ESTIMATION OF LOW SUCROSE CONCENTRATIONS AND CLASSIFICATION OF BACTERIA CONCENTRATIONS WITH MACHINE LEARNING ON SPECTROSCOPIC DATA

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Bahadır MEZGİL

June 2019
İZMİR

We approve the thesis of **Bahadır MEZGİL**

Examining Committee Members:

____

Assoc. Prof. Dr. Derya BİRANT
Department of Computer Engineering, Dokuz Eylül University

____

Asst. Prof. Dr. Nesli ERDOĞMUŞ
Department of Computer Engineering, İzmir Institute of Technology

____

Assoc. Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering, İzmir Institute of Technology

27 June 2019

____

Assoc. Prof. Dr. Yalın BAŞTANLAR
Supervisor, Department of Computer Engineering
İzmir Institute of Technology

____

Assoc. Prof. Dr. Tolga AYAV
Head of the Department of
Computer Engineering

____

Prof. Dr. Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Yalın BAŞTANLAR for his continuous support, effort, and optimism during my graduate period. He always mentors me with all his sincerity and kindness. I would like to thank Assoc. Prof. Dr. Derya BİRANT and Asst. Prof. Dr. Nesli ERDOĞMUŞ for devoting their precious time to evaluate my thesis. Also, I would like to thank IZTECH Department of Bioengineering since with their help, we have conducted a study between departments of IZTECH which would not be possible without them.

I am very thankful for my wife and my family who are always behind me for being a source of morale all the time.

Finally, I would like to thank my dearest friend Furkan Eren UZYILDIRIM for his encouragement and advice during my graduate studies.

ABSTRACT

ESTIMATION OF LOW SUCROSE CONCENTRATIONS AND CLASSIFICATION OF BACTERIA CONCENTRATIONS WITH MACHINE LEARNING ON SPECTROSCOPIC DATA

Spectroscopy can be used to identify elements. In a similar way, there are recent studies that use optical spectroscopy to measure the material concentrations in chemical solutions. In this study, we employ machine learning techniques on collected ultraviolet-visible spectra to estimate the level of sucrose concentrations in solutions and to classify bacteria concentrations. Some metal nanoparticles are very sensitive to refraction index changes in the environment and this helps to detect small refraction index changes in the solution. In our study, gold nanoparticles are used and we benefited from this property to estimate sucrose concentrations. The samples in different low sucrose concentration solutions are obtained by mixing the sucrose measured with precision scales with pure water and then the UV-Vis spectrum of each sample is measured. For the bacteria concentration solutions, spectra for six different bacteria concentrations are captured. Spectra of the same solutions are also captured before adding the bacteria. For each of these solutions, four sets are prepared where gold nanoparticles are not grown (minute 0) and grown for 4 minutes, 10 minutes and 12 minutes. After the dataset preparation, these spectrum measurements are transferred into MATLAB environment as sucrose concentration dataset and bacteria solution dataset. Then the necessary preprocessing steps are performed in order to get the most informative and distinguishing information from these datasets. The raw measurement values and processed spectrum measurements are trained with shallow Artificial Neural Networks (ANN) on MATLAB Deep Learning Toolbox and Support Vector Machine (SVM) on MATLAB Statistics and Machine Learning Toolbox. When the results of the conducted machine learning experiments are examined, success rate is promising for the estimation of sucrose concentrations and very high for classification of bacteria concentrations in pure water solution.

ÖZET

SPEKTROSKOPİK VERİ ÜZERİNDE MAKİNE ÖĞRENMESİ İLE DÜŞÜK SÜKROZ KONSANTRASYONLARININ KESTİRİMİ VE BAKTERİ KONSANTRASYONLARININ SINIFLANDIRILMASI

Spektroskopi elementleri tanımlamak için kullanılabilir. Benzer şekilde, kimyasal çözeltilerdeki madde konsantrasyonlarını sınıflandırmak için optik spektroskopiyi kullanan yeni çalışmalar vardır. Bu çalışmada, çözeltilerdeki sükroz konsantrasyonunun seviyesini tahmin etmek ve bakteri konsantrasyonlarını sınıflandırmak için toplanan ultraviyole-görünür bölge (UV-Vis) spektrumlarda makine öğrenme tekniklerini kullanıyoruz. Bazı metal nanopartiküller, ortamdaki kırılma endeksi değişikliklerine karşı çok hassastır ve bu özellik çözeltideki küçük kırılma endeksi değişikliklerini tespit etmeye yardımcı olur. Çalışmamızda altın nanoparçacıkları kullanılmış ve sükroz konsantrasyonlarını tahmin etmek için bu özellikten faydalanıldı. Farklı düşük sükroz konsantrasyon çözeltilerindeki numuneler, hassas skalalarla ölçülen sükrozun saf suyla karıştırılmasıyla elde edilir ve daha sonra her bir numunenin UV-Vis spektrumu ölçülür. Bakteri konsantrasyon çözeltileri için 6 farklı bakteri konsantrasyonu spektrumları kaydedilir. Bakteriler eklenmeden önce de aynı çözeltilerin spektrumları kaydedilir. Bu çözeltilerin her biri için, altın nanoparçacıkların büyütülmediği (dakika 0) ve 4 dakika büyütüldüğü, 10 dakika büyütüldüğü ve 12 dakika büyütüldüğü dört set hazırlanır. Veri seti hazırlığından sonra, bu spektrum ölçümleri sükroz konsantrasyonu veri seti ve bakteri çözeltisi veri seti olarak MATLAB ortamına aktarılır. Daha sonra bu veri setlerinden en bilgilendirici ve ayırt edici bilgilerin elde edilmesi için gerekli ön işleme adımları uygulanmaktadır. Ham ölçüm değerleri ve işlenmiş spektrum ölçümleri MATLAB Derin Öğrenme Araç Kutusu'ndaki yapay sinir ağları (YSA) ve MATLAB İstatistik ve Makine Öğrenmesi Araç Kutusu'ndaki Destek Vektör Makineleri (DVM) ile eğitilmiştir. Yapılan makine öğrenmesi deneylerinin sonuçları incelendiğinde, başarı oranı sükroz çözeltisindeki bakteri konsantrasyonlarının sınıflandırılması için ise çok yüksektir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Spectroscopy is a methodology for measuring the interaction between matter and electromagnetic radiation. One of the advantages of spectroscopy methodology is especially by choosing ultraviolet and visible electromagnetic spectrum, needed hardware for measurement becomes very cheap. Moreover, measurements can be performed in a very short time and frequently without any overhead. But the general problem with this methodology, it is not reliable to use in certain scientific and medical fields with its raw measurements because the obtained data is not exactly informative enough at first look. It is needed to process the raw measurements in order to obtain the most informative and distinguishing information from it by applying machine learning methodologies.

Two different datasets are collected with the help of localized surface plasmon resonance of immobilized golden nanoparticles. The plasmon resonance can be altered either by the refractive index of the solution or by absorption of some species to the nanoparticle surface. The resonance frequency of the metal nanostructures is sensitive to small variations in the refractive index that may occur in the near region of these nanostructures. This sensitivity makes plasmid nanomaterials attractive to molecular bioanalytical devices. In our study, a set of liquid solutions with different sucrose concentrations and another set of solutions with different bacteria concentrations are used with the golden nanoparticles. For each of these solutions, golden nanoparticles are grown for several minutes and separate spectra are recorded to measure the effect of particle size in estimation. The first dataset is varying low sucrose concentration solutions and their UV-Vis wavelength spectra of each of the samples are measured in different minutes. The second one is varying bacteria concentrations are prepared in golden particle solutions in order to expand the surface area of adhesion of bacteria and the UV-Vis wavelength spectra of each of the samples in well plate cells are measured in different minutes.

These collected datasets are used on different structures of shallow artificial neural networks (ANN) and support vector machine (SVM) to determine exact sucrose concentration and classify the correct bacteria solution in terms of with/without bacteria solution and low/high bacteria concentration.

1.1. Motivation

As it is mentioned in the introduction, by spectroscopy and especially targeting visible spectrum, measurement operation becomes cheaper and faster and also using golden nanoparticles techniques are increased the obtaining most informative measurements from solutions to determine a sucrose concentration or bacteria contamination. However, these measurements cannot be reliable at first look. To deal with such kind of problem, machine learning techniques and algorithms can be considered in order to extract more informative and representative data from the input and use it for getting better predictions.

Unlike standard spectrum measurements, datasets are collected by choosing UV-Vis wavelengths, the spectral range, and resolution(precision) feature of the sensor. This approach provides to have cheaper estimations for sucrose concentrations and predictions for a bacteria concentration class within the UV-Vis spectrum.

Using localized surface plasmon resonance of immobilized gold nanoparticles to obtain the datasets and especially for the bacteria experiments magnifying it by the golden particles techniques are a new approach to make spectrum measurements. Also, for these two datasets, It is the first time being processed and used in estimation. Throughout the thesis work, both the methodology of collection of the datasets and the possibility to make proper predictions by using these spectrum measurements are tested.

1.2. Thesis Goals and Contributions

This thesis aims to train Artificial Neural Networks and Support Vector Machine in order to make successful predictions through spectrum measurements of low sucrose concentrations and bacteria solutions. The overall contributions in this thesis work can be summarized as follows:

Ultraviolet-Visible wavelengths are used for spectrum measurements which gives the interaction between matter and electromagnetic radiation but within these wavelengths, a variety of hardware can be found for measurements and it would be cheaper and faster to make measurements.

We handle these two different spectroscopy datasets. Sucrose concentration estimation is modeled as a regression problem, whereas bacteria concentrations are classified. For these two problems, we train different structures of shallow Artificial Neural Networks (ANN) and Support Vector Machine (SVM).

We conduct our experiments with

i. raw measurements

ii. peak values of the spectroscopy measurements

iii. Principal Component Analysis (PCA) applied dataset

iv. Linear Discriminant Analysis (LDA) applied dataset

We face the curse of dimensionality problem on the bacteria solution dataset because of the insufficient number of samples during the implementation of LDA. That is why we apply preprocessing methods as

i. Linear Discriminant Analysis with pseudo-inverse (LDAP)

ii. First PCA then LDA implementation

We use these two extra methods to overcome the curse of dimensionality problem for bacteria solution dataset experiments.

1.3. Outline of the Thesis

This thesis is organized as follows. The next chapter provides a literature overview. Chapter 3 gives background information about spectroscopy, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Artificial Neural Networks (ANN) and Support Vector Machine (SVM). Chapter 4 includes sucrose concentration regression experiments by ANN and SVM with raw measurements and preprocessing methods. Chapter 5 includes bacteria solution classification experiments by ANN and SVM with preprocessing methods. Finally, Chapter 6 provides final remarks and discusses future research.

CHAPTER 2

RELATED WORK

In the past studies, determination of different types of concentrations like glucose concentration, sucrose concentration in solutions by using optical spectroscopy has been the subject of research for many years. With the increasing success of the machine learning algorithms, especially artificial neural network is used for estimation through spectroscopy measurements. Exemplary studies are usually on blood glucose estimation. Zeng et al. (2013) work on spectroscopy measurements at 1400-1800 nm(near infrared) wavelengths and artificial neural network (ANN) are used. Three distinct wavelengths from each of two different spectra, total 6 values are used in the input layer of ANN. Again, Trabelsi et al. (2012) studied on blood glucose level by preparing different blood samples and obtained measurements in the range of 1400-2500 nm. Manually selected 6 wavelengths from the spectrum are used to train ANN. By Chua et al. (2014), this time without using ANN method, the response of LED light with only 1450 nm wavelength is used for estimation of the glucose concentration in the blood.

In a more recent study, by Gulderen et al. (2016) near infrared (950-1100 nm range) wavelengths and MATLAB Neural Network Toolbox is used to determine glucose solution. 44 different glucose concentrations are measured and 36 of them are used for training and validation datasets, 8 of them are using for testing. In the ANN input layer, only the highest spectrum measurements are used. Estimation success is calculated as the real solution of the error in the ANN regression of concentration by relative error. In most unsuccessful case, the relative error is up to 18% for estimation. Introduced by Malik et al. (2016), the concentration of glucose in the urine samples are prepared, the spectrum obtained from the large 500 vectors of the input layer is diminished to 8 value with Principal Component Analysis (PCA). Conducted by Liu et al. (2008), glucose concentrations of 2100-2400 nm wavelength measurements are given as input. This time size of the dataset is not reduced. In this study, ANN method, least square regression and principal component regression methods are compared and ANN performance is reported to be higher. Ozbalci et al. (2013), a different spectrum called the Raman spectrum is used to determine the amounts of 4 different sugars (glucose, fructose, sucrose, maltose) in honey samples, and the success of partial least squares and ANN are compared. ANN input layer is determined as 4 which is reduced by PCA and single hidden layer structure with multiple numbers of neurons in them are tested as ANN structures. Also, Vítková et al. (2012) try to identify archaeological materials as biominerals by applying linear discriminant analysis (LDA) to spectra measurements obtained by stand-off laser-induced breakdown spectroscopy and artificial neural networks (ANN) are trained with them. By Mezgil et al. (2017), low sucrose concentrations are estimated in 400 nm and 800 nm spectrum with different shallow ANN structures.

Anker et al. (2008) propose, where the metallic nanoparticles are smaller than the wavelength of the incoming light, this electron cloud release is localized on the plasmon nanoparticle surface. Because of its optical properties, gold and silver nanoparticles are used for most localized surfaces. The plasmon resonance can be altered by either solvent refractive index or by adsorption of some species onto the nanoparticle surface. Also, in Martinsson et al. (2014), the plasmon resonance frequency of metal nanostructures is highly dependent on the dielectric properties surrounding the environment, which allows for the detection of small changes in the refractive index that may occur in the immediate vicinity of the nanostructures. The refractive index sensitivity makes plasmonic nanomaterials attractive as signal transducers in bioanalytical devices to monitor molecular binding events. According to these studies, golden nanoparticles are used to collect spectroscopy measurements of sucrose and bacteria concentrations.

CHAPTER 3

RESEARCH BACKGROUND

3.1. Spectroscopy

The study of the interaction between matter and electromagnetic radiation is called Spectroscopy. Also, the name of Optical Spectroscopy can be used. The light is some form of electromagnetic radiation that is a type of energy travels in waves. That is why every electromagnetic wave has a particular wavelength. Every wavelength has a different characteristic. These characteristics can be used for determination, discrimination or condition of the matter. Such as radio waves, microwaves, infrared, visible spectrum, ultraviolet, x-rays and gamma rays are the examples of radiation types as electromagnetic spectrum. The only part of the electromagnetic spectrum can be seen by the human eye is the visible spectrum whose wavelengths are between about 400 nm and 700 nm, Tkachenko (2006). The entire range of wavelengths of electromagnetic radiation can be seen on Figure 3.1. From longest wavelength to the shortest, radiation types are radio, microwave, infrared, visible, ultraviolet, X-ray, and gamma ray.

The interaction of the spectroscopy can happen as:

- Absorption spectroscopy
- Emission spectroscopy
- Reflection spectroscopy
- Impedance spectroscopy
- Resonance spectroscopy
- Inelastic scattering

Throughout this thesis, we deal with absorption spectroscopy. The spectroscopy interaction can be measured in nature by special spectroscopy sensors. An experimental setup of absorption spectroscopy can be seen on Figure 3.2.

3.2. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique to convert a set of correlated variables by using orthogonal transformation to find a new set of ranked dimensions. PCA considers the separability of features by looking at the variance of each feature because it is reasonable assumes that features that present high variance are more likely to have a good split between classes, by Jolliffe (2002). The basic steps of PCA:

1. Calculate the covariance matrix of the dataset without using output vector.
2. By using the calculated covariance matrix, calculate corresponding eigenvectors and eigenvalues.
3. Sort the calculated eigenvalues by decreasing order.
4. Select the number of new feature set and create a matrix from eigenvectors of selected number of biggest eigenvalues.
5. Perform the transformation of new dataset by multiplying the old dataset with the matrix formed from selected eigenvectors.

By this way new dimensions that have the biggest variance in terms of feature set can be constructed. If we look at the formulation, let X be the feature set of the dataset we want to be apply PCA. X has a set of p-dimensional vectors which have k number of samples:
$w_{(k)} = (w_1, ..., w_p)_{(k)}$
$t_{(i)} = (t_1, ..., t_l)_{(i)}$, given by $t_{k(i)} = X_i \cdot w_{(k)}$ for
$i = 1, ..., n$
$k = 1, ..., l$
$T = XW$ as W is a square matrix as $p\cdot$ and this matrix columns are eigenvectors of $X^T X$.

3.3. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a feature extraction technique like PCA which is a generalization of Fisher's linear discriminant. The main difference of this method is during the feature extraction, it also considers the dataset class information. This technique is especially useful for classification problems. Because LDA considers the distance between classes and within class samples (Izenman (2008)). On Figure 3.3, the difference in handling the variance selection of LDA and PCA can be seen separately. The basic steps of LDA can be defined as:

1. Calculate the mean vectors of each class of the dataset.
2. Calculate S_b(between classes) and S_w(within classes) scatter matrices.
3. By using the scatter matrices, calculate corresponding eigenvectors and eigenvalues.
4. Sort the calculated eigenvalues by decreasing order
5. Select the number of new feature set and create a matrix from eigenvectors of selected number of biggest eigenvalues
6. Perform the transformation of new dataset by multiplying the old dataset with the matrix formed from selected eigenvectors.

That is why S_b(between classes) and S_w(within classes) scatter matrices are calculated as following:

$$S_b = \sum_{i=1}^{g} N_i(\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})^T$$

$$S_w = \sum_{i=1}^{g}(N_i - 1)S_i = \sum_{i=1}^{g}\sum_{j=1}^{N_i}(x_{i,j} - \overline{x}_i)(x_{i,j} - \overline{x}_i)^T$$

We solve the generalized eigenvalue problem for the matrix $S_w^{-1}S_b$ to obtain the linear discriminants. This means that S_w is supposed to be invertible. But sometimes in the real-world, the number of samples can be lower than the sum of the number of features and number of classes in the dataset. This leads to the curse of dimensionality problem. This means S_w becomes a singular matrix. In order to beat this problem, a few options are:

- Elimination of features manually to overcome the curse of dimensionality problem
- Using applying pseudo-inverse to singular S_w matrix
- Apply PCA and reduce the feature set just enough to beat the curse of dimensionality problem

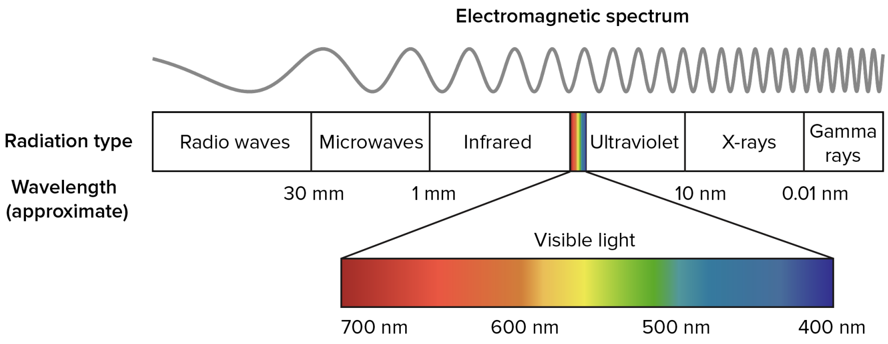


Figure 3.1. Electromagnetic spectrum and wavelengths of light, from Khan Academy (2019)

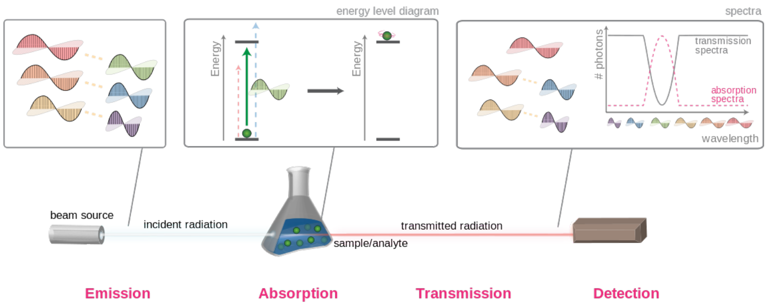


Figure 3.2. Experimental setup of absorption spectroscopy, from Wikipedia (2019)

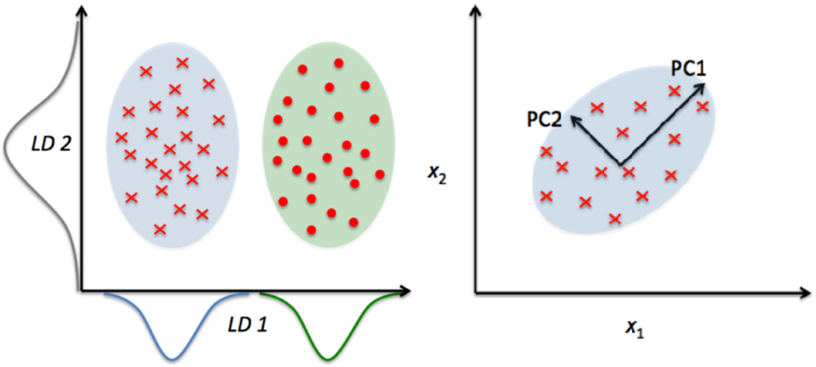


Figure 3.3. Difference of handling feature set for LDA and PCA, from Towards Data Science (2019a)

3.4. Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) is a machine learning technique which is inspired by the working principle of biological neural networks. ANN consists of a set of algorithms to interpret data through a kind of machine perception, labeling or clustering raw input. By training, it recognizes patterns as numerical values of vectors. Thus real-world data can be processes such as text, sound, an image with the condition of translation into numeric values.

Primitive implementation of a first neural network was a perceptron, which has one neuron only by accepting a different number of inputs and outputs one value. The basic structure and work principle can be seen on Figure 3.4. This neuron also has an activation function which contains a mathematical model to decide the output value according to the sum of products of input values and their weights. As each input value has a weight feature which determines the effectiveness of the input value.

A simple artificial neural network consists of three main layers as input layer, hidden layer, and output layer. This time instead of one perceptron, it has multiple neurons but it preserves the working logic. In recent neural networks, different types of activation functions can be used as sigmoid, hyperbolic tangent and rectified linear units functions. On Figure 3.5, an example of a shallow artificial neural network structure can be seen.

In order to obtain a pattern in ANN, through the dataset, we need to train it with a dataset. Just like the gradient descent algorithm also ANN can be trained by this. The training of ANN is possible with back-propagation algorithm which uses the loss function in order to update weights of the ANN till gradient descent algorithm requirements are done.

The general loss function: $L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y})^2$ as $\hat{y}$ is the output of ANN and y is the real value we want to get from ANN. By gradient descent, in each iteration of back-propagation, weights are updated by the derivative of the loss function as
$w_j = w_j - \alpha \cdot \partial \frac{L}{\partial w_j}$
α corresponds to learning rate which decides the gradient descent momentum.

3.5. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning technique which aims to build hyper planes to make classification or estimation about given labeled dataset, Wang (2005). For regression problems we will call it as Support Vector Regression (SVR). A basic example to show how SVM builds a classifier hyper planes can be seen on Figure 3.6. It is obvious in the figure, SVM builds its classifier hyper plane by choosing the hyper planes that have maximum margin in terms of groups separability. This SVM hyper plane functions are called as kernels. SVM can have Linear, Polynomial or Radial basis function kernel. As a linear kernel example for two class separation, let there are n number of samples as
$(\overrightarrow{x}_1, y_1), ..., (\overrightarrow{x}_n, y_n)$
where y_i indicates the class information as -1 or 1 and $\overrightarrow{x}_i$ is a two dimensional vector. SVM will have a classifier hyper plane as:
$\overrightarrow{w} \cdot \overrightarrow{x} - b = 0$
Also, it will have auxiliary hyper planes to help for determination of optimal hyper plane as
$\overrightarrow{w} \cdot \overrightarrow{x} - b = -1$
$\overrightarrow{w} \cdot \overrightarrow{x} - b = 1$
These auxiliary hyper planes helps to determine the sample class information as
$\overrightarrow{w} \cdot \overrightarrow{x}_i - b \leq -1, y_i = -1$
$\overrightarrow{w} \cdot \overrightarrow{x}_i - b \geq 1, y_i = 1$
Geometrically the distance between two hyper planes $\frac{2}{\|\overrightarrow{w}\|}$
By these two decision boundaries, we get:
$y_i(\overrightarrow{w} \cdot \overrightarrow{x}_i - b) \geq 1$ and $1 \leq i \leq n$
The objective of SVM becomes in order to maximize $\frac{2}{\|\overrightarrow{w}\|}$ minimize $\|\overrightarrow{w}\|$.

3.5.1. MATLAB Environment

MATLAB is a closed source programming language and computing development environment which is developed by MathWorks. It has variety of toolboxes which provide ready to use functions and applications. Especially Parallel Computing Toolbox, Deep Learning Toolbox and MATLAB Statistics and Machine Learning Toolbox are used for the experiments of this thesis on MATLAB 2018b. Deep Learning Toolbox is useful to create shallow neural networks or deep neural networks to perform regression or classification experiments and MATLAB Statistics and Machine Learning Toolbox is used to apply support vector machine (SVM) for classification experiments and support vector regression (SVR) for regression experiments, from MATLAB (2019a). To speed up the training time, Parallel Computing Toolbox is used which enables to benefit from GPU computation power, from MATLAB (2019b).

As we mention, MATLAB Deep Learning Toolbox is very useful to create shallow artificial neural networks. Also, it has a lot of options to determine the training function, neuron activation function, training function, performance function. All these options come with default ones. In this thesis we use hyperbolic tangent(tanh) transfer function option as activation function, scaled conjugate gradient backpropagation option which updates weight and bias values according to the scaled conjugate gradient method as a training function, mean squared normalized error option as performance function for regression and cross entropy option as performance function for classification.

For MATLAB Statistics and Machine Learning Toolbox comes with a lot of option to apply SVM and SVR algorithms. All these options come with default ones. In this thesis for SVM implementation, we use linear kernel function and for SVR implementation, we user polynomial kernel function with polynomial order of 3. Also, for SVM and SVR we set ”Standardize” flag as true which scales and centers each sample according to their weighted column mean and standard deviation.

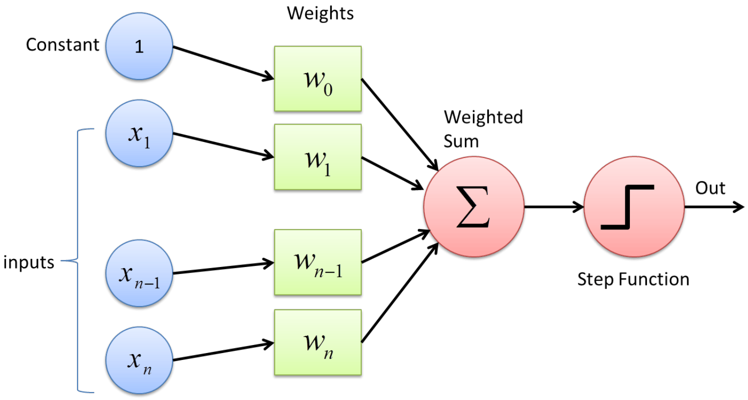


Figure 3.4. Perceptron structure, from Towards Data Science (2017)

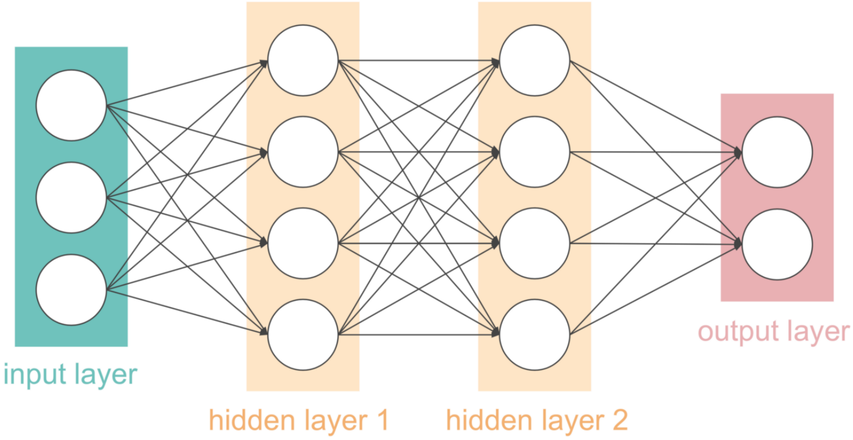


Figure 3.5. An example of a shallow artificial neural network structure, from Data Wow (2018)

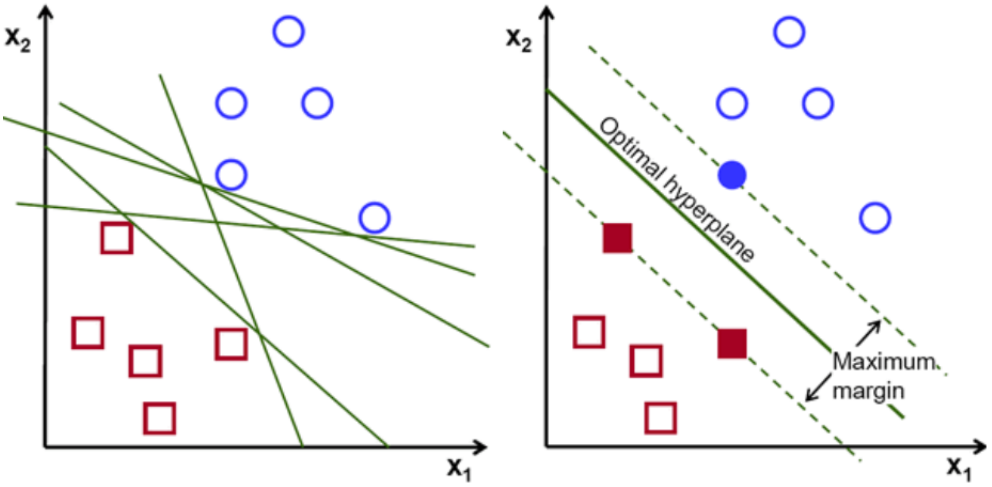


Figure 3.6. An example of support vector machine classification approach, from Towards Data Science (2019b)

CHAPTER 4

ESTIMATION OF SUCROSE CONCENTRATION

4.1. Sucrose Concentration Dataset

Thanks to IZTECH Department of Bioengineering, a different number of low sucrose concentrations in solutions with ultrapure water is prepared in well plate cells. These low sucrose concentration solutions are determined by mass as 0%, 10% (100mg/mL), 20% (200mg/mL), 30% (300mg/mL), 40% (400mg/mL), 50% (500mg/mL). The solid sucrose samples are weighed using analytical balance dissolved in ultrapure water (conductivity = $18M\Omega$) with the help of 100 mL measure and made into stock solutions. After that, each well plate is measured by Ultraviolet-Visible(UV-Vis) spectroscopy with the help of localized surface plasmon resonance of immobilized gold nanoparticles. The selected wavelengths are between 300 nm and 800 nm with a precision of 1 nm. This means for each prepared sucrose concentration in a well plate cell, we got 501 different measurements. All these measurements are performed at different times as minute 4, minute 5, minute 6, minute 7, minute 8, minute 9, minute 10, minute 11, minute 12 and minute 13. These times are the durations of gold growth in solution. The selected wavelengths and measurement times are determined by IZTECH Department of Bioengineering to determine which is, within these spectra, the most distinguishing and informative spectrum values. We investigate this in this thesis by machine learning methods.

After all the experiment setups and measurements, we get our sucrose dataset number of samples as seen Table 4.1. Each sample contains 501 different measurements which mean 501 different features. In order to examine the sucrose solution dataset, spectroscopy measurement values at each wavelength are drawn with different colors for each different sucrose concentration.

As seen in Figures 4.1, 4.2 and 4.3, determining the sucrose concentration by the specific minute of spectroscopy measurements is not possible by the naked eye because of all lines overlapping with each other. Especially for some minutes, overlapping lines rates are less than the other minutes. But it is clear that for all minutes nearly after 600 nm wavelength, sucrose concentrations start to spread. This means measurements at specific wavelengths can be more informative to distinguish sucrose concentration.

In Figures 4.1, 4.2 and 4.3 graphs, we try to explain the discrimination of sucrose concentrations. But actually, it is not a classification problem because in the real world challenges, it is intended to predict a scalar value for sucrose concentration, for example, blood glucose level problem. So it becomes a regression problem for us.

First, we try to solve this problem by previous researches approaches like using entire spectrum values or using the peak value of spectrum measurements. Then we investigate different feature extraction techniques to build a system to get the most informative and distinguishing values from these features.

Table 4.1. Sucrose concentration dataset number of samples for each minute

	0% (0mg/mL)	10% (100mg/mL)	20% (200mg/mL)	30% (300mg/mL)	40% (400mg/mL)	50% (500mg/mL)	TOTAL
Minute 4	141	141	141	141	141	141	846
Minute 5	141	141	141	141	141	141	846
Minute 6	141	141	141	127	141	141	832
Minute 7	141	141	141	87	132	141	783
Minute 8	152	141	141	141	181	141	904
Minute 9	141	141	141	141	188	141	893
Minute 10	141	141	123	141	188	141	875
Minute 11	134	141	94	141	188	141	839
Minute 12	141	135	152	141	47	141	757
Minute 13	141	94	141	141	47	141	705

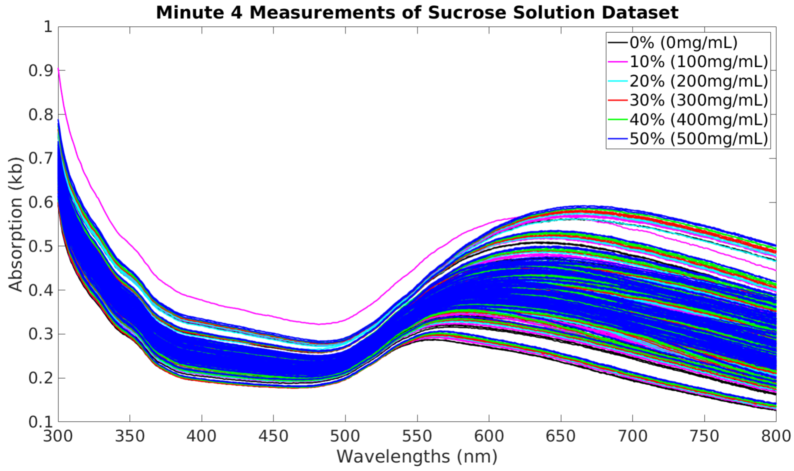


Figure 4.1. Sucrose Concentrations Measurements at Minute 4 Graph

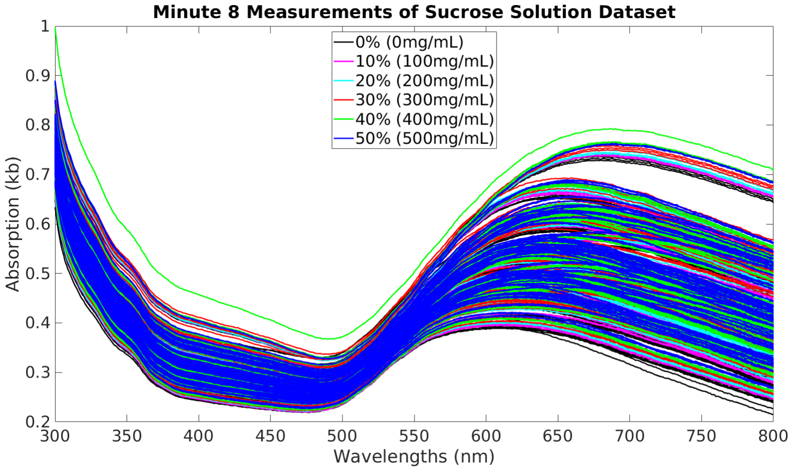


Figure 4.2. Sucrose Concentrations Measurements at Minute 8 Graph

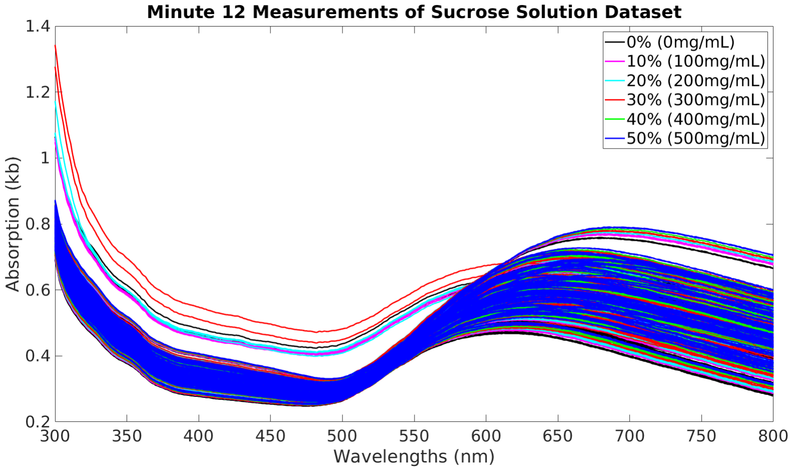


Figure 4.3. Sucrose Concentrations Measurements at Minute 12 Graph

4.2. Sucrose Concentration Experiments

To solve the sucrose concentration regression problem, we treat separately each minute samples between minute 4 and minute 13. Because of its popularity and usage in previous researches like Zeng et al. (2013) and Trabelsi et al. (2012), we want to use feed-forward shallow Artificial Neural Network (ANN) method. Thanks to MATLAB Deep Learning Toolbox, we use its environment to create and train multiple different structures of ANNs. For ANN activation function in each neuron, hyperbolic tangent(tanh) transfer function is used. As training function, scaled conjugate gradient backpropagation option is selected which updates weight and bias values according to the scaled conjugate gradient method. For ANN performance function, Mean squared normalized error performance function is used. Regularization parameter can be set to any value between 0 and 1. So 0.7 is decided by trial and error. Also in each ANN structure, input values of input neurons and output values of output neuron values are normalized with mapminmax option which normalizes the minimum and maximum values between -1 and 1 accordingly. Also in order to compare the results wit different machine learning technique, Support Vector Regression on MATLAB environment with MATLAB Statistics and Machine Learning Toolbox is used. As a kernel function polynomial kernel is selected by trial and error. Polynomial kernel function order is determined as 3 which is the default value. But again with trial and error, the final decision is the keep the default option. In addition, input values are scaled by the corresponding weighted column mean and standard deviation by passing ”Standardize” parameter to the related training function.

In the sucrose dataset, we have 501 different features. But also with the help of different preprocessing methods given in 4.2.1, the feature set size is changed. That is why the number of inputs for each machine learning methods are changed accordingly.

For the ANN hidden layer, we want to try a different number of hidden layers and neurons to observe the hidden layer structure effect on the regression success. These hidden layer types consist of two main structures as one layer hidden layer structure and two layers hidden layer structure. These hidden layer structures can be seen in Tables 4.2 and 4.3.

Before starting training of each different structures of ANNs, the separate minute measurements of the dataset are divided into 3 parts as 60% for the training set, 20% for the validation set and 20% for the test set. Also the separate minute measurements of the dataset are divided into 2 parts as 80% for the training set, 20% for the test set during the SVR experiments because SVM training does not consistent of multiple different structures. During this division, the samples are shuffled by supporting that each dataset group had an almost equal ratio of different samples as 0%, 10% (100mg/mL), 20% (200mg/mL), 30% (300mg/mL), 40% (400mg/mL), 50% (500mg/mL). In some cases, this rule is slightly broken because the total number of sucrose concentration samples are changing for each minute can be seen in Table 4.1. This dataset division operation is conducted at the beginning of each machine learning training.

Table 4.2. ANN Hidden Layer Structures with 1 Hidden Layer

Number of Neurons in the Hidden Layer
5
10
15

Table 4.3. ANN Hidden Layer Structures with 2 Hidden Layers

Number of Neurons in the First Hidden Layer	Number of Neurons in the Second Hidden Layer
5	5
5	10
10	10

4.2.1. Dataset Preprocessing

As it is mentioned above, before the regression experiments, based on the previous researches like Gulderen et al. (2016) and Malik et al. (2016), sucrose concentration dataset is approached with different techniques since we try to obtain the most informative and distinguishing values from the sucrose concentration feature set. Also applying the same regression algorithm on these preprocessed datasets, we intend to compare the effectiveness of these methods on the spectroscopy datasets.

4.2.1.1. Using Entire Feature Set

The first option is using all spectroscopy wavelength measurements without using any feature extraction technique. The feature set consists of 501 wavelength measurements between 300 nm and 800 nm with a precision of 1 nm. They are treated as each of them are distinct and meaningful features to predict sucrose concentration. Thus for the sucrose concentration dataset, we tried to make sucrose concentration estimations by using these 501 different spectroscopy measurement values. Which means, one sample consisted of 501 input values as spectroscopy measurements and one output value as the sucrose concentration.

4.2.1.2. Using Peak Values

As it is performed in Gulderen et al. (2016), for each sample the biggest spectroscopy measurement (peak value) from 501 different spectroscopy measurements between 300 nm and 800 nm. is selected. For the sucrose concentration dataset, we try to make sucrose concentration estimations by using this one wavelength value. For each sample, the wavelength that gives the highest absorption is selected. Thus in the new dataset, one sample consists of one input value as the biggest spectroscopy measurement and one output value as the sucrose concentration.

4.2.1.3. Applying Principal Component Analysis

Dimensionality reduction is one of the essential tasks in machine learning. It not only solves the curse of dimensionality problem (as we will face in Chapter 5) but it also eliminate non-discriminative features which may deteriorate the estimation/classification performance (Baştanlar and Özuysal (2014)). Thus in our sucrose dataset, we applied reduction to 501 features. Instead of direct elimination of features, we benefited from a feature extraction technique called Principal Component Analysis (PCA) similar to studies of Malik et al. (2016) and Ozbalci et al. (2013).

Just after the dataset division operation on sucrose solution dataset as 3 parts as 60% for the training set, 20% for the validation set and 20% for the test set during the ANN experiments. Also, the dataset division operation on sucrose solution dataset is conducted as 2 parts as 80% for the training set, 20% for the test set during the SVR experiments because SVR training does not consistent of multiple different structures. PCA is performed on training and validation set only. Because we do not want our machine learning algorithm to receive any preliminary information from test samples.

For each minute samples in sucrose solution dataset, PCA is applied as explained above and for 501 feature, 501 different eigenvalue and eigenvectors are calculated. These eigenvalues are ordered by their value because bigger eigenvalue indicates a bigger effect on the whole dataset. According to the sum of total eigenvalues, first 10 biggest eigenvalues give us over than 99.99% of variation in the dataset. Thus the eigenvectors of the first 10 biggest eigenvalues are used to convert the dataset from 501 features to 10 features.

4.2.1.4. Applying Linear Discriminant Analysis

Again we continue with that some of the features can mislead the sucrose concentration estimation idea. This time we apply Linear Discriminant Analysis (LDA) which is used before on spectroscopy measurements in the study of Vítková et al. (2012). LDA is differentiating from PCA by also preserving sample class information through forming the new features. That is why we expect better results from it comparing the PCA results.

Just after the dataset division operation on sucrose solution dataset, test set samples are not included in LDA computation just like the 4.2.1.3 implementation.

For each minute samples in sucrose solution dataset, LDA is applied as explained above and for 501 feature, 501 different eigenvalue and eigenvectors are calculated. These eigenvalues are ordered by their value because bigger eigenvalue indicates a bigger effect on the whole dataset. According to the sum of total eigenvalues, first 10 biggest eigenvalues give us over than 99.99% of variation in the dataset. Thus the eigenvectors of the first 10 biggest eigenvalues are used to convert the dataset from 501 features to 10 features.

4.2.2. Artificial Neural Network (ANN) Training Results

The different number of ANN structures are trained 100 times from scratch. The number of total experiments for each ANN structure is determined to eliminate the chance factor. For the determination of the error of the current trained ANN structure, mean absolute error (MAE) and root mean square error (RMSE) of regression results of the test set are used. Mean of 100 MAE and 100 RMSE from each training is used to evaluate an ANN structure for a specific minute dataset. MAE is used to determine ANN error to clearly indicate how trained ANN is predicting with fault. RMSE is used as an alternative to indicate large errors in test set to show the steadiness of MAE. Thanks to these two error metrics, we are able to compare the reliability of training results. For each minute, these dataset preparation and training operations are conducted separately. Their results can be seen as MAE in Table 4.4 and RMSE in Table 4.5. Also the standard deviation of 100 MAE and 100 RMSE are calculated and shown in these tables. A mean of MAE of 4.14 refers to that the estimated concentration has an offset of 4.14% with respect to the actual concentration (which is between 0% and 50%).

The duration of ANN regression experiments are measured in terms of seconds. Minute 4 dataset experiments with two hidden layers which has 10 neurons in each layer structure duration measurements can be seen in Table 4.6.

Performance graphs of minute 4 ANN training examples can be seen on Figures 4.4, 4.5, 4.6 and 4.7. In these performance graphs, we expect that validation and train mean squared error (MSE) values are reduced faster then test MSE value in each epoch.

Regression graphs of minute 4 ANN training examples can be seen on Figures 4.8, 4.9, 4.10 and 4.11. In these regression graphs, we mostly focused on the determination of test sample results since we evaluate the preprocessing methods and ANN hidden layer structures according to mean of the test set mean absolute errors. In the graphs, we expect that the line of crossing the samples fit exactly so the R value will be 1 in the perfect case. As a general overview of the results, we see that from using entire feature set and using peak values methods to applying PCA and LDA in order, the crossing line in the test sample result graphs become closer and closer to 1 as line fitting value of R.

Table 4.4. The regression results of sucrose concentration dataset. MAE of 100 training with their ANN hidden layer structures are shown as mean / standard deviation.

	Entire Feature Set	Hidden Layer Structure	Peak Values	Hidden Layer Structure	PCA	Hidden Layer Structure	LDA	Hidden Layer Structure
Minute 4	8.28 / 1.35	5 - 10	14.70 / 0.22	10 - 10	6.00 / 0.39	10 - 10	5.14 / 0.32	5 - 5
Minute 5	7.63 / 1.73	15	14.44 / 0.20	10 - 10	6.21 / 0.41	10 - 10	4.82 / 0.33	5 - 10
Minute 6	8.53 / 1.38	5	14.53 / 0.20	10 - 10	4.67 / 0.29	10 - 10	5.13 / 0.31	5 - 5
Minute 7	9.62 / 1.44	10	15.47 / 0.16	10 - 10	5.47 / 0.37	10 - 10	5.54 / 0.40	5 - 5
Minute 8	8.82 / 0.81	10	15.06 / 0.08	10 - 10	6.59 / 0.46	10 - 10	4.31 / 0.28	5 - 5
Minute 9	8.85 / 2.07	15	14.79 / 0.13	10 - 10	4.38 / 0.28	10 - 10	4.14 / 0.26	5 - 5
Minute 10	9.07 / 1.41	15	14.35 / 0.18	10 - 10	5.07 / 0.32	10 - 10	5.19 / 0.36	5 - 5
Minute 11	8.04 / 1.31	10 - 10	14.84 / 0.22	10 - 10	5.69 / 0.33	10 - 10	5.05 / 0.31	5 - 5
Minute 12	8.81 / 1.30	10	14.63 / 0.01	10 - 10	5.15 / 0.30	10 - 10	6.09 / 0.43	5
Minute 13	7.98 / 1.70	10	14.47 / 0.22	10 - 10	4.89 / 0.31	10 - 10	7.18 / 0.62	5

Table 4.5. The regression results of sucrose concentration dataset. RMSE of 100 training with their ANN hidden layer structures are shown as mean / standard deviation.

	Entire Feature Set	Hidden Layer Structure	Peak Values	Hidden Layer Structure	PCA	Hidden Layer Structure	LDA	Hidden Layer Structure
Minute 4	10.36 / 1.59	5 - 10	16.79 / 0.21	10 - 10	7.77 / 0.54	10 - 10	6.50 / 0.42	5 - 5
Minute 5	9.50 / 1.99	15	16.60 / 0.20	10 - 10	7.93 / 0.60	10 - 10	6.15 / 0.42	5 - 10
Minute 6	10.51 / 1.52	5	16.65 / 0.24	10 - 10	5.80 / 0.34	10 - 10	6.46 / 0.39	5 - 5
Minute 7	11.77 / 1.57	10	17.46 / 0.15	10 - 10	6.95 / 0.42	10 - 10	6.96 / 0.50	5 - 5
Minute 8	10.88 / 0.86	10	17.05 / 0.06	10 - 10	8.17 / 0.55	10 - 10	5.46 / 0.33	5 - 5
Minute 9	10.76 / 2.38	15	16.81 / 0.11	10 - 10	5.49 / 0.34	10 - 10	5.24 / 0.31	5 - 5
Minute 10	11.15 / 1.53	15	16.43 / 0.21	10 - 10	6.42 / 0.36	10 - 10	6.52 / 0.43	5 - 5
Minute 11	10.11 / 1.54	10 - 10	16.84 / 0.20	10 - 10	7.25 / 0.40	10 - 10	6.31 / 0.36	5 - 5
Minute 12	11.06 / 1.46	10	17.13 / 0.01	10 - 10	6.48 / 0.41	10 - 10	7.70 / 0.53	5
Minute 13	10.07 / 2.17	10	17.03 / 0.20	10 - 10	6.25 / 0.58	10 - 10	9.02 / 0.77	5

Table 4.6. Total 100 ANN training duration and approximate duration of each ANN training in terms of seconds of different preprocessing methods on Minute 4 dataset with two hidden layers structure while each layer has 10 neurons

	Entire Feature Set	Peak Values	PCA	LDA
Total Duration of 100 Training	68.60	45.63	53.19	85.28
Approximate Duration of Each Training	0.68	0.45	0.53	0.85

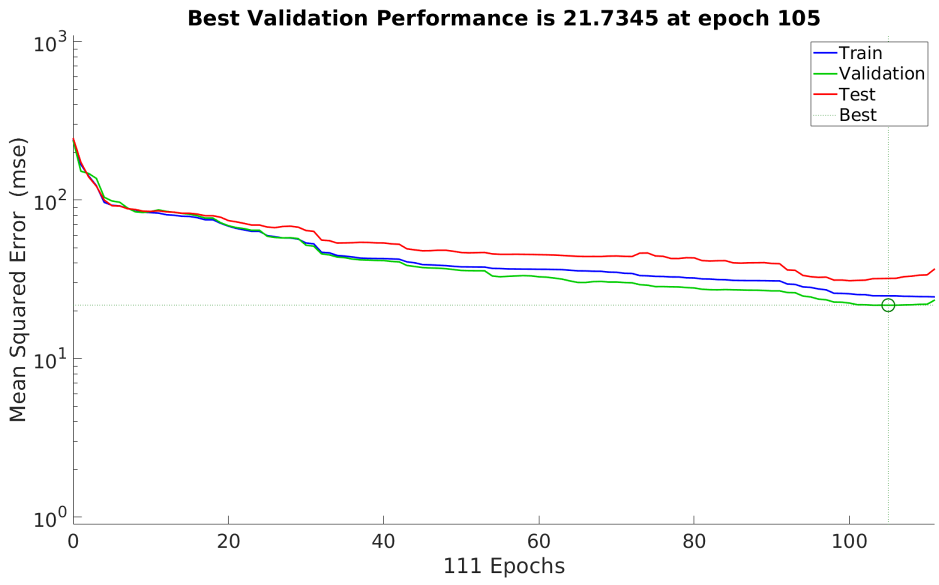


Figure 4.4. Performance graph of a sucrose concentration regression experiment by using entire feature set on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

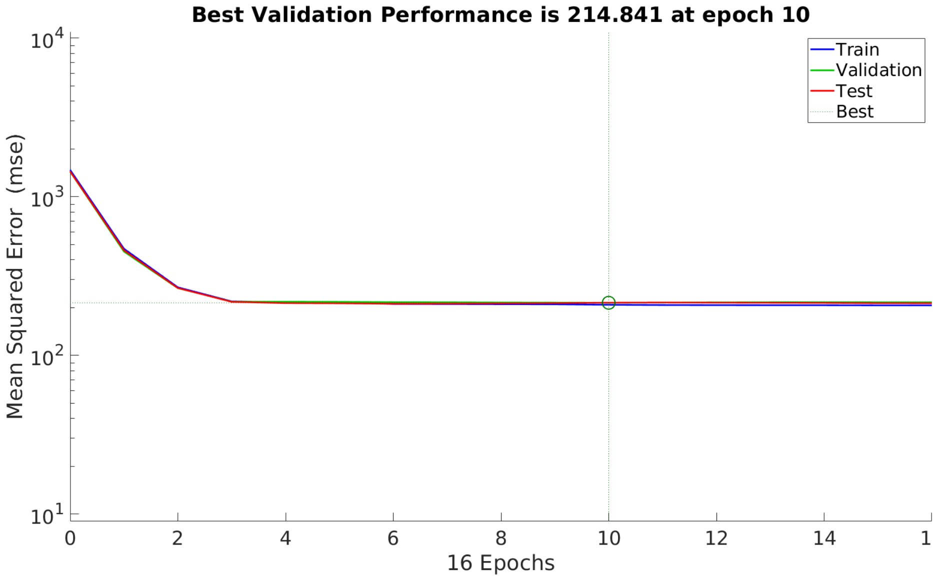


Figure 4.5. Performance graph of a sucrose concentration regression experiment by using peak values on minute 4 samples with ANN structure of a single hidden layer with 15 neurons

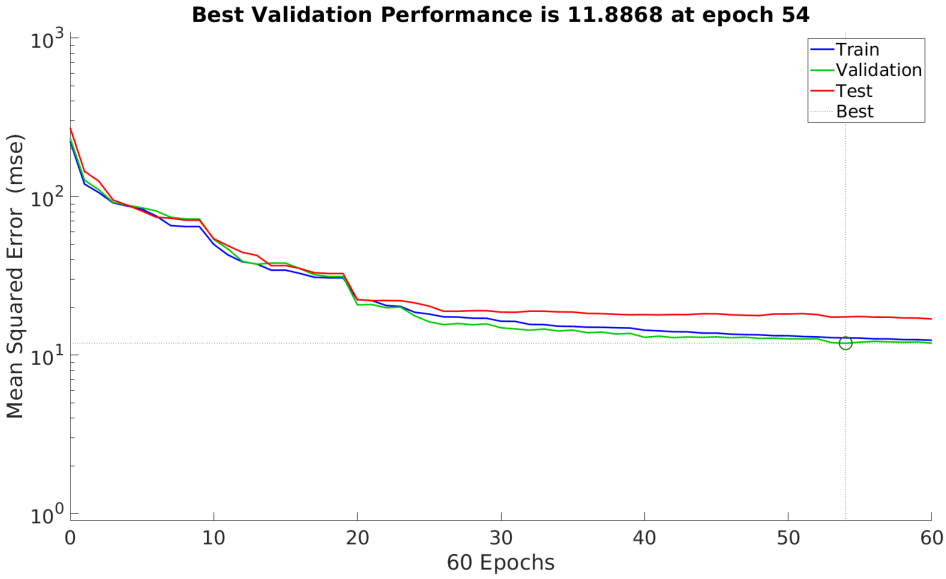


Figure 4.6. Performance graph of a sucrose concentration regression experiment by applying PCA on minute 4 samples with ANN structure of double hidden layers with 10 neurons in each layer

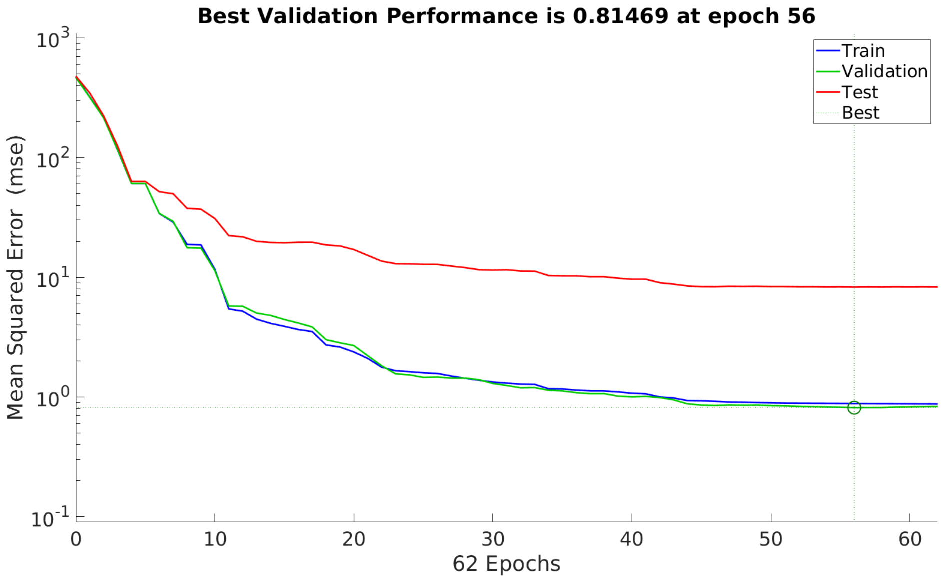


Figure 4.7. Performance graph of a sucrose concentration regression experiment by applying LDA on minute 4 samples with ANN structure of double hidden layers with 10 neurons in each layer

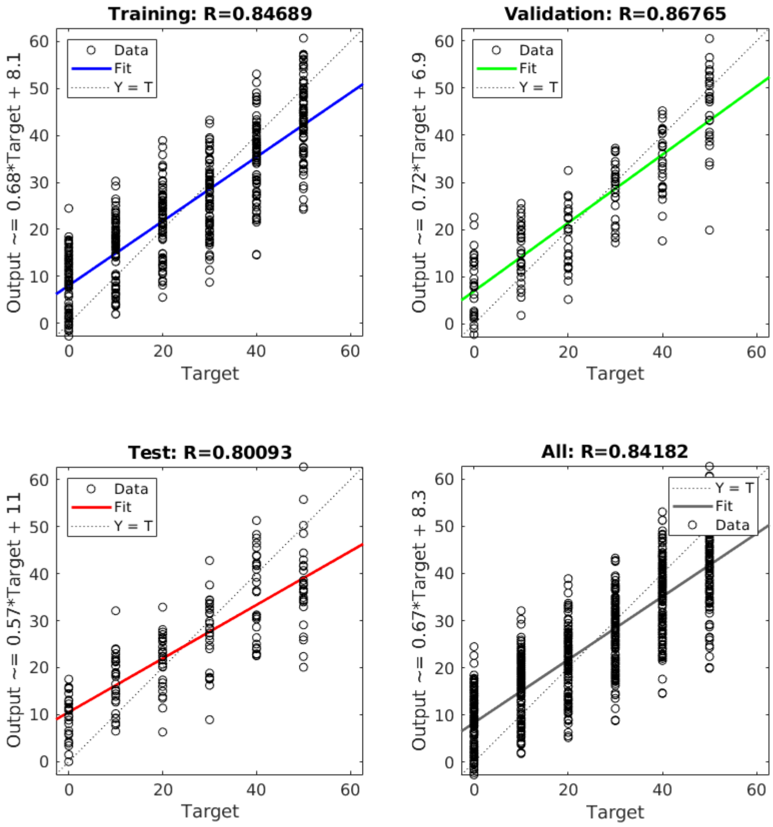


Figure 4.8. Regression results graph of a sucrose concentration regression experiment by using entire feature set on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

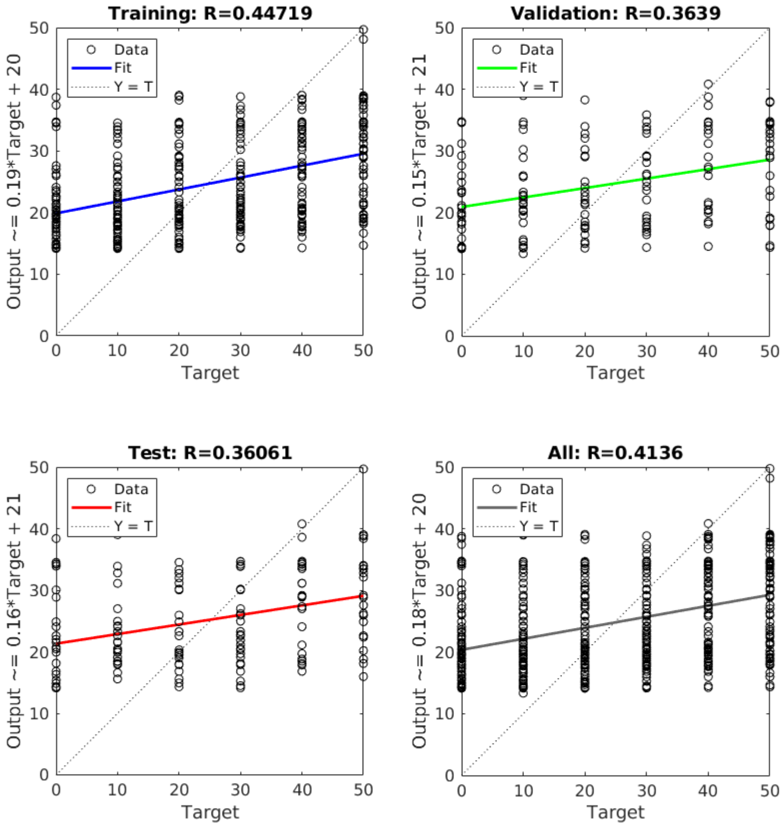


Figure 4.9. Regression results graph of a sucrose concentration regression experiment by using peak values on minute 4 samples with ANN structure of a single hidden layer with 15 neurons

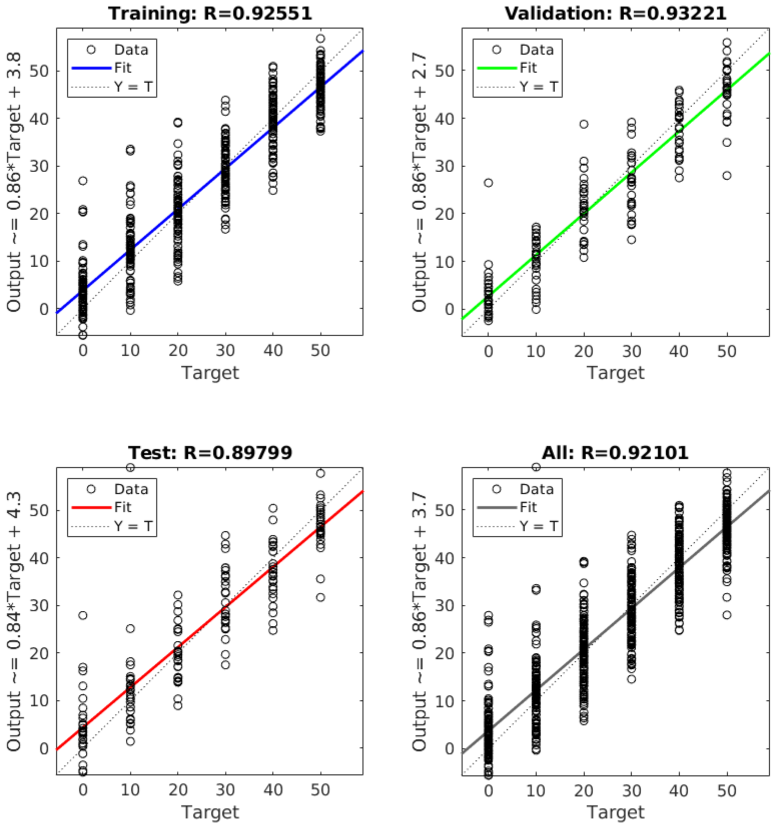


Figure 4.10. Regression results graph of a sucrose concentration regression experiment by applying PCA on minute 4 samples with ANN structure of double hidden layers with 10 neurons in each layer

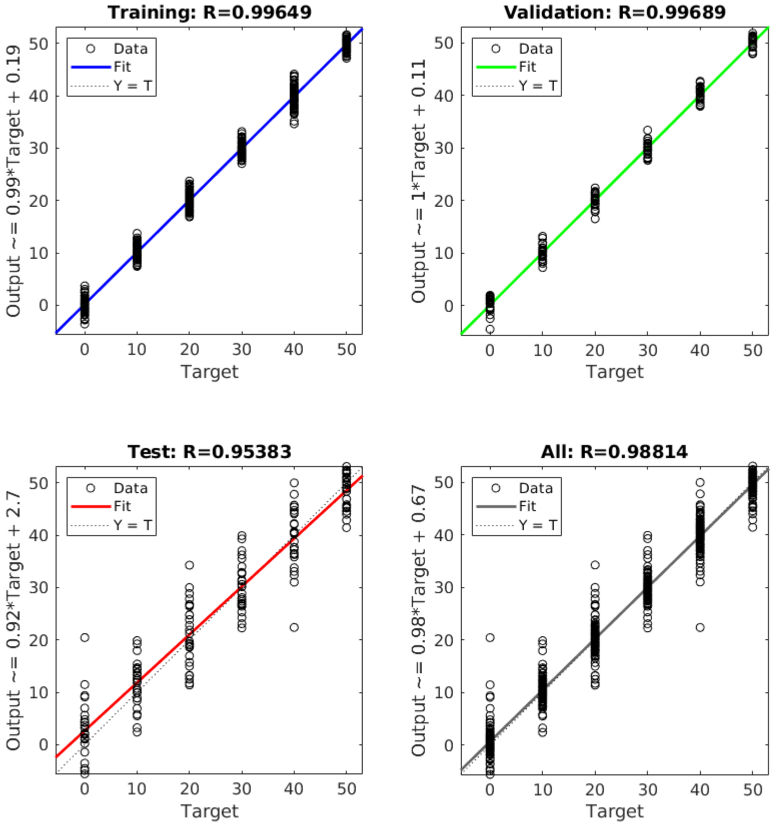


Figure 4.11. Regression results graph of a sucrose concentration regression experiment by applying LDA on minute 4 samples with ANN structure of double hidden layers with 10 neurons in each layer

4.2.3. Support Vector Regression (SVR) Training Results

SVR with polynomial kernel is trained 100 times from scratch. The number of total experiments is determined to eliminate the chance factor. For the determination of the error of the trained SVR, mean absolute error (MAE) and root mean square error (RMSE) of regression results of the test set are used. Mean of 100 MAE and 100 RMSE from each training is used to evaluation of a specific minute dataset. Thanks to these two error metrics, we are able to compare the reliability of training results. For each minute, these dataset preparation and training operations are conducted separately. Their results can be seen as MAE in Table 4.7 and RMSE in Table 4.8. Also the standard deviation of 100 MAE and 100 RMSE are calculated and shown in these tables. A mean error of 4.18 refers to that the estimated concentration has an offset of 4.18% with respect to the actual concentration (which is between 0% and 50%).

The duration of SVR experiments are measured in terms of seconds. Minute 4 dataset experiments duration measurements can be seen in Table 4.9.

Regression graphs of minute 4 SVR training examples can be seen on Figures 4.12, 4.13, 4.14 and 4.15. In these SVR regression graphs, we mostly focused on the determination of test sample results since we evaluate the preprocessing methods according to mean of the test set mean absolute errors. In the graphs, we expect that the line of crossing the samples fit exactly so the R value will be 1 in the perfect case. As a general overview of the results, we see that from using entire feature set and using peak values methods to applying PCA and LDA in order, the crossing line in the test sample result graphs become closer and closer to 1 as line fitting value of R.

Table 4.7. The regression results of sucrose concentration dataset. MAE of 100 training with SVR are shown as mean / standard deviation.

	Entire Feature Set	Peak Values	PCA	LDA
Minute 4	495.26 / 566.52	13.91 / 0.38	5.21 / 4.88	5.32 / 0.61
Minute 5	57.60 / 29.29	14.22 / 0.30	3.30 / 0.27	5.20 / 0.34
Minute 6	226.41 / 257.21	14.15 / 0.45	3.24 / 0.39	5.20 / 0.39
Minute 7	273.53 / 317.19	14.86 / 0.41	3.99 / 1.25	6.50 / 0.50
Minute 8	167.28 / 180.98	14.31 / 0.33	5.14 / 2.97	4.18 / 0.31
Minute 9	34.30 / 11.16	14.38 / 0.30	3.34 / 0.29	4.48 / 0.32
Minute 10	88.17 / 54.01	13.53 / 0.40	3.07 / 0.41	5.18 / 0.38
Minute 11	91.17 / 75.46	14.34 / 0.38	4.07 / 1.79	5.35 / 0.44
Minute 12	322.09 / 448.25	14.07 / 0.31	6.09 / 4.38	6.09 / 0.73
Minute 13	54.15 / 39.23	14.33 / 0.39	5.06 / 5.81	8.36 / 0.79

Table 4.8. The regression results of sucrose concentration dataset. RMSE of 100 training with SVR are shown as mean / standard deviation.

	Entire Feature Set	Peak Values	PCA	LDA
Minute 4	879.89 / 865.78	16.43 / 0.64	28.14 / 61.07	7.39 / 2.63
Minute 5	105.79 / 41.77	16.64 / 0.36	4.76 / 0.96	6.81 / 0.51
Minute 6	408.94 / 325.80	16.65 / 0.55	5.08 / 3.07	6.85 / 0.59
Minute 7	624.15 / 473.91	17.27 / 0.49	10.88 / 10.65	8.59 / 0.74
Minute 8	273.91 / 202.65	16.71 / 0.36	19.60 / 33.12	5.51 / 0.56
Minute 9	60.43 / 21.72	16.66 / 0.31	5.16 / 1.28	5.92 / 0.50
Minute 10	152.45 / 85.06	16.09 / 0.42	5.55 / 3.61	6.89 / 0.55
Minute 11	165.21 / 128.96	16.78 / 0.42	13.12 / 20.01	7.10 / 0.56
Minute 12	968.37 / 1140.77	16.70 / 0.35	27.74 / 37.04	8.74 / 2.03
Minute 13	96.61 / 70.74	16.96 / 0.44	17.01 / 47.82	11.50 / 1.31

Table 4.9. Total 100 SVR training duration and approximate duration of each SVR training in terms of seconds of different preprocessing methods on Minute 4 dataset

	Entire Feature Set	Peak Values	PCA	LDA
Total Duration of 100 Training	2174.70	3.56	697.69	134.24
Approximate Duration of Each Training	21.74	0.03	6.97	1.34

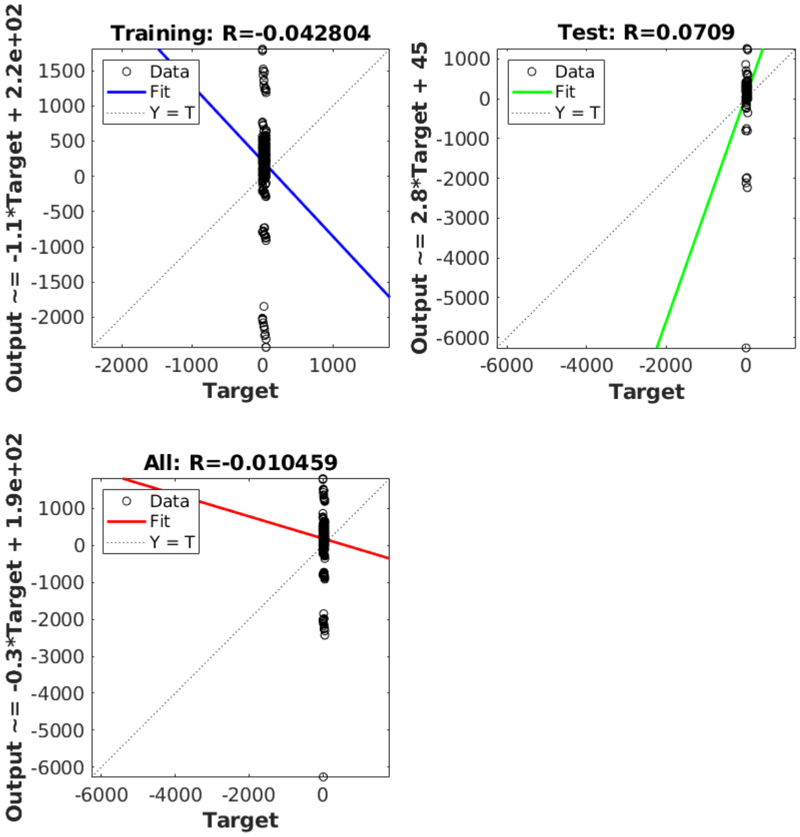


Figure 4.12. Regression results graph of a sucrose concentration regression experiment by using entire feature set on minute 4 samples with SVR

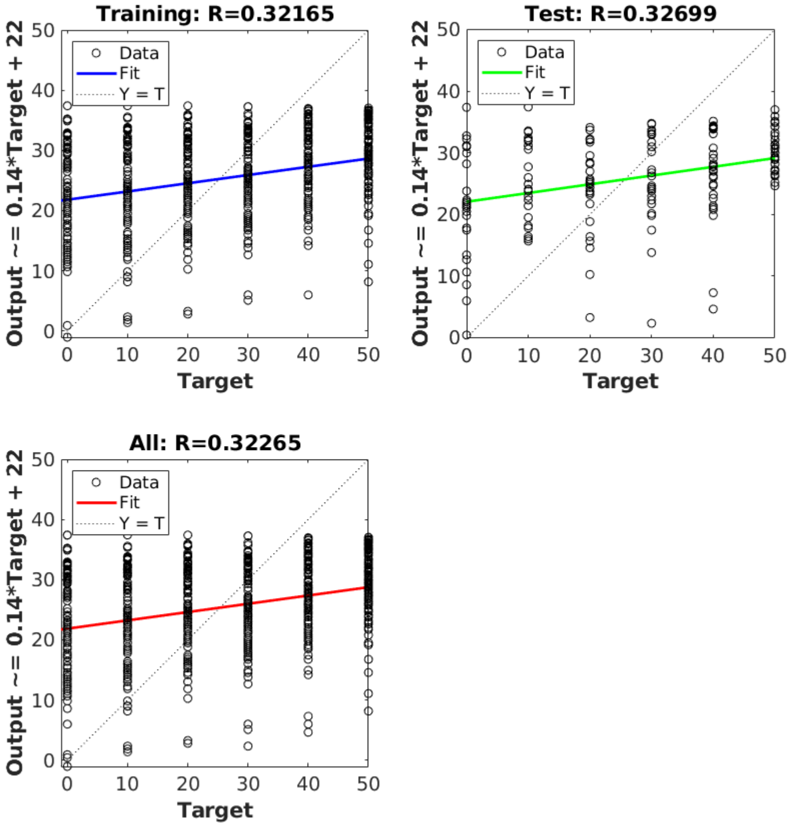


Figure 4.13. Regression results graph of a sucrose concentration regression experiment by using peak values on minute 4 samples with SVR

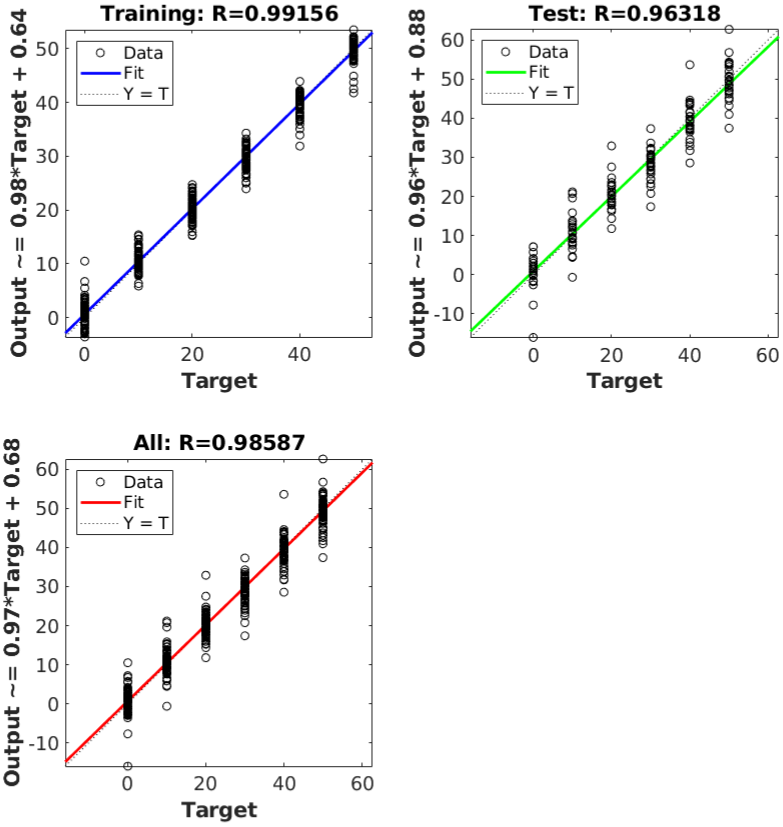


Figure 4.14. Regression results graph of a sucrose concentration regression experiment by applying PCA on minute 4 samples with SVR

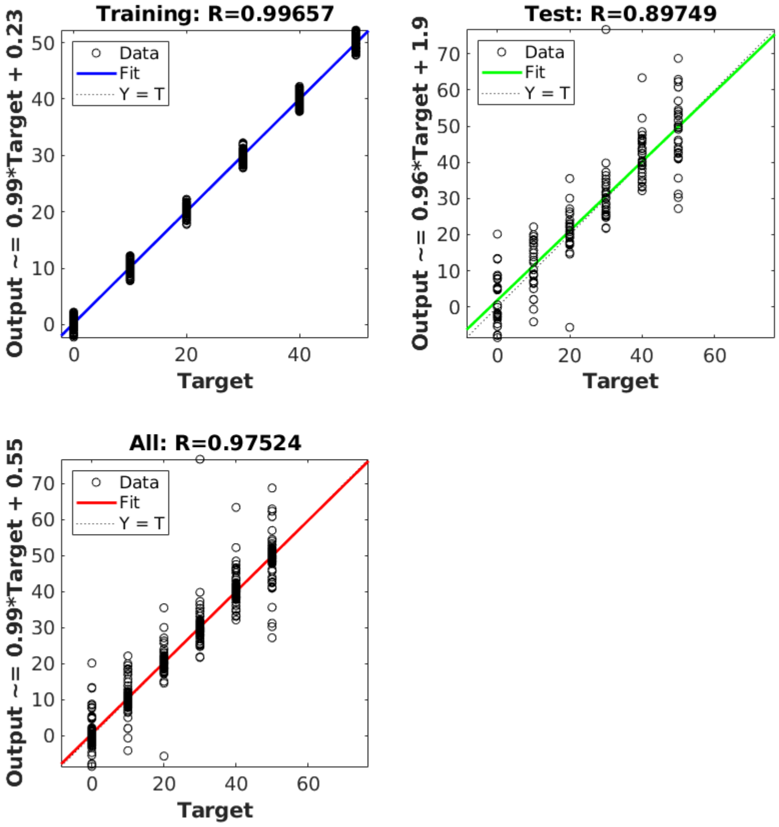


Figure 4.15. Regression results graph of a sucrose concentration regression experiment by applying LDA on minute 4 samples with SVR

4.2.4. Discussion and Conclusion

After performing all experiments with ANNs and SVR on raw and preprocessed sucrose concentration dataset, we obtain 100 MAE and 100 RMSE of each trained ANN structure and SVR separately. Then we calculate their means and standard deviation in order to evaluate results effectively.

The minimum number of error values of each minute with its ANN structure are listed as MAE in Table 4.4 and RMSE in Table 4.5. When we look at the ANN regression results, it is clear that we get the best results from LDA preprocessing method and PCA implementation results are almost close to that. Although it is a frequently used approach in the previous studies, using the peak values method give us the worst results among the others. Even using the entire feature set method is better than using peak values method but it is not sufficient as its approximate means of MAE changes between 9.50 mg/mL and 11.77 mg/mL for ANN experiments.

Among the sucrose concentration experiments, we get the best ANN result as MAE of 4.14 mg/mL and as RMSE of 5.24 mg/mL from LDA implementation of minute 9 samples on two hidden layers structure which has 5 neurons in each layer. 4.14 mg/mL error corresponds to predicting a 30 mg/mL sucrose solution as 25.86 mg/mL or 34.14 mg/mL. We get the worst ANN result as MAE of 15.47 mg/mL and RMSE of 17.46 mg/mL from using peak values method of minute 7 samples on multiple hidden layers structure which has 10 neurons in the first layer and 10 neurons in the second layer.

Among all different ANN hidden layer structures, two hidden layers structure which has 10 neurons or 5 nuerons in each layer generally gives us the best results. We also observe that the other hidden layer structure results are very close to each other in terms of the mean of MAE and RMSE. Also we get the best and worst results from the same number of hidden layer structures which is two hidden layers structure. That is why we can not say there is a direct benefit from the hidden layer structure complexity to get better results. The main effect is supported by preprocessing steps by especially PCA and LDA. We get the best results of LDA and PCA implementation from minute 9 samples. But for the other preprocessing methods, it could not accomplished by minute 9 samples. That is why, we can not say that some minute samples are superior to others by giving the extra information about the sucrose concentration. In addition, it is clear that each preprocessing method with ANN training give us reliable test results by looking at their standard deviation values. Because generally, they are changing between 0.01 and 2.07

for MAE, 0.01 and 2.38 for RMSE.

The means of 100 MAE values of each minute with SVR are listed as MAE in Table 4.7 and RMSE in Table 4.8. When we look at the SVR regression results, we get the best results from PCA and LDA preprocessing methods. For some minutes PCA is better than LDA for others LDA is better than PCA. These two preprocessing methods results are very close to each other. Just like the ANN results, using the peak values method give us the almost the same results. But this time using the entire feature set method is resulted with unreliably worst results.

Among the sucrose concentration SVR experiments, we get the best result as MAE of 3.07 mg/mL with 0.41 standard deviation from PCA implementation of minute 10 samples. 3.07 mg/mL error corresponds to predicting a 30 mg/mL sucrose solution as 26.93 mg/mL or 33.07 mg/mL. As contrast we get the best result as RMSE of 4.76 mg/mL with 0.96 standard deviation from PCA implementation of minute 5 samples. We get the worst SVR result as MAE of 495.26 mg/mL from using entire feature set method of minute 4 samples and as RMSE 968.37 mg/mL from using entire feature set method of minute 12 samples. Also by looking at the standard deviation values of using entire feature set preprocessing method, it is obvious that they are not reliable.

From the perspective of training duration, ANN handles the entire feature set faster than SVR thanks to the GPU computation power. But also ANN has a shorter duration for PCA and LDA implementations. But when it comes to the peak value preprocessing method, SVR trains in much less time.

Sucrose experiments results of ANN and SVR results are shown a significant similarity with each other.

CHAPTER 5

ESTIMATION OF BACTERIA CONCENTRATION

5.1. Bacteria Concentration Dataset

Thanks to IZTECH Department of Bioengineering, different bacteria concentrations with ultrapure water is prepared in well plate cells. Concentrations of bacteria solutions by size is determined as 10^2, 10^3, 10^4, 10^5, 10^6, 10^7 cells per unit(cpu). After that, each well plate cell is measured by Ultraviolet-Visible(UV-Vis) spectroscopy with the help of golden nanoparticles. The golden nanoparticles are grown for several minutes and used for increasing the adhesion of bacteria colonies in order to increase sensitivity to small variations in the refractive index. Also the same number of well plate cells are prepared without bacteria solutions and spectrum measurements performed on them too.

The selected wavelengths are between 400 nm and 800 nm with a precision of 1 nm. This means for each prepared bacteria concentration in a well plate cell, we get 401 different measurements. All these measurements are performed at different times as minute 0, minute 4, minute 10, minute 12. These times are the durations of gold growth in solution. The selected wavelengths and measurement times are determined by IZTECH Department of Bioengineering to determine which is, within these spectra and at these specific times, the most distinguishing and informative spectrum. We investigate this in this thesis by machine learning methods.

After all the experiment setups and measurements, we have a bacteria solution dataset whom samples are labeled as with bacteria for 10^2, 10^3, 10^4, 10^5, 10^6, 10^7 cpu concentrations and without bacteria for without bacteria solution measurements. Moreover, in the same dataset, we labeled 10^2, 10^3, 10^4 cpu concentrations as low bacteria concentrations and 10^5, 10^6, 10^7 cpu concentrations as high bacteria concentrations. Number of samples is given Table 5.1. Each sample contains 401 measurements corresponding to 401 features. To examine the bacteria concentration dataset, spectroscopy measurement values at each wavelength are drawn with different colors for each different bacteria concentration on Figures 5.1, 5.2, 5.3 and 5.4.

As seen in Figures 5.1 and 5.2, determining the low bacteria concentration and high bacteria concentration by the specific minute of spectroscopy measurements is not possible by the naked eye because of all lines overlapping with each other. It can be seen that low bacteria concentration lines are messier and high bacteria concentration lines are more organized. It is clear that for all minutes nearly after 550 nm wavelength, spectrum measurement values started to spread. Just like the 4.1, the measurements at specific wavelengths can be more informative to distinguish the bacteria concentration.

As seen in Figure 5.3 and 5.4, determining the with and without bacteria concentration in solutions by the specific minute of spectroscopy measurements is not possible by the naked eye because of the lines overlapping with each other. Even this time it is messier than low and high bacteria concentration measurements. It is clear that for all minutes nearly after 550 nm wavelength, spectrum measurement values started to spread. This means measurements at specific wavelengths can be more informative to distinguish with and without bacteria concentration again.

In the graphs, we tried to explain the classification of bacteria concentration in solutions. But this time with the knowledge of sucrose concentration regression results, previous works approaches are not helped us as much as we expected. Thus we tried to solve this problem directly by building a system to get the most informative and distinguishing values from these features which can be done by feature extractions techniques.

Table 5.1. Number of samples in the bacteria concentration dataset

	Minute 0	Minute 4	Minute 10	Minute 12
With 10^2 cpu Bacteria	21	17	21	19
With 10^3 cpu Bacteria	23	20	21	21
With 10^4 cpu Bacteria	22	20	21	20
With 10^5 cpu Bacteria	23	14	14	17
With 10^6 cpu Bacteria	18	20	18	19
With 10^7 cpu Bacteria	38	38	37	30
Low Bacteria Concentration	66	57	63	60
High Bacteria Concentration	79	72	69	66
With Bacteria	145	129	132	126
Without Bacteria	144	129	132	126

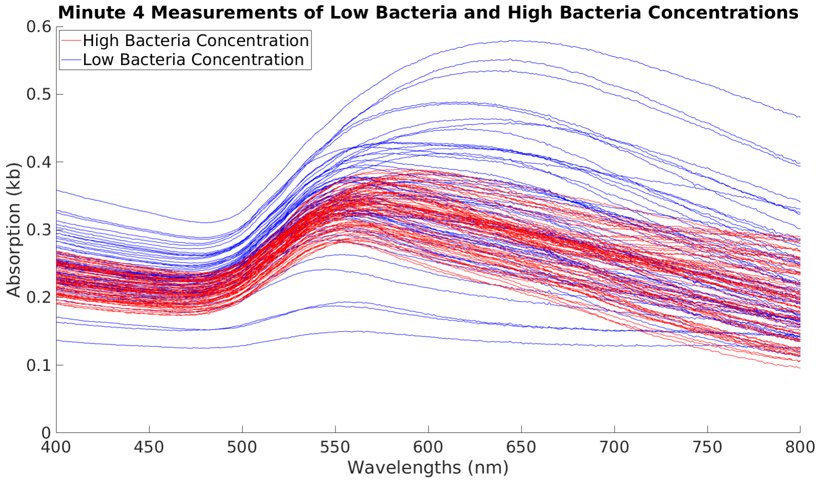


Figure 5.1. Low Bacteria Concentration (10^2, 10^3, 10^4) and High Bacteria Concentration (10^5, 10^6, 10^7) Measurements at Minute 4

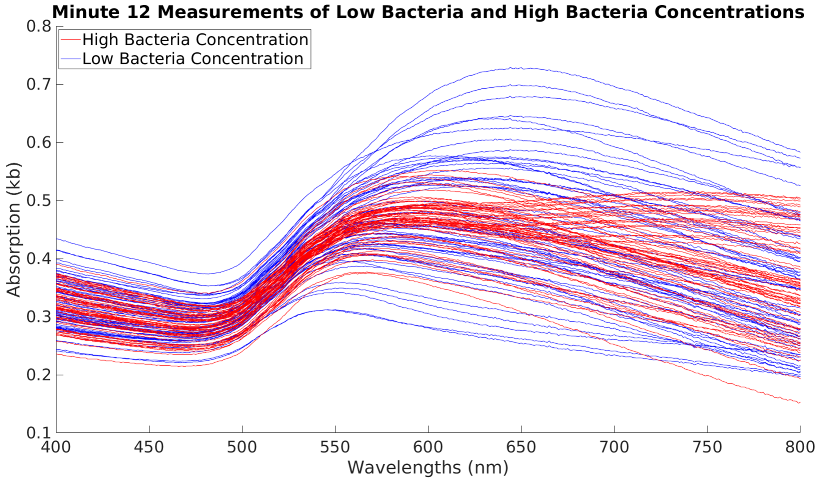


Figure 5.2. Low Bacteria Concentration (10^2, 10^3, 10^4) and High Bacteria Concentration (10^5, 10^6, 10^7) Measurements at Minute 12

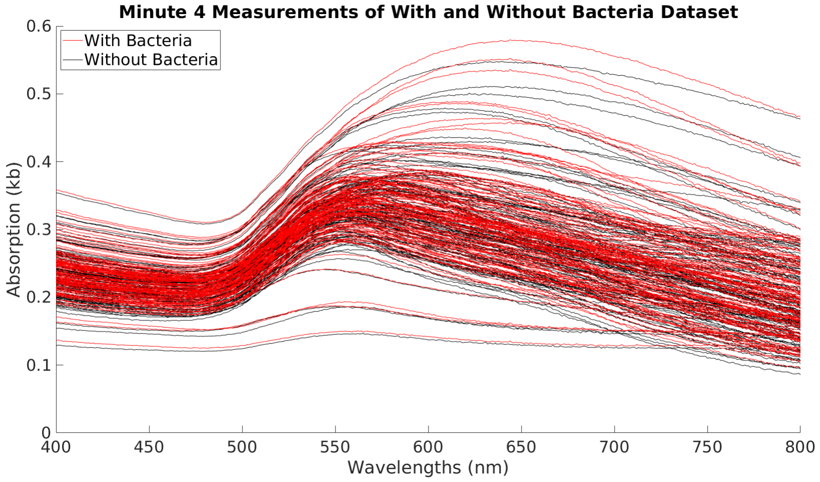


Figure 5.3. With and Without Bacteria Solution Measurements at Minute 4

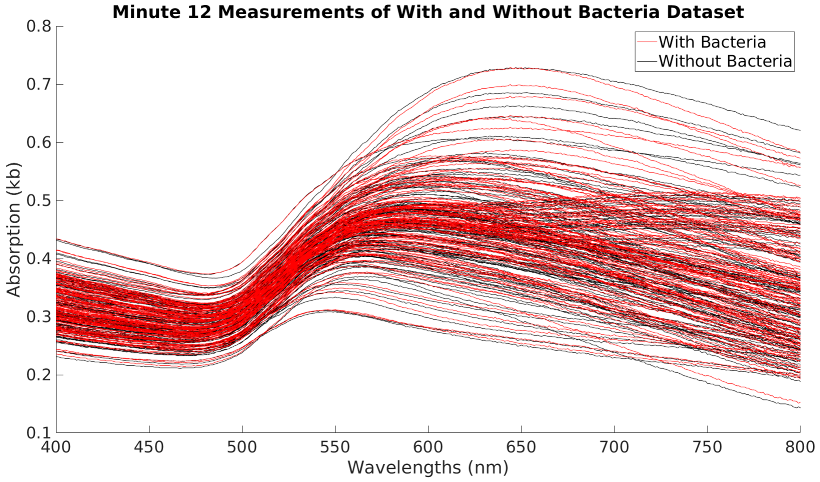


Figure 5.4. With and Without Bacteria Solution Measurements at Minute 12

5.2. Bacteria Concentration Experiments

To deal with the bacteria concentration classification problem, we treat separately each minute samples as minute 0, minute 4, minute 10 and minute 12. Just like in Section 4.2, we want to use feedforward shallow Artificial Neural Network (ANN) with the help of MATLAB Deep Learning Toolbox. For ANN activation function in each neuron, hyperbolic tangent(tanh) transfer function is used. For ANN training function, scaled conjugate gradient backpropagation option is selected which updates weight and bias values according to the scaled conjugate gradient method. But in order to deal with this classification problem, for ANN performance function, cross entropy performance function is used which heavily penalizes extremely inaccurate outputs and lightly penalizes fairly correct outputs. Again in each ANN structure, input values of input neurons and output values of output neuron values are normalized with mapminmax option which normalizes the minimum and maximum values between -1 and 1 accordingly. Also in order to compare the results wit different machine learning technique, Support Vector Machine on MATLAB environment with MATLAB Statistics and Machine Learning Toolbox is used. As a kernel function linear kernel is selected by trial and error. In addition, input values are scaled by the corresponding weighted column mean and standard deviation by passing ”Standardize” parameter to the related training function.

In the bacteria solution dataset, we have 401 different features. With the help of different preprocessing methods like PCA and LDA, the feature set size is changed. But it is mentioned in 5.2.1, the bacteria solution dataset is not suitable for direct LDA implementation due to the low number of samples. That is why, different approaches are applied in 5.2.1.2 and 5.2.1.3. The number of inputs in the ANN structures are changed accordingly. But the number of output neurons for each ANN structure remains the same since the number of classes for classification is the same for low and high bacteria concentration classification problem, with and without bacteria classification problem. Thus two neurons are used in the output layer of each ANN structure. For the hidden layer, we want to try a different number of hidden layers and neurons to observe the hidden layer structure effect on classification success. These hidden layer types are remained same just like in the Tables 4.2 and 4.3.

Before starting the training of each different structure of ANNs, the separate minute measurements of the dataset are divided into 3 parts as 60% for the training set, 20% for the validation set and 20% for the test set. Also the separate minute measurements of the dataset are divided into 2 parts as 80% for the training set, 20% for the test set during the SVM experiments because SVM training does not consistent of multiple different structures. During this division, the samples are shuffled by supporting that each dataset group had an almost equal ratio of different samples as low/high bacteria concentration and with/without bacteria solution. In some cases, this rule is slightly broken because the total number of bacteria solution samples are changing for each minute can be seen in Table 5.1. This dataset division operation is conducted at the beginning of each machine learning training.

5.2.1. Dataset Preprocessing

For the bacteria solution dataset, we try to obtain the most informative and distinguishing values from the bacteria solution feature set with the same approaches. But this time we deal with classification problem as low/high bacteria concentration and with/without bacteria solution in the bacteria solution dataset. Thus we apply these techniques for two different classification problem separately. Also, we apply the same classification algorithm on these preprocessed datasets to compare the effectiveness of these methods on the spectroscopy datasets.

5.2.1.1. Applying Principal Component Analysis

Same as in 4.2.1.3, we want to use PCA to extract the most informative and distinguishing information from bacteria solution dataset by applying to low/high bacteria concentration and with/without bacteria solution samples separately. Thus for 401 features of each bacteria sub-dataset, we apply a feature extraction technique as PCA.

The dataset division operation is applied of each bacteria sub-dataset for each minute samples as 3 parts as 60% for the training set, 20% for the validation set and 20% for the test set during the ANN experiments. Also the dataset division operation is applied of each bacteria sub-dataset for each minute samples as 2 parts as 80% for the training set, 20% for the test set during the SVM experiments because SVM training does not consistent of multiple different structures. Also, LDAP is performed on training and validation set only because again we do not want our machine learning system to receive any preliminary information from test samples.

For each minute samples in the bacteria solution dataset, PCA is applied and for 401 feature, 401 different eigenvalue and eigenvectors are calculated. These eigenvalues are ordered by their value because bigger eigenvalue indicates a bigger effect on the whole dataset. According to the sum of total eigenvalues, first 10 biggest eigenvalues give us over than 99.99% of variation in the dataset. Thus the eigenvectors of the first 10 biggest eigenvalues are used to convert the dataset from 401 features to 10 features.

5.2.1.2. Applying Linear Discriminant Analysis with Pseudo Inverse

Same as in 4.2.1.4, we apply LDA to classify low/high bacteria concentration and with/without bacteria solution. But we face with a problem when calculating S_b (between classes scatter matrix) and S_w (within classes scatter matrix), S_w become singular matrix because of the curse of dimensionality problem which causes an insufficient number of samples compared to the number of features. Most of the face recognition datasets also suffer from the same problem just like in the studies of Rui Huang et al. (2002), Sahoolizadeh and Aliyari Ghassabeh (2008) and Zhao et al. (2011). As required by the formula it becomes impossible to take the inverse of S_w matrix. In order to solve this problem, instead of calculating the inverse of the S_w matrix, we calculated pseudo-inverse which can be applied only for singular matrices. Thus our LDA application becomes Linear Discriminant Analysis with pseudo-inverse (LDAP) which is previously used by Liu et al. (2007) and Gorecki and Luczak (2013).

The dataset division operation is applied of each bacteria sub-dataset for each minute samples as 3 parts as 60% for the training set, 20% for the validation set and 20% for the test set during the ANN experiments. Also the dataset division operation is applied of each bacteria sub-dataset for each minute samples as 2 parts as 80% for the training set, 20% for the test set during the SVM experiments because SVM training does not consistent of multiple different structures. Also, LDAP is performed on training and validation set only because again we do not want our machine learning system to receive any preliminary information from test samples.

For each minute samples in bacteria sub-datasets, LDAP is applied as explained above and for 401 feature, 401 different eigenvalue and eigenvectors are calculated. These eigenvalues are ordered by their value because bigger eigenvalue indicates a bigger effect on the whole dataset. According to the sum of total eigenvalues, first 10 biggest eigenvalues give us over than 99.99% of variation in the dataset. Thus the eigenvectors of the first 10 biggest eigenvalues are used to convert the dataset from 401 features to 10 features.

5.2.1.3. Applying First PCA Then LDA

We want to extract the most informative and distinguishing information from the bacteria solution sub-datasets, As in the previous studies of Zhao et al. (2011) and Sahoolizadeh and Aliyari Ghassabeh (2008), the approach consists of first applying PCA since direct LDA implementation suffers from the curse of dimensionality problem. After the dataset number of features are diminished enough to overcome the problem, LDA implementation is performed on the PCA applied sub-datasets as low/high bacteria concentration and with/without bacteria solution. Thus 401 features of the sub-datasets, first are reduced to 90 which is determined by the total common number of samples of low and high bacteria concentrations, with and without bacteria solutions. Then LDA is implemented to form the new dataset into 10 features.

The dataset division operation is applied of each bacteria sub-dataset for each minute samples as 3 parts as 60% for the training set, 20% for the validation set and 20% for the test set during the ANN experiments. Also the dataset division operation is applied of each bacteria sub-dataset for each minute samples as 2 parts as 80% for the training set, 20% for the test set during the SVM experiments because SVM training does not consistent of multiple different structures. Also, LDAP is performed on training and validation set only because again we do not want our machine learning system to receive any preliminary information from test samples.

For each minute samples in bacteria sub-datasets, first PCA then LDA method is applied as explained above. For 401 features, 401 different eigenvalue and eigenvectors are calculated by PCA application. Then these eigenvalues are ordered by their value because bigger eigenvalue indicates a bigger effect on the whole dataset. According to the sum of total eigenvalues, first 10 biggest eigenvalues give us over than 99.99% of variation in the dataset. But in order to apply LDA, the first 90 biggest eigenvalues are used to convert the dataset from 401 features to 90 features. After that LDA is applied just like in Section 4.2.1.4. But this time instead of calculating 401 eigenvectors and 401 eigenvalues, we deal with 90 eigenvectors and 90 eigenvalues. By using the first 10 biggest eigenvalues, 90 features are formed into 10 feature in the new dataset.

5.2.2. Artificial Neural Network (ANN) Training Results

For each bacteria sub-dataset, the different number of ANN structures trained 100 times from scratch. The number of total experiments for each ANN structure is determined to eliminate the chance factor. For the determination of the error of the current trained ANN structure, the percentage of the number of misclassifications of test samples are used. Mean of 100 test set misclassification from each training is used to evaluate one ANN structure for a specific minute samples. Also the standard deviation of 100 traning is calculated to show the variation of experiment results.

For each minute, these dataset preparation and training operations are conducted separately. We obtain means of 100 error percentages of the test set of each trained ANN structure. The mean of test error percentages of each minute with its ANN structure are listed in Table 5.2 as with and without bacteria solution experiment results and in Table 5.3 low and high bacteria concentration experiment results.

The duration of ANN experiments are measured in terms of seconds. Minute 4 dataset experiments with two hidden layers structure which has 10 neurons in each layer duration measurements can be seen in Table 5.4.

Performance graphs of minute 4 ANN training examples of with and without bacteria solution classification can be seen on Figures 5.5, 5.6 and 5.7. Also, performance graphs of minute 4 ANN training examples of low and high bacteria concentration classification can be seen on Figures 5.8, 5.9 and 5.10. In these performance graphs, we expect that validation and train cross entropy error values are reduced faster then test cross entropy error value in each epoch.

Confusion graphs of minute 4 ANN training examples of with and without bacteria solution classification can be seen on Figures 5.11, 5.12 and 5.13. Also confusion graphs of minute 4 ANN training examples of low and high bacteria concentration classification can be seen on Figures 5.14. In these confusion graphs, we mostly focused on the prediction percentage of test sample results since we evaluate the preprocessing methods and ANN hidden layer structures according to error percentage of test set. In the graphs, we expect that misclassified number of samples should be zero in the perfect case. As a general overview of the results, we see that as applying PCA, LDAP and first PCA then

LDA methods in order, we get fewer and fewer misclassified number of test samples.

Table 5.2. The with and without bacteria solution classification results are shown as mean / standard deviation of test set error percentage of 100 training with their ANN hidden layer structures.

	PCA	Hidden Layer Structure	LDAP	Hidden Layer Structure	First PCA Then LDA	Hidden Layer Structure
Minute 0	12.05 / 3.73	10	4.22 / 2.96	10	1.66 / 1.58	10 - 10
Minute 4	26.74 / 6.61	15	2.70 / 2.31	5 - 5	0.22 / 0.62	10 - 10
Minute 10	21.36 / 5.54	5	15.94 / 4.20	15	14.94 / 4.05	5
Minute 12	14.58 / 5.27	15	2.26 / 2.30	10	0.18 / 0.64	5

Table 5.3. The low and high bacteria concentration classification results are shown as mean / standard deviation of test set error percentage of 100 training with their ANN hidden layer structures.

	PCA	Hidden Layer Structure	LDAP	Hidden Layer Structure	First PCA Then LDA	Hidden Layer Structure
Minute 0	0,58 / 1.39	15	0.48 / 1.20	10	0 / 0	10 and 15 and 10 - 10
Minute 4	0.20 / 1.04	15	0 / 0	10 and 10 - 10	0 / 0	10 and 15 and 10 - 10
Minute 10	1.60 / 2.34	15	0.16 / 0.78	10	0 / 0	10 and 15
Minute 12	1.32 / 2.34	10 - 10	0.64 / 1.47	15	0 / 0	10 and 15

Table 5.4. Total 100 ANN training duration and approximate duration of each ANN training in terms of seconds of different preprocessing methods on Minute 4 dataset with two hidden layers structure while each layer has 10 neurons

	PCA	LDAP	First PCA Then LDA
Total Duration of 100 Training of With/Without Bacteria Dataset	37.69	52.68	38.01
Approximate Duration of Each Training of With/Without Bacteria Dataset	0.37	0.52	0.38
Total Duration of 100 Training of Low/High Bacteria Dataset	40.14	40.82	35.16
Approximate Duration of Each Training of Low/High Bacteria Dataset	0.40	0.40	0.35

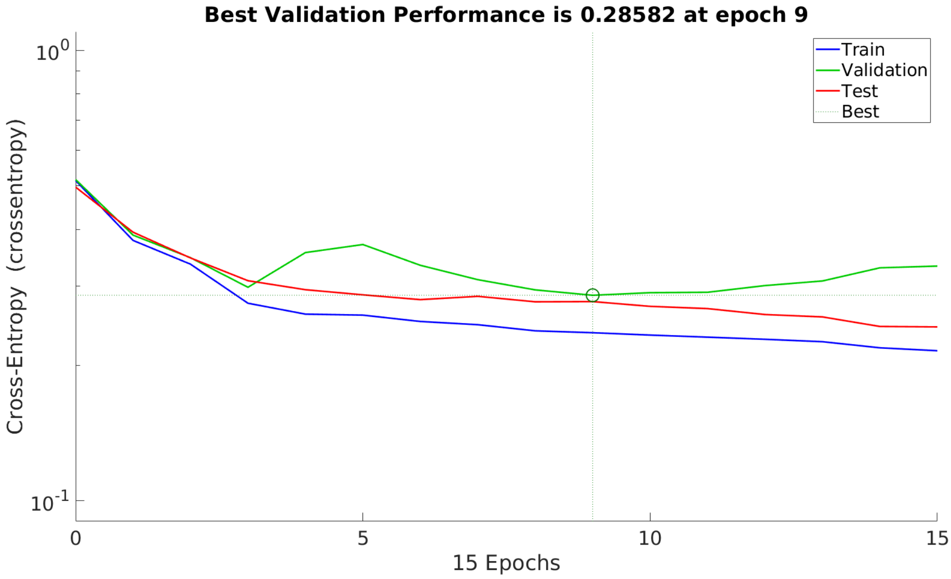


Figure 5.5. Performance graph of a with and without bacteria solution classification experiment by applying PCA on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

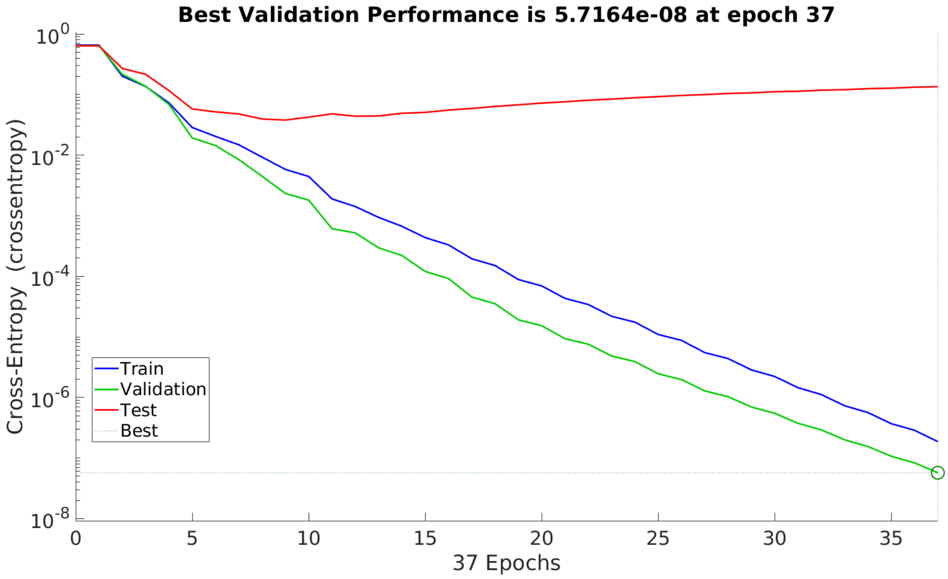


Figure 5.6. Performance graph of a with and without bacteria solution classification experiment by applying LDAP on minute 4 samples with ANN structure of double hidden layers with 10 neurons in each layer

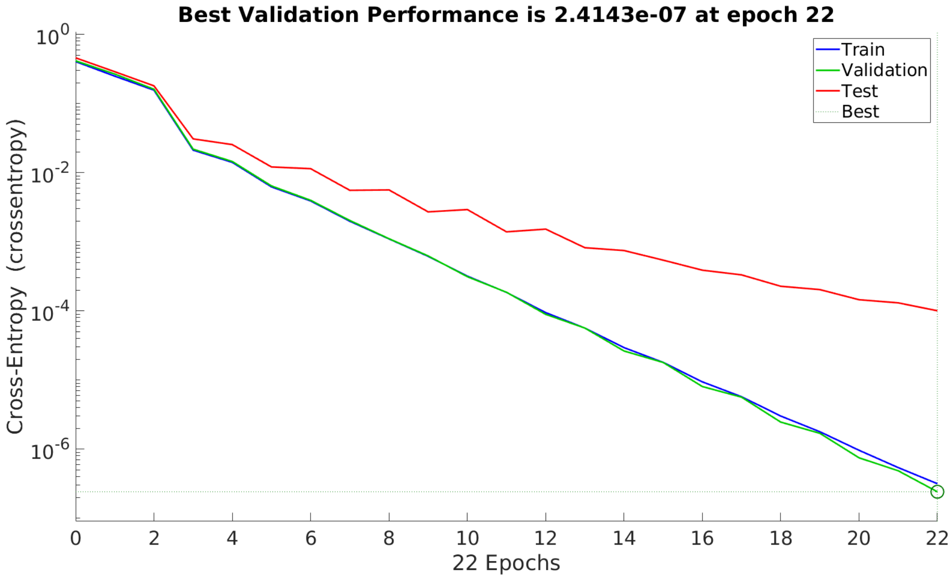


Figure 5.7. Performance graph of a with and without bacteria solution classification experiment by applying first PCA then LDA on minute 4 samples with ANN structure of double hidden layers with 10 neurons

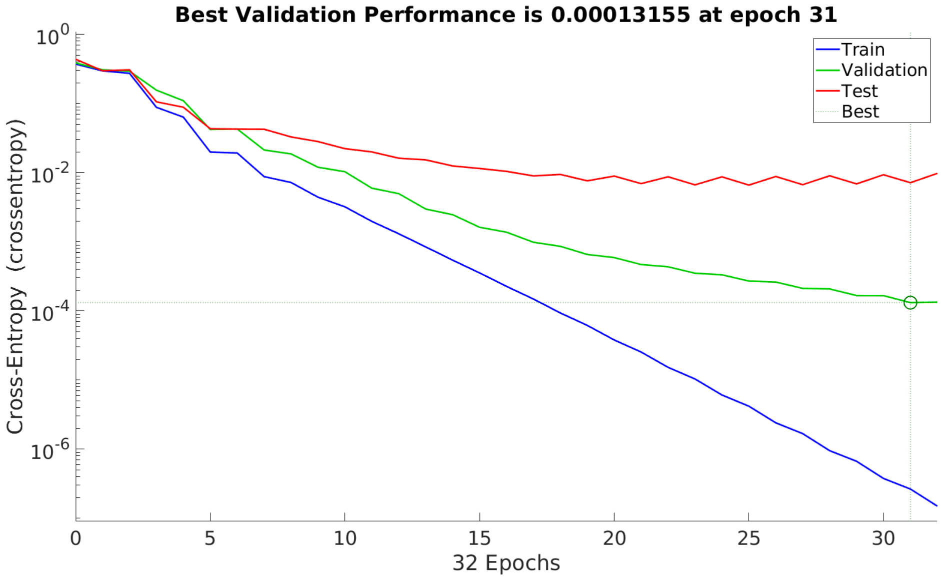


Figure 5.8. Performance graph of a low and high bacteria concentration classification experiment by applying PCA on minute 4 samples with ANN structure of double hidden layers with 10 neurons in each layer

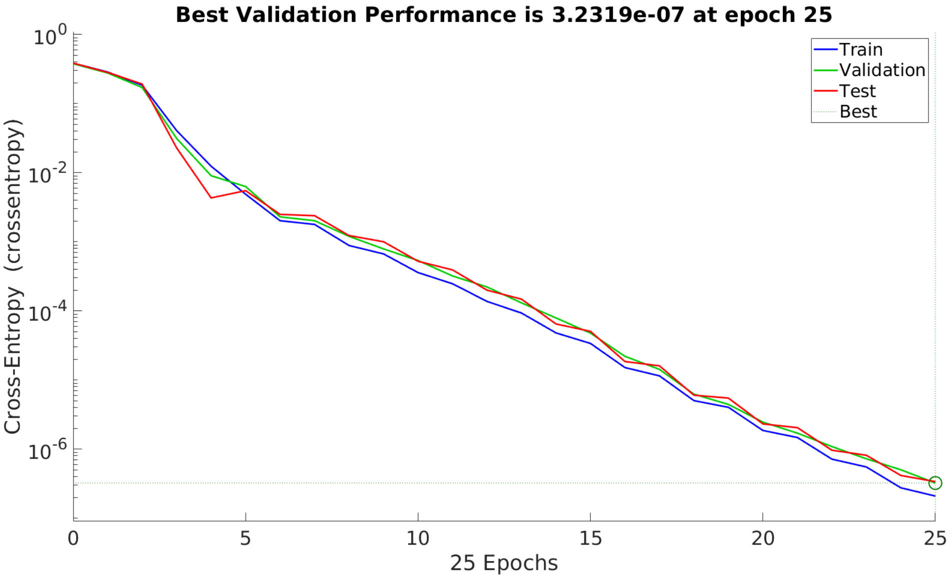


Figure 5.9. Performance graph of a low and high bacteria concentration classification experiment by applying LDAP on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

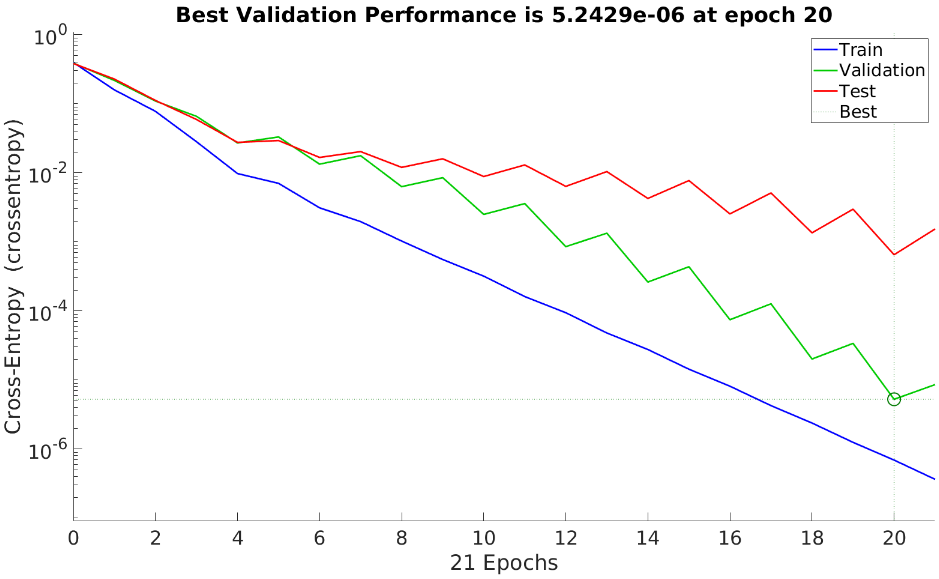


Figure 5.10. Performance graph of a low and high bacteria concentration classification experiment by applying first PCA then LDA on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

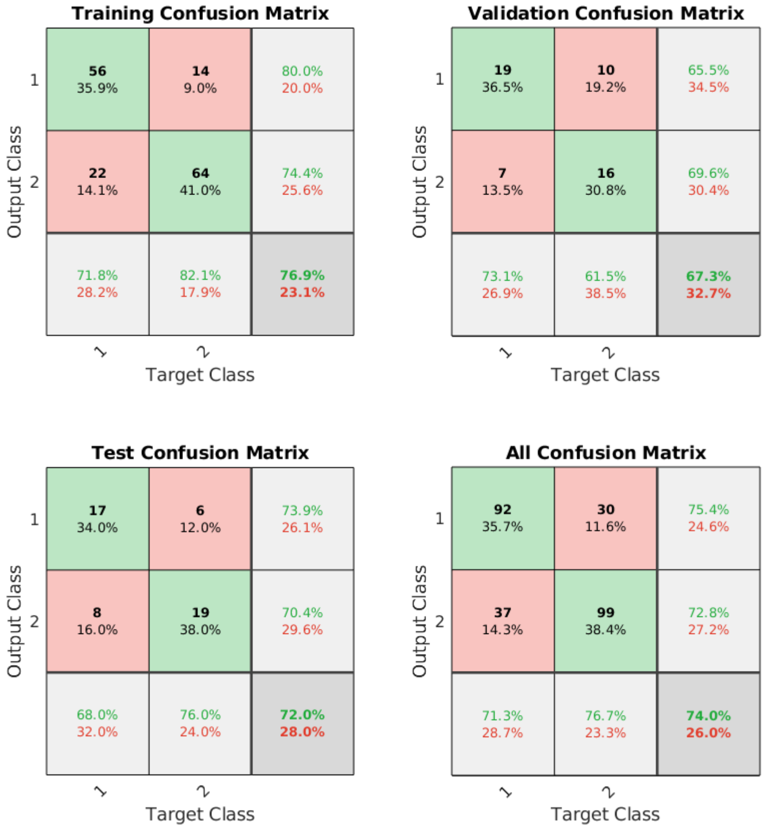


Figure 5.11. Confusion graph of a with and without bacteria solution classification experiment by applying PCA on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

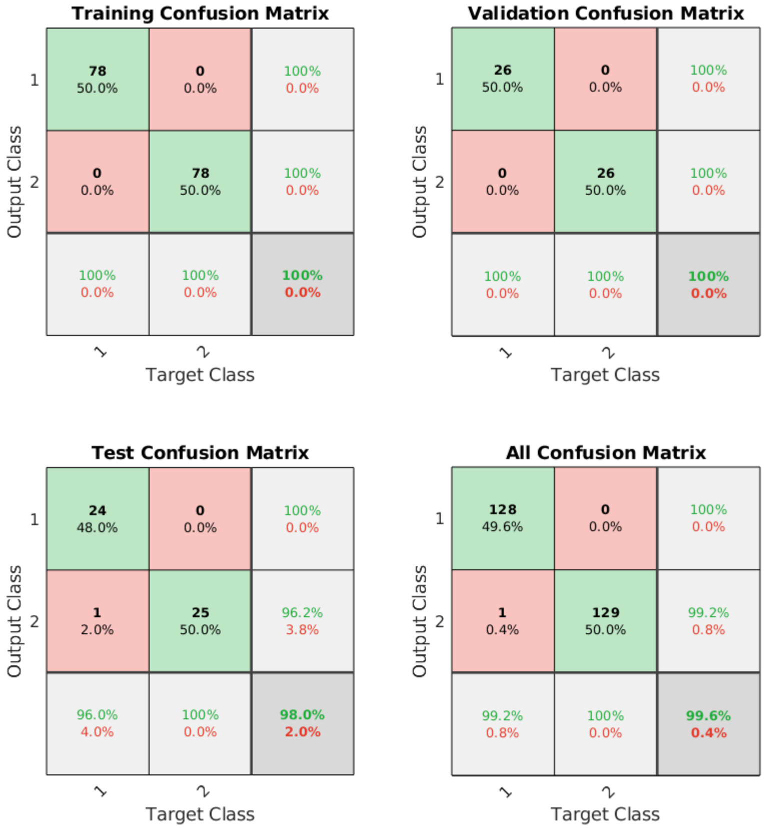


Figure 5.12. Confusion graph of a with and without bacteria solution classification experiment by applying LDAP on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

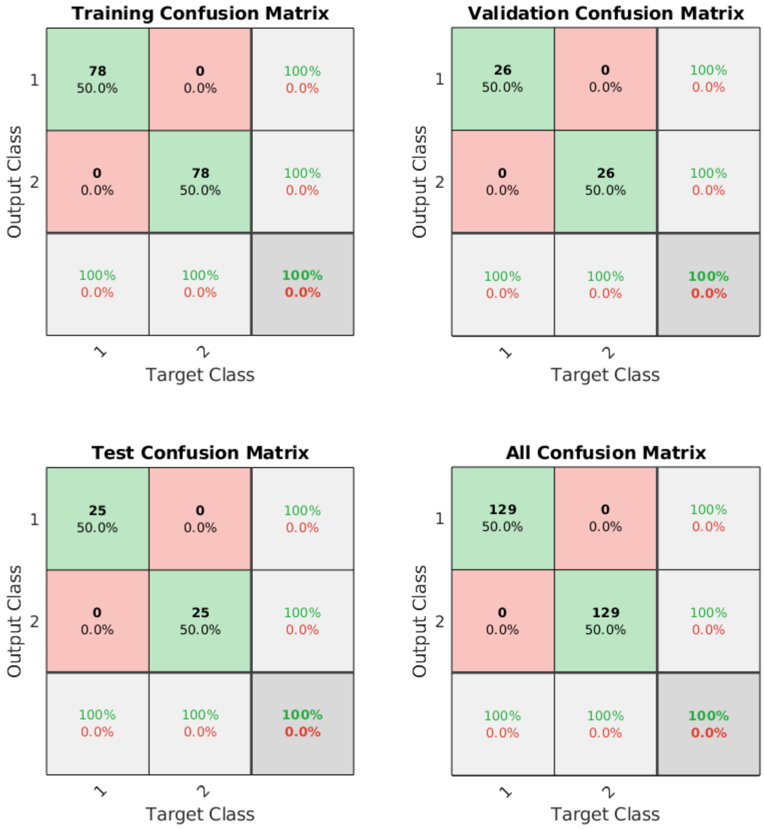


Figure 5.13. Confusion graph of a with and without bacteria solution classification experiment by applying first PCA then LDA on minute 4 samples with ANN structure of a single hidden layer with 5 neurons

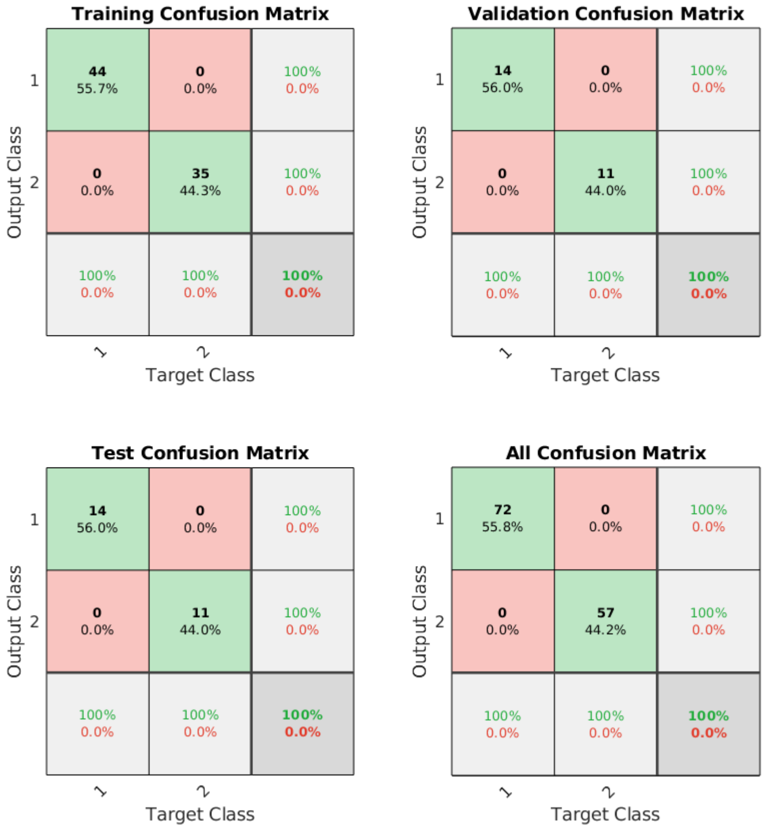


Figure 5.14. Confusion graph of a low and high bacteria concentration classification experiment on minute 4 samples with ANN structure of a single hidden layer with 5 neurons. Perfect result is obtained with PCA, LDAP and first PCA then LDA approaches.

5.2.3. Support Vector Machine (SVM) Training Results

For each bacteria sub-dataset, SVM trained 100 times from scratch. The number of total experiments is determined to eliminate the chance factor. For the determination of the error of the SVM, the percentage of the number of misclassifications of test samples are used. Mean of 100 test set misclassification from each training is used to evaluate SVM results for a specific minute samples.

For each minute, these dataset preparation and training operations are conducted separately. We obtain means of 100 error percentages from the test set of trained SVM. The mean of test error percentages of each minute are listed in Table 5.5 as with and without bacteria solution experiment results and in Table 5.6 low and high bacteria concentration experiment results.

The duration of SVM experiments are measured in terms of seconds. Minute 4 dataset experiments duration measurements can be seen in Table 5.7.

Confusion graphs of minute 4 SVM training examples of with and without bacteria solution classification can be seen on Figures 5.15, 5.16 and 5.17. Also confusion graphs of minute 4 SVM training examples of low and high bacteria concentration classification can be seen on Figures 5.18. In these confusion graphs, we mostly focused on the prediction percentage of test sample results since we evaluate the preprocessing methods and trained SVM error percentage of test set. In the graphs, we expect that misclassified number of samples should be zero in the perfect case. As a general overview of the results, we see that as applying PCA, LDAP and first PCA then LDA methods in order, we get fewer and fewer misclassified number of test samples just like the ANN experiments results.

Table 5.5. The with and without bacteria solution classification results are shown as mean / standard deviation of test set error percentage of 100 training with SVM.

	PCA	LDAP	First PCA Then LDA
Minute 0	14.54 / 4.87	4.64 / 2.53	1.70 / 1.64
Minute 4	26.10 / 5.95	2.26 / 2.32	0.10 / 0.43
Minute 10	23 / 6.32	15.67 / 4.30	15.23 / 3.88
Minute 12	11.86 / 4.53	2.70 / 2.22	0.24 / 0.65

Table 5.6. The low and high bacteria concentration classification results are shown as mean / standard deviation of test set error percentage of 100 training with SVM.

	PCA	LDAP	First PCA Then LDA
Minute 0	0.75 / 1.99	0.34 / 1.03	0 / 0
Minute 4	0.32 / 1.09	0.08 / 0.8	0 / 0
Minute 10	1.4 / 2.07	0.12 / 0.68	0 / 0
Minute 12	1.68 / 2.28	0.64 / 1.47	0 / 0

Table 5.7. Total 100 SVM training duration and approximate duration of each SVM training in terms of seconds of different preprocessing methods on Minute 4 dataset

	PCA	LDAP	First PCA Then LDA
Total Duration of 100 Training of With/Without Bacteria Dataset	4.45	11.53	5.07
Approximate Duration of Each Training of With/Without Bacteria Dataset	0.04	0.11	0.05
Total Duration of 100 Training of Low/High Bacteria Dataset	3.37	8.85	3.53
Approximate Duration of Each Training of Low/High Bacteria Dataset	0.03	0.08	0.03

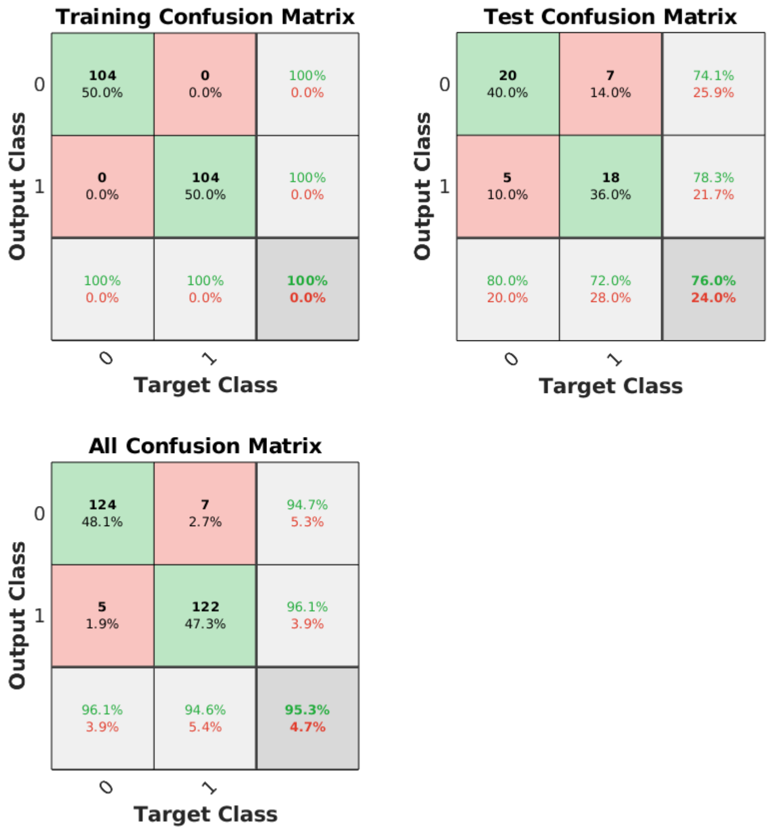


Figure 5.15. Confusion graph of a with and without bacteria solution classification experiment by applying PCA on minute 4 samples with SVM

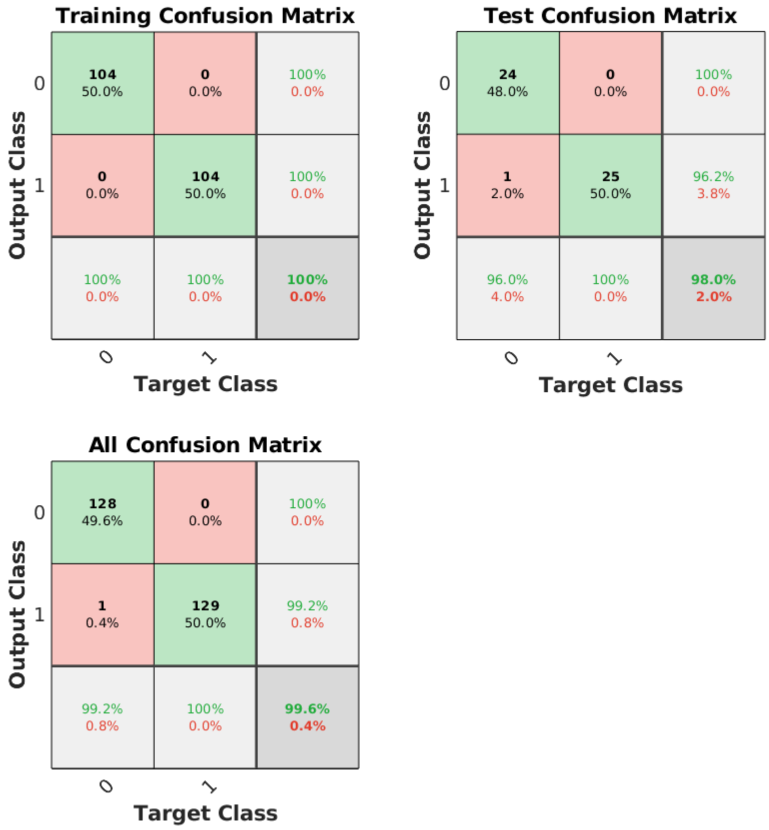
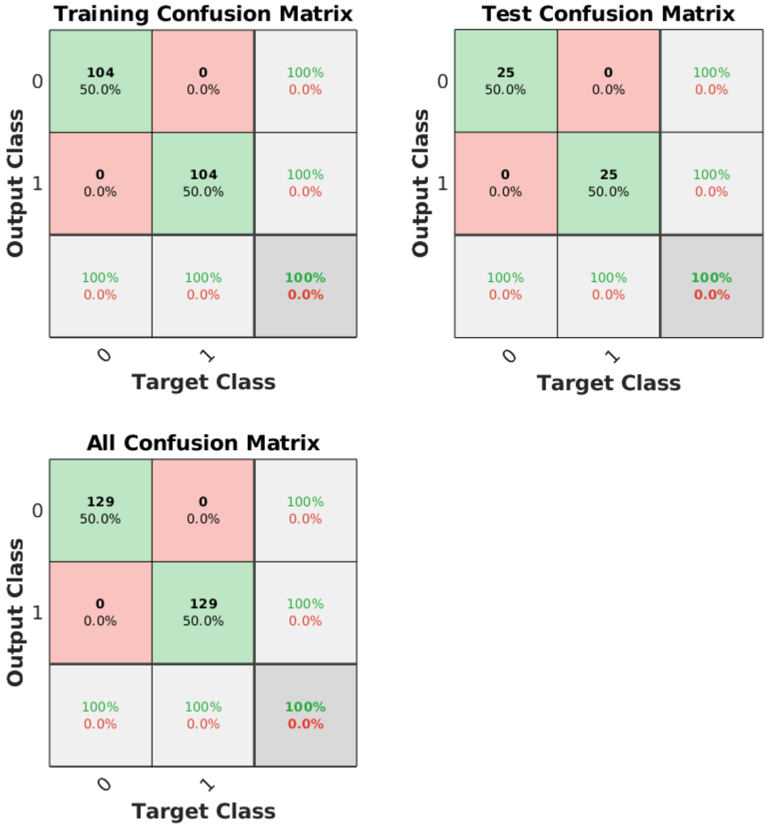


Figure 5.16. Confusion graph of a with and without bacteria solution classification experiment by applying LDAP on minute 4 samples with SVM



Training Confusion Matrix

Output Class \ Target Class	0	1	
0	**104** 50.0%	**0** 0.0%	100% 0.0%
1	**0** 0.0%	**104** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

Test Confusion Matrix

Output Class \ Target Class	0	1	
0	**25** 50.0%	**0** 0.0%	100% 0.0%
1	**0** 0.0%	**25** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

All Confusion Matrix

Output Class \ Target Class	0	1	
0	**129** 50.0%	**0** 0.0%	100% 0.0%
1	**0** 0.0%	**129** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

Figure 5.17. Confusion graph of a with and without bacteria solution classification experiment by applying first PCA then LDA on minute 4 samples with SVM

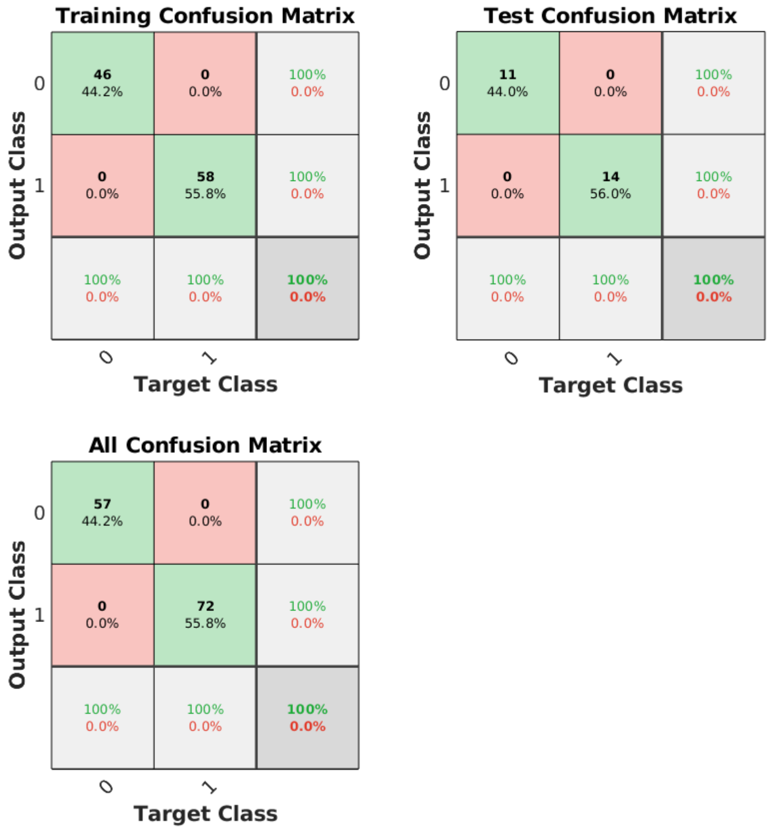


Figure 5.18. Confusion graph of a low and high bacteria concentration classification experiment on minute 4 samples with SVM. Perfect result is obtained with PCA, LDAP and first PCA then LDA approaches.

5.2.4. Discussion and Conclusion

After performing all ANN and SVM experiments with preprocessed with and without bacteria solution dataset results evaluation is given in 5.2.4.1 and low and high bacteria concentration sub-dataset results evaluation is given in 5.2.4.2.

5.2.4.1. With and Without Bacteria Solution Experiment Results

When we look at the results for ANN experiments, it is clear that we get the best results from first PCA then LDA implementation preprocessing method as 0.18% classification error with 0.64 standard deviation. The LDAP method gives us the second best results which we can say still acceptable as 2.26% classification error with 2.30 standard deviation. But the PCA implementation results are the worst among all preprocessing methods as its error percentages are changing between 12.05% and 26.74%.

Within all results, it is clear that minute 10 samples always give the worst unacceptable results without regarding the preprocessing methods. For PCA its error is 21.36%, for LDAP it is 15.94% and for first PCA then LDA implementation it is 14.94%. For that reason, it indicates us there is a problem about the minute 10 samples. Also in all preprocessing methods of minute 10 samples, the standard deviation is always high.

Among the with and without bacteria solution ANN experiments, we get the best result as 0.18% from first PCA then LDA implementation of minute 12 samples on single hidden layer structure which has 5 neurons the layer. 0.18% corresponds to predicting with or without bacteria solution with 0.18% classification error. We get the worst ANN result as 26.74% from PCA implementation method of minute 4 samples on a single hidden layer structure which has 15 neurons in the layer.

Among all different hidden layer structures, single hidden layer structure with 5 neurons give us the best result in the first PCA then LDA implementation. But also the same method implementation with two hidden layers structure which has 10 neurons in each layer generally gives us close results. Also for PCA implementation and LDAP implementation, we get the best results from a single hidden layer with 10 neurons structure. That is why we can not say there is a direct benefit from hidden layer structure complexity to get better results. The main effect is supported from preprocessing steps by especially first PCA then LDA implementation. Also, we can say that as the minutes pass to grow golden nanoparticles for increasing the adhesion of bacteria, we get better and better results for with and without bacteria solution classification because for PCA implementation minute 0 and minute 12 give the best results, for LDAP it is minute 4 and minute 12, for first PCA then LDA implementation it is again minute 4 and minute 12. Thus throughout the experiments, minute 12 samples give the best results.

When we look at the SVM results, it is clear that we get the best results again from first PCA then LDA implementation preprocessing method as 0.10% classification error with 0.43 standard deviation. The LDAP method gives us the second best results which we can say still acceptable as 2.26% classification error with 2.32 standard deviation just like the ANN experiments results. But as we expected, the PCA implementation results are the worst among all preprocessing methods as its error percentages are changing between 11.86% and 26.10%.

Again it is clear that minute 10 samples always give the worst unacceptable results without regarding the preprocessing methods. For PCA its error is 23.00%, for LDAP it is 15.67% and for first PCA then LDA implementation it is 15.23%. For that reason with the support of the ANN experiments results, it indicates us there is a problem about the minute 10 samples.

Among the with and without bacteria solution SVM experiments, we get the best result as 0.10% from first PCA then LDA implementation of minute 4 samples. 0.10% corresponds to predicting with or without bacteria solution with 0.10% classification error. We get the worst ANN result as 26.10% from PCA implementation method of minute 4 samples.

Again, we can say that as the minutes pass to grow golden nanoparticles for increasing the adhesion of bacteria, we get better and better results for with and without bacteria solution classification because for PCA implementation minute 0 and minute 12 give the best results, for LDAP it is minute 4 and minute 12, for first PCA then LDA implementation it is again minute 4 and minute 12. Throughout the experiments, minute 4 and minute 12 samples give the best results.

5.2.4.2. Low and High Bacteria Concentration Experiment Results

When we look at the results for ANN experiments, we obtain means of 100 test error percentages of each trained ANN structure. The minimum test error percentage of each minute with its ANN structure are listed in Table 5.3. When we look at the results as a whole, this time all preprocessing methods give us acceptable and successful results. But especially first PCA then LDA implementation gives us error-free classification for all minutes. Also, we see that LDAP implementation of minute 4 gives us an error-free result. The standard deviation of 100 training support the reliability of the error free results.

Among the low and high bacteria concentration ANN experiments, we get the best ANN result as 0% from first PCA then LDA implementation of all minute samples on single hidden layer structure with 10 neurons, single hidden layer structure with 15 neurons and two hidden layers structure with 10 neurons in each layer. Which means we can predict the low or high bacteria concentrations as error-free. Also, we get the worst ANN result as 1.60% from PCA implementation method of minute 10 samples on a single hidden layer structure which has 15 neurons in the layer. Which is still acceptable and close to the other results for low and high bacteria concentration prediction.

Among all different hidden layer structures, two hidden layers structure with 10 neurons in each layer, single hidden layer structure with 10 neurons and single hidden layer structure with 15 neurons give us better results. That is why we can not say there is a direct benefit from hidden layer structure complexity to get better results. Main effect is supported by preprocessing steps by especially first PCA then LDA implementation. We also observe that at the results from the minute perspective as the minutes pass to grow golden nanoparticles for increasing the adhesion of bacteria, we get better and better results for the low and high bacteria concentration classification. For PCA implementation minute 4 give the best results, for LDAP it is minute 4, for first PCA then LDA implementation it is all minutes. Thus throughout the experiments, minute 4 samples give the best results. But we cannot say it is superior to other minutes since all error percentages are almost close to the zero.

From the perspective of training duration, because of the small size of feature sets, ANN training duration is slower than SVM. It shows us how SVM can handle simple solutions much faster which can be seen in Table 5.4 and Table 5.7 for the comparison.

For SVM experiments, we obtain means of 100 test error percentages of trained SVM. The mean test error percentage of each minute are listed in Table 5.6. Again all preprocessing methods give us acceptable and successful results. But especially first PCA then LDA implementation gives us error-free classification for all minutes just like the ANN experiments results.

Among the low and high bacteria concentration SVM experiments, we get the best result as 0% from first PCA then LDA implementation of all minute samples. Which means we can predict the low or high bacteria concentrations as error-free. Also, we get the worst SVM result as 1.68% from PCA implementation method of minute 12 samples. Which is still acceptable and close to the other results for low and high bacteria concentration prediction.

We also observe that for PCA implementation minute 4 give the best results, for LDAP it is minute 4, for first PCA then LDA implementation it is all minutes. Thus throughout the experiments, minute 4 samples give the best results. But we cannot say it is superior to other minutes since all error percentages are almost close to the zero for LDAP and first PCA then LDA implementations.

From the perspective of training duration, because of the small size of feature sets, ANN training duration is slower than SVM. It shows us how SVM can handle simple solutions much faster which can be seen in Table 5.4 and Table 5.7 for the comparison.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

In this thesis, we try to make estimations for sucrose concentrations on the sucrose concentration dataset which is obtained by localized surface plasmon resonance of immobilized golden nanoparticles with spectroscopy methodology. Also we try to make classifications for bacteria concentrations on the bacteria solution dataset which is obtained by spectroscopy measurements with the help of golden particles that are used for increasing the surface areas of bacteria colonies in order to increase the interaction between bacteria and electromagnetic radiation. The bacteria classification is conducted for with/without bacteria solution and low/high bacteria concentration separately. This research is a pioneer to use golden nanoparticles for sucrose concentration and bacteria concentration in solutions. Also, these obtained datasets are used as the first time in any machine learning methodologies.

As a result of the sucrose concentration experiments, with low mean absolute errors and root mean square errors, it is possible to predict the sucrose concentration in a pure water solution. It is the first step of the development of an optical biosensor. But still, it can not be suitable to use in scientific or medical purposes since it is not close to the perfect estimation. We believe that the minimum MAE as 3.07 mg/mL can be improved by an enriched dataset. Because in the current sucrose concentration dataset, we have sample labels as 0 mg/mL, 10 mg/mL, 20 mg/mL, 30 mg/mL, 40 mg/mL and 50 mg/mL which can be problematic for predicting continuous concentration values such as 0.1 mg/mL, 15.95 mg/mL etc. Also, we tested all common preprocessing approaches as using peak values, applying PCA and applying LDA, in addition, using the entire feature set. We see that how to deal with spectroscopy measurements in order to make proper predictions. Especially PCA and LDA implementations give the best results. Because of the supervised approach of LDA, we get slightly better results compared to PCA in ANN experiments. But for SVR experiments we get close error values for PCA and LDA implementations. According to our evaluation, LDA preprocessing method is preferred for sucrose concentration estimation because it is observed that RMSE values are low and standard deviations of MAE and RMSE are small. Moreover, using peak values in the spectroscopy measurements is not reliable since we are dealing with 501 different wavelength measurement, environmental noise, and human error. So it is now a good idea to eliminate features manually for spectroscopy datasets. We also experience that for the sucrose spectroscopy datasets during ANN experiments, it is not beneficial to increase the complexity of hidden layer structure as we get MAEs and RMSEs of from different hidden layer structures are very close to each other. But it is a good thing to try different structures of hidden layers for the ANN training.

As a result of the bacteria solution experiments, it is possible to predict the with/without bacteria solution with an error percentage near the error-free and low/high bacteria concentration with completely error free. Thus it is an almost excellent outcome for the bacteria dataset since for the with/without bacteria sub-dataset, experiment results for the minute 10 abnormally bad for both ANN and SVM experiments. That is why we think there should be some mistakes through the minute 10 sample measurements. But even if minute 10 results are not reasoned from the faulty measurements, still it is an outstanding result for the bacteria classifications. Again it is the first step of the development of an optical biosensor. With the outcome of the experiments, it may be used in scientific or medical purposes. We try to test all common preprocessing approaches as applying PCA and applying LDA excluding using peak values and using entire feature set methods since we already tested them on sucrose concentrations. But for the LDA implementation, we face the curse of dimensionality problem which happens when the number of dataset samples is lower than the sum of the number of features and the number of classes. Because compared to the sucrose concentration dataset, the number of samples for each class is fewer in bacteria solution dataset. This problem is very common among face recognition datasets. That is why we used different approaches as LDA with pseudo-inverse function and first PCA then LDA method in order to beat the curse of dimensionality problem. Especially first PCA then LDA implementations give the best results. As we expected thanks to the supervised approach of LDA, we get the best results from LDAP and first PCA then LDA implementation compared to PCA. For with/without bacteria classification PCA give us unacceptable results as between 12.05% and 26.74% for ANN experiments, between 11.86% and 26.10% in SVM experiments. But for the low/high bacteria concentration classifications, all preprocessing approaches and both ANN and

SVM trainings give us error rates close to zero. For ANN experiments, we experience that it is not beneficial to increase the complexity of hidden layer structure as we get test error percentages from different hidden layer structures that are very close to each other. But again it is a good thing to try different structures of hidden layers for the ANN training.

6.2. Future Work

Using golden nanoparticles to help spectroscopy measurement is first time used on sucrose and bacteria solutions. This method can be also used on different chemical or biological solutions in order to determine the concentration.

For the sucrose concentration experiments, it is still needed an enriched dataset which will help to build a system for perfect prediction since current sample labels are more likely to be used in classification instead of regression. We hope that a more comprehensive dataset in terms of concentration samples will improve the results of the estimation of sucrose concentration.

We accomplished error-free detection of with/without bacteria solution and low/ high bacteria concentration with laboratory measurements. According to these results, a proper functional sensor can be built to determine the with/without bacteria solution and low/high bacteria concentration. Moreover, it can be used in real-world problems such as milk spoilage, water pollution, diseased blood, etc.

REFERENCES

Anker, J., W. Hall, O. Lyandres, N. Shah, J. Zhao, and R. Van Duyne (2008). Biosensing with plasmonic nanosensors. *Nature materials 7*(6), 442–453.

Baştanlar, Y. and M. Özuysal (2014). Introduction to machine learning. In *miRNomics: MicroRNA Biology and Computational Analysis*, pp. 105–128. Springer.

Chua, C. D., I. M. Gonzales, E. M. Manzano, and M. Manzano (2014). Design and fabrication of a non-invasive blood glucometer using paired photo-emitter and detector near-infrared leds.

Data Wow (2018). A neural network with 2 hidden layers. `https://cdn-images-1.medium.com/max/1600/1*GM2i5bVgcmlQOEN4lP81JQ.png`. [Online; accessed May 20, 2019].

Gorecki, T. and M. Luczak (2013). Linear discriminant analysis with a generalization of the moore–penrose pseudoinverse. *International Journal of Applied Mathematics and Computer Science 23*(2), 463 – 471.

Gulderen, A., T. Anutgan, and M. Anutgan (2016). Estimation of glucose concentration in solution using near infrared spectroscopy and artificial neural network. *2016 24th Signal Processing and Communication Application Conference (SIU)*, 1197–1200.

Izenman, A. J. (2008). *Modern Multivariate Statistical Techniques Regression, Classification, and Manifold Learning*. Springer.

Jolliffe, I. (2002). *Principal Component Analysis*. Springer.

Khan Academy (2019). Electromagnetic spectrum. `https://cdn.kastatic.org/ka-perseus-images/1f69f2373d9136ed9a061a3a1b64cbffe3abc9b2.png`. [Online; accessed May 20, 2019].

Liu, J., S. Chen, X. Tan, and D. Zhang (2007). Efficient pseudoinverse linear discriminant analysis and its nonlinear form for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence 21*(8), 1265–1278.

Liu, W., W. Yang, L. Liu, and Q. Yu (2008). Use of artificial neural networks in near-infrared spectroscopy calibrations for predicting glucose concentration in urine. In *International Conference on Intelligent Computing*, pp. 1040–1046. Springer.

Malik, B. A., A. Naqash, and G. M. Bhat (2016, Sep.). Backpropagation artificial neural network for determination of glucose concentration from near-infrared spectra. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2688–2691.

Martinsson, E., B. Sepulveda, P. Chen, A. Elfwing, B. Liedberg, and D. Aili (2014, Aug). Optimizing the refractive index sensitivity of plasmonically coupled gold nanoparticles. *Plasmonics 9*(4), 773–780.

MATLAB (2019a). Matlab deep learning toolbox documentation. `https://www.mathworks.com/help/deeplearning/index.html`. [Online; accessed May 20, 2019].

MATLAB (2019b). Matlab parallel computing toolbox documentation. `https://www.mathworks.com/help/parallel-computing/index.html`. [Online; accessed May 20, 2019].

Mezgil, B., D. Erdoğan, Y. Alduran, Ü. H. Yıldız, A. A. Yıldız, and Y. Baştanlar (2017). Estimation of low sucrose concentrations by uv-vis spectroscopy and artificial neural networks. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4. IEEE.

Ozbalci, B., I. Boyaci, A. Topcu, C. Kadilar, and U. Tamer (2013, 02). Rapid analysis of sugars in honey by processing raman spectrum using chemometric methods and artificial neural networks. *Food chemistry 136*, 1444–52.

Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma (2002, Aug). Solving the small sample size problem of lda. In *Object recognition supported by user interaction for service robots*, Volume 3, pp. 29–32 vol.3.

Sahoolizadeh, H. and Y. Aliyari Ghassabeh (2008, Sep.). Face recognition using eigen-faces, fisher-faces and neural networks. In *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, pp. 1–6.

Tkachenko, N. V. (2006). *Optical Spectroscopy: Methods and Instrumentations*. Elsevier.

Towards Data Science (2017). Perceptron structure. `https://cdn-images-1.medium.com/max/800/1*n6sJ4yZQzwKL9wnF5wnVNg.png`. [Online; accessed May 20, 2019].

Towards Data Science (2019a). PCA orients data along the direction of the component with maximum variance whereas LDA projects the data to signify the class separability. `https://cdn-images-1.medium.com/max/800/1*4ibdHcy6xlV7-HU3KjonsQ.png`. [Online; accessed May 20, 2019].

Towards Data Science (2019b). SVM Classification. `https://cdn-images-1.medium.com/max/1600/1*06GSco3ItM3gwW2scY6Tmg.png`. [Online; accessed June 3, 2019].

Trabelsi, A., M. Boukadoum, and M. Siaj (2012, 12). A preliminary investigation into the design of an implantable optical blood glucose sensor. *American Journal of Biomedical Engineering 1*, 62–67.

Vítková, G., K. Novotnỳ, L. Prokeš, A. Hrdlička, J. Kaiser, J. Novotnỳ, R. Malina, and D. Prochazka (2012). Fast identification of biominerals by means of stand-off laser-induced breakdown spectroscopy using linear discriminant analysis and artificial neural networks. *Spectrochimica Acta Part B: Atomic Spectroscopy 73*, 1–6.

Wang, L. (2005). *Support Vector Machines: Theory and Applications*. Springer.

Wikipedia (2019). Absorption spectroscopy experimental setup. `https://upload.wikimedia.org/wikipedia/commons/f/f2/Spectroscopy_overview.svg`. [Online; accessed May 20, 2019].

Zeng, B., W. Wang, N. Wang, F. Li, F. Zhai, and L. Hu (2013, 01). Noninvasive blood glucose monitoring system based on distributed multi-sensors information fusion of multi-wavelength nir. *Engineering 05*, 553–560.

Zhao, N., W. Mio, and X. Liu (2011). A hybrid PCA-LDA model for dimension reduction. In *The 2011 International Joint Conference on Neural Networks*, pp. 2184–2190. IEEE.

APPENDIX A

SPECTROSCOPY MEASUREMENTS

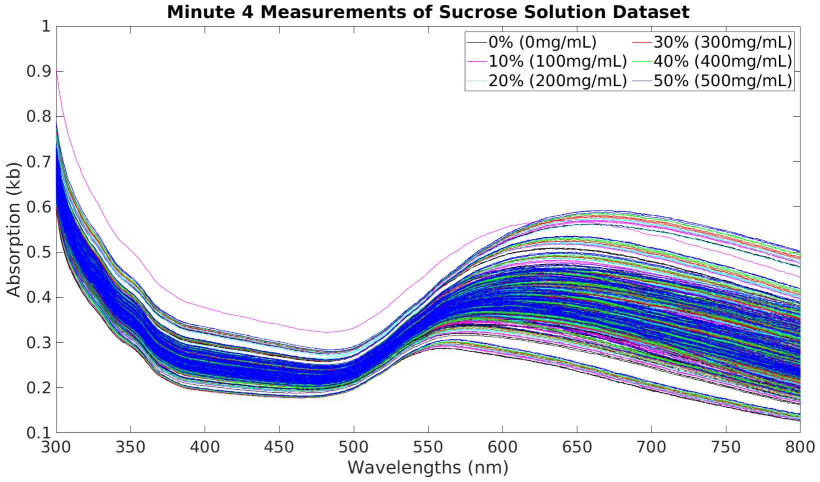


Figure A.1. Sucrose Concentrations Measurements at Minute 4 Graph

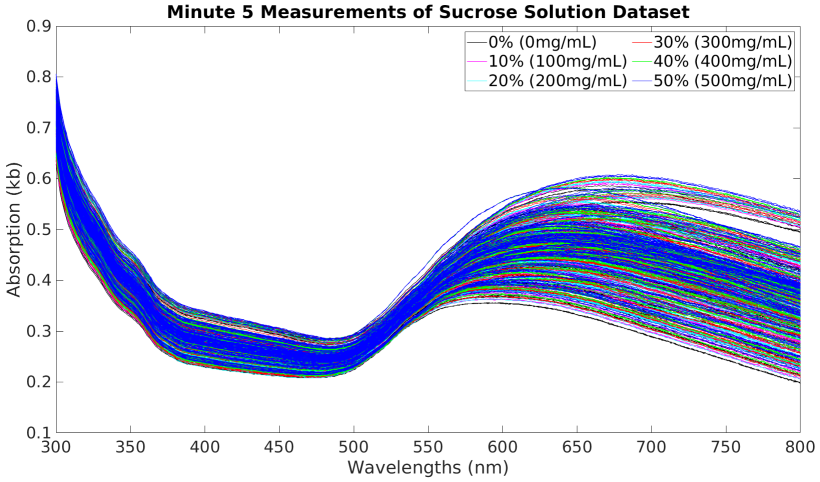


Figure A.2. Sucrose Concentrations Measurements at Minute 5 Graph

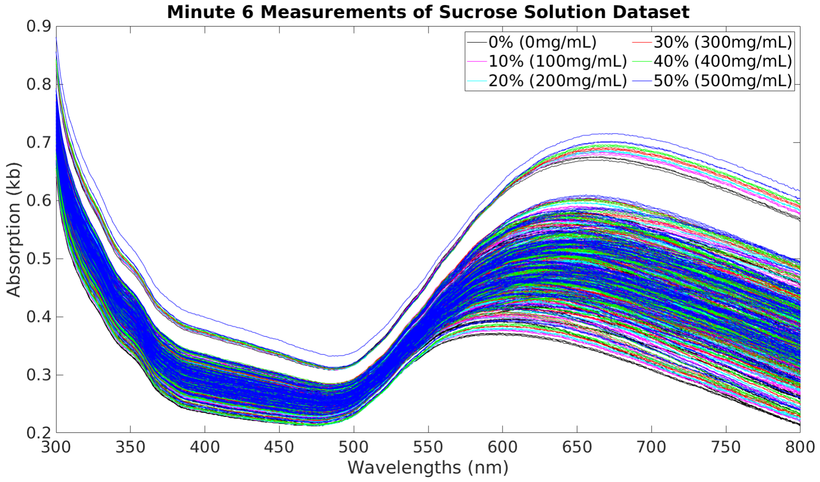


Figure A.3. Sucrose Concentrations Measurements at Minute 6 Graph

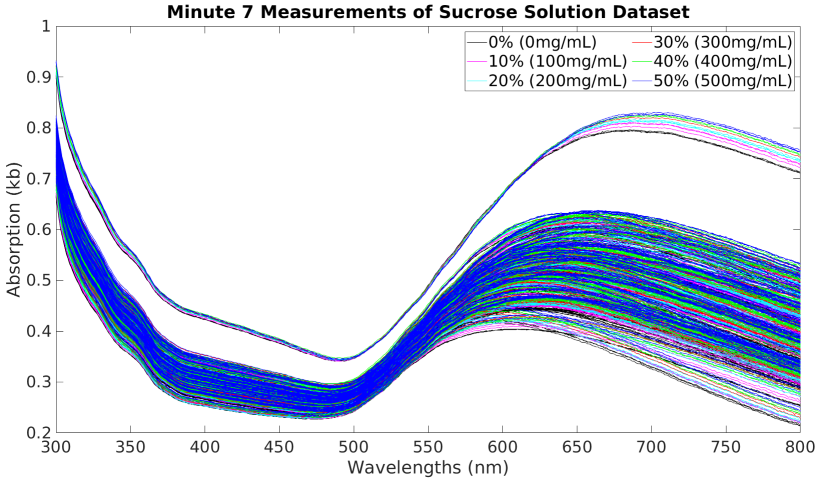


Figure A.4. Sucrose Concentrations Measurements at Minute 7 Graph

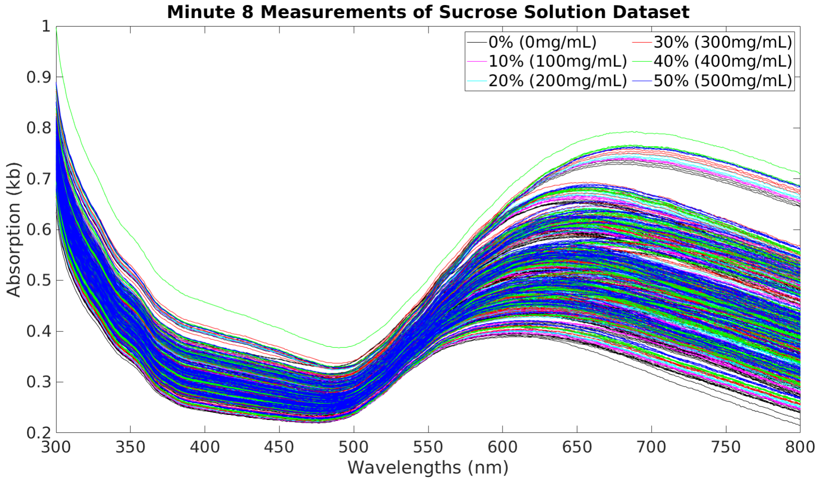


Figure A.5. Sucrose Concentrations Measurements at Minute 8 Graph

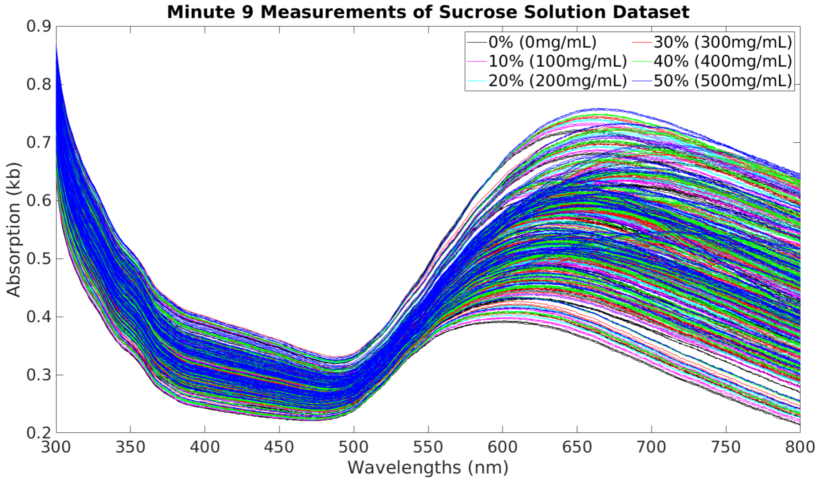


Figure A.6. Sucrose Concentrations Measurements at Minute 9 Graph

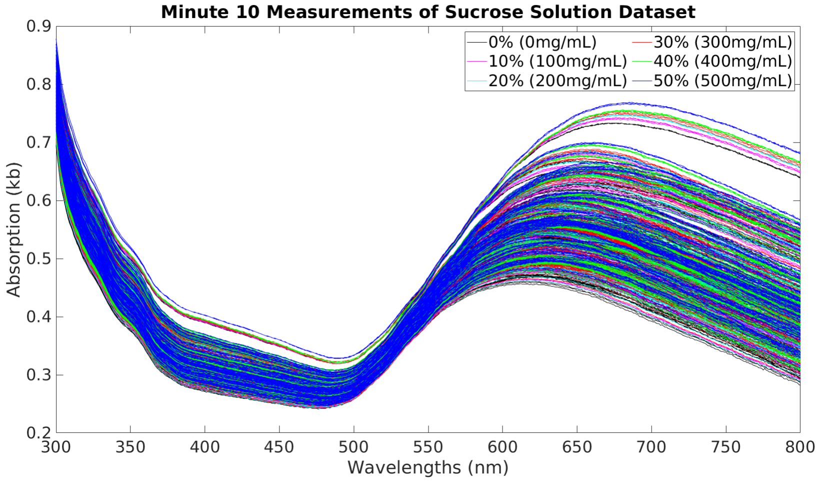


Figure A.7. Sucrose Concentrations Measurements at Minute 10 Graph

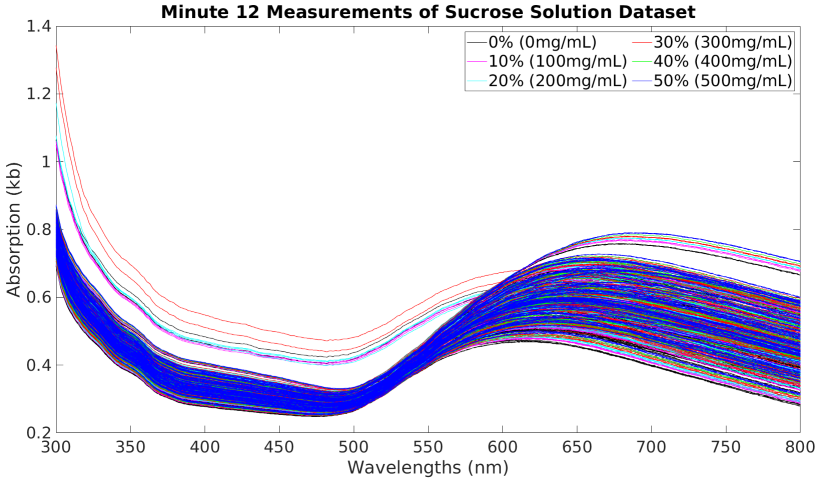


Figure A.8. Sucrose Concentrations Measurements at Minute 12 Graph

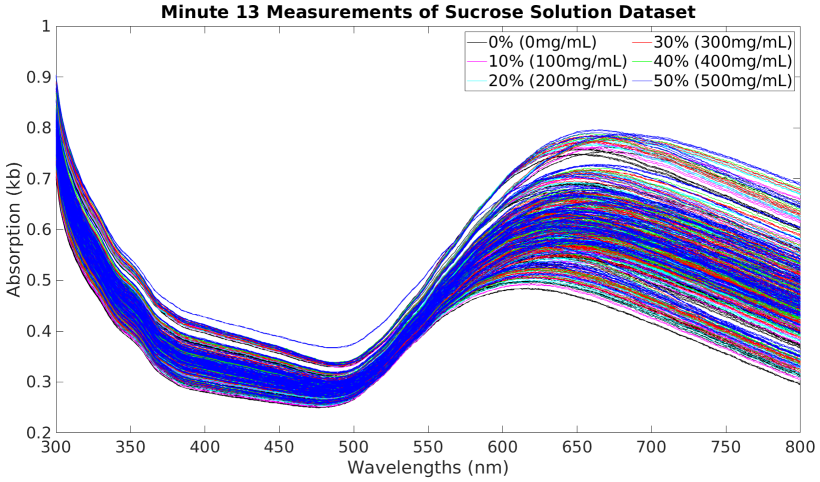


Figure A.9. Sucrose Concentrations Measurements at Minute 13 Graph

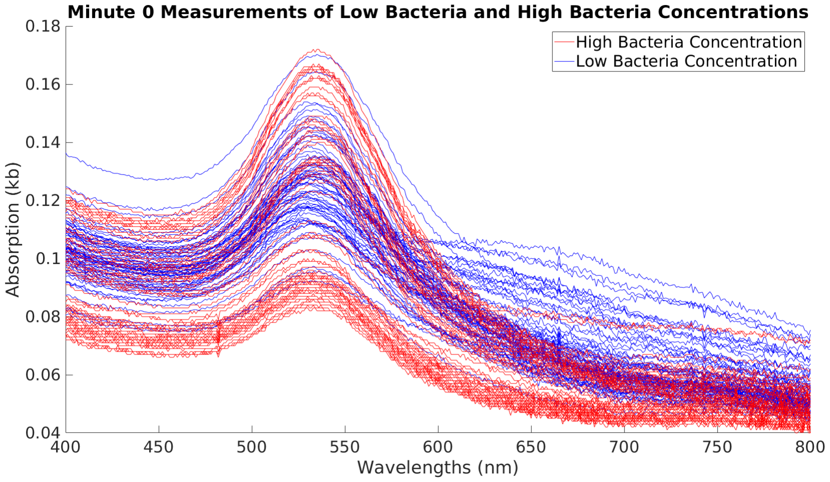


Figure A.10. Low Bacteria Concentration(10^2, 10^3, 10^4) and High Bacteria Concentration(10^5, 10^6, 10^7) Measurements at Minute 0

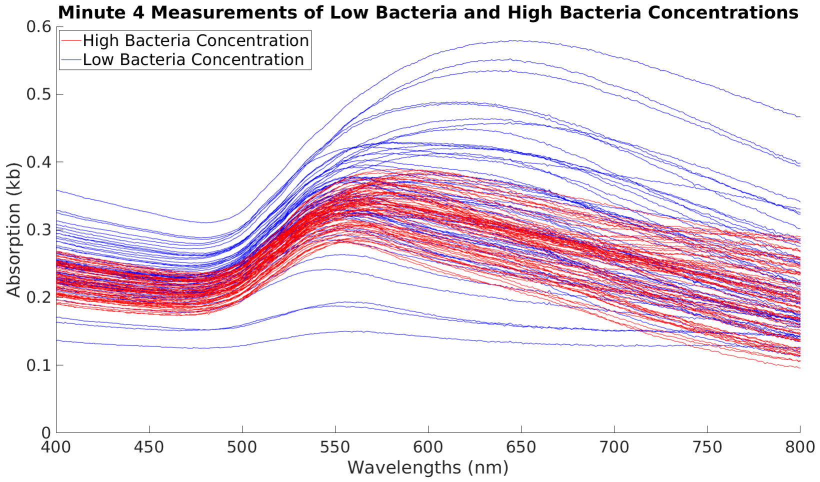


Figure A.11. Low Bacteria Concentration(10^2, 10^3, 10^4) and High Bacteria Concentration(10^5, 10^6, 10^7) Measurements at Minute 4

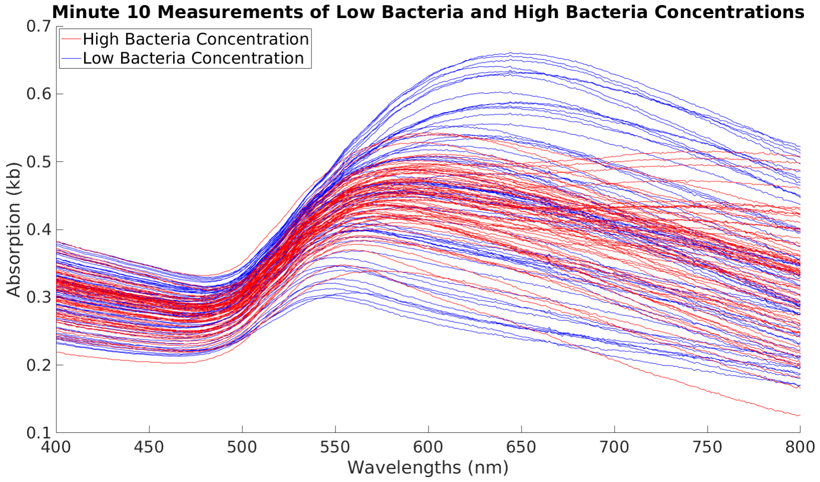


Figure A.12. Low Bacteria Concentration(10^2, 10^3, 10^4) and High Bacteria Concentration(10^5, 10^6, 10^7) Measurements at Minute 10

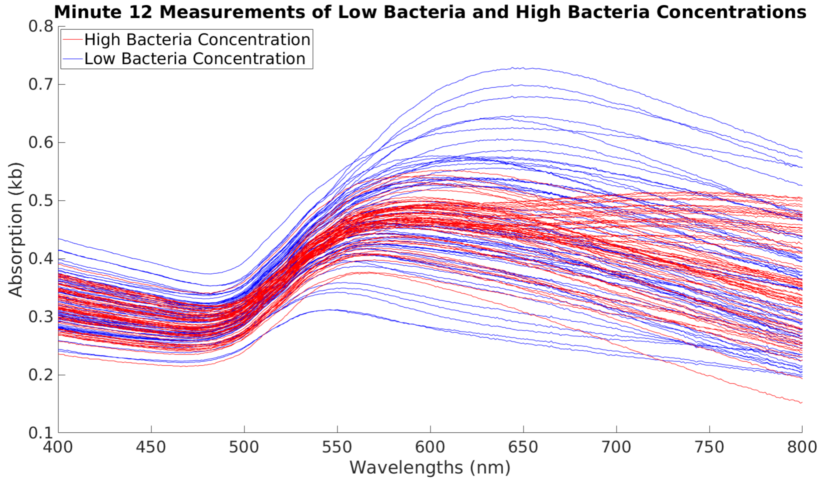


Figure A.13. Low Bacteria Concentration(10^2, 10^3, 10^4) and High Bacteria Concentration(10^5, 10^6, 10^7) Measurements at Minute 12

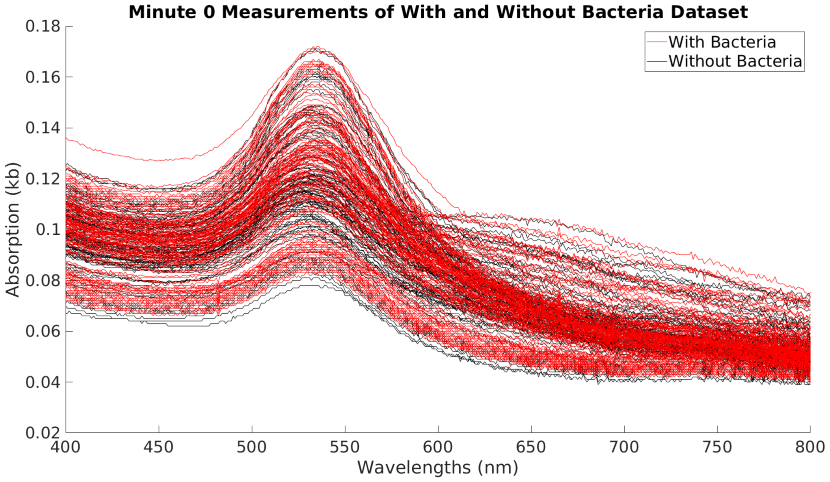


Figure A.14. With and Without Bacteria Solution Measurements at Minute 0

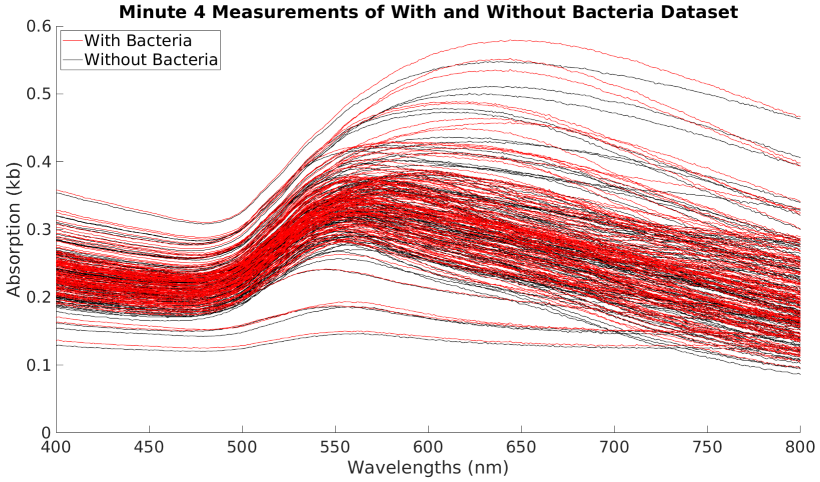


Figure A.15. With and Without Bacteria Solution Measurements at Minute 4

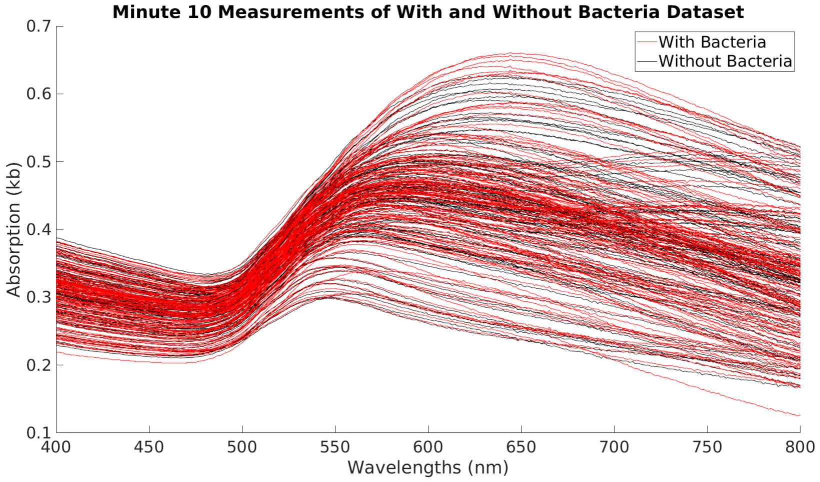


Figure A.16. With and Without Bacteria Solution Measurements at Minute 10

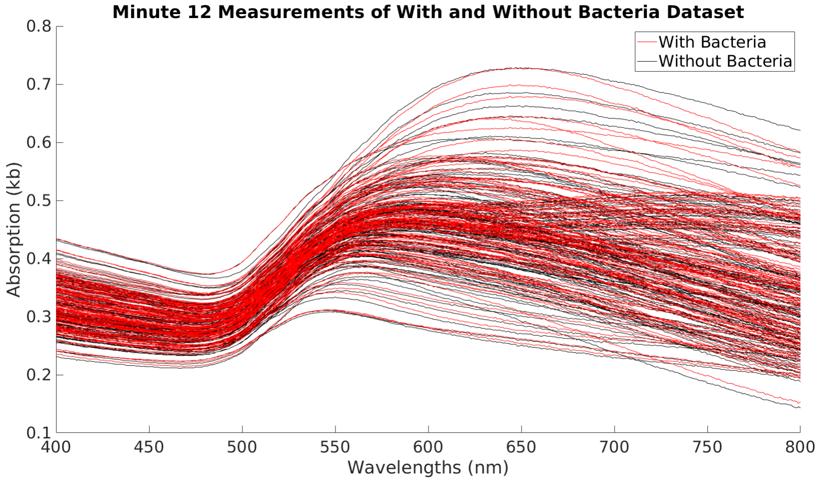


Figure A.17. With and Without Bacteria Solution Measurements at Minute 12