by the effective attenuation coefficient μ_{eff} [22]:

$$I = I_0 e^{-\mu_{eff} l} \tag{1}$$

where: I - intensity of reflected light

I_0 - intensity of incident light

μ_{eff} - effective attenuation coefficient

l - effective path length in the medium

On the other hand, μ_{eff} can be expressed as a function [22]:

$$\mu_{eff} = \sqrt{(3\mu_a[\mu_a + \mu_s(1 - g)} \tag{2}$$

where, μ_a is the absorption coefficient and μ_s is the scattering coefficient, g = cos(θ) and θ is a partial deflection angle.

Glucose can affect the measured signal by its light absorption. Changes in glucose concentration can influence the coefficient μ_a through the changes in absorption corresponding to water displacement or changes in its intrinsic absorption [22]. Therefore, it can be inferred that using specific wavelengths and measuring the changes in reflected signal by absorption of light,

glucose concentration can be accurately approximated. In other words, a level of glucose concentration can be quantitatively mapped to a certain signal value.

According to the absorptivity of glucose and other biological materials calculated by Amerov et al. in [23], there are overlaps between the absorption peaks of glucose and other molecules, so a glucose concentration would influence not only one but many wavelengths. In other words, many wavelengths are correlated to a glucose concentration and a combination of spectral information across a range of wavelengths is necessary to determine a glucose concentration.

1.3 Literature Review

Previous sections show that absorption spectroscopy is considered to have high potential for noninvasive glucose monitoring and that a combination of different wavelengths is desirable to accurately determine a glucose concentration. The first half of this section will review researches on noninvasive glucose monitoring regarding recommended ranges of wavelengths that contain most information for glucose prediction; methods for extracting the most informative wavelengths; and the analysis models as well as metrics for predicting glucose concentration and evaluation of the prediction performance. After acquiring information regarding the analysis models, the second half of this section will review the strengths and weaknesses of common analysis models in the literature.

1.3.1 Literature Review on the Informative Wavelengths

A number of different wavelengths have been investigated by different researchers [19-30]. They are summarized in Table 1-2.

Malin et al., Jeon et al., Amerov et al., and Jun Chen et al. in [21, 23-25] attempted to determine the most informative wavelengths by investigating wavelengths in the first overtone region and the combination overtone region of near-infrared (NIR) spectrum, i.e. 1100-1850nm and 2000-2500nm, where the absorption of water is believed to be lower than the absorption of glucose. They all used Partial least-squares regression (PLSR) multivariate analysis model for prediction and evaluation. The metrics used for evaluation of all model in all of the studies were standard error of cross-validation (SECV) and standard error of prediction (SEP).

Specifically, Malin et al. in [21] conducted experiment on 3 datasets measured by using 3 different wavelengths regions, i.e. 1100-1380nm, 1450-1850nm, and 2050-2375nm. There are a total of 14 NIR wavelengths that were used. The first dataset was obtained from a diabetic subject with associated blood glucose concentrations ranging from 91 to 446mg/dl. Similarly, the ranges of blood glucose concentration for the second and third dataset were 82-294mg/dl and 97-171mg/dl, respectively. Each dataset was split into training and testing sets with the ratio of 7/3. For the first dataset, the SECV value was 28.82 mg/dl; the SEP value was 138mg/dl. For the second dataset, the SECV value was 30.63 mg/dl; the SEP value was 44mg/dl. For the third dataset, the SECV value was 17.1mg/dl; the SEP value was 41mg/dl. It was concluded that the wavelengths in the range 1100-2375 have a cautious optimism for noninvasive glucose monitoring since the model achieve reasonable SECV values. However, it was also stated that there was difficulty in identifying or extracting unique wavelengths, which would be necessary for more accurate prediction.

Jeon et al. in [24] conducted experiment on three wavelength regions: 1100-1850nm, 2200-2500nm, and the entire region of 1100-2500nm. They used a dataset of 63 observations with glucose 7 different glucose concentrations varied between 0-1000mg/dl. The dataset was split

into training and testing set with the ratio of 6/4. Results showed that the range between 1100-1850nm has the best SECV of 33.51mg/dl compared to 108.04mg/dl of the range 2200-2500nm and 69.58mg/dl of the range 1100-2500nm. However, the SEP of the range 1100-1850nm was high, having the value of 437.54 mg/dl.

Amerov et al. in [64] conducted experiment on two wavelength regions: 1111-1851nm, and 2000-2500nm. They used a dataset of 80 observations with glucose concentrations varied between 54-540mg/dl. The dataset was split into training and testing set with the ratio of 7.5/2.5. For each aforementioned region, the best spectral range was determined by using grid search to examine 5100 combinations. For the region 1111-1851nm, the best spectral range was 1550-1750nm. The standard error of calibration (SEC) and SEP of the PLSR model for this spectral range were 20.54mg/dl and 21.62mg/dl, respectively. For the region 2000-2500nm, the best spectral range was 2061-2380nm. The SEC and SEP of the PLSR model for this spectral range were 10.09mg/dl and 17.3mg/dl, respectively.

Similar to Amerov et al., Jun Chen et al. in [25] conducted experiment on two wavelengths regions: 1658-1769nm, and 2192-2439nm. They also used PLSR as model for regression and prediction. Their results demonstrated that the range 2000-2500nm was particularly better compared to the range 1538-1818nm.The SEC values for the range 2192-2439nm was 5.58mg/dl and 22.34mg/dl for the range 1658-1769nm. The SEP values are approximately 3 times lower for the glucose model generated from the range 2192-2439nm versus from the range 1658-1769nm, i.e. 8.11mg/dl versus 20.18mg/dl.

Instead of PLSR model, Al-Mbaideen et al. in [26] investigated the use of principal component regression (PCR) multivariate analysis model to predict glucose concentration. A total of 90 wavelengths from 2100-2400nm were used to form a dataset from 30 mixture samples

of glucose, urea, and triacetin. The glucose concentration of the prepared samples was ranged in between 20-500mg/dl; triacetin concentration ranged from 10-190mg/dl; and urea ranged from 0-50mg/dl. From the 90 input wavelengths, the model managed to extract 17 principal components. The study did not validate the regression of the proposed model but directly evaluated the prediction ability. Two third of the dataset were used for building the regression model while the rest were used for prediction. Results showed that the developed model using solely PCR had the SEP of 40mg/dl.

M. Habibullah et al. in [27] also conducted experiment using the range of wavelength between 1300nm to 2600nm which cover most of the first and combination overtone region. Instead of the biased regression analysis model PLSR and PCR, the team made new idea to utilize two machine learning techniques called Random Forest and Support Vector Machine (SVM) to establish links between measured wavelengths signals and the glucose concentration. For evaluation of models using the two techniques, the accuracy and precision metrics were used. It should be noted that, unlike other studies that built regression models, this work developed a classification model and hence the differences in evaluation metrics. The Random Forest technique obtained the accuracy of 67.5% and precision 70%. The SVM technique obtained the accuracy of 77.5% and precision of 82%. The drawback of their approach was that they did not remove uncorrelated wavelengths, or features in machine learning parlance, and trained the model on the large range; hence the unstable results with low accuracy and precision.

In contrast to the aforementioned studies, Haider Ali et al. in [28] determined the optimum wavelengths by passing visible and NIR light in range 500-1200nm through water and then utilized Snell's law and other mathematical analysis. As the results, they claimed that the 650nm wavelength was suitable for the development of noninvasive blood glucose monitoring device.

The project used Clark Error Grid (CEG) analysis model to verify the accuracy and repeatability. Measured glucose concentrations of 45 subjects using their device were plotted compared to the reference blood glucose measured by another commercial device. It was showed that glucose monitoring device using 650nm was able to achieve overall accuracy of 90-92%.

Kasahara et al. in [29] decided to investigate the effectiveness of mid-infrared (MIR) in the range 8333-10204nm for glucose monitoring. In this study, since the penetration depth of MIR radiation is limited to a few microns, glucose signals from ISF were measured rather than from blood. A simple multiple linear regression (MLR) analysis model was used to regress spectral data to glucose concentrations. The study also attempted to find the most informative wavelength in the proposed MIR range by extracting all possible subsets of wavelengths with minimum size of 1 wavelength and maximum size of 3 wavelengths then applying cross-validation technique. The result claimed that the MLR model using three wavelengths 9523nm, 9345nm, and 9090nm had comparable outcome to those obtained by reference PLS model using a larger number of wavelengths.

Table 1-2. Literature Summary

Ref.	Sample Concentration Range	Spectral Range	Model Type		Metric			
					Regression		Classification	
			Regression	Classification	SECV/SEC/r	SEP	Accuracy	F1 Score
Malin et al., 1999 [21]	1st: 91-446mg/dl, 2nd: 82-294mg/dl, 3rd: 97-171mg/dl	1100-1380nm, 1450-1850nm, 2050-2375nm	PLSR		28.82mg/dl 30.63mg/dl 17.1mg/dl	138mg/dl 44mg/dl 41mg/dl		
Jeon et al., 2006 [24]	0-1000mg/dl	1100-1850nm, 2200-2500nm, 1100-2500nm	PLSR		33.51mg/dl 108.04mg/dl 69.58mg/dl	437.54mg/dl		
Amerov et al., 2004 [23]	54-540mg/dl	1111-1851nm, 2000-2500nm	PLSR		*20.5mg/dl *10.09mg/dl	21.62mg/dl 17.3mg/dl		
Jun Chen et al., 2004 [25]		1658-1769nm, 2192-2439nm	PLSR		*22.34mg/dl *5.58mg/dl	20.18mg/dl 8.11mg/dl		
Al-Mbaideen et al., 2010 [26]	20-500mg/dl	2100-2400nm	PCR		N/A	40mg/dl		

Table 1-2. Literature Summary (cont.)

Ref.	Sample Concentration Range	Spectral Range	Model Type		Metric			
					Regression		Classification	
			Regression	Classification	SECV/SEC/r	SEP	Accuracy	F1 Score
Habibullah et al., 2019 [27]	72-360mg/dl	1300-2600nm		SVM			77.5%	76%
Haider et al., 2017 [28]	0-450mg/dl	500-1200nm	***CEG				*** 90-92%	N/A
Kasahara et al., 2018 [29]		8333-10204nm	MLR		**0.36			

*: the metric used here is Standard Error of Calibration (SEC)

**: the metric used here is correlation coefficient r

***: this study did not build a regression model to predict glucose concentration level but using CEG to verify the accuracy of their monitoring device

1.3.2 Literature Review on the Strengths and Weaknesses of Common Analysis Models

The most straightforward analysis model is MLR model in which the dependent variable is modeled as a linear combination of the independent variables and the regression coefficients are estimated with the least squares criterion [30]. An MLR model can be represented as:

$$y = c_0 + c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \quad (3)$$

where: y – estimated property (e.g. glucose concentration)

x_i – wavelength variables

c_i – regression coefficients

The problem with this model is that there is no unique solution of the regression coefficients, which is known as "exact multicollinearity", when the number of variable is larger than the number of samples [31]. Even in case the number of variables is smaller than the number of samples, MLR may lead to poor prediction performance when the different variables are highly correlated.

The most widely used modeling technique is PLSR [32-33]. In a PLSR [34], dependent variables are described as function of a few principal components (or latent variables) [35], which are derived from the original variables as linear combinations that maximally capture the covariance between the independent variables and the dependent variables. The equations for PLSR are shown as follow:

$$X = TP^T + E \quad (4)$$

$$y = Tq + e = Xb + e \quad (5)$$

$$b = W(P^T W)^{-1} q \quad (6)$$

where: X – data matrix

T – score matrix. Scores are the projection of samples onto the principal components

P – loading matrix. Loadings are the projection of the principal components onto the original variables

E – matrix of residuals that represents the differences between the observed and predicted X

e – vector of residuals that represents the differences between the observed and predicted y

b – vector of regression coefficients

W – matrix of loadings

P – result of the projection of X onto the LVs

The PLSR analysis model is sensitive to its number of principal components and requires tuning in order to achieve optimal results. Increasing the number of principal components would decrease the bias and increase the variance [36]. Bias relates to the prediction-accuracy level of a model. Variance relates to the estimation error, i.e. the level of uncertainty of the predicted values [37]. Therefore, the use of cross-validation techniques is necessary for tuning the principal components numbers of model. Furthermore, it is known that even though PLSR model can capture specific covariance between the acquired spectral signals and the glucose

concentrations, it is also very sensitive to unspecific correlation, which results in a not robust enough model [38].

Because of the limitations of PLSR, other modeling techniques have been proposed. Similar to PLSR, PCR construct new predictor variables, known as principal components, as linear combinations of the original predictor variables. PCR, however, creates components without considering the response variable unlike PLSR. Therefore, PCR often needs more principal components to fit the response variable. It should be noted that PCR is based on principal component analysis (PCA) technique. PCA is a method of data reduction, representing a large number of variables by a smaller number, each of which is a linear combination of the original variables. One output of PCA is principal component scores. PCR uses those scores as independent variables in a regression.

Although SVM has been mentioned in the literature regarding the investigation on noninvasive glucose monitoring, very few research works have actually examined the use of SVM. The experiments that utilized SVM led to results of not very reliable models and outcomes [27]. However, regarding other topics, e.g. NIR spectroscopy for food, analysis of chemical data, SVM has become increasingly popular [39] both as a classification technique and the ability to solve multivariate regression problems. Furthermore, unlike other aforementioned techniques, which are effective against datasets with linearity or weak non-linearity, SVM has functions that are designed specifically for solving both linearity and non-linearity [39-40]. Another advantage of SVM is its ability to deal with high dimensional inputs [39-40]. Thissen et al. in [40] compared the performance of SVM and PLSR in two applications, i.e. the determination of two monomer masses during a copolymerization reaction and the determination of ethanol, water, and iso-propanol mole fractions in a ternary mixture. The result was that SVM outperformed

PLSR in both applications. In another study, Bulent Ustun in [41] conducted several experiments to compare the performance of SVM and PLSR. The result also indicated that, in most of the cases, SVM outperformed PLSR.

1.4 Challenges and Assumptions

Studies have shown that main reasons which limit the use of absorption spectroscopy based glucose monitoring are absorption by other biological materials, interferences due to scattering and other properties of measurement surface, calibration issues, instrument drift, time drift, baseline drift, thermal noise, physiological factors, environmental factors and proper selection of wavelength [43-47]. Therefore, the assumptions made in this study are as follows:

- Water, which is the main constituent of most living organism, is the main biological material that interferes and produces uncertainty of signals obtained in addition to the significant signals. Characteristics of glucose in distilled water solution should be investigated in detail first before other biological materials can be added in future work.
- Hardware for data acquisition does not suffer significant deviation due to both short-term and long-term machine drift. Long-term machine drift can be proved to be insignificant by replicating an experiment from a paper and obtaining similar results.
- The environmental temperature is maintained at 25^{o}C. Sample containers not heated up by the incoming light. It is the same as assuming the skin temperature is maintained and does not show variation during measurements.
- The photo probe and sample holder are designed to prevent repositioning that cause baseline drift.
- Glucose does not undergo glycolysis in distilled water.

- The sample containers are made of the same materials and would have the same optical characteristics for the same wavelengths. It is the same as assuming optical characteristics of human tissue show insignificant change for different individuals.
- Other time dependent factors, i.e. blood pressure, blood flow, atmospheric pressure, humidity, skin hydration will cause minimal effect and not considered.
- For the purpose of experimental designs, the utilized wavelengths are assumed to be both continuous and discrete. Depending on the situation, the former or latter definition will be applied.

1.5 Problems Statement and Objectives

Literature reviews in section 1.3 shows that the PLSR analysis modeling technique is by far the most popular technique used in investigating noninvasive blood glucose monitoring. In spite of its simplicity to use, speed, relative good performance, the PLSR technique still possesses some limitations, including sensitivity to unspecific correlation between dependent and independent variables, ineffectiveness against non-linear datasets, which make models built with this technique not robust enough practical usage. On the other hand, SVM is hardly known and used even though it has many advantages compared to PLSR. Studies in other similar topic also show that SVM is better than PLSR in term of performance. Therefore, it might be worth to investigate further on the use of SVM as a candidate for modeling blood glucose monitoring system.

The current few results of experiments utilized SVM indicates that using solely SVM technique is insufficient for a reliable of glucose prediction system. It is stated in [39] that when SVM being applied to highly multivariable datasets, a prior variable reduction step (such as

PCA) should be applied first. Moreover, reviews in section 1.3.1 shows that although many researchers have aimed to find the most informative wavelengths for blood glucose monitoring, they came to different conclusions; the fact suggests that some wavelengths might contain spectral interferences that affect performances. By eliminating those irrelevant wavelengths the prediction performance and robustness can be improved [31, 42]. Therefore, techniques to select relevant wavelengths or features in machine learning parlance should be investigated further in the developing of glucose monitoring system.

Furthermore, literature reviews also show that preprocessing and cross-validation techniques can also help to improve the performance and robustness as well as serve a counter for weakness of modeling techniques. It is necessary to develop a framework utilizing different techniques to optimize the performance and ensure its validity.

Motivated by these gaps as well as potentials, this thesis is dedicated to investigate the use of SVM and feature selection techniques as core algorithms to develop an automated and computationally efficient system to facilitate selecting the most informative wavelengths for accurate glucose monitoring. To achieve the ultimate goal, the thesis will focus on completing the following objectives:

1. Developing a reliable framework for data synthesis, data acquisition for further investigation on glucose levels determination model.
2. Developing a framework that allows efficient preprocessing, tuning, and assessing multiple techniques (feature selection techniques, PCA, SVM, etc.) for optimal performance
3. Investigating the effectiveness of SVM techniques by considering two different approaches: regression and classification.

4. Investigating the effectiveness and limit of the proposed techniques for dimensionality reduction to improve performance and extract informative wavelengths
5. Utilizing feature selection techniques for extracting informative wavelengths for glucose monitoring

1.6 Scope and Limitations

Since water is assumed to have the most significant interference to the signals is required for the preparation of every type of samples, the research focuses on investigating the effectiveness of the proposed machine learning techniques using glucose in distilled solution as samples. Effects of other biological materials, e.g. hemoglobin, fat, etc. are considered beyond the scope of this study and left for future work.

Furthermore, due to inaccessibility to verified datasets that contained suitable information, we chose to use a published dataset from [27] and prepare a new dataset using a set of sample prepared by provided information in [27]. Although there are two different datasets, they actually represent measurements from the same set of samples twice with differences in time and conditions. In other words, it is similar to testing a return patient. This is a typical short coming that most of the promising models have, in which results are better only for single subject [5]. Therefore, it is important to take into account the assumptions in the previous section when verifying the validity of this thesis' proposed model.

1.7 Summary of Contributions

In this study, we hypothesize and proceed to investigate whether the use of SVM incorporating other machine learning techniques will be effective enough to establish the relation between spectral wavelengths and glucose concentration for future application on noninvasive monitoring. The main contributions of the work are as follow:

- SVM is adapted to calibrate the relationship between wavelengths and glucose concentration and predict a glucose concentration based on provided wavelength signal. SVM is demonstrated to be able to achieve up to 91% testing accuracy and 90.97% f1 score when used in combination with preprocessing, mutual information, select k best, sequential forward floating selection (SFFS), pipeline, nested cross-validation techniques, etc. Details of this contribution can be found in section 5.1 and 5.4
- Wrapper methods of feature selection are essential for glucose monitoring model using SVM. However, each of the method can be used alone to extract informative wavelengths for noninvasive glucose monitoring as well as provide much more efficient and accurate performance. Details of this contribution can be found in section 5.2.
- Filter methods of feature selection offer a trade-off between speed and performance for the proposed models when used in combination with wrapper methods. If time is an important constraint, then techniques of filter methods should be added to the system for much faster speed with slight drop of performance. Details of this contribution can be found in section 5.2 and 5.4

- The assumption whether features are discrete or continuous does not have much influence because it only makes the accuracy or standard error to vary within 1-2 percent or unit of mg/dl whereas choosing the approach of either classification or regression (i.e. assuming whether glucose concentrations are discrete classes or continuous values) when performing Sequential Forward Floating Selection (SFFS) technique of wrapper methods has a great effect on the accurate performance of the system. Base on the obtained results, it is recommended follow classification approach when perform SFFS technique. Details of this contribution can be found in section 5.2
- A framework consisted of 3 major phases is proposed to allow tuning and assessing the performance of SVM and other techniques. The framework proves promising in optimizing the prediction of glucose concentration while minimizing bias due to information leakage. Details of this contribution can be found in section 5.4
- The proposed system model was show through experiments to possess an ability to extract the relevant information out of any random input wavelength. For the examined range between 1300-2600, the informative wavelengths fall in the ranges should fall in the ranges of 1300-1600nm and 1800-2400nm or 2000-2600nm. Details of this contribution can be found in section 5.3

1.8 Organization of the Thesis

Chapter 1 Introduction introduces the background information and motivation for the study; reviews current findings regarding the noninvasive approach for glucose monitoring; provides the remaining problems, the objectives, the assumptions, and the overall contributions. Chapter 2

Methodology describes and explains the rationale for utilizing each chosen method, technique and what they aim to solve; introduces the overview of system model and the process of developing the framework for data synthesis, data acquisition, and software structure; describes the development of the proposed models. Chapter 3 focus on the optimization of proposed models by providing strategies, techniques for hyperparameter tuning and validation; describe and explain the selection of hyperparameter values. Chapter 4 Experimental Design describes and explains a variety of models based on the system model; explains how datasets and the proposed were used to evaluate the effectiveness of introduced methods, techniques; evaluates the effectiveness of selected features. Chapter 5 Results presents the results and analyzes and discusses the results. Chapter 6 Conclusion and Future Work summarizes and gives conclusion on the study; discusses future work.

Chapter 2 Methodology

This chapter introduces the techniques that were used to build the proposed system model and also explains why each of the technique was selected. This chapter also provides the overview workflow of the system model and the details, including concepts, important settings, parameters, procedure, of each step of the system model.

2.1 Rationale for Utilized Machine Learning Techniques

Support Vector Machine (SVM) is a supervised learning method whose applications have been successful in several areas. Although SVM has been proved to have performance advantages compared to other techniques, e.g. PLSR, in other fields, SVM is hardly known and used on absorption spectroscopy for establishing the relation between glucose concentrations and spectral wavelengths. Reviews on the few studies of SVM for glucose monitoring indicate that using solely SVM is insufficient for a reliable of glucose prediction system. Therefore, it is hypothesized that the effectiveness of SVM on predicting glucose concentration can be improved by incorporating other machine learning techniques into the model.

SVM is known to not be scale invariant, i.e. difference distances between feature data points would affect the performance of SVM in different ways. Therefore, it was decided a scaling step shall be added at the very start of the model. In addition, since most machine learning algorithms, including SVM and even scaling techniques are incompatible with datasets that have missing values, often encoded as blanks or not-a-number (NaN), an extra technique called data imputation was implemented together with scaling techniques. These techniques were assembled

into Pre-processing step which was deployed immediately after datasets were formed. More details on the Pre-processing techniques can be found in section 2.2.3.

Although SVM has an advantage that it can still be effective in high dimensional space, studies have suggested that it would be a good practice to include a dimension reduction step prior to SVM for optimal performance. This is related to a machine learning phenomenon called curse of dimensionality [48] in which a larger number of feature, while providing extra information, also may:

1. introduce a loss in performance due to poor correlation or noise;
2. increase the risk of overfitting when there are more features than observations;
3. increase computational complexity due to observations to appear equidistant from others.

There are two types of dimensional reduction technique available, i.e. supervised and unsupervised. In supervised techniques, relevant features are selected from labeled input features which mean selected features can be traced back and identified; whereas, this is impossible for unsupervised techniques. Moreover, selected features are also the features that contain the most information regarding the studied phenomena, which is the relation between spectral wavelengths and glucose concentration in this case. Therefore, we believed that a dimensional reduction step consists of both supervised and unsupervised techniques was necessary and implemented into the system model. In this dissertation, supervised techniques will be further referred as feature selection and unsupervised techniques will be referred as feature extraction due to the nature of the utilized techniques. More details on the feature selection and feature extraction techniques can be found in sections 2.2.4.

It should be noted that most techniques in the literature tried to determine glucose concentration by using multivariate regression techniques, i.e. to develop multivariate regression models to predict glucose concentration based on input wavelengths. SVM, however, is traditionally a classification technique, i.e. it decides which predefined level of glucose concentration a sample belong based on input wavelength, but can be extended to solve regression problems. Therefore, in order to enable direct comparisons between SVM and other techniques, this study investigates SVM using two approaches: regression and classification. Besides, it is beneficial to investigate whether different approaches would actually bring forth valuable results. Nevertheless, it should be noted that the goals and applications of the two approaches are slightly different. A successful regression model would enable the ability to predict specific glucose concentration values based on future input while a successful classification would allow us to decide the condition of a subject, e.g. whether a patient has diabetes, has high, low, or normal blood glucose, etc. A classification model can also predict specific value with enough number of glucose concentration levels (classes), precision, and recall. For regression, popular metrics such as Standard Error of Cross-Validation (SECV) and Standard Error of Prediction (SEP) would be used for evaluation. For classification, since there is not a strong literature, different metrics, i.e. accuracy and f1 score, are chosen for the evaluation purpose. More details on the evaluation metrics can be found in section 3.1

One important consideration is that SVM is very sensitive to several essential model parameters, such as C, gamma, kernel, etc. In machine learning parlance, these parameters are called hyperparameters whose values are not directly learnt within the estimator but must be provided as arguments. In order to achieve the optimal performance of SVM, it is recommended to try various values for these hyperparameters. However, it is not helpful to investigate an

influence of a single hyperparameter since different values of all hyperparameters would interact with each other and lead to different results. It is tedious and difficult to manage a large number of combinations of hyperparameters values when arguments are passed manually. Therefore, it was decided a hyperparameter-value searching algorithm must be added to system model.

The addition of hyperparameter-value searching algorithm, in turn, raises another problem. For each combination of hyperparameters' values, the model must be evaluated to find the optimal one. In other words, the input dataset must be used repeatedly that it increases the risk of overfitting when the final prediction is conducted because information has been leaked and the model has already learnt the true results. Ideally, the solution to this problem is to use different datasets for training and testing. However, because we lack access to available data, it was decided different strategy called nested cross-validation, which basically spit the input dataset into several training and testing sets without reducing available number of observations for sufficient training, would be deployed. In order to simplify and combine together hyperparameter tuning algorithm, nested cross-validation, SVM, and other machine learning technique, another technique called pipeline was utilized. More details on the hyperparameters-value searching, nested cross-validation and pipeline are discussed further in chapter 3.

2.2 Descriptions of the System Model and Utilized Machine Learning Techniques

The following subsections will start by giving the overview of the system model, what it consists of, and then go on to describe with more details each step, i.e. data acquisition, pre-processing, dimensional reduction, and support vector machine, as well as the utilized techniques of that step.

2.2.1 Overview of the System Model

Based on the provided rationale in the previous section, the general system model was developed and illustrated in Figure 2.1. The general system model consists of three phases. The first phase is Data Acquisition, in which samples were prepared; hardware and procedure for measurement were decided; and datasets were formed. The second phase, which is also the main focus, is the training phase which consists of four main steps: pre-processing, dimensional reduction, SVM, and hyperparameter tuning. The dimensional reduction steps include two sub-steps: feature selection and feature extraction. The SVM step is divided into two approaches: regression and classification. The hyperparameter tuning comprises the hyperparameter-value searching technique and nested cross-validation technique. The optimal model output from SVM from phase 2 was then deployed into phase 3, i.e., a testing pipeline, so that it can be used for predicting glucose concentration. Future input data will go through similar pre-processing step in phase 2 before it can be passed to the deployed model. In this study, however, a reserved portion from the original input dataset for phase 2 was used as the testing set. More details on each step are described in following sections.

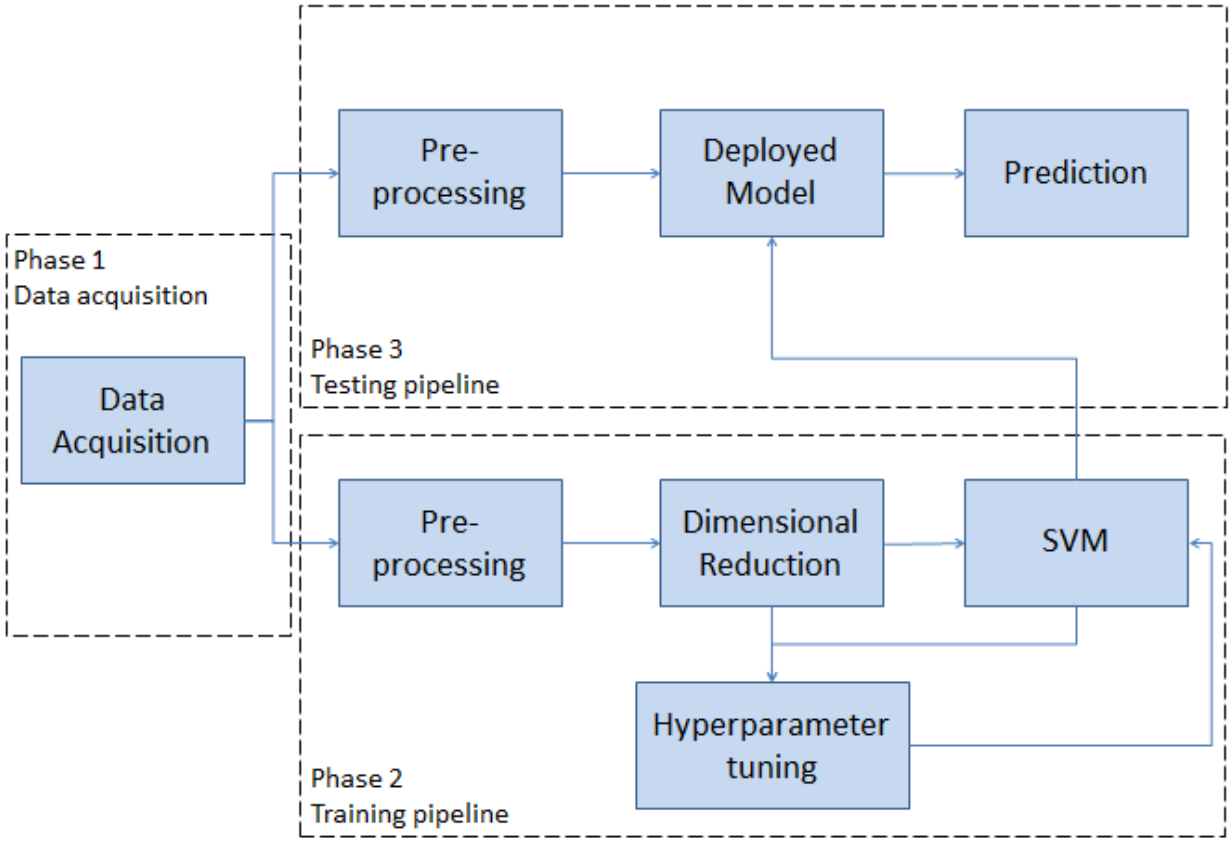


Figure 2-1. Overview of the system model.

2.2.2 Data Acquisition

Data acquisition has three important aspects to be reviewed. The first one would be the selection of hardware to use for measuring and processing signals. The second one would be the design of datasets, i.e. how signals are measured and stored. The last one would be the preparation of samples and the use of hardware to form datasets as designed.

2.2.2.1 Hardware

A literature review shows that most of the studies regarding noninvasive blood glucose monitoring conduct experiment on the first overtone region and the combination overtone region

of near-infrared (NIR) spectrum, i.e. 1100-1850nm and 2000-2500nm. Therefore, in order to make comparison to other works, it was decided to also utilize wavelengths belonging to the aforementioned region of NIR spectrum. However, it should be noted that this selection of wavelengths is not necessary as one of the goals of this study is to identify the most informative wavelengths from the input wavelengths. In other words, the proposed system would try to identify the most informative wavelengths for blood glucose monitoring from any random wavelength inputs.

In order to minimize machine drift and other technical issues, this study utilized the Neospectra Micro Development Kit, a product of Neospectra Si-ware System Company [49]. The kit provides a light source capable of emitting light radiation between 1300-2617nm in a span of few seconds. There is not a fixed value between each step of the radiation. However, there are always 158 steps which mean 158 wavelengths in total. The kit also comes with an optical core module that can received reflected light from samples and calculate the absorbance of the sample. The optical core module can be connected directly to any single-board computer that support SPI interface. In this study, the module was integrated to a Raspberry Pi board.

The remaining steps were conducted on a personal laptop, DELL PRECISION series M6700. The system configuration that was used is as below:

- Processor speed: 2.70 GHz
- Number of Processors: 8 logical processors
- Total number of cores: 4 cores
- System version: Microsoft Windows 7 ver. 6.1.7601
- Installed physical memory (RAM): 8.00 GB
- Utilized language and library: Python and Scikit-learn [50], Mlxtend [51]

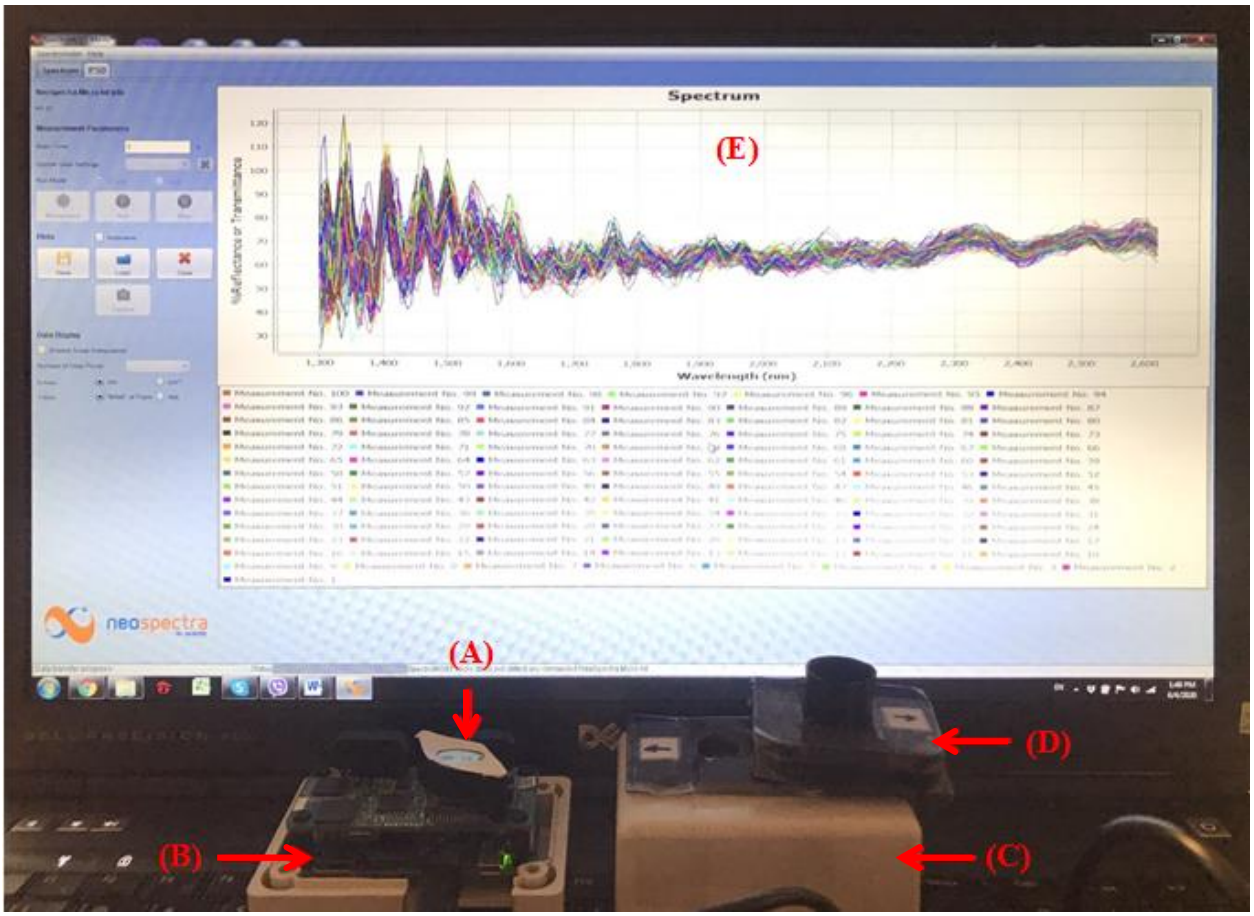


Figure 2-2. The Neospectra Micro Development Kit (A) connects to the Raspberry Pi board (B). The box (C) ensures that there is no optical interference from the ambient environment. The black accessories (D) ensures each sample container is place in the same place with each other and prevent baseline drift.
Each measurement of a sample is recorded and plotted via a software on a computer (E).

2.2.2.2 Datasets Description

In this research work, two datasets have been used: dataset C0 and dataset C1. Dataset C0 was obtained from [27] to serve as ground truth for validating the result. The dataset consists of

158 columns and 100 rows. Each column represents the wavelengths or features in machine learning parlance. All of the wavelengths are in the range between 1300-2600nm, which belong to the short-wave infrared (SWIR) band. Each row represents an observation. Therefore, there are 15,800 data points. Each data point is the value of absorbance of glucose solutions in the distilled. In addition, there is extra column contained information regarding the classes. Each value of classes represents a distinct concentration of glucose solution. There are 10 different glucose concentrations represent 10 different samples. Each sample was measured 10 times. Therefore, the first 10 rows contain information regarding the absorbance of sample 1 or class 1; the next 10 rows contain information regarding the absorbance of sample 2 or class 2 and so on.

There exists a risk that the number of observations of 100 observations in dataset C0 was too small and would not provide sufficient information which could lead to several problems such as high bias or high variance results. Therefore, the dataset C1 was formed as a solution to the problem. Similar to dataset C0, dataset C1 consists of 158 columns and 1000 rows. All of the wavelengths are the same as those of dataset C0. The samples are prepared by the same information provided in [27]. Table 2-1 shows the detailed values to prepare samples for both datasets. The 10 samples are now measured 100 times each. Therefore, there are 158,000 data points. The first 100 rows contain information regarding the absorbance of sample 1 or class 1; the next 100 rows contain information regarding the absorbance of sample 2 or class 2 and so on.

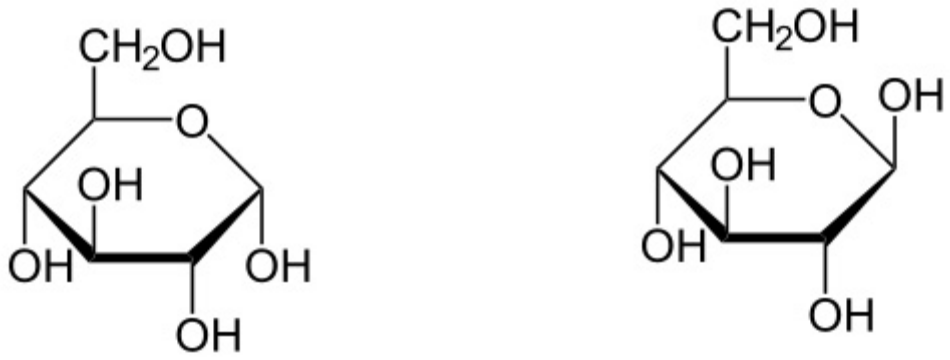
2.2.2.3 Samples Preparation and Measurement Procedure

The samples are prepared by dissolving glucose in distilled water. There are two isomer of glucose called D-glucose and L-glucose. Only D-glucose is biologically active while L-glucose cannot be used by cells. The D-glucose can exist in two forms: α-D-glucose and β-D-glucose.

They differ only in the direction that -H and -OH groups point on carbon 1. When α-D-glucose molecules are joined chemically to form a polymer, starch is formed. When β-D-glucose molecules are joined to form a polymer, cellulose is formed. Most of the β-D-glucose remains in the blood stream because human body does not have an enzyme to break down cellulose. Conventional invasive glucose monitoring devices use an enzyme called glucose oxidase to catalyze the oxidation of β-D-glucose into D-gluconic acid and generated hydrogen peroxide. The hydrogen peroxide can then be used to oxidize a chromogen or the consumption of oxygen measured to estimate the amount of glucose present [52]. Therefore, in this study, β-D-glucose was used to prepare the samples to minimize any unforeseen complication.

Table 2-1. Concentration for each sample together with their required volume of distilled water and mass of D-glucose powder

Sample	**Concentration (mol/l)**	**Concentration (mg/dl)**	**Mass of β-D-glucose (g)**	**Volume (l)**
1	0.0040	72.072	0.180156	0.250
2	0.0060	108.108	0.270230	0.250
3	0.0070	126.126	0.315270	0.250
4	0.0082	147.748	0.369300	0.250
5	0.0100	180.180	0.450390	0.250
6	0.0110	198.198	0.495400	0.250
7	0.0137	246.847	0.617030	0.250
8	0.0156	281.081	0.702600	0.250
9	0.0174	313.513	0.783600	0.250
10	0.0200	360.360	0.900780	0.250



(a) (b)

Figure 2-3. (a). A α-D-glucose molecule. (b). A β-D-glucose molecule

In order to prepare the 10 samples with 10 different glucose concentration provided in Table 2-1, we need to calculate the mass of β-D-glucose and the volume of distilled water to used. Firstly, the following formula was used:

$$Glucose\ mass\ (gram) = Concentration\ \left(\frac{mol}{L}\right) \times Molar\ Mass\left(\frac{gram}{mol}\right) \quad (7)$$

Where Molar Mass is the glucose molar mass. Considering we have the formula of glucose as $C_6H_{12}O_6$ and the atomic mass of Carbon, Hydron, and Oxygen as 12.011, 1.008, and 15.999 respectively, the molar mass for glucose would be:

$$Molar\ Mass = (6 \times 12.011) + (12 \times 1.008) + (6 \times 15.999) = 180.156\ \left(\frac{gram}{mol}\right) \quad (8)$$

Therefore, for instance, to prepare sample 1 with the concentration 0.0040 mol/L, we need:

$$Glucose\ mass\ (gram) = 0.0040\ \left(\frac{mol}{L}\right) \times 180.156\left(\frac{gram}{mol}\right) = 0.720624\left(\frac{gram}{L}\right)$$

Because of a restriction on available of laboratory equipment, the volume of 0.250mL of distilled water was used instead. Hence, for preparing sample 1 with glucose concentration of

0.0040 mol/L, we need $\frac{0.720624}{4} = 0.180156$g of β-D-glucose and 0.250mL of distilled water. Other samples were prepared similarly by the same method.

The procedure for preparing the samples can be summarized as follow:

- Put weighing paper on the molecular balance then press tare to calibrate the weight back to 0
- Weight the exact mass of glucose using the above table
- Transfer the amount of glucose to a 500ml beaker
- Use a cylinder to take 200ml distilled water
- Pour the amount of water into the 500ml beaker that contains glucose
- Mix the solution using magnetic bar and magnetic stirrer
- Pour 200ml glucose solution from the 500ml beaker into a 250ml volumetric flask
- Use a pipette to add more distilled water until the volume reaches exactly 250ml
- Invert the flask 2-3 times
- Transfer the solution into a container
- Label the container

Repeat the procedure for all concentration levels. It is important make sure all of the containers are air-tight since glucose can be oxidized when exposed to the environments. All of the samples should be stored in dark, dry place in room temperature since glucose solution can freeze easily and form crystals which would interfere with the measurement process.

The measuring procedure was done in the dark room to prevent ambient light from interfere with the results. The temperature was set to 25^{o}C. Each sample was transferred to a cuvette with a flat surface in contact with the surface of the light source and transducer to prevent scattering

and reflection due to poor angle positioning from happening. Using the hardware and software described in section 2.2., each sample was measured 100 times. Each sample generated an excel file (mini dataset) contained 159 columns and 100 rows. The 10 mini datasets were then joined to formed dataset C1.

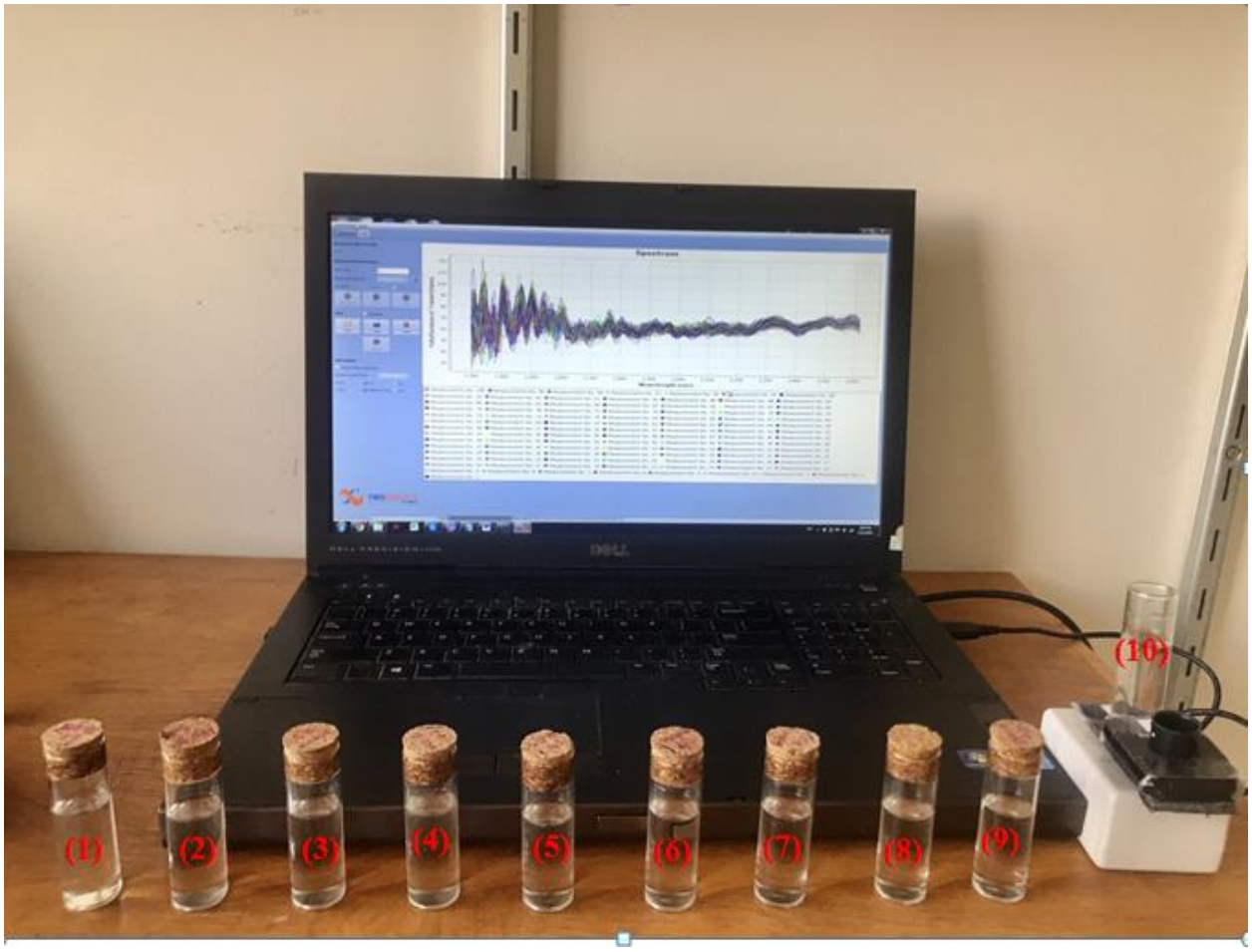


Figure 2-4. D-Glucose samples with 10 different concentrations and the hardware setup

2.2.3 Pre-processing

Pre-processing is an important step that is required immediately after data acquisition. As shown in Figure 2.4, after all samples have been measured and a dataset is properly formed, the dataset is passed to pre-processing step. The objective of the pre-processing step is to complete

two tasks: fixing the datasets off any missing or NaN values, and scaling the datasets to prevent high dynamic values causing bias.

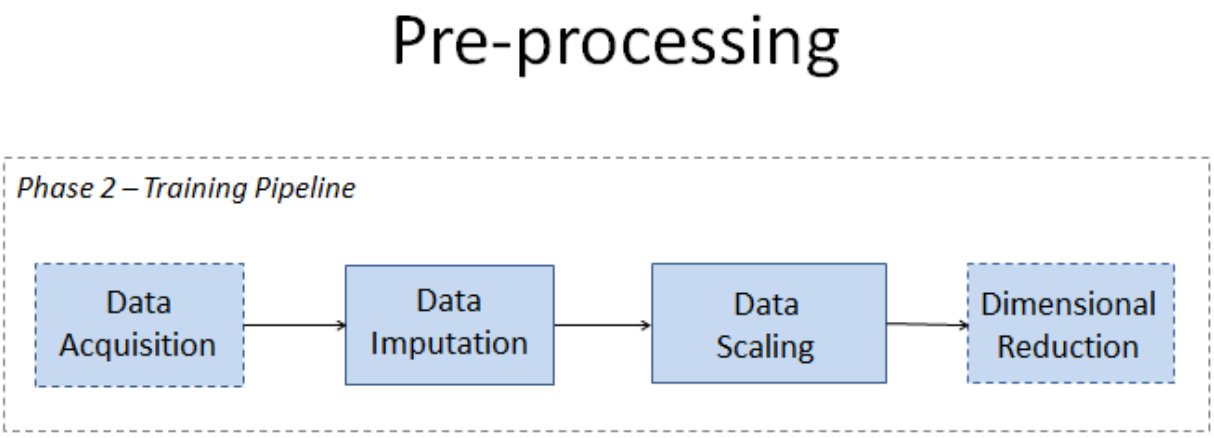


Figure 2-5. Techniques in the Pre-processing step

Firstly, missing values, often encoded as blanks, not-a-number (NaN) values, are identified and removed or replaced. This act is important because any further machine learning algorithms are incompatible with datasets that have such values. A basic strategy is to discard the entire row contained missing value. Nonetheless, this might come at the price of losing value information. A better strategy is to impute the missing values, i.e. to infer them from the known part of the dataset. There are two types of imputation algorithm: univariate and multivariate. Univariate imputation imputes values in the i-th feature dimension using only non-missing values in that feature dimension. In contrast, multivariate imputation uses the entire set of available feature dimensions to estimate the missing values. Because each feature in the study represents a distinct wavelength, we want to replace the missing values using neighboring information from the feature containing the missing value only. Hence, *SimpleImputer*, a univariate imputation technique provided by scikit-learn library, was utilized. *SimpleImputer* can impute missing values using three different statistics, i.e. mean, median, and most frequent. The first one is to

pick the mean value of neighboring values for the missing value. The second option is to calculate the median of neighboring values and replace the missing value with that value. The last solution is to simply choose the most frequent value in the dataset to be in place of the missing value. Since we had not known the range and variation of the neighboring values, the median option was chosen. To use the median option, the parameter *strategy* was set to *strategy='median'*.

In this particular study, our datasets have approximate 10 missing values, which are caused by the process of joining several small files of separate measurements to form a complete dataset. Each and every missing value was replaced by a value obtained from calculating the median of neighboring values. After fixing all of the missing values, all values in the dataset are normalized into the same dynamic range to prevent large weights creating biased results which might lead to problems in latter steps. Because each sample has different level of glucose concentration that impacts the amount of NIR light intensity absorbed, the measured values in each observation might have a different dynamic range. Computations do on dataset with various dynamic ranges can create large weights and pull other values toward it producing biased results.

Three techniques for scaling the data were considered: *MinMaxScaler, MaxAbsScaler, and StandardScaler*. The first two techniques work in a similar fashion. *MinMaxScaler* scale features to lie between a given minimum and maximum number, often between 0 and 1. The formula for the technique is:

If

$$Standard\ deviation\ of\ X = \frac{(X - \min(X))}{(\max(X) - \min(X))} \tag{9}$$

Then

$$Scaled\ X = Standard\ deviation\ of\ X * (MAX - MIN) + MIN \quad (10)$$

Where MAX is the upper limit and MIN is the lower limit of the new feature range.

MaxAbsScaler scales in a way that the training data lies within the range [-1, 1] by dividing through the largest maximum value in each feature. The advantages of *MaxAbsScaler* over *MinMaxScaler* are its robustness to very small standard deviation and its preservation of zero entries in sparse data, i.e. data with lots of gaps information, or zero values, of different features.

In real practice, features often do not look like standard normally distributed data, i.e. Gaussian with zero mean and unit variance. However, many elements used in objective function of a learning algorithm (such as RBF kernel of Support Vector Machine) assume that all features are centered around zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected. Therefore, *StandardScaler* technique that transforms the data to center it by removing the mean value of each feature then scale it by dividing non-constant features by their standard deviation is also considered.

StandardScaler standardize features by removing the mean and scaling to unit variance and then scale it by dividing non-constant features by their standard deviation. Since many elements used in objective function of a learning algorithm assume that all features are centered around zero and have variance in the same order, it is necessary to apply *StandardScaler* to prevent a feature's variance with high magnitude to dominate the classifier to make them less accurate.

Because of the experimental design, each sample was measured continuously until the light source stop operating; both the datasets C0 and C1 would not be sparse data. Therefore, *MaxAbsScaler* was considered not necessary. On the other hand, it is believed that

MinMaxScaler would not be suitable for real-life datasets with many possible outliers. This left us with the choice of *StandardScaler* for scaling datasets.

2.2.4 Dimensional Reduction

The dimensional reduction step consists of two sub-steps: feature selection and feature extraction. The goal of both sub-steps is to reduce the feature-space dimension while retained related information and hence reduces the complexity and enhances the performance in term of increased accuracy, precision, recall, or decreased standard error. Feature extraction is distinguished from feature selection based on how it reduces the dimensional space. Feature selection examines possible combinations of input feature and creates the subset that would produce best result. Feature extract, in other hand, "extracts" the essential information from input feature and transform and project them in a lower dimensional space. This study utilized both techniques to further reduce the already filtered input for optimal performance.

2.2.4.1 Feature Selection

Techniques in Feature Selection method are further categorized into 2 main groups: filter methods and wrapper methods. In contrast to filter methods, which prove to have excellent speed, wrapper methods produce much more accurate performance [53, 54]. Therefore, experiments in this study were designed so that the efficiency of only wrapper methods could be examined first and then the efficiency of the combination of both methods later. More details on the experimental design can be found in chapter 4.

2.2.4.1.1 Filter Methods

When used in combination with wrapper methods, filter methods were adopted and applied first to conduct initial screening; filter out unimportant features using fast and simple statistical techniques to lessen the burden for a more accurate wrapper method later Figure 2.5 demonstrates filter methods that were considered.

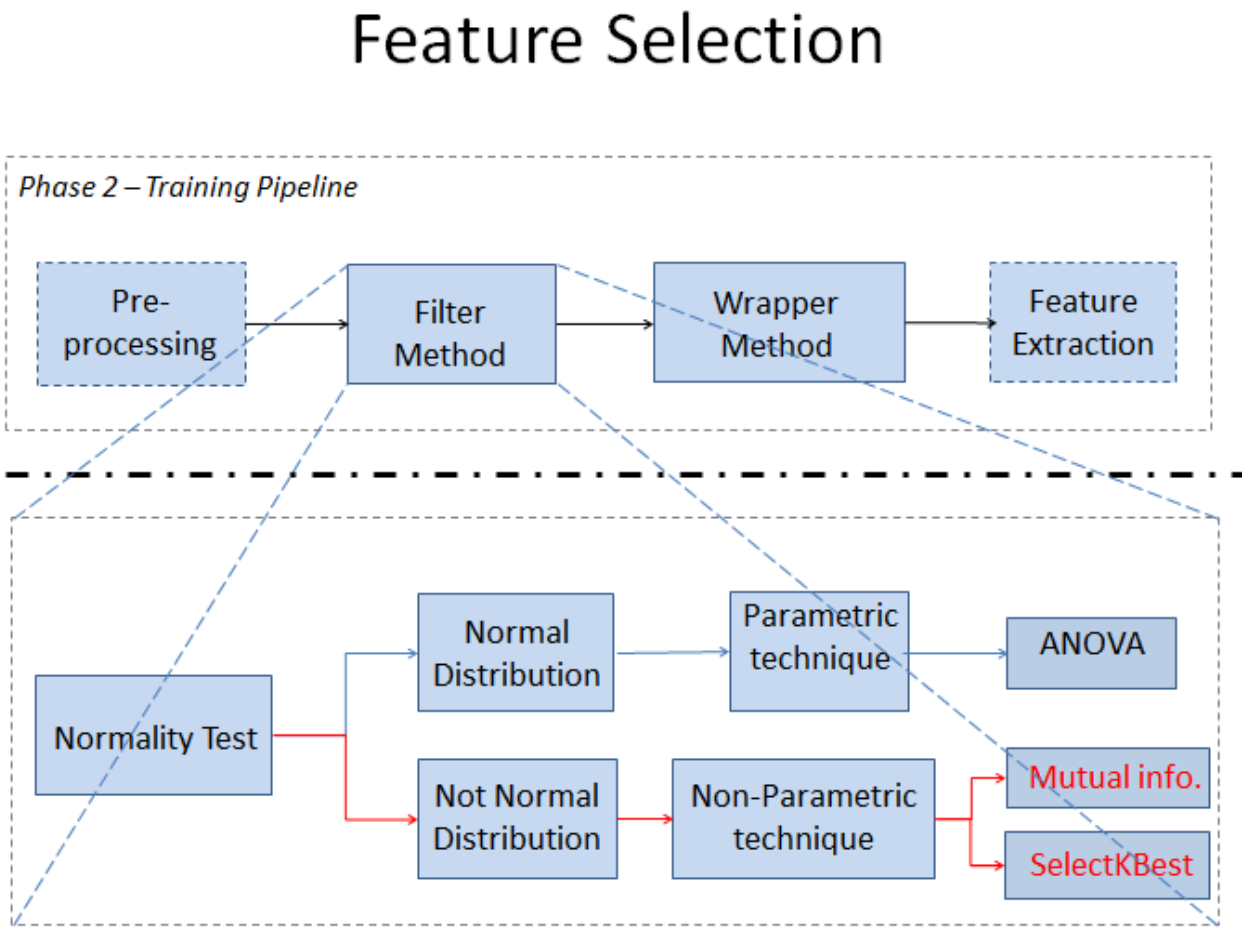


Figure 2-6. Filter methods of feature selection

It should be noted that the performance of each type of filter method technique depends on the distribution of the data; hence, a normality test was applied beforehand. There are many techniques to test for a normal distribution which can be divided into two groups: visualization

techniques, and statistical inference techniques. Visualization techniques include histogram, box plot, QQ plot. These test draw plots based on the dataset and require human interaction to decide on the conclusion. Since they are rather subjective and not automatic, they were not used in this study. Among available Statistical inference techniques Shapiro Wilk test is the most powerful test when testing for a normal distribution as it has been developed specifically for this distribution. To use the test, we can simply import a build-in function from *scipy* library called *"shapiro"*. If the p-value of the Shapiro Wilk test is larger than 0.05 then a dataset is normally distributed and vice versa. The test results indicated that both datasets C0 and C1 in this study did not have normal distribution.

If the data were normally distributed, parametric techniques, e.g. *ANOVA*, would have been used. Since the datasets were not normally distributed, non-parametric techniques were used. In this study, *mutual information* (*MI*) technique [55] and *SelectKBest (SKB)* were selected as non-parametric techniques for feature selection. *MI* worked as a scoring function, outputting scores reflecting in the importance of the features. *SKB* technique used the output of *MI* to judge which features should be chosen.

MI is said to be a perfect statistic for measuring the degree of relatedness between data [56]. Let (X, Y) be a pair of random variables, *MI* is defined as:

$$MI(X,Y) = \int\int dxdy\mu(x,y)log\frac{\mu(x,y)}{\mu_x(x)\mu_y(y)} \quad (11)$$

where $\mu_x(x) = \int dy\mu(x,y)$ and $\mu_y(y) = \int dx\mu(x,y)$ are the marginal densities of X and Y.

The advantages of *MI* include:

- *MI* can detect any kind of relationship between variables, whether it involves the mean values or the variances.

- *MI* is insensitive to the size of the dataset
- *MI* has a straightforward interpretation as the amount of shared information between variables (unit in bits)

However, calculating *MI* is not always easy. *MI* can be calculated if the underlying probability distribution is known, which is not usually the case. Therefore, *MI* must be estimated form the statistics of our dataset. A. Kraskov et al. [57] suggested *MI* estimator based on entropy estimated from k-nearest neighbor distances which, compared to conventional estimators based on binnings, are more data efficient, adaptive (higher resolution with more numerous data), and minimal bias. Then, *MI* in Equation 11 can be estimated using the following equation:

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{12}$$

where H is the function to estimate the entropy.

This study utilized function *mutual_info,* which offers use of k-nearest neighbor of scikit-learn library to estimate the mutual information between each feature (wavelength) and the glucose concentration. On the other hand, Ross et al. [56] said that the procedure for estimating *MI* depends on whether X and Y take discrete values or are continuous variables. Because this study aims to investigate SVM using both approaches of regression and classification, glucose concentrations (which are Y in this case) are considered as both discrete and continuous values. Similarly, wavelengths (which are X) can be considered as discrete or continuous. Therefore, the function needs to be tweaked and tuned.

There are two variations of *mutual_info*: mutual_info_classif and mutual_info_regression. The first one is used when glucose concentration is considered discrete. The second is used when glucose concentration is considered continuous. Each *MI* function variation has *MI* two important hyperparameters: *discrete_features* and n_*neighbor*. The hyperparameter

discrete_features is used to control whether wavelengths are considered discrete or continuous. The remaining hyperparameter n_*neighbor* determines the k number of nearest neighbors to use for *MI* estimation. SKB has one important hyperparameters: *k*. These hyperparameters, like the hyperparameters of SVM, need to be to be tuned and validated in order to achieve the best performance. More details on the hyperparameters and their tuning process can be found in chapter 3.

2.2.4.1.2 Wrapper Methods

The surviving features from filter methods were then input into wrapper method for further optimization. Figure 2.4 demonstrates popular techniques in wrapper method including exhaustive search (ES), sequential forward selection (SFS) and sequential backward selection (SBS), sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) [51, 53, 54].

In terms of optimal performance, ES is the best, followed by SFFS, SBFS, and then SFS and SBS. ES technique calculates all possible combinations of features and uses a simple classification/regression algorithm as to evaluate performance of each combination. Features within the combination with the best score are chosen while the rest are eliminated. However, because the smallest combination consists of 1 feature and the largest combination consists of all 159 features, it is obvious the total number of possible options would be gigantic and take heavy toll on computational resources; hence, not practical for wearable applications.

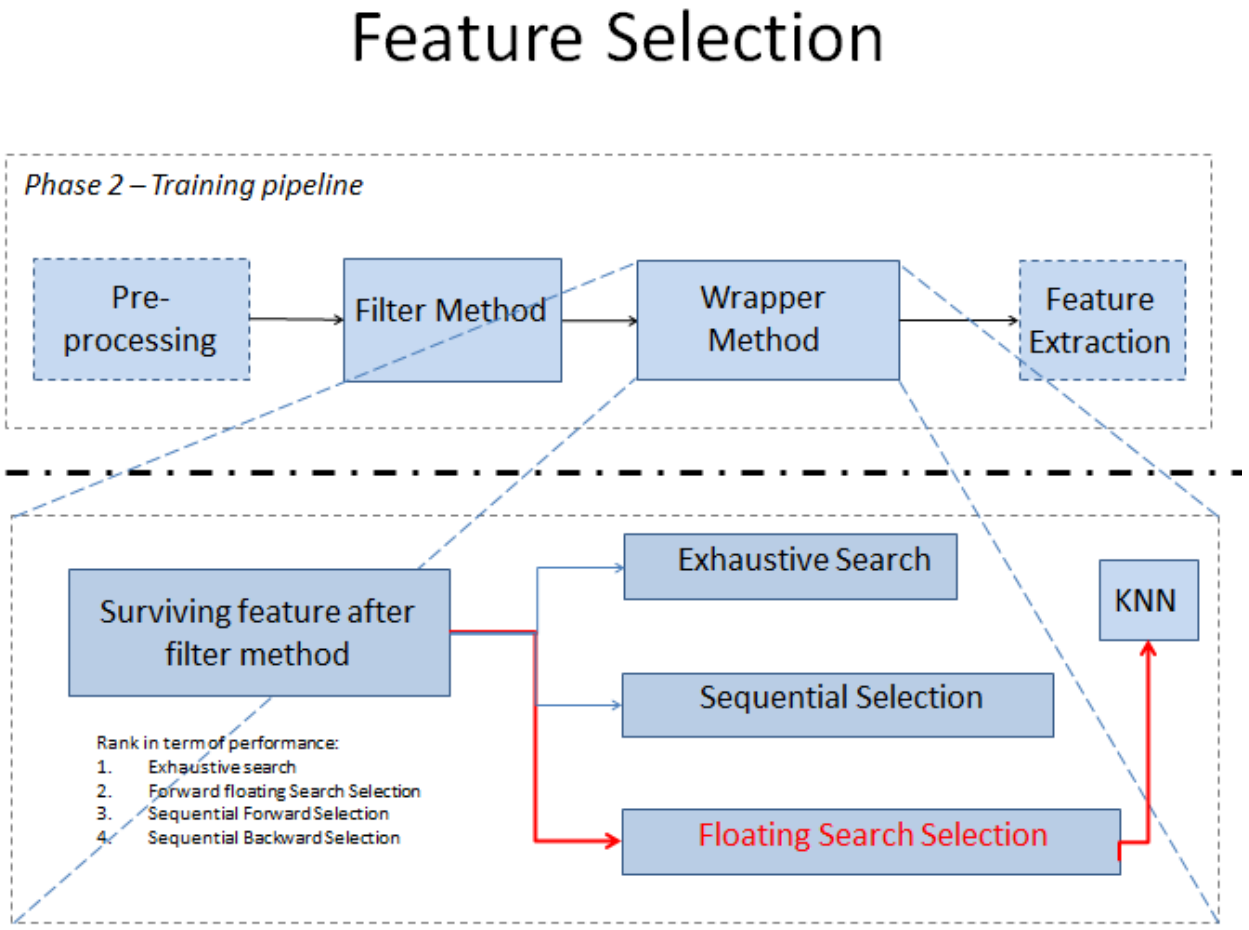


Figure 2-7. Wrapper methods of feature selection

In terms of time efficiency, SFS and SBS techniques are the fastest. Fundamentally, the techniques add or remove one feature at the time based on the simple classifier performance, which is KNN's in this case, until a feature subset of desired size *k* is reached. Nevertheless, their drawback is the lack of accuracy [53, 54]. To counter this issue, SFFS and SBFS techniques were developed. These floating variants can be considered as extensions to the simpler SFS and SBS algorithms. The floating algorithms have an additional exclusion or inclusion step to remove features once they were included (or excluded), so that a larger number of feature subset combinations can be samples. It is important to emphasize that this step is conditional and only

occurs if the resulting feature subset is assessed as "better" by the criterion function after removal (or addition) of a particular feature. Therefore, the optimal choice is actually the sub-optimal solution of SFFS and SBFS with some limited in the accuracy but have reasonable speed. Since the only difference between SFFS and SBFS is the starting point, SFFS was chosen as the technique for feature selection step. The output of SFFS is the reduced number of wavelength. These are also the informative wavelength that needed to be determined in the objective 5.

This study utilized the MLxtend library [51] to perform the SFFS for informative wavelengths. Similar to filter method, SFFS also needs an estimator to evaluate and produce score as a foundation for feature selection process. K-nearest Neighbors (KNN) [58] algorithm was used as utilized for this study due to its low complexity and fast speed. The scoring metric for KNN were accuracy for classification approach and MSE for regression approach. Details on the setting and tuning of hyperparameters of wrapper methods are described in section 3.4.2.

2.2.4.2 Feature Extraction

After the informative set of features was determined, their essential information known as principal components was then extracted allowing even smaller dimensions and enhanced performance. Principal Component Analysis (PCA) [59] is one of the most popular techniques in feature extraction. PCA technique was implemented to identify the similarity that still remained in the subset output after feature selection. The desired goal is to reduce the dimensions of a $\boldsymbol{d}$-dimensional dataset by projecting it onto a ($\boldsymbol{k}$)-dimensional subspace (where $\boldsymbol{k} < \boldsymbol{d}$) in order to increase the computational efficiency while retaining most of the information. PCA do this by computing eigenvectors (the principal components) of a dataset and collect them in a projection matrix. Each of those eigenvectors is associated with an eigenvalue which can be interpreted as

the "length" or "magnitude" of the corresponding eigenvector. If some eigenvalues have a significantly larger magnitude than others then PCA will drop the "less informative" eigenpairs. The procedure is summarized as follow:

- Standardize the data.
- Obtain the Eigenvectors and Eigenvalues from the covariance matrix or correlation matrix, or perform Singular Vector Decomposition.
- Sort eigenvalues in descending order and choose the **k** eigenvectors that correspond to the **k** largest eigenvalues where **k** is the number of dimensions of the new feature subspace ($\boldsymbol{k} < \boldsymbol{d}$).
- Construct the projection matrix **W** from the selected **k** eigenvectors.
- Transform the original dataset **X** via **W** to obtain a **k**-dimensional feature subspace **Y**.

Essentially, PCA technique further reduced the dimensionality in order to enhance the accuracy in the training step but it does not actually involve in the determination of suitable wavelengths. The number of k eigenvectors can be selected by modifying a hyperparameter *n_components*. The output of PCA is the transformed dataset with a further reduced feature subspace. More information can be found in section 3.4.3.

2.2.5 Support Vector Machine

The following section will attempt to expand the concept of classification and regression approaches as mentioned in section 2.1 by reviewing the principle of SVM technique.

2.2.5.1 For Classification

First of all, we need to understand the concept of SVM in order to deploy and tune for optimal performance. Traditionally, the objective of the SVM algorithm is to find a hyperplane in an N-dimensional space (N is the number of features) that distinctly classifies the data points. In its simplest form, SVM is a linear classifier of a two-class problem. Samples are of two classes g (=A or B) with c = +1 for class A and -1 for class B and are perfectly linearly separable. These samples can be used to determine a decision function to separate two classes, which in its simplest form can be expressed by a linear boundary:

$$g(x_i) = sgn(wx_i' + b) = sgn(b + \sum_{j=1}^{J} w_j x_{ij}) \quad (13)$$

where w and b are often called weight and bias parameters that are determined from the training set.

The sign of g determines which class a sample is assigned to: if positive class A and if negative class B. Any generic hyperplane (w, b) can be defined by coordinates (w, b) satisfying the condition $\mathrm{wx}' + \mathrm{b} = 0$ which divides the data space into two regions opposite in sign. If the two classes are separable we can define a 'margin' between the two classes, such that

$$\mathrm{wx}' + \mathrm{b} \geq 1, \mathrm{c} = +1 \text{ and } \mathrm{wx}' + \mathrm{b} \leq -1, \mathrm{c} = -1 \quad (14)$$

The hyperplane should be equidistant from the two extreme samples in each class. For no error to occur, the hyperplane must satisfy the following condition for all the samples providing that they are perfectly linearly separable.

$$c_i(wx_i' + b) \geq 1 \quad (15)$$

However, there are an infinite number of possible hyperplane (w, b) satisfying this so there needs to be a further rule to determine which of these hyperplanes is best. Therefore, the optimal

separating hyperplane is the one which has the largest margin between the most similar samples in each group. The samples on the margins are called support vectors (SVs). Note that this hyperplane now depends only on the SVs, and other samples have no influence over the hyperplane. The optimization task of finding the largest margin can be expressed by the structure error function:

$$\varphi(w, b, \alpha) = \frac{1}{2}(ww') - \sum_{i \in N_{sv}} \alpha_i (c_i[wx_i' + b] - 1) \quad (16)$$

where α is called a Lagrange multiplier, N_{sv} is the number of SVs for which both $c_i(wx_i' + \mathrm{b}) \geq 1$ and $\alpha_\mathrm{i} > 0$.

In the context of SVMs, the value of φ has to be minimized with respect to w and b and maximized with respect to the Lagrange multipliers α_i. The minimum of φ with respect to w and b is given by:

$$\frac{\partial \varphi}{\partial b} = 0 \Rightarrow \sum_{i \in N_{sv}} \alpha_i c_i = 0 \quad (17)$$

and

$$\frac{\partial \varphi}{\partial w} = 0 \Rightarrow w - \sum_{i \in N_{sv}} \alpha_i c_i x_i = 0 \quad (18)$$

where the samples i are formally SV. Hence:

$$\varphi(\alpha) = \frac{1}{2} \sum_{i \in N_{sv}} \sum_{l \in N_{sv}} \alpha_i c_i (x_i x_l') c_l \alpha_l - \sum_{i \in N_{sv}}^{l} \alpha_i \quad (19)$$

The optimization task is that of minimizing $\varphi(\alpha)$with respect to α, satisfying the constraints:

$$\alpha_i \geq 0 \; and \; \sum_{i \in N_{sv}} \alpha_i c_i = 0 \quad (20)$$

Finally, the optimal $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{N_{sv}})$ allows determination of the weight vector w of the optimal separating hyperplane:

$$w = \sum_{i \in N_{sv}} \alpha_i c_i x_i \quad (21)$$

while the offset b can be calculated from any pair of samples of opposite classes satisfying the conditions that their values of α are greater than 0.

It can be seen that SVM is rather too complex for the case with low dimension and the classes are nearly or completely linearly separable, for which other methods could satisfy. SVM shows its appeal when the classes are not linearly separable. SVM adds an extra step into the procedure described above in which the samples are projected on a new higher dimensional space by means of a feature function $\phi(x)$. The back projection of the optimal separating boundary (in the form of a hyperplane) from this new feature space to the original variable space will then result in a non-linear boundary of given complexity that better suits the distribution in the original variable space.

The boundary is found by reformulating the optimization task (Equation 19):

$$\varphi(\alpha) = \frac{1}{2}\sum_{i\in N_{sv}}\sum_{l\in N_{sv}}\alpha_i c_i \langle\phi(x_i),\phi(x_l)\rangle c_l\alpha_l - \sum_{i\in N_{sv}}^{l}\alpha_i \qquad (22)$$

where $\langle\phi(x_i),\phi(x_l)\rangle$ is the scalar product of the respective feature functions.

An important concept in SVMs is that there exist kernel functions K in the original variable space that corresponds to the dot product of functions in the new feature space:

$$K(x_i, x_l) = \langle\phi(x_i),\phi(x_l)\rangle \qquad (23)$$

By using kernel function $K(x_i, x_l)$ rather than feature function $\phi(x)$, it is possible to solve the optimization task without creating the feature space but working only in the original data space. This is known as the "kernel trick" and makes SVM effective in solving complex tasks. Some of the most common kernels are: Linear function, Radial basis function (RBF), Polynomial

function (PF), and Sigmoidal function (SF). In practice, which type of kernel function to used is controlled by the hyperparameter *kernel*.

Because the kernel trick allows SVM to define complex boundaries and it is possible to define almost any boundary around training set samples even if there is no particular significance, there is a risk of overfitting by creating a perfect but over-complicated boundary with no real predictive power. Therefore, in practice, the design of SVM involves the setting of a value call *C*. High *C* values means that SVM is set to hard margin in which the optimal boundary must be found that exactly separates the classes with the maximum possible margin between classes. Low *C* values means that SVM is set to soft margin in which a degree of misclassification is tolerated so that the classification error against the complexity of the model is balanced. In addition to the hyperparameter *C*, there is a hyperparameter called *gamma* which defines how far the influence of a single training point reaches. A high value of *gamma* gives low variance and high bias and vice versa for low *gamma*. The different combinations of values of *C* and *gamma* also have different influence on the performance of SVM. Therefore, in addition to select the suitable kernel function, it is important to find and select appropriate values of *C* and *gamma* to achieve the optimal performance of SVM. More information on the tuning of these hyperparameters will be described in section 3.4.4.

2.2.5.2 For Regression

Support Vector Regression (SVR) [60, 61] is an extension of SVM for regression. For simple linear regression, the aim is to form a model between x and y:

$$\hat{y} = b + wx \tag{24}$$

where $\hat{y}$ (glucose concentration) is predicted from x (spectral intensity).

The linear model between two variables is similar to the boundary between two groups in SVM; however, the aim here is to enclose all samples within the margin. For that, a new hyperparameter ε, which is the error tolerance, is defined. In order to enclose all samples between the margins, there will be a maximum value and a minimum value of ε. For different value of ε, there will be several different possible straight lines. Samples on the margin are SVs that define the margin, and those outside are analoguous to bounded SVs. Samples between the margins are not SVs. The number of SVs will limit the number of possible margins. The task now is to minimize:

$$\varphi(w, b, \xi) = \frac{1}{2}(ww') + C\sum_{i \in N_{sv}} \xi_i \tag{25}$$

Where ξ_i is the slack variable for sample i which defines the distance from sample to the margin; C is a penalty error, which is similar to C in SVM, that determines how important it is to ensure all variables are within the margin.

Similar to SVM, when the relationship is no longer linear, it is necessary to use a kernel function. There is also a hyperparameter *gamma* that needs to be tuned. It is important to find the combination of ε, C and *gamma* that gives the lowest prediction error. More information regarding the tuning and optimization process can be found in section 3.4.4.

Chapter 3 Hyperparameter Tuning & Model Validation

As mentioned, chapter 2 provides information regarding the important hyperparameters of each utilized technique. These hyperparameters need to be selected carefully as different values of each hyperparameter would lead to different performance of the model. Furthermore, the combination of these hyperparameters also has significant effects on the end results. Chapter 3 will focus on strategies to tune these hyperparameters and find the best combination for optimized results. Chapter 3 also covers the techniques and metrics used to assess the effectiveness of each combination of values of the hyperparameters as well as the final prediction ability of the model. In addition, chapter 3 will discuss further the hyperparameters of each technique; the possible values for each hyperparameter.

3.1 Evaluation Metrics

In order to decide which value of a hyperparameter would lead to an enhanced model, we need to a scoring metric to use as a basis for comparison. For classification approach, the following metrics were used:

$$Accuracy = \frac{True\ Positive+True\ Negative}{True\ Positive+False\ Positive+\ True\ Negative+False\ Negative} \quad (26)$$

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive} \quad (27)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative} \quad (28)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (29)$$

Accuracy is an evaluation metric that allows us to measure the total number of predictions a model gets right. However, a high accuracy by itself does not guarantee a good model, especially with unbalanced dataset accuracy. For example, we want to predict whether 30 patients have diabetes or not by using a noninvasive glucose monitoring meter. If most of the patients actually do not have diabetes and the model predicts negative every time, then we will achieve high accuracy. However, in real practice, where the numbers of patient with and without diabetes are the same, then the model in this example will not be able to recognize patients with the disease. In other words, the high accuracy alone in this case is not reliable. Therefore, there is a need for other metrics.

Precision evaluates how precise a model is in predicting positive labels. In other words, precision shows out of those predicted positive, how many of them are actually positive. Precision is a good evaluation metric to use when the cost of a false positive is high.

Recall calculates the percentage of actual positive labels a model correctly identified. Recall is recommended when the cost of false negative is high.

In this study, since the utilized datasets are balanced and the training and testing sets are split so that every set should include data points from all the classes, therefore, it is safe to use accuracy as a metric for evaluating the regression process of proposed model. However, when evaluating the final prediction ability, other metrics must be used together with accuracy. Since both the costs of false positive and false negative are high, both precision and recall should be examined. Considering the previous example, there would be bad consequences if a patient was predicted with diabetes when they are healthy or if a patient was predicted healthy when they actually have diabetes. F1 score is a metric to used when a balance between precision and recall are needed. Therefore, in this study, F1 score would be used as scoring metrics together with

accuracy for evaluating the performance of proposed models as well as tuning and selecting the hyperparameters when following the classification approach.

For regression approach, we will follow the literature and use the following metrics:

- Standard error of cross-validation (SECV)
- Standard error of prediction (SEP)

Both metrics are calculated as the Root mean square error. SECV is based on an iterative algorithm that selects samples from a sample set population to develop the regression model and then predicts on the remaining unselected samples. It can be said that SECV is an estimate of the SEP. SECV will be computed for several different combination of hyperparameter values, the one with the lowest SECV will be selected as the best regression model. SEP allows for comparison between predicted values and the actual or web laboratory values.

3.2 Validation Techniques

In the previous section, we have decided on the scoring metrics that can be used to judge the performance of our model. However, we still need to have a strategy that produces and uses the metrics to decide the best hyperparameter values for the models. There are several options that can be chosen.

The first option is to randomly split the complete dataset intro training and testing sets. It is advised against testing and training on the same dataset since there is a risk of overfitting since the model has already memorized the dataset. Ideally, a model should be tested using an entirely different dataset. However, in real life, accessing to additional dataset can be difficult due to limited resources. This problem is solved by splitting the original dataset into subsets called

training set and testing set. This is commonly done in a ratio of 80/20 for training and testing respectively. Nevertheless, because a testing set is used several times for evaluating different settings of the hyperparameters, the risk of overfitting on the test set still remains.

The second option is to split a whole dataset into three subsets: training, validation, and testing. Training proceeds on the training set, after which evaluation is done on the validation set, and when the experiment seems to be successful, final evaluation can be done on the test set. However, by partitioning the available data into three sets, we drastically reduce the number of samples which can be used for learning the model, and the results can depend on a particular random choice for the pair of (train, validation) sets. In other words, there is not enough input information or information is incomplete for the model to learn to come up with a complete answer.

The third option is to use a technique called Cross-validation (CV). Firstly, the whole dataset is split into training and testing sets using technique of option 1. Then, the training set is split into k folds (smaller set). The following procedure is followed for each of the k folds:

- Train the model using $k-1$ folds (e.g. $k = 5$);
- Validate the model using the remaining kth fold.
- Repeat until every k fold serves as the test set.

The average scoring metrics will be the performance metric for the model. The setting of hyperparameters that produces the best scoring metric will be selected.

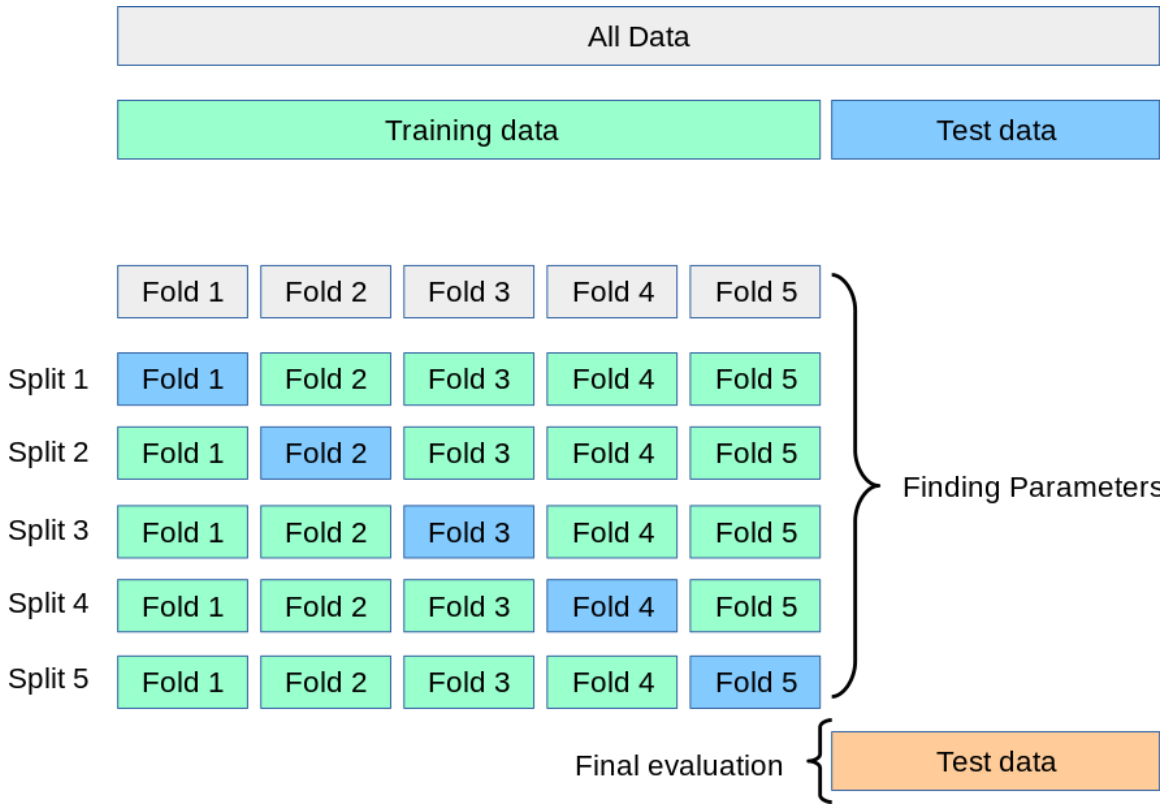


Figure 3-1. Illustration of cross-validation technique. [50]

This study utilized the third option which included the first option. However, because of a high number of hyperparameter values that needed to be tested, there is still a chance of error or instability due to a particular partitioning. Therefore, an extra technique called nested cross-validation (Nested CV) was implemented to estimate the generalization error of the developing models and their stability. Nested CV technique is illustrated in Figure 3-2. The technique consists of two loops: inner loop CV and outer loop CV. The inner loop is just a basic CV technique, as described in option three, which is responsible for model selection/hyperparameter tuning. The outer loop is another CV technique responsible for error estimation. For example, a dataset is split into training and testing set. Then:

- The training set is divided into 5 folds.

- 1 fold is hold as outer validation set for outer loop; the remaining 4 folds are used in the inner loop.
    - 3 folds of the 4 inner folds are used as an inner training set for a model with a specific hyperparameter setting
        - Remaining 1 fold of the 4 inner folds is used as an inner validation set to validate the performance of the model
            - Repeat until every inner fold serves as the inner validation set
            - After all combinations of models are validated, the best-performing model is selected.
                - Train the model again using all 4 inner folds.
- Validate the selected model from the inner loop using the reserve outer fold.
- Repeat the whole process again until every fold serves as outer validation set.

Note that nested CV does not make the hyperparameter optimization any more successful but to provide an honest estimate of the performance that can be achieved with that particular optimization strategy. In other words, nested CV helps to determine when a model's performance happens to be overoptimistic but does not improve this performance.

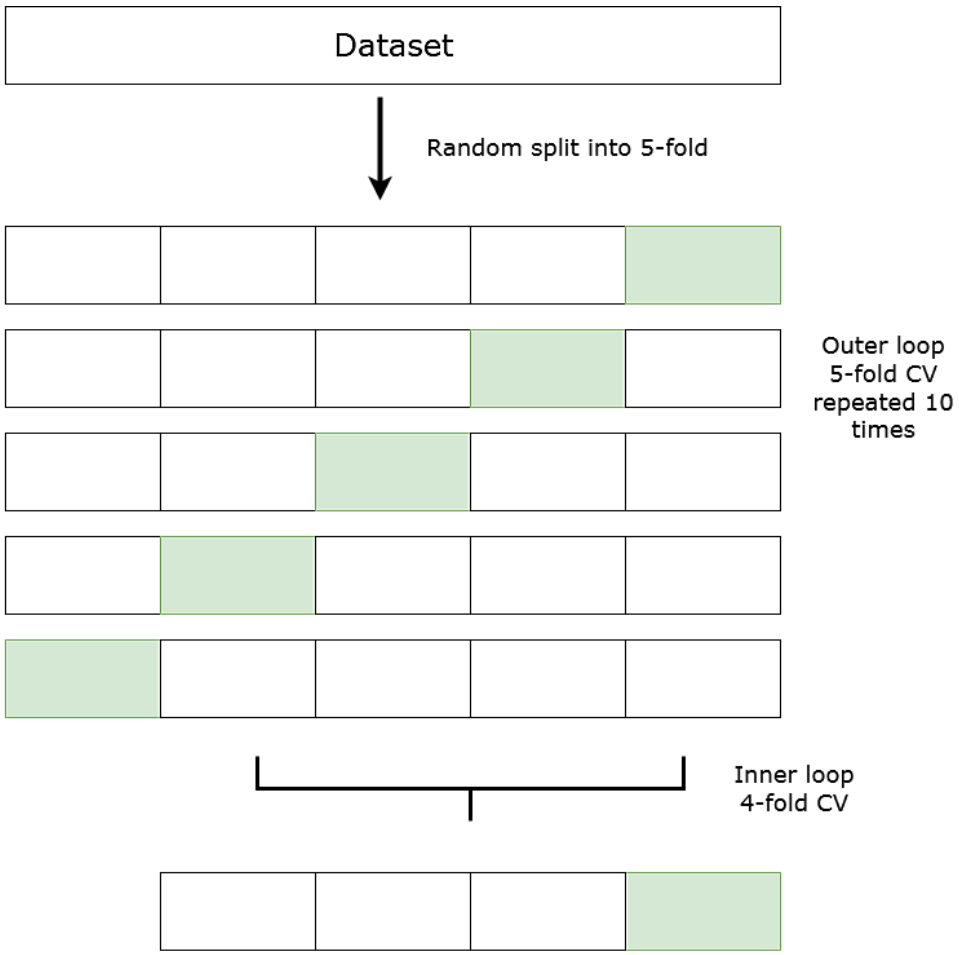


Figure 3-2. Illustration of nested cross-validation technique.

Another important issue to consider is that when dividing training set into different folds, we need to ensure that the same group is not represented in both testing and training sets. Because each glucose sample was measured several times, there is a risk that the model would learn from highly sample specific features and could fail to generalize to new subjects.

In this study, both the inner loop and outer loops of the nested CV were set to 5-fold for C0 and 10-fold for C1.

3.3 Brute Force Hyperparameter Search

After obtaining the metrics and techniques to validate each setting or combination of hyperparameters to build the optimized model, we need a tool to pass values of the hyperparameters to the validation technique. This study utilized a function from scikit-learn library called *GridSearchCV* that used brute force to exhaustively examine all hyperparameter values combination.

Normally, since not only SVM but other techniques have several hyperparameters that need to be tuned, the searching process needs to be repeated many times. This is not only inconvenient but also cause overfitting due to the fact that the dataset has to be fit, or read, repeatedly. Information regarding the dataset might leak from the previous steps. Therefore, as a solution, a technique called *pipeline* was deployed. The purpose of the pipeline is to assemble several steps that can be cross-validated together with different hyperparameter values. Another advantage of pipeline technique is that it allows a more organized coding structure.

In summary, the *GridSearchCV* technique would consists of the following elements:

- An estimator: a pipeline contains all other techniques, including the a classifier such as SVM or KNN
- An parameter space: a grid, or set, contains every parameter values that needed to be tested
- A cross-validation scheme: a number of k fold
- A scoring function: a function that decide the metric to evaluate the performance of each value or a combination of values of different hyperparameters

3.4 Hyperparameters Settings

3.4.1 Filter methods

As mentioned in section 2.2.4.1.1, the focus of filter methods is on the two techniques MI and SKB. *MI* technique has two function variations that are *mutual_info_classif and mutual_info_regression.* The first variation was used when developing models following the classification approach while the second variation was used when developing models following the regression approach. In other words, when glucose concentration is considered to be discrete, the first variation is used; when glucose concentration is considered to be continuous, the second variation is used. Each *MI* function variation has *MI* two important hyperparameters: *discrete_features* and n*_neighbor*. The *discrete_features* was set to False when developing models that considered spectral wavelengths to be continuous and set to True when developing models that considered spectral wavelengths to be discrete. This setting of *discrete_features* is important because treating a continuous variable as discrete and vice versa will usually give incorrect results. Since spectral wavelength can be appropriate for both cases, it is necessary to investigate the effect of both scenarios. The hyperparameter n*_neighbor* decides the number of neighbors to use for *MI* estimation. The higher values reduce would reduce variance of the estimation but could introduce a bias. Therefore, the *GridSearchCV* technique was used to test the performance of *MI* technique with increasing values of n*_neighbor* from 3, 7, 15, 21, 27, and 31.

SKB technique uses the output of *MI* technique to decide which features have the most relevant information to keep and remove the remaining. The number of features to keep is controlled by the hyperparameter *k.* Since *MI* between two random variables is a non-

negative value that measures the dependency between the variable and higher values mean higher dependency, in order to find a number *k*, we would cut off any features that has:

- Option 1: *MI* < 1
- Option 2: MI < 0.8

The survived features would be then counted and passed as a value of the hyperparameter *k*. The summary of values for the hyperparameters *discrete_features*, n*_neighbor*, *k* is shown in Table 3-1.

3.4.2 Wrapper Methods

SFFS is a kind of greedy search algorithms that reduces an initial d-dimensional feature space to a k-dimensional feature subspace where k < d and chooses a subset of feature that is most relevant. The pseudo code for SFFS is outlined as follow [62, 63]:

Input: the set of all features, $Y = \{y_1, y_2, \dots, y_d\}$

- The *SFFS* algorithm takes the whole feature set as input, if our feature space consists of, e.g. 100 then dimensions (*d = 100*).

Output: a subset of features, $X_k = \{x_j | j = 1,2, \dots, k;\ x_j \in Y\}$, where $k = (0,1,2, \dots, d)$

- The returned output of the algorithm is a subset of the feature space of a specified size. For instance, a subset of 55 features from a 100-dimensional feature space (k = 55, d = 100).

Initialization: $X_0 = \emptyset, k = 0$

- We initialize the algorithm with an empty set ("null set") so that the $k = 0$ (where k is the size of the subset)

Step 1 (Inclusion):

$x^+ = \text{argmax}\, J(x_k + x)$, where $x \in Y - X_k$

$X_{k+1} = X_k + x^+$

$k = k + 1$

Go to Step 2

Step 2 (Conditional Exclusion):

$x^- = \text{argmax}\, J(x_k - x)$, where $x \in X_k$

if $J(x_k - x) > J(x_k - x)$:

$X_{k-1} = X_k - x^-$

$k = k - 1$

Go to Step 1

- In step 1, we include the feature from the feature space that leads to the best performance increase for our feature subset (assessed by the criterion function). Then, we go over to step 2
- In step 2, we only remove a feature if the resulting subset would gain an increase in performance. If $k = 2$ or an improvement cannot be made (i.e., such feature x^+ cannot be found), go back to step 1; else, repeat this step.
- Steps 1 and 2 are repeated until the Termination criterion is reached.

Termination:* stop when k*** equals the number of desired features

In real practice, the numbers of features to be selected are determined by the hyperparameter *k_features*. We can pass a tuple which contains two numbers to *k_features* so that SFFS will

return any feature combination between the two numbers that scored highest in cross-validation using the scoring function KNN. In this study, *k_features* was set to $(1, k)$, where k is:

- Option 1: 158 (the total number of features) when only wrapper method is utilized
- Option 2: the output number of features of *SKB* of filter method.

In addition, since SFFS use KNN as a scoring function, the hyperparameter n_*neighbor* of KNN also needs to be tuned using *GridSearchCV* technique. Similar to the hyperparameter n_*neighbor* of *MI* techniques, the higher values reduce would reduce variance of the estimation but could introduce a bias. Therefore, a set of values from 3 to 30 were tested.

The summary of values for the hyperparameters *k_features*, n_*neighbor* is shown in Table 3-1.

3.4.3 PCA

As mentioned in section 2.2.4.2, PCA reduces the dimensional space by extracting the essential information known as principal components (eigenvectors), sorting them in decreasing order, then keeping the desire percentage while eliminating the rest. To choose the amount of information to be retained, the hyperparameter n_*components* needs to be set to a specific value, e.g. n_*components* = 0.90 means that 90% of the information will be retained. However, in real practice, it would be difficult to determine a sufficient percentage of information to keep; therefore, a set of values, i.e. 0.95, 0.93. 0.90, 0.88, were passed to the search technique for trial.

3.4.4 Support Vector Machine

As mentioned in section 2.2.5, both SVM and SVR need to tune the following hyperparameters: kernel, *C, gamma,* SVR, in addition, have an extra hyperparameter *epsilon* to be tuned.

The hyperparameter *kernel* determines the kernel function to be used. For this study, two kernel functions were tested:

- *linear*: $\langle x, x' \rangle$.
- *rbf*: $\exp(-\gamma \|x - x'\|^2)$. Where $\gamma > 0$, and γ is specified by keyword *gamma.*

The hyperparameter *C* determines the regularization or hard and soft margin for SVM, the penalty error for SVR. The values of *C* to be tested were: 0.1, 1, 10, 100, and 1000.

The hyperparameter *gamma* defines how far the influence of a single training point reaches. The values of *gamma* to be tested were: 0.0001, 0.0005, 0.001, 0.01, 0.1, and 1.

The hyperparameter *epsilon* defines the maximum error within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value. The values of *epsilon* to be tested were: 0.01, 0.1, 0.5, 1, 2, and 5.

Table 3-1. Summary of all settings of hyperparameter

Technique	Hyperparameter	Value
Nested CV	inner_cv_C0 outer_cv_C0 inner_cv_C1 outer_cv_C1	5 5 10 10
MI	discrete_features n_neighbor	True, False 3, 7, 15, 21, 27, 31
SKB	k	number of features with MI>1, number of features with MI >0.8
SFFS	KNN_n_neighbor k_features	range(3, 30) (1, SKB_k)
PCA	n_components	0.95, 0.93. 0.90, 0.88
SVM	kernel C gamma	‘linear’, ‘RBF’ 0.1, 1, 10, 100, 1000 0.0001, 0.0005, 0.001, 0.01, 0.1, and 1
SVR	kernel C gamma epsilon	‘linear’, ‘RBF’ 0.1, 1, 10, 100, 1000 0.0001, 0.0005, 0.001, 0.01, 0.1, and 1 0.01, 0.1, 0.5, 1, 2, and 5

Chapter 4 Experimental Design

This study covers the designs of experiments based on the general system model to evaluate the effectiveness each proposed step and techniques. The first section describes the overview of the experimental designs. The remaining sections describes in detail each of the experiments and their purposes.

4.1 Overview of the Experimental Design

There are in total four experiments labeled as: model 1, model 2, model 3, and model 4. Models from 1 to 3 are conducted on both datasets C0 and C1 while model 4 is conducted only on dataset C1. The models are designed so that their complexity increases from 1 to 4. The purpose of the design is to evaluate the effectiveness of each added step or combination of techniques on the calibration and the final performance. Models 3 and 4 are compared directly to each other to evaluate the efficacy of dimensional reduction techniques in finding the most informative wavelengths for noninvasive glucose monitoring. For each model, both the classification and regression approaches were investigated by applying SVM and SVR respectively.

Table 4-1. Summary of the two datasets

Datasets	Number of Features	Number of Classes	Number of Observations
C0	158	10	100
C1	158	10	1000

4.2 Model 1

Model 1 has a rather simple structure which includes just four main techniques: *StandardScaler, train_test_split, PCA, SVM/SVR*. In this model, pipeline technique, search technique, and CV or nested CV were not applied. The simple technique *train_test_split* was used in place of CV technique too as a validation technique helps to prevent information leakage. The ratio for training and testing was 6/4 in [27]; however, it was not standard and preserving too many observations for testing could lead to inefficient training process. Therefore, in addition to 6/4 ratio, an 8/2 ratio was also tested. Because there no search technique was applied, all of the hyperparameters were set to default, which is the same as in [27]. The whole process was repeated 30 times to verify the model's stability. Table 4-2 summaries all values of the hyperparameters of model 1. Figure 4.1 illustrates the workflow of model 1.

The purpose of model 1 is two folds:

1. To verify the two datasets C0 and C1 by comparing the result to reference [27];

2. To serve as a reference to check the effectiveness of proposed methods, such as *pipeline, GridSearchCV, MI, SKB, SFFS,* etc. in other models.

Model 1 has many limitations:

- It only uses the simplest technique to split dataset for training and evaluation. The simplicity nature of this technique can still result in information leakage and biased model.
- It does not consider whether data points are linearly or nonlinearly separable. The parameter-sensitive *SVM* technique is not tuned neither.

- It uses the whole datasets which might contain irrelevant features or noises, which reduce the performance of the classifier.

Table 4-2. Summary of all settings of hyperparameter for model 1

Technique	**Parameters**	**Values**
Train_test_split	test_size, train_size	Option 1: 0.6 and 0.4 Option 2: 0.8 and 0.2
PCA	n_components	0.95
SVM	kernel C gamma	‘linear’ 1 ‘scale’
SVR	kernel C gamma epsilon	‘linear’ 1 ‘scale’ 0.1

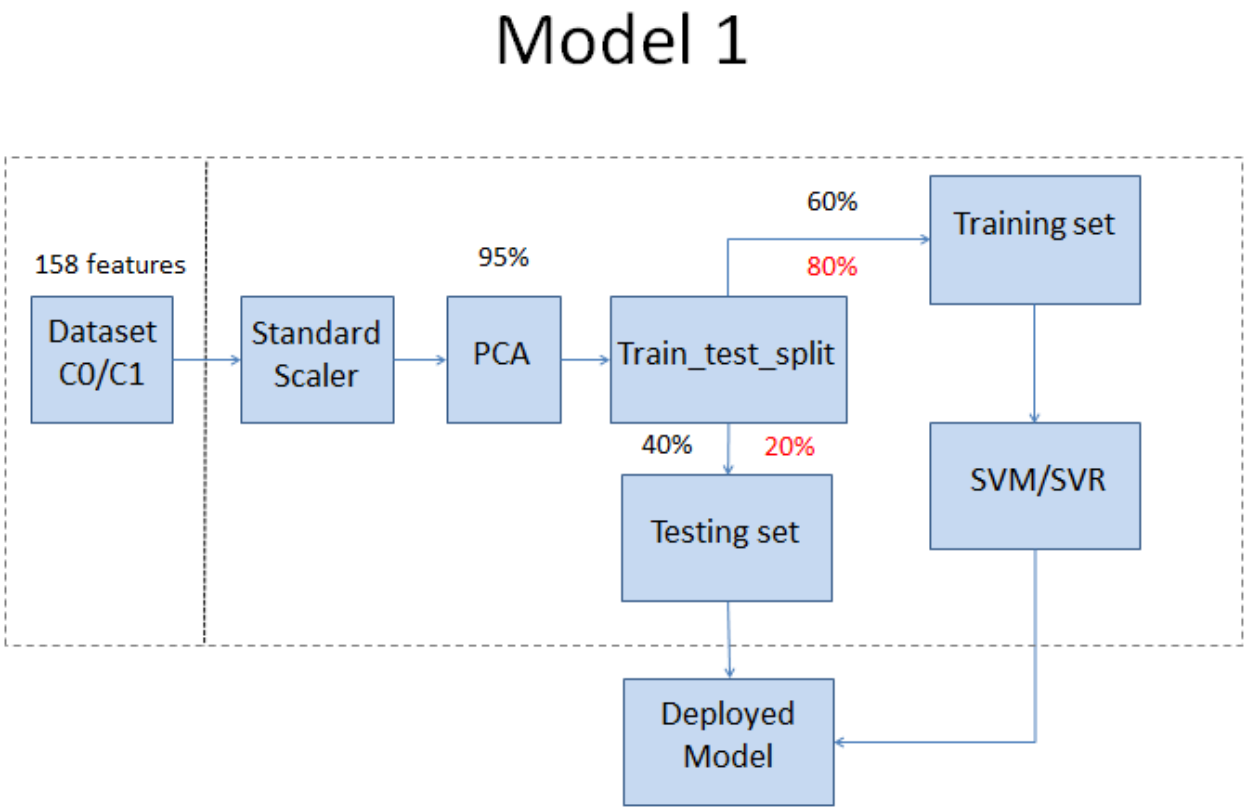


Figure 4-1. Block diagram of Model 1.

4.3 Model 2

Model 2 is an upgrade of model 1 in which pipeline and search techniques are applied. The datasets were also split into a training set and a testing set with ratio 8/2 respectively. In addition to *train_test_split* technique, *cross-validation* was also utilized. The hyperparameters of *PCA* and *SVM/SVR* were tuned using values in Table 3-1. For dataset C0, due to small number of observations, 5 fold *CV* was used. For dataset C1, 10 fold *CV* was used. Techniques of feature selection method were not utilized in this model. Figure 4.2 illustrates the workflow of model 2. The purpose of model 2 is to evaluate the effectiveness of pipeline and searching techniques as well as the cross-validation technique.

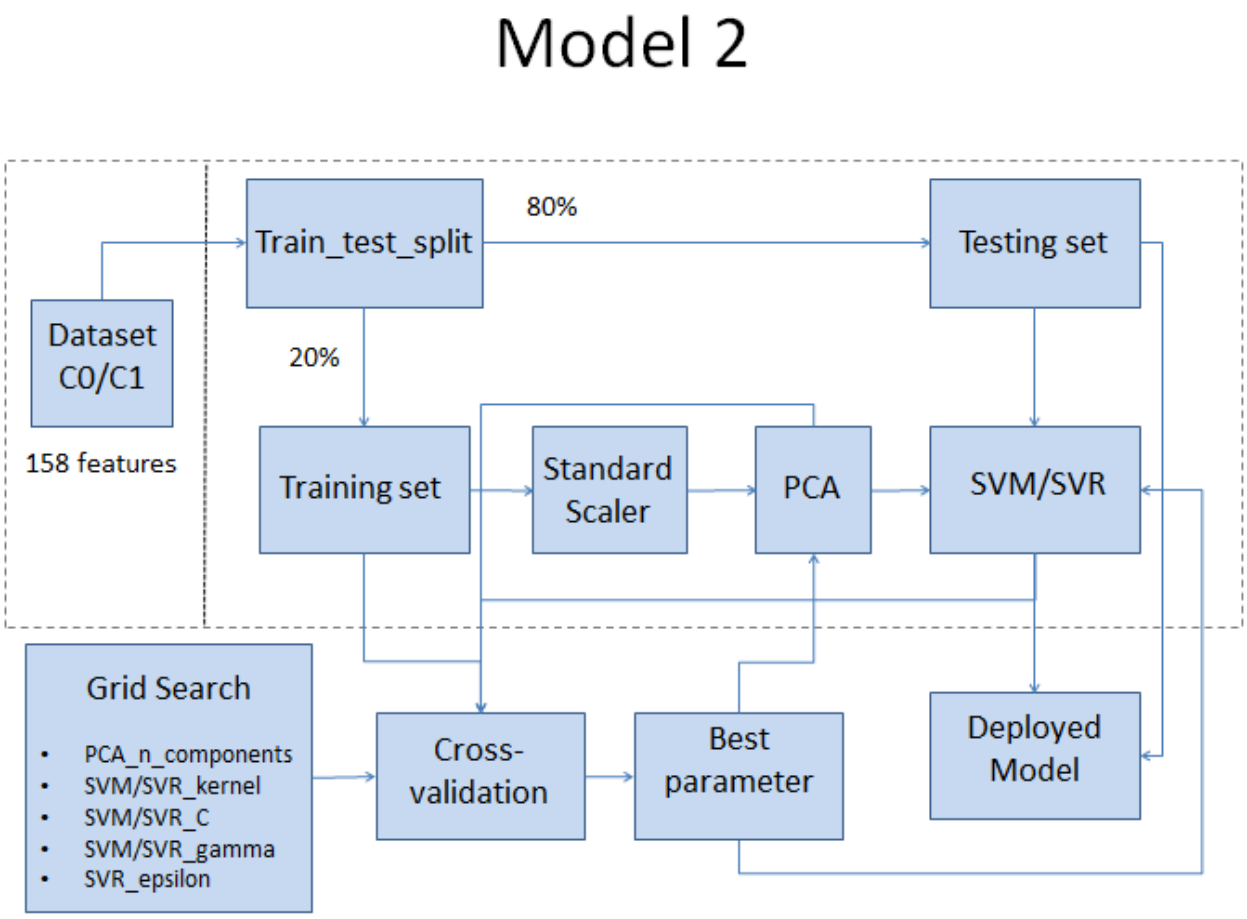


Figure 4-2. Block diagram of model 2

4.4 Model 3

In model 3, feature selection method was applied in addition to other methods in model 2. However, only *SFFS* technique of wrapper method was utilized. Additionally, model 3 utilized *nested cross-validation* technique to validate the hyperparameters. The hyperparameter tuning process used values summarized in Table 3-1. A tuple of (1, 158) was passed into the *k_features* parameter which means that the technique would try to find the best combination of features with the number of features in the range of 1 and 158. The smallest subset consists of 1 feature and the largest subset consists of 158 features. The output of the *SFFS* is the reduced number of

wavelengths. The whole procedure is repeated 30 times. The final evaluation score is averaged. Model 3 is illustrated by Figure 4.3.

The purpose of model 3 is to evaluate the infectiveness of the *SFFS* technique of wrapper method of feature selection in term of performance, speed as well as the ability to automatically select features (wavelengths).

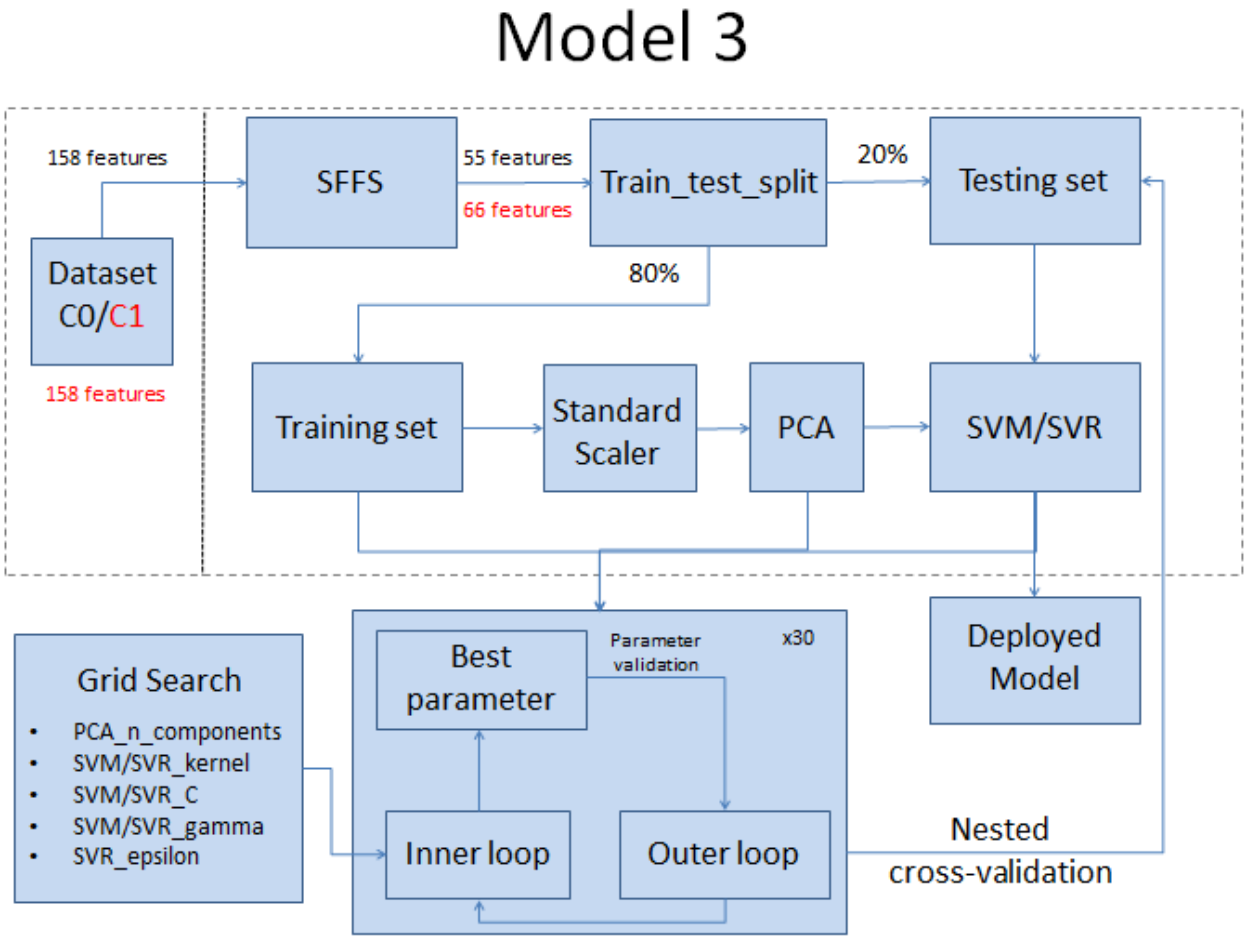


Figure 4-3. Block diagram of model 3

4.5 Model 4

Model 4 is the full model that consists of all proposed techniques: *StandardScaler, MI, SKB, SFFS, PCA, SVM, pipeline, Grid Search,* and *nested cross-validation.* The *MI* and *SKB* techniques of filter method of feature selection were added before *SFFS* in attempt to increase the speed of *SFFS* and the whole process. The hyperparameter tuning process used values summarized in Table 3-1. A tuple of (1, k) was passed into the *k_features* hyperparameter with k is the number of survived features after applying *MI* and *SKB*.

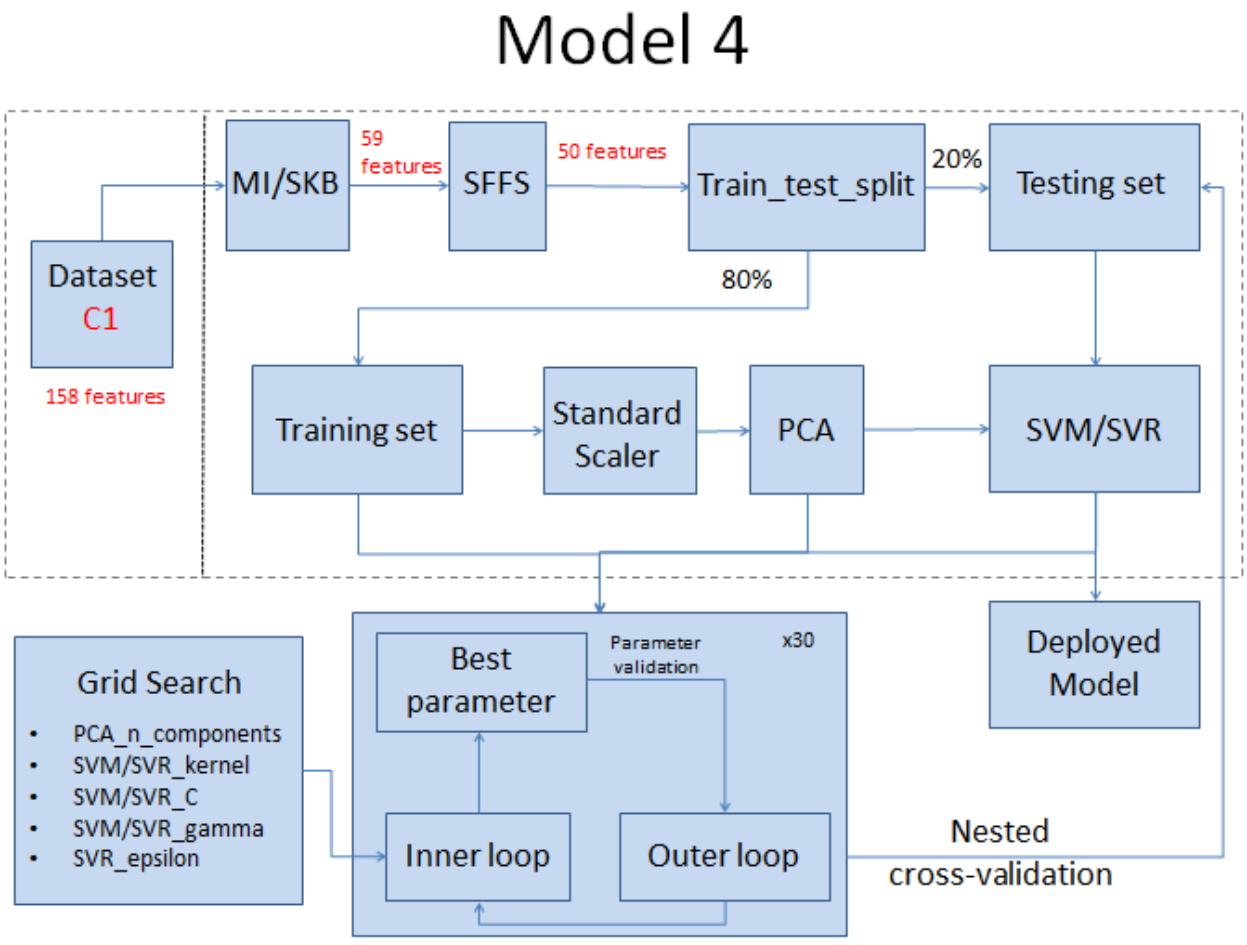


Figure 4-4. Block diagram of model 4

Chapter 5 Results and Discussion

All models and their related experiments in chapter 4 were designed and conducted with the aim to complete each and every objective stated in section 1.4. Therefore, in chapter 5, results will be presented so that they can be explicitly connected to each objective and key conclusions can be made. Please note that objective 1 was achieved by following the development process in section 2.2.2 and its results will be assessed later after analyzing the outcomes of all other objectives. Similarly, objective 2 can only be achieved and assessed after completing all other objectives. Therefore, this chapter will start by presenting results opted for objective 3.

5.1 Results of Experiments for Objective 3

In order to complete objective 3 which is the investigation on the effectiveness of SVM techniques, model 1 was analyzed using both classification and regression approaches.

5.1.1 Results of Classification Approach for Model 1

Table 5-1 shows the testing accuracy and f1 score of model 1 using two datasets C0 and C1. The values in this table were obtained using classification approach. For each of the dataset, the training and testing sets were split using two ratios 6/4 and 8/2, i.e. the majority of the data were used for training.

Table 5-1. Evaluation results of model 1 using classification approach

Dataset	Model	Note	Average non-nested CV Accuracy (%)	Average nested CV Accuracy (%)	Testing Accuracy (%)	Testing F1_score (%)
C0	1	train_test_split=6/4	N/A	N/A	71.41	71.23
	1	train_test_split=8/2	N/A	N/A	78.67	78.01
C1	1	train_test_split=6/4	N/A	N/A	84.71	84.77
	1	train_test_split=8/2	N/A	N/A	85.17	85.22

5.1.2 Results of Regression Approach for Model 1

Table 5-1 shows the testing accuracy and f1 score of model 1 using two datasets C0 and C1. The values in this table were obtained using classification approach. For each of the dataset, the training and testing sets were split using two ratios 6/4 and 8/2, i.e. the majority of the data were used for training.

Table 5-2. Evaluation results of model 1 using regression approach

Dataset	Model	Note	Non-nested SECV (mg/dl)	Nested SECV (mg/dl)	SEP (mg/dl)
C0	1	train_test_split=6/4	N/A	N/A	94.12
	1	train_test_split=8/2	N/A	N/A	93.03
C1	1	train_test_split=6/4	N/A	N/A	79.18
	1	train_test_split=8/2	N/A	N/A	78.68

5.1.3 Discussion on Objective 3 and Model 1

As stated in section 4.2, model 1 is a very simple model consists of only SVM technique and other necessary initial processing steps. Therefore, the effectiveness of model 1 can be used to assess the effectiveness of SVM technique. By comparing the values of testing accuracy, testing f1 score in Table 5-1 and SEP in Table 5-2 with values of the same metrics in Table 1-2, it can be seen that SVM technique produced worst results than PLSR technique in most of other papers. This is expected because, as stated in section 2.1, SVM often requires prior variable reduction step(s) and hyperparameters tuning process to be effective.

The fact that increasing the number of observations while keeping the same number of feature increased the accuracy, f1 score, and decrease SEP further implies that adding dimensionality reduction steps may play a crucial role in enhancing the performance of the system. In addition, simply changing the ratio of training and testing sets also increased the scores slightly. This can be inferred that more advanced techniques to control the use of information would be highly useful for improving the system too.

It can be concluded by now that using SVM technique alone would not produce adequate results. The results were the same for both classification and regression approaches. This might indicate that there is a consistency in both approaches. However, there are hints that adding extra steps or techniques into the system might lead to better results. This will be confirmed by completing other objectives and reviewing their results in later sections.

5.2 Results of Experiments for Objective 4

Objective 4 is to investigate the effectiveness and limit of the proposed techniques for dimensionality reduction to improve performance and extract informative wavelengths which was conducted by analyzing model 3 and 4. Firstly, the ability of reducing dimensionality or extracting informative wavelengths of each wrapper methods, filter methods, or the combination of both will be presented and reviewed. Note that each method type was conducted using both classification and regression approaches. Next, the actual end-performance of each model will be assessed and discussed.

5.2.1 Results of the Dimensionality Reduction Methods

5.2.1.1 Wrapper Methods

Table 5-3 and Table 5-4 show the top 3 optimal subsets together with their number of features and validation score as outputs of SFFS technique for each of the dataset. For classification approach (shown in Table 5-3), the SFFS technique used accuracy as the scoring metric. For regression approach (shown in Table 5-4), the SFFS technique used root mean square error (RMSE) as the scoring metric.

Table 5-3. The top 3 optimal subsets as output of SFFS technique for both datasets C0 and C1 for classification approach

Dataset	Feature subset ranking	Number of features	Selected subset average CV accuracy (%)
C0	Top 1	55	93
	Top 2	99	93
	Top 3	56	92
C1	Top 1	66	90.1
	Top 2	54	90
	Top 3	65	90

Table 5-4. The top 3 optimal subsets as output of SFFS technique for both datasets C0 and C1 for regression approach

Dataset	Feature subset ranking	Number of features	Selected subset average CV RMSE (mg/dl)
C0	Top 1	10	94.97
	Top 2	15	95.87
	Top 3	21	95.96
C1	Top 1	20	105.63
	Top 2	11	105.67
	Top 3	12	105.92

5.2.1.2 Filter Methods

For filter methods, the independent variable (wavelengths or features) was considered to be either discrete or continuous; the target variable (glucose concentration) was also considered to be either different classes (classification) or continuous values (regression).

Filter methods from both Table 5-5 and Table 5-6 followed classification approach. Table 5-5 and Table 5-6 shows the performance assessments of each subset created from dataset C1 by filter methods when the hyperparameter *n_neighbors* set to 3, 7, 15, 21, 27, and 31. In Table 5-5, the features (wavelengths) were considered to be continuous (discrete_features=False) and setting the threshold to be MI>1. In Table 5-6, the features (wavelengths) were considered to be discrete (discrete_features=True). When the features are considered discrete, the *MI* of each feature is very high (>2) and is similar to each other. They can only be divided into 3 groups as shown in Table 5-6. The value k in these cases represents the reduce number of features.

Filter methods from Table 5-7 followed regression approach. Table 5-7 also shows the performance assessments of each subset created from dataset C1 when the hyperparameter *n_neighbors* set to 3, 7, 15, 21, 27, and 31 with the threshold *MI*>1. Note that the filter algorithms raise errors when considering the target variable (glucose concentration) to be continuous but the independent variable (wavelengths or features) to be discrete. Therefore, only experiments those consider both the target and independent variables to be continuous were conducted and reported. The value k in these cases represents the reduce number of features.

Table 5-5. Evaluation results of filter methods (Classification, discrete_features=False, MI>1)

n_neighbor	**k**	**Average non-nested CV accuracy (%)**	**Average nested CV accuracy (%)**	**Testing Accuracy (%)**	**Testing F1_Score (%)**
3	49	88	87.25	87.5	87.44
7	52	88.62	88.12	87.5	87.49
15	52	89	88.37	88.5	88.47
21	54	88.38	87.38	86.5	86.6
27	55	88.44	87.75	89	89.05
31	55	88.44	87.75	89	89.05

Table 5-6. Evaluation results of filter methods (Classification, discrete_features=True)

MI	k	Average non-nested CV Accuracy (%)	Average nested CV Accuracy (%)	Testing Accuracy (%)	Testing F1_Score (%)
>2	158	88.13	87.56	85.5	85.4
>2.301	133	87.81	87.31	86.5	86.63
>2.302	87	88.25	87.87	89.5	89.47

Table 5-7. Evaluation results of filter methods (Regression, discrete Feature=False, MI>1)

n_neighbor	k	Average non-nested SECV (mg/dl)	Average nested SECV (mg/dl)	SEP (mg/dl)
3	49	61.37	61.57	55.21
7	50	61.55	61.97	56.34
15	47	62.73	63.02	57.27
21	45	62.92	62.99	56.94
27	42	65.60	65.77	60.67
31	39	69.65	69.66	66.27

5.2.1.3 Combination of filter and wrapper methods

Table 5-8 and Table 5-9 show the top 3 optimal subsets together with their number of features and validation score as outputs of the combination of all proposed wrapper and filter methods for dataset C1. Because these experiments applied the combination of all proposed dimensionality reduction methods, the features were also considered to be either discrete or continuous. Methods from Table 5-8 used classification approach while methods from Table 5-9 used regression approach.

Table 5-8. The top 3 optimal subsets as output of the combination of both filter and wrapper methods for dataset C1 for classification approach

Discrete Feature	Feature subset ranking	Number of features	Selected subset average CV accuracy (%)
False	Top 1	27	86.1
	Top 2	38	86
	Top 3	23	85.7
True	Top 1	43	89.2
	Top 2	45	89.2
	Top 3	44	89.2

Table 5-9. The top 3 optimal subsets as output of the combination of both filter and wrapper method for both datasets C1 for regression approach

Discrete Feature	Feature subset ranking	Number of features	Selected subset average CV RMSE (mg/dl)
False	Top 1	20	105.62
	Top 2	11	105.67
	Top 3	12	105.92

5.2.1.4 Discussion on the effectiveness and reliability of dimensionality reduction methods

By reviewing the results of Table 5-3, it can be seen that wrapper method (SFFS) alone did a good job to reduce the number of features from 158 features to 55 (down by ~65%) for dataset C0 and to 66 (down by ~58%) for dataset C1 when using classification approach. These numbers came with a high CV accuracy of 93% and 90.1% for dataset C0 and C1 respectively. The high

accuracy score indicates that the results are reliable and should be utilized further for the assessment of SVM techniques.

Table 5-4 shows that wrapper method alone managed to reduce the number of features even more when used regression approach. For dataset C0, the number of features was reduced from 158 features to 10 features (down by ~93%). For dataset C1, the number of features was reduced from 158 features to 20 features (down by ~87%). However, their RMSE(s) were relatively high, 94.97 mg/dl and 105.63 mg/dl. This means there is a risk of decreasing the final testing scores when utilizing this top subset for latter processes. Nevertheless, the ability to greatly reduce the dimensionality implies that it might be worth to trade-off some of the accurate performance to extract informative wavelengths for future applications.

By reviewing the three Table 5-5, 5-6, and 5-7, it can be seen that filter methods can be used alone and be able to produce acceptable results with both classification and regression approaches and both assumptions on the continuity of features. However, with the use of classification approach and the assumption of continuous features, filter methods can only reduce a smaller number of features compared to other options, from 158 features to 88 features compared to 55 features and 49 features of the others'. The advantage of using only filter methods is that they are very fast comparing to wrapper methods or to the combination of both filter and wrapper methods while still being able to produced fairly good accurate performance. Details on the processing time of all proposed methods and steps can be found later in section 5.4.

Table 5-8 shows that the combination of wrapper and filter methods produced better results when using classification approach with the assumption of discrete features. When using those conditions, the combination method also produced better CV accuracy than that produced by

using only filter methods, i.e. 89.2% compared to 89.05%. The combination method also managed to reduce more features, from 158 features to 43 features. When using regression approach with the assumption of continuous features, the combination method can also create a smaller subset of features than filter methods. However, the combination method produced a higher RMSE than filter methods. This might be because the wrapper methods alone already produce a relatively high RMSE.

5.2.2 Results of Model 3 and 4

5.2.2.1 Results of Classification Approach for Model 3 and 4

Table 5-10 shows summary of the performance evaluation of all proposed models on the two datasets C0 and C1 using classification approach. The metrics are average non-nested CV accuracy and average nested CV accuracy (use for assessing the training phase); testing accuracy and testing f1 score (use for assessing the testing phase). Note that the closer the values of nested and non-nested accuracy to each other, the more stable, robust the model is. The "Note" column shows the selected subset that was used for training, its number of features, the table that subset can be found, its utilized method, and its approach.

Table 5-10. Summary of the performance evaluation of all proposed models on the two datasets C0 and C1 using classification approach

Dataset	**Model**	**Note**	**Average non-nested CV Accuracy (%)**	**Average nested CV Accuracy (%)**	**Testing Accuracy (%)**	**Testing F1_score (%)**
C0	1	train_test_split=6/4	N/A	N/A	71.41	71.23
	1	train_test_split=8/2	N/A	N/A	78.67	78.01
	2		77.75	N/A	74.83	74.53
	3	# Feature =55 (Table 5-3 Wrapper-Classification)	71.99	67.5	85	84.67
	3	# Feature =10 (Table 5-4 Wrapper-Regression)	37.41	32.92	35	35.83

Table 5-10. Summary of the performance evaluation of all proposed models on the two datasets C0 and C1 using classification approach (cont.)

Dataset	**Model**	**Note**	**Average non-nested CV Accuracy (%)**	**Average nested CV Accuracy (%)**	**Testing Accuracy (%)**	**Testing F1_score (%)**
C1	1	train_test_split=6/4	N/A	N/A	84.71	84.77
	1	train_test_split=8/2	N/A	N/A	85.17	85.22
	2		87.37	N/A	87.17	87.21
	3	# Feature =66 (Table 5-3 Wrapper-Classification)	91.53	90.16	91	90.97
	3	# Feature =20 (Table 5-4 Wrapper-Regression)	43.33	39	30	33.79
	4	# Feature =27 (Table 5-8 Combination-Classification)	87.07	86.4	86	85.83
	4	# Feature =43 (Table 5-8 Combination-Classification)	87.64	86.88	88	88.02
	4	# Feature =20 (Table 5-8 Combination-Regression)	43.33	39	30	33.79

5.2.2.2 Results of Regression Approach for Model 3 and 4

Table 5-11 shows summary of the performance evaluation of all proposed models on the two datasets C0 and C1 using regression approach. The metrics are average non-nested CV accuracy and average nested CV accuracy (use for assessing the training phase); SEP (use for assessing the testing phase). Note that the closer the values of nested and non-nested accuracy to each other, the more stable, robust the model is; the lower the value of SEP, the better result obtained. The "Note" column shows the selected subset that was used for training, its number of features, the table that subset can be found, its utilized method, and its approach.

Table 5-11. Summary of the performance evaluation of all proposed models on the two datasets C0 and C1 using regression approach

Dataset	**Model**	**Note**	**Non-nested SECV (mg/dl)**	**Nested SECV (mg/dl)**	**SEP (mg/dl)**
C0	1	train_test_split=6/4	N/A	N/A	94.12
	1	train_test_split=8/2	N/A	N/A	93.03
	2		67.51	N/A	68.91
	3	# Feature =55 (Table 5-3 Wrapper-Classification)	74.1	75.86	62.14
	3	# Feature =10 (Table 5-4 Wrapper-Regression)	80.84	83.43	83.86

Table 5-11. Summary of the performance evaluation of all proposed models on the two datasets C0 and C1 using regression approach (cont.)

Dataset	**Model**	**Note**	**Non-nested SECV (mg/dl)**	**Nested SECV (mg/dl)**	**SEP (mg/dl)**
C1	1	train_test_split=6/4	N/A	N/A	79.18
	1	train_test_split=8/2	N/A	N/A	78.68
	2		61.42	N/A	60.82
	3	# Feature =66 (Table 5-3 Wrapper-Classification)	45.12	45.40	39.08
	3	# Feature =20 (Table 5-4 Wrapper-Regression)	79.09	79.3	82.93
	4	# Feature =27 (Table 5-8 Combination-Classification)	49.84	50.62	46.53
	4	# Feature =43 (Table 5-8 Combination-Classification)	47.56	48.96	44.15
	4	# Feature =20 (Table 5-8 Combination-Regression)	79.09	79.3	82.93

5.2.3 Discussion on Objective 4 and Model 2, 3, and 4

As described in section 4.4 and 4.5, model 3 utilized only wrapper method and model 4 utilized both filter methods and wrapper methods. In general, it can be seen from Table 5-10 and Table 5-11 that the scoring metrics, which represent the performance of the model, increase significantly with the use of the SFFS technique. Model 3 actually achieves the best performance results for both classification and regression approaches. For example, considering dataset C1, the average non-nested accuracy is 91.53%, average nested CV is 90.16%, testing accuracy is 91%, f1 score is 90.97% for classification approach and average non-nested SECV is 45.12mg/dl, average nested SECV is 45.40mg/dl, and SEP is 39.08mg/dl for regression. The accuracy values are much higher than those of model 1 and 2 and the error values are also much smaller than those of model 1 and 2.

However, it is not always the truth. Even though the assumption whether features are discrete or continuous does not have much influence, choosing the approach of either classification or regression (i.e. assuming whether glucose concentrations are discrete classes or continuous values) when performing SFFS technique of wrapper methods has a great effect on the accurate performance of the system. It can be seen from Table 5-10 and 5-11 that results of model using subsets created by utilizing regression approach, i.e. subsets with numbers of features 20, 10, have significantly lower testing accuracy, f1 score, or significantly higher SEP. However, it should also be noted that these subsets have very few features. It is not guaranteed that the similar scores can be achieved using a randomly picked features. In other words, despite its low scores, the features might be the informative wavelengths and might help to improve the accurate performance when being added to other subset of features. This should be investigated further in future work.

It can also be seen that model 4, which combines filter and wrapper methods, produce good scores but not as good as model 3's. However, model 4 significantly increases the speed of the training pipeline. By using filter method prior to wrapper methods, a number of features had been already screened out which then relieve the burden for wrapper methods. The processing time details will be presented later when discussing about objective 2, which is about the development of a framework that allow optimal performance.

After the evaluation of model 3 and 4 by comparing their results to the results of the simpler models, which demonstrates improvement in term of accurate performance and shows the effectiveness of the added proposed techniques, we need to examine how well the models perform compared to other works in the literature. Let us have a look at the table of summary of results from other studies in the literature, Table 1.2, in Chapter 1 again.

For classification approach, it can be seen that the best result, which was obtained using model 3, shows significant improvement compared to the result in reference [27] with higher values of both accuracy and f1 score. On the other hand, the accuracy of model 3 is comparable to the accuracy of the work in reference [28]. However, one advantage of our study is the automation of the whole operation and that any random ranges of wavelength should be possible inputs. Informative wavelengths can be extracted automatically instead of being manually calculated by first the obtaining transmittance light through water then computing optical density like in reference [28]. In addition, the model in reference [28] does not have an evaluation system to achieve the accuracy but needs to be calculated manually by comparing the obtained values with value measured by another traditional equipment to check whether they match or not. Despite the aforementioned points, it is still difficult to fully compare and evaluation the two works because the difference in samples and data used. Therefore, besides other references, we

also utilize the EN ISO 15197:2015 [65] from the International Standardization Organization as a base to validate the model performance. It is stated that for clinical usage, the accuracy of the instrumentation should be at least 95%. Nevertheless, we can still conclude the accuracy below 95% but above 90% to be acceptable and can be used as an initial screening result before any further examinations.

Similar to classification approach, it is still difficult to directly compare the results due to differences in used samples and data even though, at first glance, it is obvious that the obtained results have reasonable standards error that much lower than some studies' but higher than others'. The EN ISO 15197:2015 [65] can also be applied in this situation to assess the performance of model 3 and 4 following regression approach. It is stated that for samples with glucose concentration < 100mg/dl then the error should be $\pm$15mg/dl; for samples with glucose concentration $\geq$100mg/dl then the error should be $\pm$15% of the actual value. Therefore, for samples numbered 1 to 5, the obtained errors are higher than the acceptable standard; however, for samples numbered 6 to 10, the obtained errors are within the acceptable standard. This is considered to be a drawback in experimental design that complicates the assessment of the models performance. This drawback is also applied to other studies in the literature. Therefore, this conclusion might imply that a different range of concentration should be prepared for the samples in future works.

Table 1.2. Literature Summary (First occurrence in Chapter 1)

Ref.	Sample Concentration Range	Spectral Range	Model Type		Metric			
					Regression		Classification	
			Regression	Classification	SECV/SEC/r	SEP	Accuracy	F1 Score
Malin et al., 1999 [21]	1st: 91-446mg/dl, 2nd: 82-294mg/dl, 3rd: 97-171mg/dl	1100-1380nm, 1450-1850nm, 2050-2375nm	PLSR		28.82mg/dl 30.63mg/dl 17.1mg/dl	138mg/dl 44mg/dl 41mg/dl		
Jeon et al., 2006 [24]	0-1000mg/dl	1100-1850nm, 2200-2500nm, 1100-2500nm	PLSR		33.51mg/dl 108.04mg/dl 69.58mg/dl	437.54mg/dl		
Amerov et al., 2004 [23]	54-540mg/dl	1111-1851nm, 2000-2500nm	PLSR		*20.5mg/dl *10.09mg/dl	21.62mg/dl 17.3mg/dl		
Jun Chen et al., 2004 [25]		1658-1769nm, 2192-2439nm	PLSR		*22.34mg/dl *5.58mg/dl	20.18mg/dl 8.11mg/dl		
Al-Mbaideen et al., 2010 [26]	20-500mg/dl	2100-2400nm	PCR		N/A	40mg/dl		

Table 1.2. Literature Summary (cont.) (First occurrence in Chapter 1)

Ref.	Sample Concentration Range	Spectral Range	Model Type		Metric			
					Regression		Classification	
			Regression	Classification	SECV/SEC/r	SEP	Accuracy	F1 Score
Habibullah et al., 2019 [27]	72-360mg/dl	1300-2600nm		SVM			77.5%	76%
Haider et al., 2017 [28]	0-450mg/dl	500-1200nm	***CEG				*** 90-92%	N/A
Kasahara et al., 2018 [29]		8333-10204nm	MLR		**0.36			

By now, there are several key points that can be concluded. Firstly, both wrapper methods and filter methods can be used alone to extract informative wavelengths and improve the accurate performance of the system. Secondly, using only wrapper methods produces the best accuracy, f1 score or the smallest SEP. Using a combination of filter and wrapper methods together with classification approach and an assumption of discrete features would produce better accuracy, f1 score or smaller SEP than using only filter methods. Thirdly, using a combination of wrapper and filter methods would reduce the number of features to a smaller value and might create a trade-off between accurate performance and processing time. The next point is that the assumption whether features are discrete or continuous does not have much influence and it is recommended to choose classification rather than regression approach when performing SFFS technique of wrapper methods as it has great effect on the accurate performance of the system. Other point is that, even though choosing regression approach when performing SFFS technique lowers the scores, it can significantly reduce the number of features, up to 87% smaller. These features might contain much more information compared to other features. It can be seen that compared to certain reference, the proposed models did shown potential in improve the accurate performance though a complete analysis might not be possible to due complication in difference data. For classification, even though the accuracy is below critical threshold for important clinical usage, the proposed model can still be used to conduct initial screening. For regression, the obtained results did show improvement compared to some other works. When being compared to the standard, the error is acceptable for half of the datasets but not for the other half.

5.3 Results of Experiments for Objective 5

Objective 5 is to utilize feature selection techniques for extracting informative wavelengths for noninvasive glucose monitoring. This objective was completed by using the dimensional reduction step of model 3 and model 4. This section will first present the extracted wavelengths using only wrapper methods and using a combination of filter and wrapper methods. For each scenario, both classification approach and regression approach were tested. Then it will analyze the obtained results and draw conclusion.

5.3.1 Extracted Wavelengths Using Dimensional Reduction Step of Model 3 and Model

Table 5-12. Selected wavelengths of the top 3 optimal subsets in Table 5-3

Dataset	Feature subset ranking	Selected wavelengths (nm)
C0	Top1	1480, 1507, 1518, 1530, 1669, 1696, 1704, 1763, 1794, 1802, 1810, 1835, 1843, 1869, 1877, 1886, 1895, 1904, 1913, 1950, 1959, 1978, 1988, 1998, 2007, 2017, 2028, 2058, 2069, 2080, 2090, 2101, 2112, 2157, 2169, 2192, 2228, 2253, 2291, 2304, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2535, 2551, 2600
	Top 2	1308, 1365, 1374, 1379, 1403, 1408, 1413, 1427, 1433, 1438, 1480, 1524, 1530, 1559, 1603, 1609, 1615, 1622, 1628, 1635, 1641, 1648, 1675, 1696, 1748, 1755, 1763, 1778, 1786, 1794, 1802, 1810, 1818, 1826, 1835, 1843, 1860, 1869, 1877, 1886, 1895, 1904, 1913, 1922, 1931, 1940, 1950, 1959, 1968, 1978, 1988, 1998, 2007, 2017, 2028, 2038, 2048, 2058, 2069, 2080, 2090, 2101, 2112, 2123, 2134, 2146, 2157, 2169, 2180, 2192, 2204, 2216, 2228, 2241, 2253, 2266, 2278, 2291, 2304, 2318, 2331, 2344, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2535, 2551, 2567, 2583, 2600, 2617
	Top 3	1480, 1507, 1518, 1530, 1669, 1696, 1704, 1763, 1794, 1802, 1810, 1835, 1843, 1869, 1877, 1886, 1895, 1904, 1913, 1950, 1959, 1978, 1988, 1998, 2007, 2017, 2028, 2048, 2058, 2069, 2080, 2090, 2101, 2112, 2157, 2169, 2080, 2090, 2101, 2112, 2157, 2169, 2192, 2228, 2253, 2291, 2304, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 24888, 2504, 2519, 2535, 2551, 2600

Table 5-12. Selected wavelengths of the top 3 optimal subsets in Table 5-3 (cont.)

Dataset	Feature subset ranking	Selected wavelengths (nm)
C1	Top1	1635, 1682, 1689, 1733, 1755, 1778, 1802, 1818, 1826, 1835, 1852, 1860, 1869, 1877, 1895, 1904, 1913, 1931, 1950, 1959, 1978, 1988, 1998, 2017, 2028, 2038, 2048, 2058, 2069, 2080, 2090, 2101, 2112, 2123, 2134, 2146, 2157, 2169, 2192, 2216, 2228, 2241, 2253, 2266, 2278, 2304, 2318, 2331, 2344, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2535, 2551, 2567, 2583, 2600, 2617
	Top 2	1682, 1689, 1755, 1802, 1843, 1852, 1860, 1877, 1895, 1904, 1922, 1931, 1950, 1959, 1978, 1988, 1998, 2028, 2038, 2058, 2096, 2090, 2112, 2123, 2134, 2146, 2157, 2192, 2216, 2241, 2253, 2266, 2278, 2318, 2331, 2344, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2535, 2551, 2583, 2600, 2617
	Top 3	1635, 1682, 1689, 1733, 1755, 1778, 1802, 1818, 1826, 1835, 1852, 1860, 1869, 1877, 1895, 1904, 1913, 1931, 1950, 1959, 1968, 1978, 1988, 1998, 2017, 2028, 2038, 2048, 2058, 2069, 2080, 2090, 2101, 2112, 2123, 2134, 2146, 2157, 2169, 2192, 2216, 2228, 2241, 2253, 2266, 2291, 2304, 2318, 2331, 2344, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2535, 2551, 2567, 2583, 2600, 2617

Table 5-13. Selected wavelengths of the top 3 optimal subsets in Table 5-4

Dataset	Feature subset ranking	Selected wavelengths (nm)
C0	Top1	1365, 1427, 1433, 1474, 1502, 1524, 1530, 1536, 1553, 2600
	Top 2	1347, 1365, 1427, 1433, 1474, 1502, 1524, 1530, 1536, 1553, 1704, 1904, 1913, 2583, 2600
	Top 3	1347, 1365, 1615, 1704, 1711, 1978, 2519, 2551, 2567, 2583
C1	Top1	1299, 1303, 1329, 1333, 1351, 1360, 1365, 1370, 1384, 1388, 1393, 1417, 1458, 1464, 1469, 1474, 1513, 1542, 1584, 1590
	Top 2	1299, 1303, 1329, 1333, 1365, 1384, 1388, 1393, 1458, 1469, 1590
	Top 3	1299, 1303, 1329, 1333, 1365, 1370, 1384, 1388, 1393, 1458, 1469

Table 5-14. Selected wavelengths of the top 3 optimal subsets in Table 5-8

Discrete Feature	Feature subset ranking	Selected wavelengths (nm)
False	Top1	1968, 1998, 2017, 2038, 2069, 2101, 2146, 2228, 2241, 2253, 2278, 2291, 2304, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2504, 2519, 2567, 2583, 2617
	Top 2	1968, 1978, 1998, 2007, 2028, 2048, 2069, 2090, 2101, 2134, 2146, 2169, 2180, 2192, 2216, 2228, 2241, 2253, 2278, 2304, 2318, 2344, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2535, 2567, 2583, 2617
	Top 3	1968 , 1998, 2038, 2069, 2101, 2146, 2241, 2253, 2266, 2278, 2304, 2358, 2372, 2400, 2414, 2429, 2443, 2458, 2473, 2504, 2519, 2567, 2583
True	Top1	1895, 1950, 1968, 1978, 1998, 2007, 2017, 2028, 2038, 2048, 2069, 2101, 2112, 2134, 2146, 2157, 2169, 2180, 2192, 2204, 2253, 2266, 2278, 2291, 2318, 2331, 2344, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2473, 2488, 2504, 2535, 2551, 2567, 2583, 2600, 2617
	Top 2	1895, 1922, 1950, 1968, 1978, 1998, 2017, 2028, 2038, 2048, 2058, 2069, 2101, 2112, 2123, 2134, 2146, 2157, 2169, 2180, 2192, 2204, 2253, 2266, 2278, 2291, 2318, 2331, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2458, 2473, 2488, 2504, 2519, 2551, 2567, 2583, 2600, 2617
	Top 3	1895, 1950, 1968, 1978, 1998, 2007, 2017, 2028, 2038, 2048, 2069, 2101, 2112, 2134, 2146, 2157, 2169, 2180, 2192, 2204, 2241, 2253, 2266, 2278, 2291, 2318, 2331, 2344, 2358, 2372, 2386, 2400, 2414, 2429, 2443, 2473, 2488, 2504, 2535, 2551, 2567, 2583, 2600, 2617

Table 5-15. Selected wavelengths of the top 3 optimal subsets in Table 5-9

Discrete Feature	Feature subset ranking	Selected wavelengths (nm)
False	Top1	1299, 2303, 1329, 1333, 1351, 1360, 1365, 1370, 1384, 1388, 1393, 1417, 1458, 1464, 1469, 1474, 1513, 1542, 1584, 1590
	Top 2	1299, 1303, 1329, 1333, 1365, 1384, 1388, 1393, 1458, 1469, 1590
	Top 3	1299, 1303, 1329, 1333, 1365, 1370, 1384, 1388, 1393,1458, 1469, 1584

5.3.2 Discussion on Objective 5

Table 5-12 shows the features, in term of wavelengths (nm), of each of the top 3 optimal subsets of Table 5-3. It can be seen that most of the selected wavelengths fell in the range between 1800-2400nm. In other words, it can be inferred that the informative wavelengths would belong to the range between 1800-2400nm.

Table 5-13 shows the features, in term of wavelengths (nm), of each of the top 3 optimal subsets of Table 5-4. It can be seen that most of the selected wavelengths fell in the range between 1300-1600nm.

Table 5-14 shows the features, in term of wavelengths (nm), of each of the top 3 optimal subsets of Table 5-8. It can be seen that most of the selected wavelengths fell in the range between 2000-2600nm.

Table 5-15 shows the features, in term of wavelengths (nm), of each of the top 3 optimal subsets of Table 5-9. It can be seen that most of the selected wavelengths fell in the range between 1300-1600nm.

These wavelengths include the wavelengths, as mentioned in section 5.2, which do not produce high result but might contain relevant information. These ranges also agree with some of the finding in the literature. Therefore, it might be concluded that the proposed system model is competent in finding the informative wavelengths for glucose concentration monitoring.

5.4 Results of Experiments for Objective 2

After completing all other objectives and reviewing their results, it is now possible to present relevant results to objective 2 and analyze them. Section 5.1 shows that using only SVM

technique is inadequate and would produce results with low accuracy. By analyzing the results of model 2 in Table 5-9 and Table 5-10 in section 5.2, it can be seen that adding pipeline and search technique to simplify the flow of the program and tune the hyperparameters as well as adding cross-validation techniques to control and prevent the leakage of information can increase the performance of the system. The testing accuracy and testing f1 score increase ~2-3% and the SEP decrease up to ~18-19mg/dl. Discussion of the dimensionality reduction methods shows that SFFS technique of wrapper methods and filter methods can extract informative wavelengths and significantly improve the accuracy, f1 score or decrease the errors. The best results can be obtained by using only wrapper methods. Filter methods can be used alone and be able to produce acceptable results. The combination of both filter and wrapper method actually produces good results but not as good as using solely wrapper method. By examining Table 5.16, which show in details the processing time needed to complete a specific step or phase, it can be seen that the addition of filter methods speeds up the dimensionality reduction step and training step by roughly 17 times and 9 times respectively. Therefore, there is a trade-off between speed and performance by utilizing filter methods together prior to wrapper methods.

In conclusion, a framework for optimal results with the most accurate performance should include every steps in model 3 which consists of SFFS, train_test_split, Standard Scaler, PCA, SVM/SVR, Hyperparameter Grid Search tuning, pipeline, and nested-cross validation techniques. However, if time is an important constraint, then techniques of filter method should be added to the system, prior to the SFFS technique.

Table 5-16. The average processing time of all proposed models

Dataset	Model	Step	Technique	Time (s)
C0	1	Feature selection	Filter method	n/a
			Wrapper Method	n/a
		Training	StandardScaler + PCA + SVM/SVR	0.011
		Testing	n/a	0.070
	2	Feature Selection	Filter method	n/a
			Wrapper Method	n/a
		Training	StandardScaler + PCA + SVM/SVR + GridSearchCV + Cross-validation	16.608
		Testing	n/a	0.006
	3	Feature Selection	Filter method	n/a
			Wrapper Method	23980.882
		Training	StandardScaler + PCA + SVM/SVR + GridSearchCV + Nested cross-validation	558.586
		Testing	n/a	0.060
	4	Feature Selection	Filter method	n/a
			Wrapper Method	n/a
		Training	StandardScaler + PCA + SVM/SVR + GridSearchCV + Nested cross-validation	n/a
		Testing	n/a	n/a

Table 5-16. The average processing time of all proposed models (cont.)

Dataset	Model	Step	Technique	Time (s)
C1	1	Feature selection	Filter method	n/a
			Wrapper Method	n/a
		Training	StandardScaler + PCA + SVM/SVR	0.042
		Testing	n/a	0.018
	2	Feature Selection	Filter method	n/a
			Wrapper Method	n/a
		Training	StandardScaler + PCA + SVM/SVR + GridSearchCV + Cross-validation	248.696
		Testing	n/a	0.023
	3	Feature Selection	Filter method	n/a
			Wrapper Method	262434.207
		Training	StandardScaler + PCA + SVM/SVR + GridSearchCV + Nested cross-validation	36188.023
		Testing	n/a	0.056
	4	Feature Selection	Filter method	2.216
			Wrapper Method	15374
		Training	StandardScaler + PCA + SVM/SVR + GridSearchCV + Nested cross-validation	4153.703
		Testing	n/a	0.046

Chapter 6 Conclusion

6.1 Research Summary

In this study, SVM is adapted to calibrate the relationship between wavelengths and glucose concentration and predict a glucose concentration based on provided wavelength signal. SVM is proved to provide good results when used in combination with other machine learning techniques. A list of contributions has been presented earlier in Sec. 1.7. In this section, the key findings will be summarized.

Firstly, both wrapper methods and filter methods can be used alone to extract informative wavelengths and improve the accurate performance of the system. However, it was found that wrapper methods of feature selection are essential for glucose monitoring using an SVM-based modeling approach as it produces the best accuracy, f1 score or the smallest SEP. The SFFS technique is a good candidate for wrapper methods algorithm. Model 3, which utilized the subset created by the SFFS technique as input, produced the best results in terms of performance with average non-nested CV accuracy of 91.53%, testing accuracy of 91%, f1 score of 90.97% for classification and average non-nested SECV of 45.12mg/dl, SEP of 39.08mg/dl for regression. For classification approach, these values shown improvement, or at least comparable results, when being compared to other works the literature. Even though these results are lower than critical standard for clinical applications, they can still be used for initial screening process that might assist in diagnosing. For regression approach, the obtained error values are within acceptable standard range with certain condition. This implies that future works need to change the experimental design especially in the sample preparation and data acquisition steps.

In addition, filter methods of feature selection were found to offer a trade-off between speed and performance when used in combination with wrapper methods. Model 4, which utilized subsets created by both filter and wrapper methods as input, performed with slightly inferior scoring metrics while being significantly faster, up to 17 times for feature selection and 9 times for training. Therefore, for optimal results, it is recommended to utilize only the SFFS technique. However, if time is an important constraint, then techniques of filter method should be added to the system.

Both filter and wrapper methods of feature selection techniques are shown to exhibit promising potentials to extract the most informative wavelengths for noninvasive glucose monitoring. When conducting the feature selection process, metrics of classification approach appears to lead to subsets that can create better performances. However, metrics of regression approach can reduce the feature spaces; hence isolate informative wavelengths, more efficiently. Even though they might not provide immediate results, further investigation on the isolated wavelengths by this approach with other supporting wavelengths might provide enhanced performance.

It is found that the assumption whether features are discrete or continuous does not have much influence while choosing the approach of either classification or regression (i.e. assuming whether glucose concentrations are discrete classes or continuous values) when performing SFFS technique of wrapper methods has a great effect on the accurate performance of the system.

In conclusion, the proposed system model which consists of 3 phases and 4 main steps appears to be efficient enough in optimizing the prediction of glucose concentration while minimizing bias due to information leakage and can be utilized further in future work. It also seems that the proposed system model can extract relevant information of any random input

wavelength ranges. For this study, the variations of the system models suggested that the most informative wavelengths for noninvasive glucose monitoring might fall in the ranges: 1300-1600nm, 1800-2400nm, and 2000-2600nm which agrees with some finding in the literature.

6.2 Future work

While the characteristics of glucose in distilled water solution have been investigated in detail and the proposed models provided promising potential, a natural extension should next involve investigation on other biological materials. Future samples should be prepared by mixing β-D-glucose together with animal hemoglobin (bovine hemoglobin), or urea, etc. respectively. For example, at first, the sample set could contain only three elements, i.e. β-D-glucose, distilled water, and one other biological material. Different sample sets would be prepared to test the ability of the model to detect the extra added materials and how they interfere with the results. After proper results and understanding have been achieved, solutions of more materials would be tested. For future work, we would also need to prepare samples with a narrower range of glucose concentration to test the sensitivity and limit of the proposed techniques. Furthermore, as having reviewed in Table 1-2, visible and MIR wavelengths should also be investigated and compared with the current results. Future work should also prioritize the classification approach when further investigate advanced dimensionality reduction technique.

Reference

[1] Gojka Roglic. “WHO Global Report on Diabetes: A Summary.” *International Journal of Noncommunicable Diseases*, vol. 1, no. 1, 2016, pp. 3–8.

[2] “Diagnosis and Classification of Diabetes Mellitus.” *Diabetes Care*, vol. 35, no. 1, 2012, pp. S64–71.

[3] “Diagnosis and Classification of Diabetes Mellitus.” *Diabetes Care*, 30 Suppl 1, 2007, pp. S42–7

[4] Cho, N.H, et al. “IDF Diabetes Atlas: Global Estimates of Diabetes Prevalence for 2017 and Projections for 2045.” *Diabetes Research and Clinical Practice*, vol. 138, 2018, pp. 271–281.

[5] Yadav, Jyoti, et al. “Prospects and Limitations of Non-Invasive Blood Glucose Monitoring Using near-Infrared Spectroscopy.” *Biomedical Signal Processing and Control*, vol. 18, no. 2, 2015, pp. 214–227.

[6] Klonoff, David C. “Continuous Glucose Monitoring: Roadmap for 21st Century Diabetes Therapy.” *Diabetes Care*, vol. 28, no. 5, 2005, pp. 1231–1239.

[7] Pickup, John C, et al. “In Vivo Glucose Monitoring: the Clinical Reality and the Promise.” *Biosensors and Bioelectronics*, vol. 20, no. 10, 2005, pp. 1897–1902.

[8] Bertoni, A G, et al. “Diabetes and the Risk of Infection-Related Mortality in the U.S.” *Diabetes Care*, vol. 24, no. 6, 2001, pp. 1044–1049.

[9] Gross, T M, et al. “Performance Evaluation of the MiniMed Continuous Glucose Monitoring System during Patient Home Use.” *Diabetes Technology & Therapeutics*, vol. 2, no. 1, 2000, pp. 49–56.

[10] S. Darzi, Y. Munz, The impact of minimally invasive surgical techniques, Annu. Rev. Med. 55 (2004) 223–237.

[11] S.N. Thennadil, J.L. Rennert, B.J. Wenzel, K.H. Hazen, T.L. Ruchti, M.B. Block, Comparison of glucose concentration in interstitial fluid, and capillary and venous blood during rapid changes in blood glucose levels, Diabetes Technol. Ther. 3 (2001) 357–365.

[12] T. Koschinsky, L. Heinemann, Sensors for glucose monitoring: technical and clinical aspects, Diabetes Metab. Res. Rev. 17 (2001) 113–123.

[13] Benoît Leboulanger. "Reverse Iontophoresis for Non-Invasive Transdermal Monitoring." *Physiological Measurement*, vol. 25, no. 3, 2004, pp. R35–R50.

[14] Tierney, et al. "Design of a Biosensor for Continual, Transdermal Glucose Monitoring." *Clinical Chemistry*, vol. 45, no. 9, 1999, pp. 1681–1683.

[15] Panchagnula, Ramesh, et al. "Transdermal Iontophoresis Revisited." *Current Opinion in Chemical Biology*, vol. 4, no. 4, 2000, pp. 468–473.

[16] Park, Ho Dong, et al. "Design of a Portable Urine Glucose Monitoring System for Health Care." *Computers in Biology and Medicine*, vol. 35, no. 4, 2005, pp. 275–286.

[17] Cameron BD, Baba JS, Cot´e GL. "Optical polarimetry applied to the development of a noninvasive vivo glucose monitor." *Proceedings of SPIE*, vol. 3923, 2004, pp. 66–77.

[18] Berger, Andrew J, et al. "Feasibility of Measuring Blood Glucose Concentration by near-Infrared Raman Spectroscopy." *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 53, no. 2, 1997, pp. 287–292.

[19] A. Ergin and G. A. Thomas, "Noninvasive detection of glucose in porcine eyes," *Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference 2005.*, Hoboken, NJ, 2005, pp. 246-247.

[20] Ellis, David I., and Royston Goodacre. “Metabolic Fingerprinting in Disease Diagnosis: Biomedical Applications of Infrared and Raman Spectroscopy.” *The Analyst*, vol. 131, no. 8, 2006, pp. 875–885.

[21] Malin, S F, et al. “Noninvasive Prediction of Glucose by near-Infrared Diffuse Reflectance Spectroscopy.” *Clinical Chemistry*, vol. 45, no. 9, 1999, pp. 1651–8.

[22] Khalil, Omar S. “Non-Invasive Glucose Measurement Technologies: an Update from 1999 to the Dawn of the New Millennium.” *Diabetes Technology & Therapeutics*, vol. 6, no. 5, 2004, pp. 660–697.

[23] Amerov, Airat K, et al. “Molar Absorptivities of Glucose and Other Biological Molecules in Aqueous Solutions over the First Overtone and Combination Regions of the near-Infrared Spectrum.” *Applied Spectroscopy*, vol. 58, no. 10, 2004, pp. 1195–1204.

[24] Jeon, Kye Jin, et al. “Comparison between Transmittance and Reflectance Measurements in Glucose Determination Using near Infrared Spectroscopy.” *Journal of Biomedical Optics*, vol. 11, no. 1, 2006, p. 014022.

[25] Chen, Jun, et al. “Comparison of Combination and First Overtone Spectral Regions for near-Infrared Calibration Models for Glucose and Other Biomolecules in Aqueous Solutions.” *Analytical Chemistry*, vol. 76, no. 18, 2004, pp. 5405–5413.

[26] Al-Mbaideen, Amneh A. A, et al. “Determination of Glucose Concentration from near-Infrared Spectra Using Principle Component Regression Coupled with Digital Bandpass Filter.” *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, 2010, pp. 243–248.

[27] Habibullah, Mohammad A., et al. “NIR-Spectroscopic Classification of Blood Glucose Level Using Machine Learning Approach.” 2019 *IEEE Canadian Conference of Electrical and Computer Engineering*, CCECE 2019, 2019, pp. 1–4

[28] Ali, Haider, et al. “Novel Approach to Non-Invasive Blood Glucose Monitoring Based on Transmittance and Refraction of Visible Laser Light.” *IEEE Access*, vol. 5, 2017, pp. 9163–9174

[29] Kasahara, Ryosuke, et al. “Noninvasive Glucose Monitoring Using Mid-Infrared Absorption Spectroscopy Based on a Few Wavenumbers.” *Biomedical Optics Express*, vol. 9, no. 1, 2018, pp. 289–302.

[30] Bro, R. “Multivariate Calibration - What Is in Chemometrics for the Analytical Chemist?” *Analytica Chimica Acta*, vol. 500, no. 1-2, 2003, pp. 185–194.

[31] Goodarzi, Mohammad, et al. “Towards Better Understanding of Feature-Selection or Reduction Techniques for Quantitative Structure–Activity Relationship Models.” *Trends in Analytical Chemistry*, vol. 42, 2013, pp. 49–63.

[32] Kalivas, John H. “Multivariate Calibration, an Overview.” *Analytical Letters*, vol. 38, no. 14, 2005, pp. 2259–2279.

[33] Wentzell, Peter D, and Lorenzo Vega Montoto. “Comparison of Principal Components Regression and Partial Least Squares Regression through Generic Simulations of Complex Mixtures.” *Chemometrics and Intelligent Laboratory Systems*, vol. 65, no. 2, 2003, pp. 257–279.

[34] Wold, Svante, et al. “PLS-Regression: a Basic Tool of Chemometrics.” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, 2001, pp. 109–130.

[35] Krämer, Nicole, and Masashi Sugiyama. “The Degrees of Freedom of Partial Least Squares Regression.” *Journal of the American Statistical Association*, vol. 106, no. 494, 2011, pp. 697–705.

[36] Faber, Nicolaas M., and Gemperline, P. J. “A Closer Look at the Bias–Variance Trade-off in Multivariate Calibration.” *Journal of Chemometrics*, vol. 13, no. 2, 1999, pp. 185–192.

[37] Næs, Tormod, and Harald Martens. “Principal Component Regression in NIR Analysis: Viewpoints, Background Details and Selection of Components.” *Journal of Chemometrics*, vol. 2, no. 2, 1988, pp. 155–167.

[38] Kemps, Bart J. J, et al. “The Importance of Choosing the Right Validation Strategy in Inverse Modelling.” *Journal of Near Infrared Spectroscopy*, vol. 18, no. 4, 2010, pp. 231–237.

[39] Brereton, Richard G., and Gavin R. Lloyd. “Support Vector Machines for Classification and Regression.” *The Analyst*, vol. 135, no. 2, 2010, pp. 230–267.

[40] Thissen, U, et al. “Comparing Support Vector Machines to PLS for Spectral Regression Applications.” *Chemometrics and Intelligent Laboratory Systems*, vol. 73, no. 2, 2004, pp. 169–179.

[41] Bulent Ustun. “A comparison of Support Vector Machines and Partial Least Squares regression on spectral data.” Katholieke Universiteis Nijmegen, 2003, Master thesis.

[42] Goodarzi, Mohammad, et al. “Feature Selection Methods in QSAR Studies.” *Journal of AOAC International*, vol. 95, no. 3, 2012, pp. 636–651.

[43] D.T. Delpy et al. “Estimation of optical path length through tissue from direct time of flight measurements.” Phys. Med. Biol. 33, 1988, pp. 1433-1442

[44] Lam, S, et al. “Non-Invasive Blood Glucose Measurement by near Infrared Spectroscopy: Machine Drift, Time Drift and Physiological Effect.” *Spectroscopy*, vol. 24, no. 6, 2010, pp. 629–639.

[45] T.B. Blank et al. “The use of near-infrared diffuse reflectance for the non-invasive prediction of blood glucose levels.” LEOS Newslett. 13, 1999

[46] Lam, Chak. *Clinical Evaluation of Non-Invasive Blood Glucose Measurement by Using near Infrared Spectroscopy via Inter- and Intra-Subject Analysis.* Vol. 71, 2009.

[47] Tenhunen, Jussi, et al. "Non-Invasive Glucose Measurement Based on Selective near Infrared Absorption; Requirements on Instrumentation and Spectral Range." *Measurement,* vol. 24, no. 3, 1998, pp. 173–177.

[48] Theodoridis, Sergios, and Koutroumbas, Konstantinos. *Pattern Recognition*. 4th ed., Elsevier/Academic Press, 2009.

[49] https://www.neospectra.com/wp-content/uploads/2018/01/Neospectra-SWS62231-Datasheet.-11-12-17.pdf

[50] Pedregosa et al. "Scikit-learn: Machine Learning in Python." *JMLR 12*, 2011, pp. 2825-2830.

[51] Raschka, Sebastian. "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack." *J Open Source Softw 3*, 24, 2018

[52] J. Michael McMillin. "Clinical Methods: The History, Physical, and Laboratory Examinations." *Clinical Methods,* 3rd edition, Boston: Butterworths, chapter 141

[53] Theodoridis, Sergios, and Koutroumbas, Konstantinos. *Pattern Recognition*. 4th ed., Elsevier/Academic Press, 2009.

[54] Koutroumbas, Konstantinos, et al. *Introduction to Pattern Recognition: A Matlab Approach*. Elsevier Inc., 2010.

[55] Cover, T. M., and Thomas, Joy A. *Elements of Information Theory*. Wiley, 1991.

[56] Ross, Brian C., and Daniele Marinazzo. "Mutual Information between Discrete and Continuous Data Sets." *PLoS ONE,* vol. 9, no. 2, 2014, p. e87357.

[57] Kraskov, Alexander, et al. “Estimating Mutual Information.” *ArXiv.org*, vol. 69, no. 6 Pt 2, 2003, p. 066138.

[58] Larose, Daniel T., and Larose, Chantal D. “k -Nearest Neighbor Algorithm.” *Wiley Series on Methods and Applications in Data Mining*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2014, pp. 149–164

[59] Jolliffe, I. T. “Principal Component Analysis.” *Springer-Verlag*, 2002.

[60] Smola, Alex, and J. Schölkopf. “A Tutorial on Support Vector Regression.” *Statistics and Computing*, vol. 14, no. 3, 2004, pp. 199–222.

[61] ÜStün, B, et al. “Visualisation and Interpretation of Support Vector Regression Models.” *Analytica Chimica Acta*, vol. 595, no. 1-2, 2007, pp. 299–309.

[62] Ferri, F. J et al. "Comparative study of techniques for large-scale feature selection." Pattern Recognition in Practice IV, North Holland, 1994, pp. 403-413.

[63] Pudil, P, et al. “Floating Search Methods in Feature Selection.” *Pattern Recognition Letters*, vol. 15, no. 11, 1994, pp. 1119–1125.

[64] Amerov, Airat K., et al. “Scattering and Absorption Effects in the Determination of Glucose in Whole Blood by near-Infrared Spectroscopy.” Analytical Chemistry, vol. 77, no. 14, 2005, pp. 4587–4594.

[65] International Standardization Organization. “In vitro diagnostic test systems – Requirements for blood-glucose monitoring systems for self-testing in managing diabetes mellitus.” ISO No. 15197:2013, 2013. Retrieved from https://www.iso.org/standard/54976.html.