Improvement of Bacteria Detection Accuracy and Speed Using Raman Scattering and Machine Learning

Aseel Mandour

Thesis submitted to University of Ottawa
in partial fulfillment of
the requirements of the degree of Master of Science

Department of Electrical & Computer Engineering
Facult of Engineering
University of Ottawa

Abstract

Bacteria identification plays an essential role in preventing health complications and saving patients' lives. The most widely used method to identify bacteria, the bacterial cultural method, suffers from long processing times. Hence, an effective, rapid, and non-invasive method is needed as an alternative. Raman spectroscopy is a potential candidate for bacteria identification due to its effective and rapid results and the fact that, similar to the uniqueness of a human fingerprint, the Raman spectrum is unique for every material.

In my lab at the University of Ottawa, we focus on the use of Raman scattering for biosensing in order to achieve high identification accuracy for different types of bacteria. Based on the unique Raman fingerprint for each bacteria type, different types of bacteria can be identified successfully. However, using the Raman spectrum to identify bacteria poses a few challenges. First, the Raman signal is a weak signal, and so enhancement of the signal intensity is essential, e.g., by using surface-enhanced Raman scattering (SERS). Moreover, the Raman signal can be contaminated by different noise sources. Also, the signal consists of a large number of features, and is non-linear due to the correlation between the Raman features. Using machine learning (ML) along with SERS, we can overcome such challenges in the identification process and achieve high accuracy for the system identifying bacteria.

In this thesis, I present a method to improve the identification of different bacteria types using a support vector machine (SVM) ML algorithm based on SERS. I also present dimension reduction techniques to reduce the complexity and processing time while maintaining high identification accuracy in the classification process. I consider four bacteria types: *Escherichia coli* (EC), *Cutibacterium acnes* (CA, it was formerly known as *Propionibacterium acnes*), *methicillin-resistant Staphylococcus aureus* (MRSA), and *methicillin-sensitive Staphylococcus aureus* (MSSA). Both the MRSA and MSSA are combined in a single class named MS in the classification. We are focusing on using these types of bacteria

as they are the most common types in the joint infection disease.

Using binary classification, I present the simulation results for three binary models: EC vs CA, EC vs MS, and MS vs CA. Using the full data set, binary classification achieved a classification accuracy of more than 95% for the three models. When the samples data set was reduced, to decrease the complexity based on the samples' signal-to-noise ratio (SNR), a classification accuracy of more than 95% for the three models was achieved using less than 60% of the original data set. The recursive feature elimination (RFE) algorithm was then used to reduce the complexity in the feature dimension. Given that a small number of features were more heavily weighted than the rest of the features, the number of features used in the classification could be significantly reduced while maintaining high classification accuracy.

I also present the classification accuracy of using the multiclass one-versus-all (OVA) method, i.e., EC vs all, MS vs all, and CA vs all. Using the complete data set, the OVA method achieved classification accuracy of more than 90%. Similar to the binary classification, the dimension reduction was applied to the input samples. Using the SNR reduction, the input samples were reduced by more than 60% while maintaining classification accuracy higher than 80%. Furthermore, when the RFE algorithm was used to reduce the complexity on the features, and only the 5% top-weighted features of the full data set were used, a classification accuracy of more than 90% was achieved. Finally, by combining both reduction dimensions, the classification accuracy for the reduced data set was above 92% for a significantly reduced data set.

Both the dimension reduction and the improvement in the classification accuracy between different types of bacteria using the ML algorithm and SERS could have a significant impact in fulfilling the demand for accurate, fast, and non-destructive identification of bacteria samples in the medical field, in turn potentially reducing health complications and saving patient lives.

Acknowledgments

First, I would like to express my sincere gratitude to my supervisor Prof. Hanan Anis for her continuous support of my research, patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me through my courses, research work, and the writing of this thesis. Thank you, Prof. Hanan, for accepting me in your group and guiding me through my MSc.

I would like to thank my colleague Robert Hunter, with whom I worked throughout my MSc. I consider myself lucky to work cooperatively with such a knowledgeable colleague. He has been very helpful in sharing ideas and fruitful discussions. Thank you, Rob, for sharing your knowledge and being there for many questions. I would like to thank my colleague Mahamaya Deb for her helpful discussions and lab support. I would also like to thank the rest of our group for helping me along the way

Furthermore, I recognize that this research would not have been possible without the financial assistance of the School of Electrical Engineering and Computer Science and Graduate Studies at the University of Ottawa. Also, I would like to recognize my bursary and loan support from the government of Quebec.

I would like to express my deep appreciation for and pay humble respects to my parents. I could not imagine finishing this thesis without their support, especially my mother's prayers for me. I would like to thank my family for their support. I would like to thank my beautiful daughters, Talin El-Fiky, and Ruqayyah El-Fiky. Their beautiful smiles pushed me through hard times and towards reaching my goals. Finally, I would like to thank my lovely husband Eslam Elfiky for being there for me through the entire MSc. You have been very supportive and patient through hard times. I know how much you sacrificed during long nights of assignments and exams. I love you unconditionally.

Associated publications

The original contributions of the research work presented in this thesis resulted in the following paper [1]. Also, another paper was published during the work on this thesis and not directly related to the work in the thesis [2].

1. **Aseel Mandour**, Robert Hunter, and Hanan Anis,“ Improvement of Bacteria Detection Accuracy and Speed Using Raman Scattering and Machine Learning ” *to be submitted*, 2022.

   My contribution to this paper is developing the dimension reduction codes, performing the data processing, and writing the paper. The coauthors contributed in preparing the samples, capturing the data, implementation of the machine learning algorithm, and editing the paper.

2. Robert Hunter, Meshach Asare-Werehene, **Aseel Mandour**, Benjamin K. Tsang, and Hanan Anis,“ Determination of chemoresistance in ovarian cancer by simultaneous quantification of exosomes and exosomal cisplatin with surface enhanced raman scattering ” *Sensors and Actuators B: Chemical*, vol. 354, p. 131237, 2022.

   My contribution to this paper is capturing the Raman spectrum for the prepared samples by the other coauthors.

Contents

List of Figures

List of Tables

List of common acronyms

airPLS	**A**daptive **I**teratively **R**eweighted **P**enalized **L**east **S**quare
AU	**A**rbitrary **U**nits
BA	**B**lood **A**gar
BHI	**B**rain **H**eart **I**nfusion
CARS	**C**oherent **A**nti-Stokes **R**aman **S**cattering
CCD	**C**oupled **C**harged **D**evice
CMOS	**C**omplementary **M**etal **O**xide **S**emicondictor
cts/s	**C**oun**ts** per **s**econd
CW	**C**ontinuous **W**ave
DNN	**D**eep **N**eural **N**etwork
EC	**E**scherichia **C**oli
EnRFE	**En**hanced **R**ecursive **F**eature **E**limination
FS	**F**eature **S**election
FE	**F**eature **E**xtraction
GA-SVM	**G**enetic **A**lgorithm **S**upport **V**ector **M**achine
LIDAR	**LI**ght **D**etection **A**nd **R**anging
LS	**L**east **S**quare
ML	**M**achine **L**earning
MRMR	**M**inimum **R**edundancy **M**aximum **R**elevancy
MRSA	**M**ethicillin **R**esistant **S**taphylococcus **A**ureus
MSSA	**M**ethicillin **S**ensitive **S**taphylococcus **A**ureus
OVA	**O**ne **V**ersus **A**ll
OVO	**O**ne **V**ersus **O**ne
PA	**P**ropionibacteria **A**cnes (reclassified as *Cutibacterium acnes*)
PCA	**P**rincipal **C**omponent **A**nalysis

PLS	**P**enalized **L**east **S**quare
PLS-DA	**P**rojection to **L**atent **S**tructure **D**iscriminant **A**nalysis
PVP	**P**oly **V**inyl **P**yrrolidone
QF	**Q**uality **F**actor
R6G	**R**rhodamine 6G
RBF	**R**adial **B**asis **F**unction
RFE	**R**ecursive **F**eature **E**limination
RQK	**R**ational **Q**uadratic **K**ernel
SBS	**S**timulated **B**rillouin **S**cattering
SERS	**S**urface **E**nhanced **R**aman **S**pectrum
SG	**S**avitzky **G**olay
SMO	**S**equential **M**inimal **O**ptimization
SNR	**S**ignal to **N**oise **R**atio
SNV	**S**tandard **N**ormal **V**ariant
SOLIS	**SOL**ution for **I**maging and **S**pectroscopy
SRS	**S**timulated **R**aman **S**cattering
SSE	**S**um **S**quared **E**rror
SV	**S**upport **V**ector
SVM	**S**upport **V**ector **M**achine
TERS	**T**ip **E**nhanced **R**aman **S**pectrum
TSA	**T**ryptic to **S**oy **A**gar
VTPspline	**V**ector **T**ransformation **P**enalized spline

Chapter 1

Introduction

1.1 Motivation

Biosensors are sensors used for detecting a biological component and converting it into a measurable signal, e.g., electrical or optical signals. In the 1960s, Clark and Lyons pioneered the work of biosensors, using the first biosensor for oxygen detection [1]. The use of biosensors has since expanded to include applications such as biomedical diagnosis, food control, environmental monitoring, and drug discovery [2].

Raman spectroscopy has gained significant attention and is widely used in many different fields due to its effective and rapid results. In addition, it provides a unique spectrum for every material, similar to the uniqueness of a human fingerprint. This unique spectrum contains detailed information about the material's structure, chemical components, and molecular interactions. As a result, Raman spectroscopy is used in the fields and applications where a non-destructive image, chemical decomposition, and analysis are needed either for classification/regression purposes or quantitative/qualitative analysis. For example, Raman spectroscopy is used in the following fields: raw material verification [3–5], in vivo analysis and skin depth profiling [6–8], and doping effects [9–11].

Recently, Raman scattering for biosensing has been widely studied for medical appli-

cations since it is a non-destructive and non-invasive method. For example, the growing number of diabetic patients poses a significant problem in the medical field. Typically, blood glucose needs to be checked a few times per day. The current method, based on taking a blood sample and analyzing it, is not convenient. Significant progress has been made in developing in vivo glucose sensors based on enhanced Raman scattering, enabling low concentrations of blood to be measured consistently [12]. Such results are paving the way for Raman-based sensors for diabetic patients.

Bacteria are microorganisms that are everywhere around us. Some are good and vital for human existence. But others are harmful and can cause dangerous infectious diseases for humans and animals. As the number and types of harmful bacteria increase, it is crucial to be able to rapidly identify the bacteria type so the correct antibiotic can be used to treat a bacterial infection.

In this thesis, I focus on using Raman spectrum to identify different types of bacteria. This technique overcomes the processing time disadvantage of bacterial cultural methods. Indeed, using Raman spectrum provides nearly instantaneous results, compared with the days needed when using the bacterial culture method [13]. Such savings in the processing-time can result in the reduction of health complications for patients.

Using Raman spectroscopy to identify bacteria does pose a few challenges. First, the Raman scattered signal is typically a very weak signal. Hence, the feasibility of different applications depends on significant enhancement of the Raman spectrum. Different mechanisms can be used for the enhancement, such as the surface-enhanced Raman scattering (SERS) mechanism, which provides a significant gain to the scattered signal that is proportional to the fourth power of the input field. Details on the enhancement mechanism are discussed in Chapter 2.

Also, the Raman spectrum is typically captured for different bacteria types to enable multiplexing. Moreover, the Raman spectra are typically non-linear due to the correlation between the Raman peaks and least squares (LS) techniques used for regression may fail

for such problem or result in an overfitted solution. Furthermore, many samples are captured with a large number of wavenumbers or dimensions, which adds to the complexity of the classification process of different bacteria types. Hence, advanced multi-variate signal analysis algorithms are needed for the classification process. Partial least squares or projection to latent structures (PLS) technique is one of the most popular multi-variate techniques used for Raman spectroscopy; it reduces the dimensions of the Raman spectra by projecting the wavenumbers into a smaller subset of predictors. While the PLS is a powerful technique for regression models, it may fail in the presence of the non-linearities of the Raman spectrum and not achieve an acceptable identification accuracy for different bacteria types.

Machine learning (ML) algorithms based on kernel methods have been proposed for the identification of the non-linear data. In the kernel methods, the non-linear data is mapped from the non-linear space into a higher dimensional space where the data can be linearly separated. Among different ML algorithms, the support vector machine (SVM) algorithm has been widely studied in the analysis of Raman spectra in the biological field as it is known for its robustness in the prediction process. For example, it has been used in breast cancer diagnosis and the monitoring of blood glucose levels [14, 15]. While the SVM ML algorithm is a robust algorithm, it suffers from high complexity for large data set sizes.

In this thesis, I use the SVM ML algorithm for the multivariate analysis of the Raman spectra for different bacteria types. In my research group at the University of Ottawa, we are working on two main ways to improve classification accuracy in identifying bacteria using SERS: first, by using SVMs and pre-processing using the adaptive iteratively reweighted penalized least square (airPLS) algorithm; second, by reducing the system complexity and the processing time of the ML algorithm by performing dimension reduction on the data set. The dimension reduction is performed on the samples via quantitative analysis, and on the features using the recursive feature elimination (RFE) technique, as explained in more detail in the following chapters.

1.2 Organization of the thesis

The rest of this thesis is organized as follows.

In Chapter 2, I introduce the basics of Raman scattering and mention a few applications. Then, I explain the enhancement of the Raman scattering. Biosensors are also briefly discussed.

Chapter 3 details the main steps performed in the identification process of different types of bacteria. First, I explain the SVM algorithm used for the bacteria classification. Also, the data pre-processing steps are explained, including normalization, baseline correction, and blank subtraction. In addition, I discuss binary and multiclass classification processes. Finally, I explain dimension reduction techniques for both the sample size and feature (or, wavenumber dimensions).

In Chapter 4, I present the classification results for binary classification and multiclass classification for four bacteria types: *Escherichia coli* (EC), *Cutibacterium acnes* (CA, it was formerly known as *Propionibacteri acnes*), *methicillin-resistant Staphylococcus aureus* (MRSA), and *methicillin-sensitive Staphylococcus aureus* (MSSA). Also, I present the classification accuracy after applying dimension reduction techniques on the processed samples.

Finally, Chapter 5 concludes the thesis by summarizing the key contributions made and lists some of the potential improvements and future research work.

1.3 Original contributions

The original contributions of this thesis are summarized as follows:

- Using binary classification, I present the simulation results for three binary models: EC vs CA, EC vs MS, and MS vs CA. Using the full data set, a classification accuracy of more than 95% was achieved for all three models. Then, results based on complexity reduction in both the sample and feature dimensions are presented. First, using less than 60% of the original data set sample size, classification accuracy was maintained

at more than 95% for all three models. Then, the feature reduction was applied to calculate the feature weights and reduce the complexity in the feature dimension. A small number of features were heavily weighted compared with the rest of the features. As a result, the number of features used in the classification could be significantly reduced, to 10 features, for the classification. Using both reduction dimensions, classification accuracy was 96.35% for the EC vs CA model, 91.85% for the EC vs MS model, and 97.67% for the MS vs CA model. Moreover, the full data set and the reduced data set needed approximately 120 mins and 2.6 mins, respectively, to train and process 100 iterations. This corresponds to more than 98% reduction in the processing time. Similarly, we achieve more than 98% reduction in the memory requirements.

- Furthermore, I present the classification accuracy using multiclass classification. Using the complete data set, the classification accuracy was more than 90%. By applying sample reduction, the input samples could be reduced by more than 60% and maintain a classification accuracy higher than 80%. By applying the reduction on the feature dimension to reduce the complexity on the features and using only the 5% top-weighted features, a classification accuracy of more than 92% could be achieved. Finally, by combining both the reduction dimensions and the classification accuracy for the reduced data set, the classification accuracy was more than 90% for a significantly smaller data set. Moreover, we achieve more than 98% reduction in the processing time and memory requirements using the reduced data set compared to the full data set for the multiclass cass.

Chapter 2

Introduction to Raman spectroscopy

2.1 Overview

In the previous chapter, I discussed the importance of Raman spectroscopy in the biological field, where it can be used to focus on the insight of the bacteria cell structure [16, 17]. In this way, Raman spectroscopy can be used to classify and identify different types of bacteria based on the fact that every bacteria has its unique Raman fingerprint. In this chapter, I review the basics of the Raman scattering and the enhancement of the Raman signal.

The rest of this chapter is organized as follows. In Section 2.2, light scattering and more specifically Raman scattering is discussed. Then, enhancement of Raman scattering is explained in Section 2.3. Finally, biosensors are briefly discussed in Section 2.4.

2.2 Raman scattering

Raman spectroscopy is based on light scattering upon falling on a medium. When a photon hits a molecule, the light scatters in two possible ways as shown in Figure 2.1. The first type of scattering is elastic scattering, e.g., Rayleigh scattering, where the energy of the scattered photon equals the energy of the incident photon. The energy has a direct relation

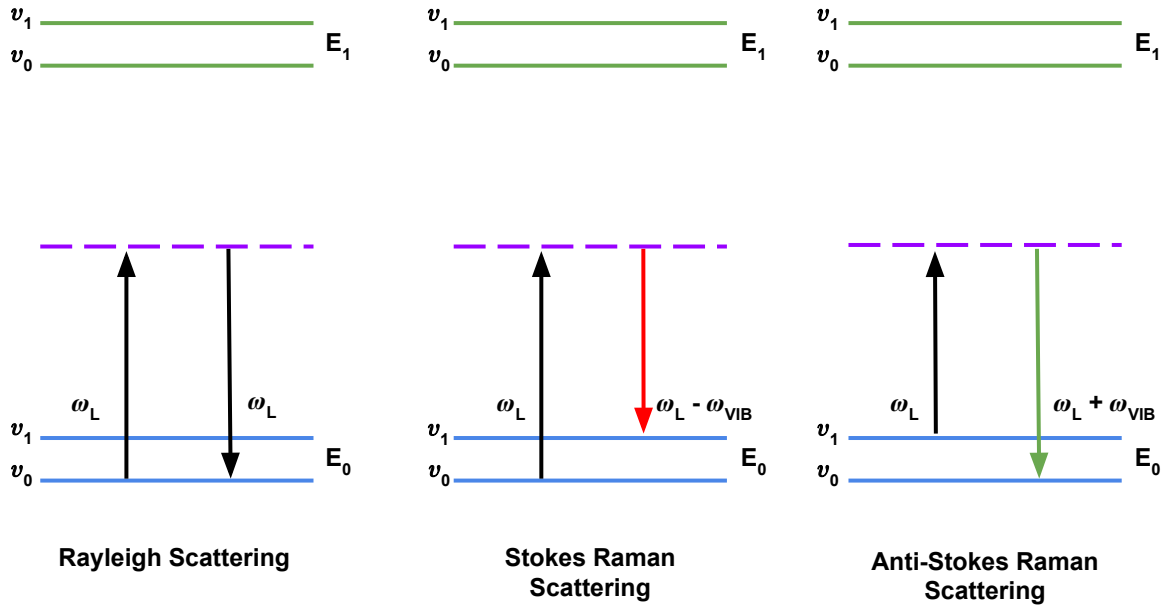


Figure 2.1: Rayleigh scattering, Stokes Raman scattering and anti-Stokes Raman scattering [18]

with the light's frequency given by the following equation:

$$E = h\omega \tag{2.1}$$

where E is the light's energy, h is plank's constant, and ω is the light's frequency. As a result, we can say that the scattered photons have the same frequency as the incident ones.

Inelastic scattering occurs when the energy of the scattered photons differs from the incident ones. The scattered light can have lower energy as the material's molecules absorb an amount of the incident energy and result in forming Stokes components. The scattering can also result in higher energy, where the material's molecules lose some energy, and result in forming anti-Stokes components. In other words, the scattered light can have either lower or higher frequency from the incident light due to the inelastic scattering. Raman scattering is an inelastic scattering first discovered by the Indian physicist C.V. Raman in 1928 and named after him.

Raman spectroscopy is an application based on Raman scattering for an incident frequency of ν_s. The Raman scattered photons from the vibrating molecules form the Raman spectrum consisting of Stokes and anti-Stokes shifts with absolute frequency of $\nu_s \pm \nu_m$. Under typical conditions, most of the molecules are in the ground state, and hence the Stokes signal is stronger than the anti-Stokes signal. So, the Raman spectroscopy is based on the Stokes waves. The captured Raman spectra are usually plotted versus the shifts instead of the absolute frequencies which are called wavenumbers or features having units of cm^{-1}.

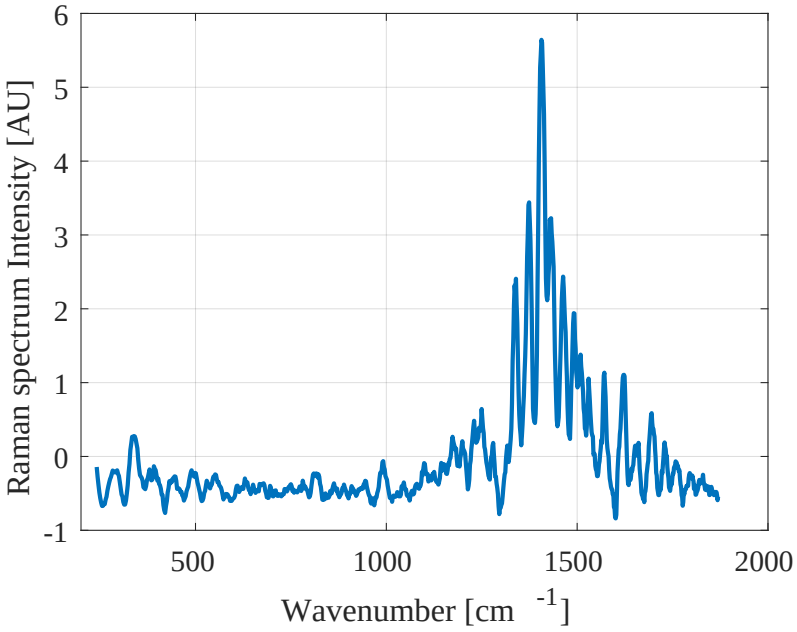


Figure 2.2: Processed Raman spectrum of the E. coli bacteria

Figure 2.2 shows an example of the Raman spectrum, after processing, for the E. coli sample where the y-axis represents the intensity of the spectrum in arbitrary units (AU) and the wavenumbers on the x-axis in cm^{-1}. The spectrum is distributed over a large number of wavenumbers/features. Each wavenumber represents the wavelength position of the scattered Raman photons that is characteristic of a certain molecular bond in the molecule [19]. Hence, it contains important information about the chemical contents, chemical bonds

and structure of the molecules, such as protein, lipids, acids, etc. [20, 21]. As a result, the resulting spectrum will have information that can be used to classify different types of bacteria. The Raman spectrum consists of a large number of features where for biological samples approximately 90% of the peaks are found in the "fingerprint" spectral region, which covers the wavenumbers in the range of 250 cm^{-1} to $\sim$1800 cm^{-1} [22]. Based on our experimental setup, this corresponds to 885 features; such a large number of features affects the training time and complexity of the ML algorithm, which will be addressed in the next chapter.

The intensity of the Stokes Raman scattering signal is proportional to the following [23]:

1. Intensity of the incident light at frequency ν_s.
2. Number of scattering molecules.
3. Frequency term of the form $(\nu_s - \nu_m)^4$.
4. State of polarization of the incident signal.

Hence, we should use a high-intensity and high-frequency laser source and a large sample size to achieve a high-intensity scattered signal. However, using a high-intensity laser source is not possible for Raman spectroscopy, since it can damage the sample under test. Moreover, using a high-frequency or short wavelength depends on the absorption frequency band for the sample. In our results, a laser with wavelength of 785 nm is used in the experimental setup. Also, using a very large sample size will render the system unpractical. Furthermore, the Raman scattering is 2-3 orders of magnitude lower than the Rayleigh scattering. As a result, the resultant Raman spectrum is a very weak signal, which has severely limited its usage for many applications in the past.

Several techniques have been proposed to enhance the Raman signal [24–26]. The highest amplification of the Raman signal comes from SERS, in which single molecules can be detected due to the large enhancement. In our work, the SERS method is used to enhance the spectrum collected from the bacteria, as discussed further in the next section.

2.3 Surface-enhanced Raman scattering (SERS)

In 1974, SERS was first observed in the Raman spectra of pyridine on roughened silver electrode [12]. However, it was not recognized whether such high intensity of the Raman signals was enhanced or due to a new phenomenon. It was believed to be attributed to the increased surface of the silver electrode. In 1977, two groups independently confirmed that the concentration of scattering species could not account for the enhanced signal; these findings led to the discovery of SERS. Each group proposed a mechanism for the observed enhancement: Jeanmaire and Van Duyne proposed an electromagnetic effect, while Albrecht and Creighton proposed a chemical theory [27]. The electromagnetic theory is based on the excitation of localized surface plasmons, and the chemical theory is based on the formation of charge-transfer complexes.

After years of debate about the exact mechanism of the enhancement effect of SERS in the literature, it is now accepted that the dominant mechanism for the enhancement is the electromagnetic mechanism. This is attributed to the theoretical enhancement factor of approximately 10^{11} and 10^{3} for the electromagnetic mechanism and chemical theory, respectively. Hence, we will focus on the explanation of the electromagnetic mechanism in more detail.

Figure 2.3 illustrates a basic example for SERS where the incident light strikes a sample and metal particles attached to the surface. When an electric field is incident on metal particles, multipoles with different orders are induced. When the particles have a diameter that is much smaller than the wavelength of the incident field, e.g., gold nanoparticles, only the simple dipolar plasmon contribution is considered.

When an incident field E_i strikes the surface of the sphere, the induced field E_d at the surface of the sphere is given by [28]:

$$E_d = \frac{\epsilon_1 - \epsilon_2}{\epsilon_1 + 2\epsilon_2} E_i, \tag{2.2}$$

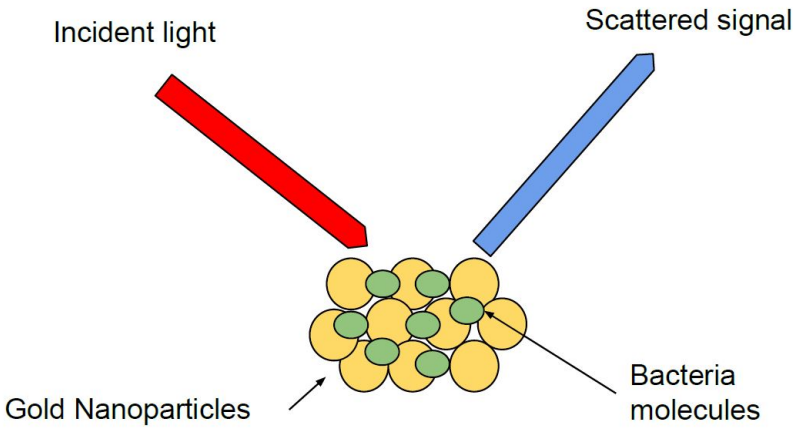


Figure 2.3: Basic example for SERS

where ϵ_1 and ϵ_2 are the complex frequency-dependent permittivity of the metal and the relative permittivity of the ambient phase, respectively. At a certain frequency, we can find the resonance condition where $\epsilon_1 = -2\epsilon_2$. Hence, the induced field by the metal particle can be represented as:

$$E_d = gE_i, \tag{2.3}$$

As a result, the sample molecule adsorbed at the surface of the metal nanoparticle will be excited by E_d. So, the Raman scattered light E_s produced by the molecule will be proportional to E_d. It should be noted that the field is enhanced twice where the Raman scattered light is also enhanced by the metal nanoparticles due to the same mechanism that excited the incident light. Therefore, the enhanced scattered light is given by:

$$E_{SERS} \propto g'E_d, \tag{2.4}$$

where g' represents the gain for the Raman shifted signal where it is not generally the same value as g. Then, the total SERS scattered signal will be given by:

$$E_{SERS} \propto gg'E_i, \tag{2.5}$$

When the frequency shift of the Raman signal is relatively small, g can be approximated to be equal to g'. Also, the intensity is proportional to the square of the absolute value of the field. Hence, the intensity of the SERS signal is given by:

$$I_{SERS} \propto |g|^4 I_i, \tag{2.6}$$

where I_i is the intensity of the input field. From that, we can understand the significant increase in the intensity where a small input field will be enhanced by a factor of $|g|^4$. The following factors need to be taken into account in SERS:

- The choice of metal surface is governed by the plasmon resonance frequency. For the visible and near-infrared radiation, silver and gold are typically used since their plasmon resonance frequencies are within these frequency bands. Although copper has also been used as a metal surface, it is reactive and more difficult to handle.

- The size of the metal surface affects the enhancement. If the size of the metal surface is comparable to the incident wavelength, multipoles will be excited which are non-radiative, and the overall enhancement will be degraded. Hence, the upper bound on the size of the metal is based on the incident wavelength. Also, if the metal particle is too small and comparable to the size of the molecule, the conductance will be reduced and radiation will also be degraded. Typically, the optimum size of the nanoparticles is in the range of 10 nm to 100 nm.

- The polarization of the incident field with respect of the metal particle greatly affects the enhancement. As shown in Figure 2.4, when the incident field is polarized along the inter-particle axis of a metal particle, huge enhancement can occur in the gap between the particles. This effect doesn't occur if the polarization is orthogonal.

Finally, two principal configurations are used for SERS: intrinsic SERS and extrinsic SERS. Figure 2.5 illustrates the difference between both configurations [29].

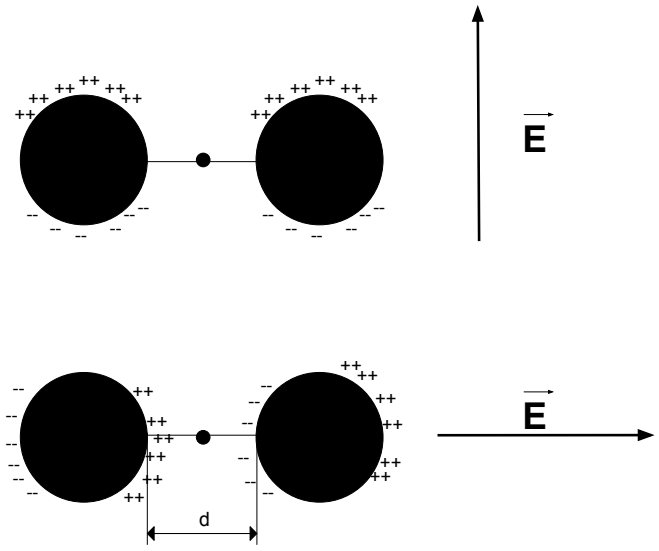
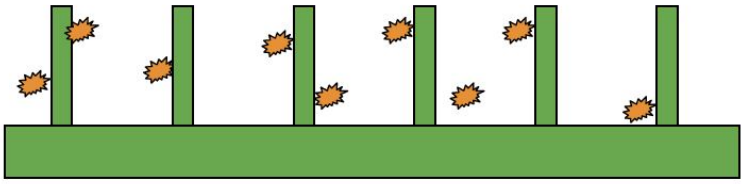
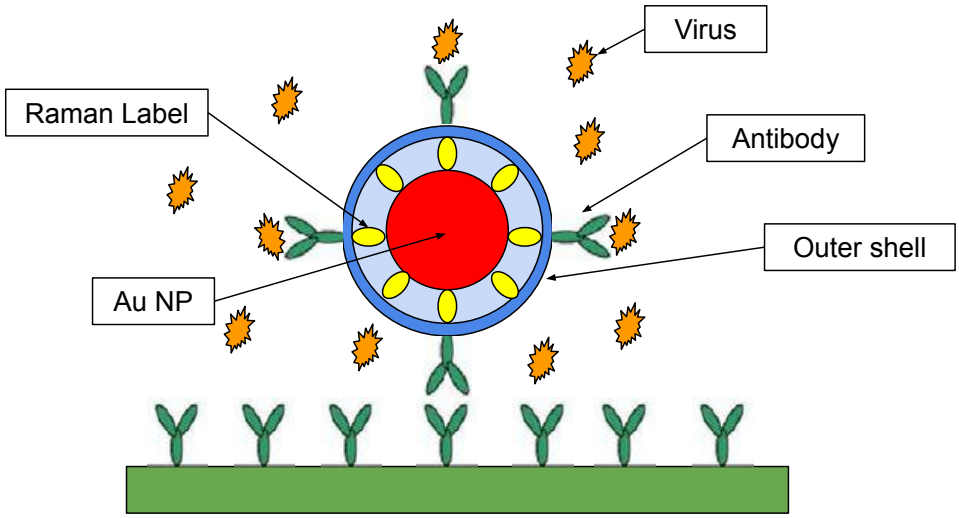


Figure 2.4: Illustration of the polarization orientation of the incident field on the metal nanoparticles.

In intrinsic SERS, the analyte is directly applied to the nanoparticles. Thus, the measured Raman spectrum can be directly used to identify the sample. On the other hand, extrinsic SERS can be used when intrinsic SERS is susceptible to failure. In extrinsic SERS, a Raman reporter molecule is used to generate a signal for detection. For example, a Raman reporter molecule is immobilized between the nanoparticle and an outer shell. Then, the virus specimens are detected using a sandwich structure. In our work, the measurements are based on an intrinsic SERS configuration.



(a) Intrinsic SERS



(b) Extrinsic SERS

Figure 2.5: Principal configurations of SERS [29].

2.4 Biosensors

SERS has been used in applications and fields such as biosensors, chemical detection, and single molecules SERS. My focus in this thesis is on the biosensor application of SERS for bacteria identification.

A biosensor is a sensor used for detecting a biological component and converting it into a measurable signal such as electrical or optical signals. Biosensor applications include biomedical diagnosis, food control, environmental monitoring, and drug discovery [2]. Currently, SERS-based biosensors have been proposed for the detection of various biological

samples and diseases such as Alzheimer's disease [30, 31], cancers [32, 33], and Parkinson's disease [34, 35]. A typical biosensor consists of the following stages [1]:

1. Bioreceptor: The bioreceptor role is to interact with the analyte under test and convert the interaction to a measurable signal for the transduce, e.g., enzymes, cells, deoxyribonucleic acid (DNA) and antibodies. This process is called bio-recognition.

2. Transducer: The transducer is needed to convert the bio-recognition event into a measurable signal, e.g., piezo-electric effect and optical effects. In our work, SERS based on gold nanoparticles is part of the transducer.

3. Electronics: Electronics are needed to process the converted signal and prepare it for display, e.g., using a Raspberry Pi.

Chapter 3

Machine learning for bacteria identification

3.1 Overview

In the previous chapter, Raman spectroscopy and its importance in the medical field, specifically as a biosensor, were discussed. In this chapter, I explain the ML algorithm used in the process to differentiate between different types of classes, e.g., bacteria types. The ML algorithm is needed to overcome the challenges associated with the processing of the captured Raman spectra where the signal is contaminated by noise and has high dimensionality. More importantly, the Raman features are correlated, i.e., non- linear, and so least square (LS) techniques can fail to provide a solution or can be overfitted. Hence, using ML enables rapid and accurate identification accuracy.

This chapter also discusses the classification process steps in detail. This classification process consists of several steps, starting with capturing the Raman spectrum and ending with the classification stage as shown in Figure 3.1. The spectrum capturing stage is discussed in Chapter 4 because it is specific to bacteria identification and not generic to any type of classes. The classification process is based on the SVM ML algorithm based on

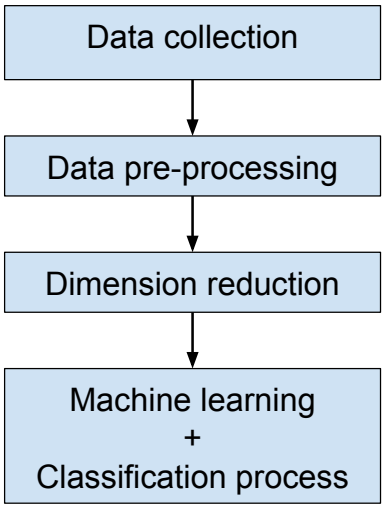


Figure 3.1: Main steps for the classification process.

the training using the input Raman spectra or data set. However, it should be noted that the raw data set is corrupted with different noise sources, which necessitates adding the data pre-processing stage to overcome the noise effects on the efficacy of the classification. Moreover, the data set is typically large, given the large number of samples and wavenumbers. As a result, the complexity of the classification process can be significantly increased. Hence, dimension reduction techniques are used to reduce the size or dimensions of the input data set without affecting the accuracy of the system while achieving a significant reduction in the complexity of the system.

The rest of this chapter is organized as follows. Section 3.2 explains the SVM ML algorithm. In Section 3.3, the data pre-processing steps are explained. In Section 3.4, different classification types are explained. Finally, dimension reduction techniques are explained in Section 3.5.

3.2 Machine learning algorithm

Machine learning can be seen as that part of artificial intelligence that aims to emulate the behaviour of human learning or human intelligence. ML algorithms are trained on sample data and then use the trained data pattern to make a future prediction or a decision and improve through experience. In the last decade, different ML algorithms have been proposed in the literature that can be used in applications such as disease diagnostics [36,37], speech recognition [38,39], image recognition [40,41], traffic prediction [42,43], and optical network planning [44,45].

ML algorithms differ from each other based on whether the data used is linear or nonlinear, i.e., where the data can be separated using a linear boundary or a non-linear boundary. Figure 3.2 illustrates the difference between a linear boundary and a non-linear boundary for two input classes.

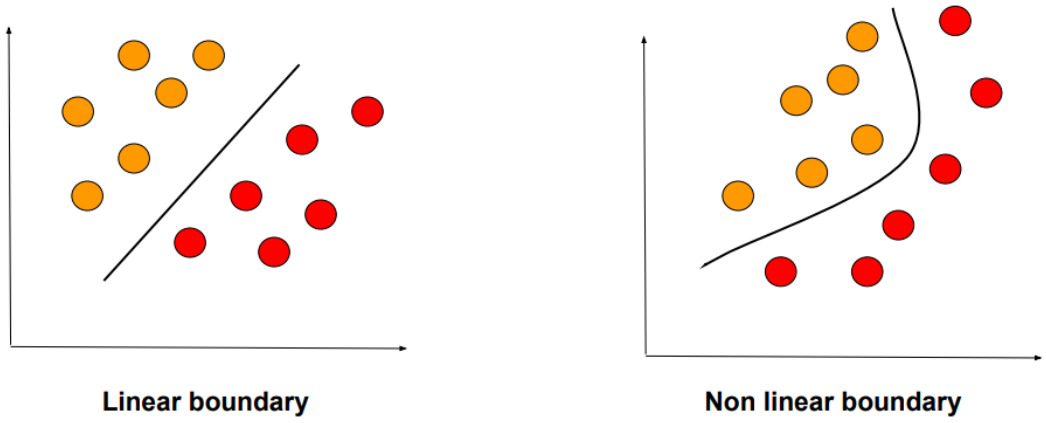


Figure 3.2: Difference between linear boundary and non-linear boundary for binary classes.

ML algorithms can also be classified depending on whether they are supervised or not. In a supervised algorithm, each input or observation in the data set has a corresponding output or label that is used in the training of the ML algorithm, e.g., a decision tree algorithm [46], and neural networks [47]. In an unsupervised algorithm, the labels are not used in the training of the ML algorithm, e.g., principal component analysis (PCA) [48],

K-Means clustering [47]. Furthermore, ML algorithms can be divided depending on the nature of the desired output of the algorithm. In a classification algorithm, the classification process predicts a discrete output; in a regression algorithm, the regression process predicts a continuous one [47, 49, 50].

In this thesis, I use the SVM ML algorithm for the classification results. SVMs have been proposed by Vladimir N. Vapnik as a supervised learning model to analyze data as a linear classifier [51], with a modification that an SVM can be used as a non-linear classifier [52]. SVMs have been applied to applications such as handwritten digit and text recognition [51,53], classification of images [54,55], and classification of satellite data [56,57].

3.2.1 Support vector learning as a linear classifier

An SVM is a supervised ML algorithm where the output y of each observation x is known. This knowledge is used in the training process of the algorithm where the boundary function is designed to clearly separate classes.

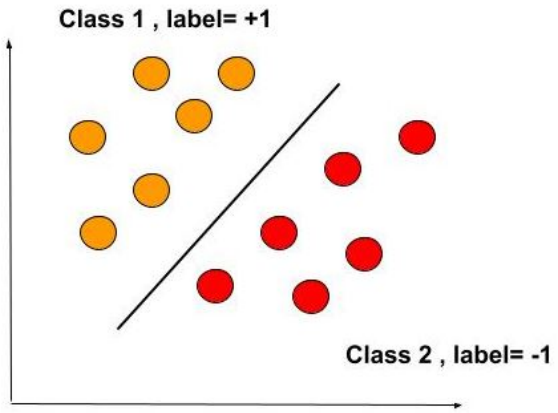


Figure 3.3: An example of a binary classification model.

Assume that we have a data set for two different classes, e.g., bacteria types, that are linearly separable and arranged in a matrix $X_{n\times m}$, where n is the number of observations or samples and m is the number of features. Each observation x is assigned to a label or output y where $y \in$ [-1,+1] as shown in Figure 3.3. This data set is then divided into two

sets: a training and a testing set, each with a selected percentage, e.g., 60% and 40%. It should be noted that the training set should contain more samples than the testing set, so the MLA model will be well trained and achieve accurate results.

As there are multiple possibilities for the linear boundary or the decision function between the two classes, it is important to find the optimum decision function. The decision function can be expressed as follows [58, 59]:

$$f(x) = w.x + b, \tag{3.1}$$

w and b are the weight vector and the offset of the boundary, respectively. Figure 3.4 shows

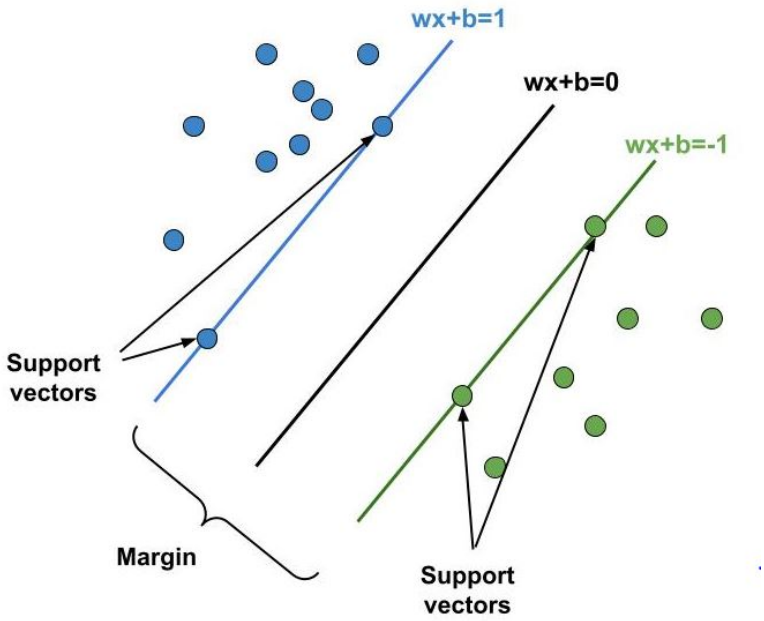


Figure 3.4: Example of a binary model showing the linear hyperplane and the margin between the two classes [60, 61].

the boundary margin for two classes. The optimum boundary should achieve the maximum margin and minimum risk or loss between the two classes. The loss can be simply defined as the difference between the actual and the predicted output of a certain input x_i. It can

be expressed as follows:

$$L = \sum_{i=1}^{N} max[0, 1 - y_i f(x_i)] \tag{3.2}$$

For example, if the predicted label for an observation x is correct for binary classification, the empirical loss will be equal to zero since both of $f(x_i)$ and y_i will have the same sign. If the observation is not correct, the loss will be greater than zero.

Using the margin's equations of both classes as shown in Figure 3.4 [61]:

$$f(x) = \begin{cases} wx + b \geq +1 & \text{for class } 1, y = +1 \\ wx + b = 0 & \text{at decision boundary} \\ wx + b \leq -1 & \text{for class } 2, y = -1 \end{cases} \tag{3.3}$$

By subtracting both equations for the two classes, we can have a simple expression for the maximum margin as $\frac{2}{||w||}$. Hence, the weight vector must be minimized, which is obtained by the Lagrange multipliers or the support vectors α based on the dual form, which is expressed by the following equation [58, 62]:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{3.4}$$

By using Eq. (3.4) into Eq. (3.1), the final form of the decision function is [58]:

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i (x_i.x) + b \tag{3.5}$$

where $x_i.x$ refers to the dot product between x and x_i.

3.2.2 Support vector machine as a non-linear classifier

The Raman spectrum consists of correlated features, so they cannot be separated linearly. In order to use the SVM as a non-linear classifier, the samples need to transferred to a higher

dimensional space called the feature space where they can be linearly separated [63–66].

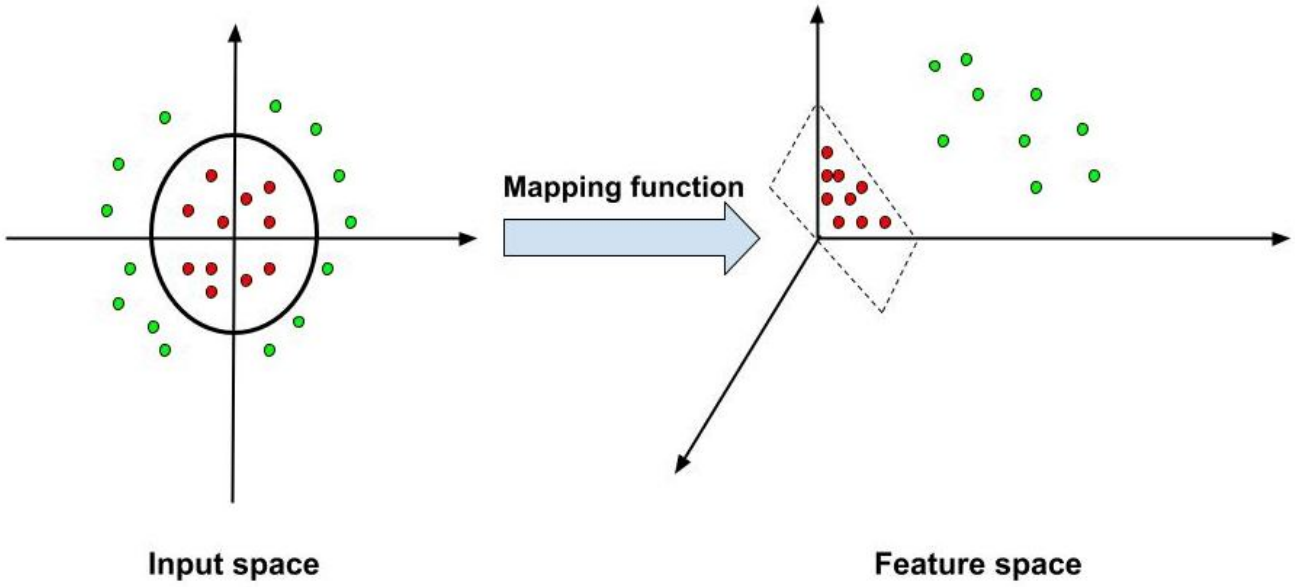


Figure 3.5: Illustration of transforming the data from the input space to the feature space using mapping function

Figure 3.5 illustrate the mapping of the non-linear data to the new space where a mapping function Φ is applied to the input spectrum. As a result, the decision function in Eq. (3.5) will be adjusted to the following form:

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i (\Phi(x_i).\Phi(x)) + b \tag{3.6}$$

However, this method has a high complexity even for only two classes as the mapping function Φ is a non-linear function. Also, it should be noted that the Raman spectrum has high dimensionality in the input space. Hence, it is complex to transform it into the feature space using the mapping function. As a result, another method is used based on using a kernel function called the kernel trick [52, 58, 67]. This process calculates the dot product in the feature space between different observations in the data matrix set [66, 68].

Several types of kernels have been proposed in the literature, such as linear, polyno-

mial kernel, radial basis function (RBF), and Gaussian kernel, etc. [69–71]. The rational quadratic kernel (RQK) is the kernel used in the design of the ML algorithm used in our results. It can be expressed using the following equation [72]:

$$k_{RQK} = \sigma^2(1 + \frac{(x_i - x_j)^2}{2\beta l})^{-\beta} \tag{3.7}$$

where x is the collected Raman spectrum, σ^2, β, l represent the kernel hyper-parameters. Therefore, the decision function can be written as:

$$f(x_i) = \sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b, \tag{3.8}$$

where m is the number of features, α and b are the optimized parameters from the SVM training function, and K is the kernel matrix.

The training process of the SVM algorithm is divided into two stages:

1. Finding the optimum values of the kernel hyper-parameters.

2. Finding the best hyperplane that separates between the two classes achieving the maximum margin between the two classes.

These two stages are dependent on each other and performed through an iterative process. In other words, the kernel parameters are optimized when the best hyperplane is found and vice-versa.

For the optimization of the hyper-parameters, different methods have been proposed in the literature, such as grid search, particle-swarm, and genetic algorithm [73]. The grid search is an exhaustive search algorithm that explores a specified space until a pre-defined threshold is achieved. In our work, the genetic algorithm method is used in the optimization process because it has less complexity and is faster than the grid search method. The GA-SVM code is implemented using a MATLAB code [69]. An iterative process is used to find the hyper-parameters where an initial set is generated using random hyper-parameters

values for each member of the set. Then, subsets are created, mutations are applied to the members, and new populations are created. The main idea is that the best fitness function will survive such process, and the algorithm is stopped once the best fitness, defined as four-times better than the original random population, is achieved.

Moreover, the sequential minimal optimization (SMO) algorithm is used to optimize the Lagrange multipliers and find the system's coefficients and its support vectors. This algorithm has been proposed by John Platt for the training phase of the SVM [69, 74, 75]. The SMO is used to solve the quadratic programming optimization problem related to training of the SVM algorithm where it divides the large problem into smaller ones. More specifically, it solves for only 2 support vectors at a time which can be solved analytically and results in less training time, less memory requirements, and less complexity [74].

The outputs of the decision function are:

1. The sign of the output $[+, -]$.

2. The absolute value of the output, which represents the confidence value.

The output is used differently in the classification process depending on the classification type, i.e., binary or multiclass.

3.3 Data pre-processing

The collected data consists of the Raman spectra that contain the vibrational information about the sample under test and can be used for the classification process. The spectrum also contains different sources of variation and noise, as shown in Figure 3.6. The top plot shows the Raman signal plus all noise contributions. The noise can be divided into background noise, cosmic spikes, and white noise [76, 77]. The background noise is due to different contributors, including fluorescence background noise from the experimental setup, stray laser light and reflections that are not totally suppressed by the filters, and detected photons from the sample that are not Raman photons, e.g., Rayleigh scattering.

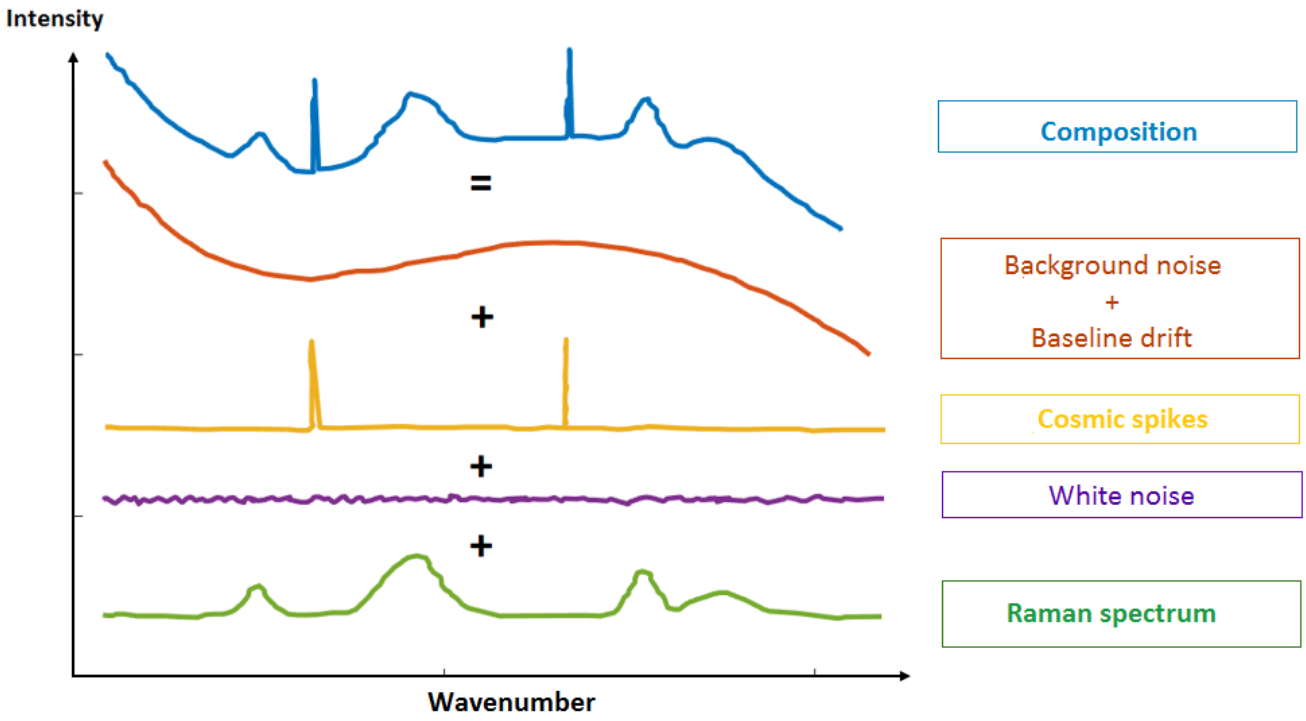


Figure 3.6: Raman signal spectrum composition [78]

The cosmic spikes are caused by cosmic rays when the photons hits the coupled charged detector (CCD) randomly during collecting the spectrum [76,78]. The white noise contains the rest of the noise contributions. As expected, the noise added to the signal, especially the baseline drift, can significantly affect the classification accuracy. Therefore, it is important to reduce the noise associated with the Raman spectrum to improve the quality of the spectrum and to achieve high classification accuracy. This process is called data pre-processing. The data pre-processing can be divided into three main stages, as shown in Figure 3.7.

3.3.1 Normalization

Different captures of the signal spectra have different amplitudes. These differences are due to the different conditions of the measurements and the surrounding conditions. Hence, the

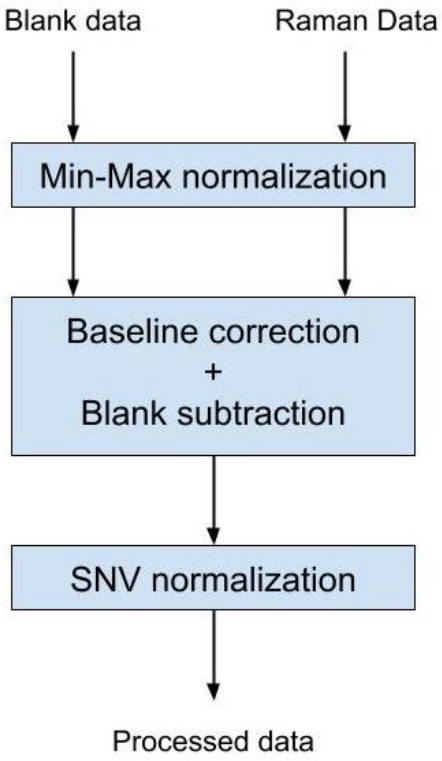


Figure 3.7: Data pre-processing main stages.

first stage of the pre-processing is the signal normalization to ensure the input signals have a uniform scaling. Methods of normalizing the spectrum include min-max normalization, vector normalization, and standard normal variant (SNV) [78–81]. As shown in Figure 3.7, two types of normalization are used in our work: the min-max and the SNV normalization.

Assume a spectrum $x = x_1, x_2, ...x_m$ where m is the number of features. The min-max normalization is applied using the following equation:

$$x_{normalized} = \frac{x - min(x)}{max(x) - min(x)} \tag{3.9}$$

where $min(x)$ is the minimum value of the spectrum x, and $max(x)$ is the maximum value of the spectrum. As a result, the output spectrum will range between 0 and 1.

The SNV normalization removes the mean of the spectrum first. Then, the result-

ing signal is scaled by the standard deviation of the spectrum as shown in the following equation:

$$x_{normalized} = \frac{x - \mu_x}{\sigma_x}, \tag{3.10}$$

where μ_x and σ_x are the mean and the standard deviation of the signal, respectively.

In our classification process, we apply min-max normalization to both the Raman and blank signals. Then, the output signals are passed to the baseline correction block, as explained in the next section. After the baseline correction and blank subtraction steps, we apply a second normalization step before the ML algorithm training.

3.3.2 Baseline correction

The baseline correction task is to fix the baseline drift that occurs in the signal spectrum due to the background fluorescence and external light sources. It should be noted that this process should be carefully applied in order to not alter the main information contained in the spectrum. This correction must be applied before the dimension reduction or ML classification steps. To mitigate the effect of the baseline drift, we perform the following steps:

1. The experiment and data collection occur in a totally dark room to reduce the effects due to the external sources.

2. A baseline correction technique is applied to the signal to correct the residual baseline drift.

3. Blank subtraction: The blank spectrum is collected from the empty spots on the sample under test. Then, this spectrum is subtracted after baseline correction to remove the blank contribution.

Recently, different baseline correction techniques have been proposed and used in the literature, such as simple and modified polynomial fitting [82–85], wavelet transform [86–89],

penalized least square (PLS) and weighted penalized least square [90–92], vector transformation penalized spline (VTPspline) [93], cubic spline algorithm [94, 95], and asymmetric penalized least squares [96, 97].

In our work, we consider using the airPLS technique [98–101]. An open source MATLAB code is used for the implementation of the airPLS [102]. This technique aims to calculate a LS fit curve for the input signal spectrum. Different parameters can control the fit as explained next. After calculating the fitted curve, we subtract the fitted curve from the original signal spectrum to calculate the baseline corrected signal.

The main idea of airPLS is based on the PLS that creates a fitted curve z based on the sum square error (SSE) between the original spectrum x and z. This relation is called the fidelity F to the original spectrum x and can be expressed by the following equation [98]:

$$F = \sum_{i=1}^{m} (x_i - z_i)^2 \tag{3.11}$$

where m is the total number of features. Another important term called the roughness R, which expresses the roughness of the fitted curve as shown in the following equation [98] :

$$R = \sum_{i=2}^{m} (z_i - z_{i-1})^2 = \sum_{i=2}^{m-1} (\Delta z_i)^2 \tag{3.12}$$

The balance between the roughness and the fidelity is expressed in the following equation [98]:

$$Q = F + \lambda R = ||x - z||^2 + \lambda ||Dz||^2, \tag{3.13}$$

where D is the derivative of the matrix z.

The roughness contribution is controlled by a penalty parameter λ. The λ parameter controls the smoothness of the curve; it is typically in the range from 1 to 10^6. Figure 3.8 shows an example of the raw spectrum for E. coli and the effect of different values of λ on the smoothness of the fitted curve. The figure shows the signal intensity versus the

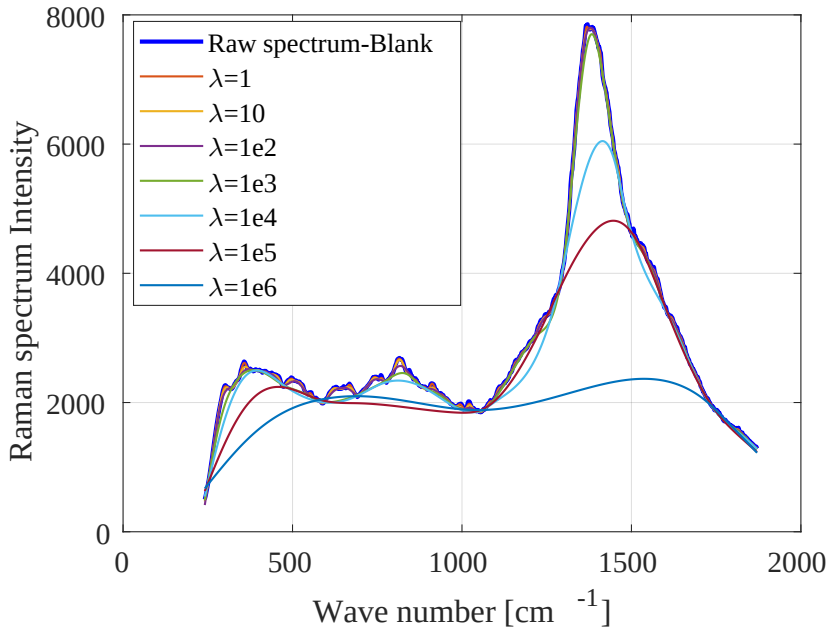


Figure 3.8: Raw spectrum of E. coli bacteria and the fitted curve at different values of λ.

wavenumber of the signal. It can be observed that as λ increases, the fitted curve will be smoother, e.g., $\lambda = 10^5$. However, at larger increases, the signal spectrum can be under fit. Smaller values will result in a curve that closely tracks the spectrum and that may cause removing important information from the spectrum e.g., $\lambda = 10^1$. Hence, an optimum value for λ exists and needs to be found empirically by optimizing the classification accuracy. More details on the results are presented in the next chapter.

Finally, the fitted curve z is calculated by minimizing Eq. (3.13). This method has a main disadvantage for Raman spectrum in that it assumes the background noise is constant along the spectrum [90,91]. However, the Raman spectrum contains peaks that are used to differentiate between bacteria types. As a result, a modified PLS called weighted PLS has been proposed in [87,103]. In the modified PLS, a weight parameter w is used to penalize the fidelity F. So, the fidelity is equal to zero in the peak's position of the signal x, and

the baseline can be tracked. The modified fidelity can be expressed as [98]:

$$F = \sum_{i=1}^{m} w_i (x_i - z_i)^2 = (x - z)' W (x - z) \tag{3.14}$$

where w_i is the diagonal of the diagonal matrix W.

It should be noted that the weighted PLS technique suffers from a major disadvantage: prior to the baseline correction, the user must have previous knowledge about the position of the peaks in the Raman spectrum to be able to set the weight vector to zeros at the correct position. Therefore, the airPLS algorithm has been proposed to overcome such disadvantage [98].

This technique uses an iterative process to generate an adaptive weight vector w that assumes an initial value of $w^0 = 1$. Then, a certain weight value w^t is calculated based on the value of the signal x^t and the value of fitted curve from the previous iteration z^{t-1} for each iteration t. The balance equation used to get the value of the fitted curve point z^t is given by [98]:

$$Q^t = \sum_{i=1}^{m} w_i^t |x_i - z_i^t|^2 + \lambda \sum_{j=2}^{m} |z_j - z_{j-1}^t|^2 \tag{3.15}$$

The update equation for the weight is given by [98]:

$$w_i^t = \begin{cases} 0 & x_i \geq z_i^{t-1} \\ e^{\frac{t(x_i - z_i^{t-1})}{|\mathbf{d}^t|}} & x_i < z_i^{t-1} \end{cases} \tag{3.16}$$

where t is the current iteration, $t - 1$ is the previous iteration, and z_i^{t-1} is a baseline candidate. The d^t vector containing the negative values resulting from calculating the difference between x and z^{t-1} at iteration t. This iterative process overcomes the peak detection problem found in the weighted PLS method, where at value i of the signal x and at iteration t, the weight is set to zero when the value of the x_i is larger than the fitted value at the previous iteration z^{t-1} since it is within the peak. So, this value will be

ignored in the next iteration. Hence, the airPLS algorithm will automatically detect and not eliminate the peak points in the Raman spectrum [98]. It should be noted that this behaviour doesn't occur at non-optimum λ values, where much smaller λ values will cause the airPLS to detect the peaks as part of the signals and remove them as shown in Figure 3.8. Two stopping criteria for this iteration process are considered:

1. Reaching the fixed maximum number of iterations, which is set to 20 in our results.
2. Reaching the stopping condition defined as $|d^t| < l \times |x|$, where l represents the tolerance and is set to 0.001 [98].

After correcting the baseline, the blank spectrum is subtracted from the signal spectrum. Then, the last step of the data pre-processing is the normalization, where the resulting spectrum is scaled using SNV normalization, so all the spectrum captures have zero mean since it will cause the ML algorithm be trained on the mean of different captures [104]. Finally, dimension reduction is performed, as explained in more detail in the following sections.

3.4 Classification types

This section explains the different classification types that can be applied to different data sets.

3.4.1 Binary classification

This type of classification is used to distinguish between two different classes, e.g., two different bacteria types, as shown in Figure 3.3. The binary classification process follows the following steps:

1. The data set is formed from samples or observations from both classes, where each class is assigned a different label or output either 1 or -1.

2. Then, the data set is divided into a training set with 60% of the original data set and a testing set with 40% of the original data set [64].

3. The training data is used to train the ML algorithm, finding the optimal values of the kernel hyper-parameters, and to find the decision function that separates the two classes.

4. Finally, the testing set is passed with the decision function to calculate the classification accuracy of the input data set.

3.4.2 Multiclass classification

The SVM algorithm has usually been used for binary classification problems that can be extended to multiclass classification. Multiclass classification is a generalized case of binary classification where multiple classes are compared, e.g., three different bacteria types. Generally, multiclass classification is divided into several binary models. There are two approaches for the multiclass classification type: one-vs-one (OVO) and one-vs-all (OVA) [59, 105, 106].

One-vs-one (OVO)

The OVO algorithm is one of the methods used to perform the multiclass classification [59], and can be explained in the following steps:

- In the case of having C classes, the OVO generates M binary models defined by [59]:

$$M = \frac{C \times (C-1)}{2} \tag{3.17}$$

- For every class i, a binary model is built with each of the other classes with two different labels [+1,−1]. An example of building three binary models corresponding to three classes is shown in Figure 3.9.

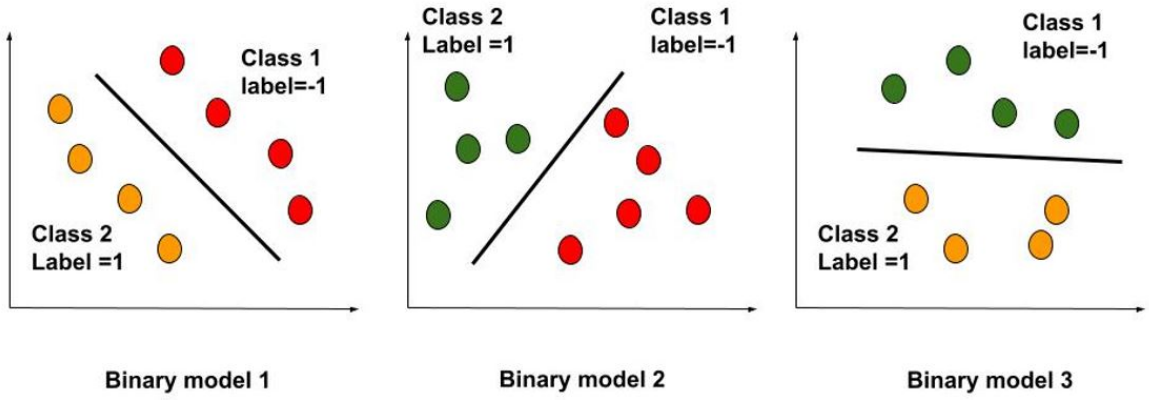


Figure 3.9: Example of three OVO binary models for three classes.

- Then, the data set is divided into a training set with 60% of the original data set size and a testing set with 40%.

- Then, the SVM binary classifier is trained with the training set for each binary model i and all the SVM coefficients are saved to be used in the testing/prediction stage.

- In the testing stage, the saved coefficients are used in Eq. (3.8) to predict the class label y_p. The prediction is performed in three steps. First, the maximum confidence value determines the model where the sample belongs. Then, the sign of the confidence value determines the class within the model. Finally, the predicted class for each file, which can contain multiple capture iterations, is determined based on the majority of the predicted class within each file. This process is summarized in Figure 3.10 for the three different classes.

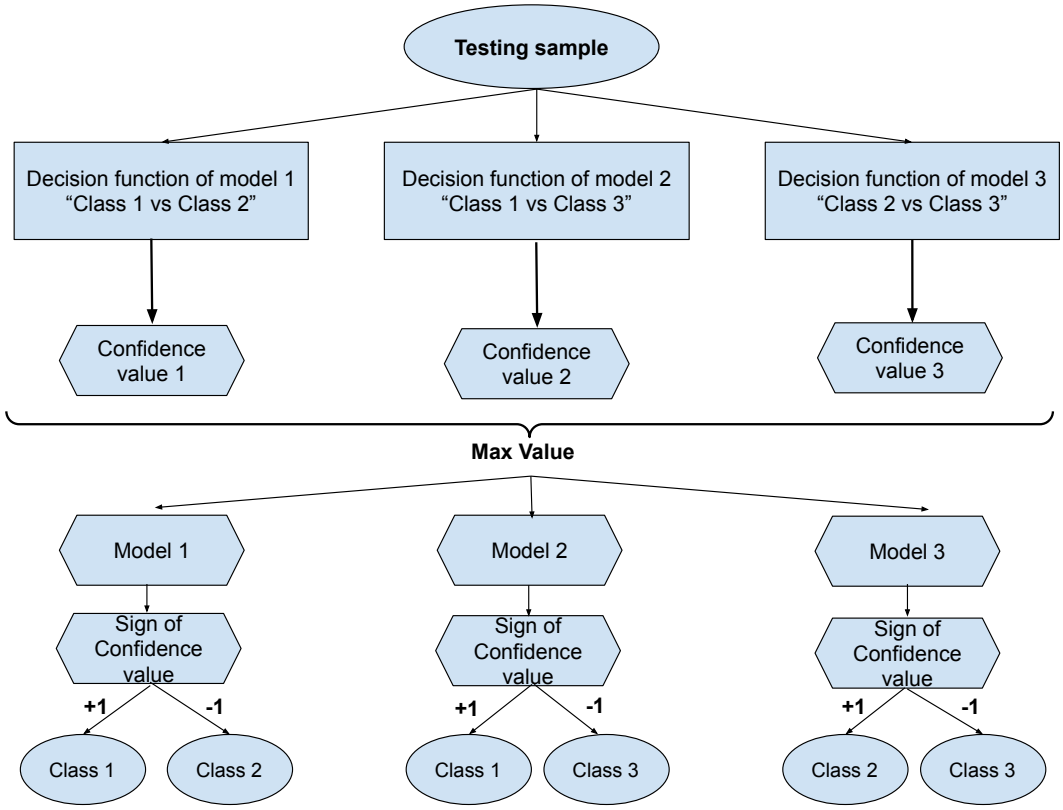


Figure 3.10: OVO classification process

3.4.3 One-vs-all (OVA)

OVA is another popular technique used to perform multiclass classification [59, 105, 107]. OVA is considered less complex than the OVO technique. OVA generates M binary models equal to the number of classes C. Hence, there are less comparison cases than with OVO when number of classes exceeds three classes. For example, if we consider a four-class scenario, we need six cases for OVO and four cases for OVA.

Figure 3.11 shows an example for building three OVA binary models for three classes. The OVA algorithm can be explained in the following steps:

- In case of having C classes, the OVA generates M=C binary models corresponding to the number of classes.

- For each class i, a binary model is built with two labels [+1,−1] where class i will have the positive label and all the other classes will have the negative label, as shown in Fig 3.11.

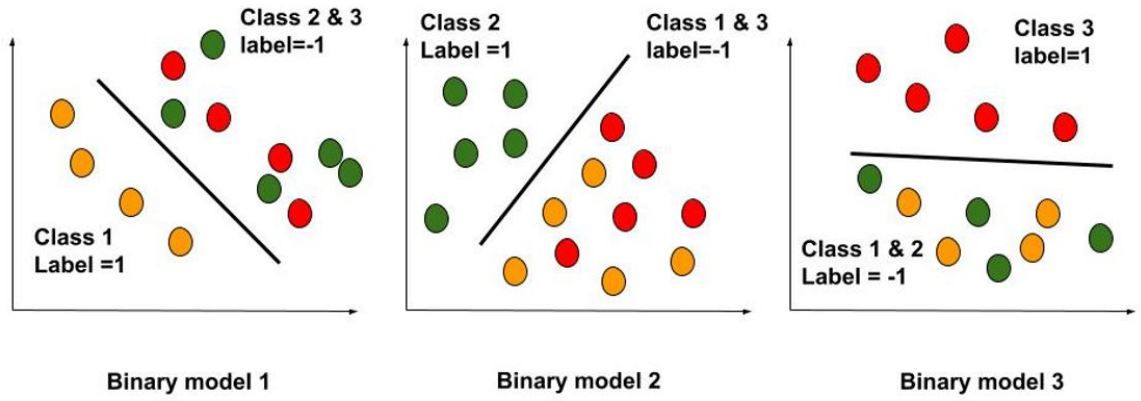


Figure 3.11: Example of three OVA binary models for three classes

- Then, the data set is divided into a training set with 60% of the input data set size and a testing set with and 40%.

- Then, the SVM binary classifier is trained with the training set for each binary model i and all the SVM coefficients are saved to be used in the testing/prediction stage.

- The testing stage is similar to the OVO steps explained previously. The saved coefficients are used in Eq. (3.8) to predict the class label y_p. This happens in a three-stage process. First, the maximum confidence value determines the model where the sample belongs. Second, its sign determines the class within the model. Finally, the predicted class for each file, which can contain multiple capture iterations, is determined based on the most predicted class within each file. The only difference for OVA compared with the OVO testing stage is that we focus only on the positive sign in each model that belongs to the individual class being compared to the other classes. Figure 3.12 illustrates the main steps to perform the classification process for each sample for the three different classes.

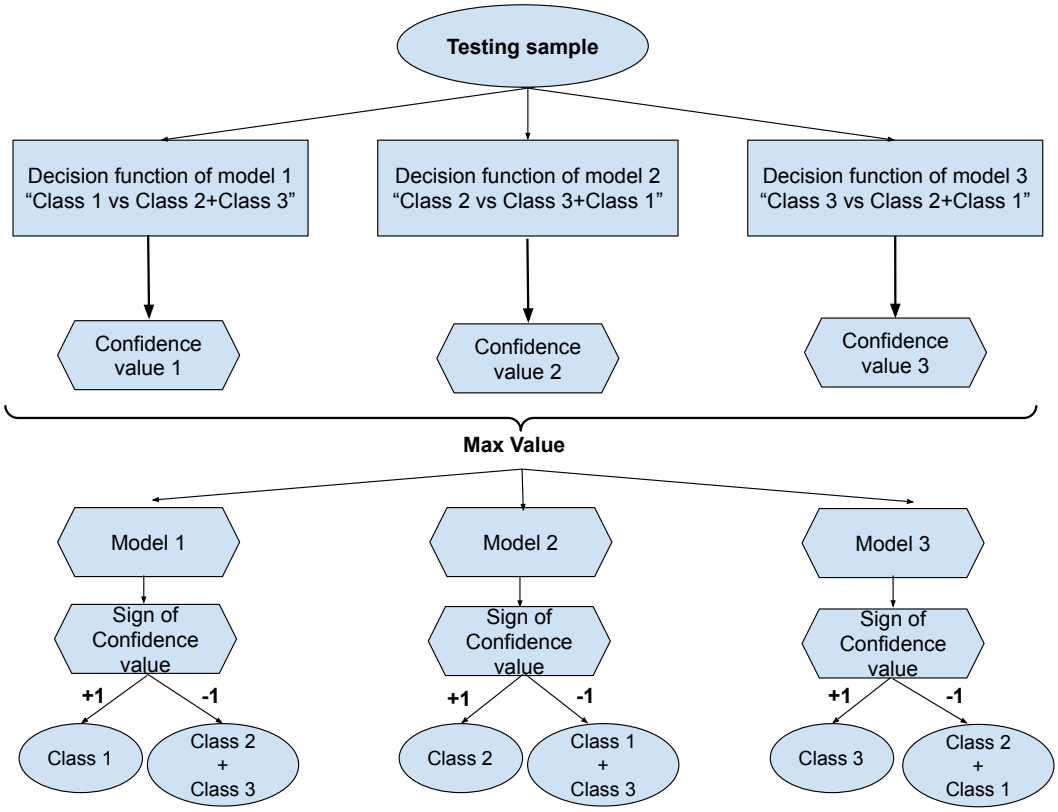


Figure 3.12: Classification process of three models using the OVA technique

3.5 Dimension reduction techniques

Normally, the data set results of Raman spectrum consist of a large number of samples and features. The dimension reduction process has a significant importance in processing the data used with ML algorithms. First, it can significantly reduce the system's complexity, the training time, and the training memory of the ML algorithm [108]. Also, it can improve the classification accuracy if the sample or feature is causing a degradation to the classification accuracy and is removed.

In our work, we consider performing the reduction in both dimensions, i.e., the reduction of the samples and features, as explained in the following subsections in more detail [109–112].

3.5.1 Sample reduction

In focusing on the quantitative analysis of the data set, we aim to reduce its size by removing the samples that are not significant or can be degrading the classification accuracy. We consider the sample reduction based on the sample signal-to-noise ratio (SNR), where the SNR of the collected spectrum is used to exclude samples with a low SNR that can affect the classification accuracy [113].

The SNR for each feature i of the spectrum is calculated using the following equation [113]:

$$SNR_i = \sqrt{N * T * P} * \frac{R_i}{\sqrt{R_i + noise}}, \tag{3.18}$$

Where R_i is the Raman contribution, i is the feature number, N is the number of captures, T is the acquisition time, and P is the laser power.

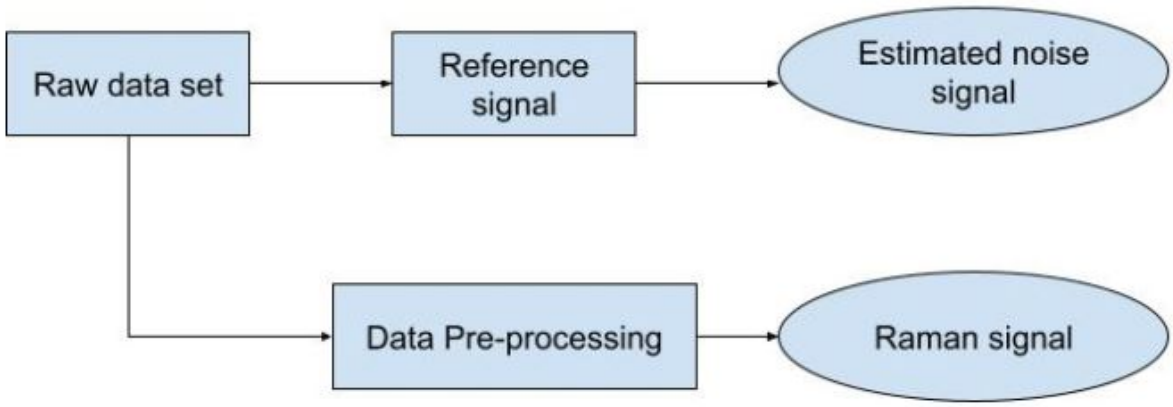


Figure 3.13: Main steps of the Raman and noise signal calculation.

First, we need to calculate both the signal power and the noise power to compute the SNR. Figure 3.13 shows the main steps to calculate the SNR. The signal power is calculated based on the signal after passing through the pre-processing stages explained in Section 3.3. Then, the reference signal is generated by filtering the raw spectrum using a Savitzky–Golay (SG) smoothing filter of order 3 and a window size of 9, which is implemented using a built-

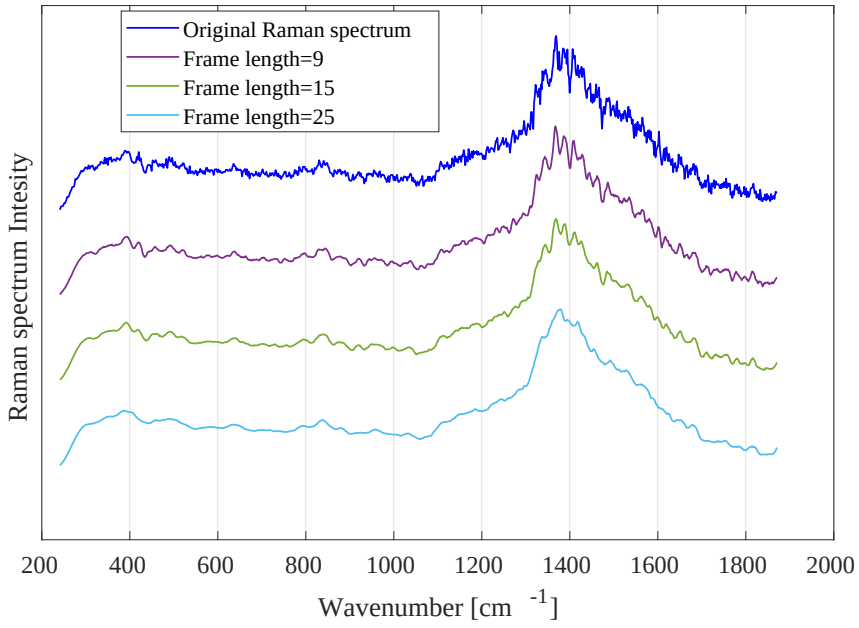


Figure 3.14: The effect of the SG window size on the Raman bands.

in MATLAB function. To avoid affecting the bands of the Raman spectrum, it is important to choose the SG filter carefully, generally by choosing a small window size [114]. Figure 3.14 shows how the Raman bands change as the window size increases. Then, the estimated noise signal is calculated by subtracting the raw signal and the reference signal [115]. An example of the raw, reference, and noise signals for the CA bacteria is shown in Figure 3.15.

Finally, we calculate the quantitative quality factor (QF) metric, which is defined as:

$$Q_F = \sum_i SNR_i, \tag{3.19}$$

which is the sum of the SNR calculated for each feature i. Then, this metric is used to select or neglect the samples from the captures based on a chosen threshold where the samples with the lowest QF are first removed. The optimum threshold is found empirically to optimize the classification accuracy and complexity, as presented with the results in the next chapter.

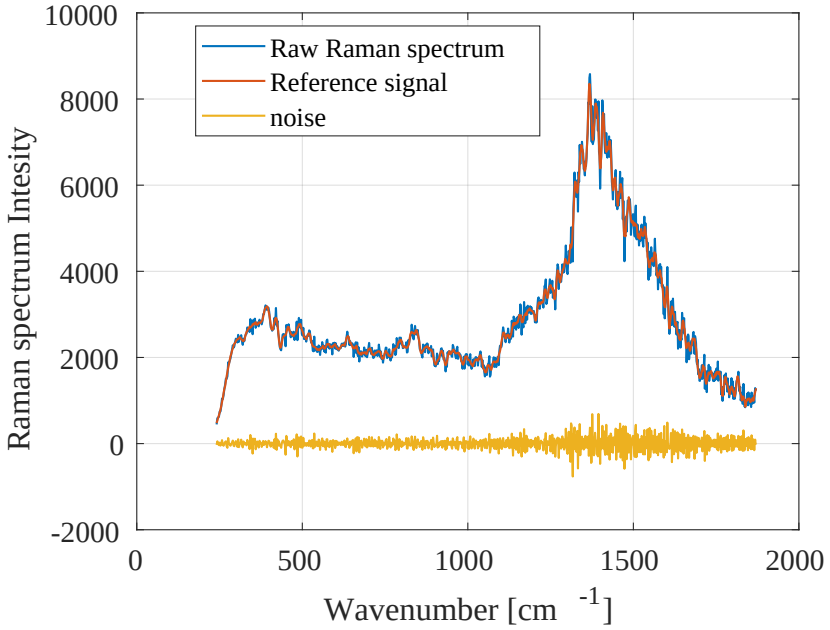


Figure 3.15: Raw, reference, and noise signals for PVP-EC.

3.5.2 Feature reduction

The captured Raman spectra usually consists of many features, e.g., 885 features in our results, which are not all needed for the classification and can even degrade the performance. Hence, feature reduction is important for reducing the complexity and/or improving the classification accuracy.

The multiple reduction techniques proposed in the literature can be categorized into two groups: feature extraction (FE) techniques and feature selection (FS) techniques. FE techniques transform the original high-dimensional features into a new less-dimensional feature set through matrix transformation such as principal component analysis (PCA) and independent component analysis [70, 116–119]. Although, the FE techniques project the data into a new space with lower dimension size, every feature doesn't necessarily represent meaningful information on its own. As a result, the mapped set must all be used within the MLA model.

FS techniques select a new feature set from the original set based on its importance and

its effect on the classification accuracy without any change in the features' nature or the amount of information they contain unlike the FE techniques. As a result, more dimension reduction can be achieved.The FS can be further divided into three techniques: filter, wrapper, and embedded [108,109,120]. For the filter technique, the feature selection process is dependent on the data set only and independent of the classifier ML algorithm used. Also, it is mainly used with linear data. For both the embedded and wrapper techniques, the feature selection process is dependant on the ML algorithm, where the parameters used to measure the feature's importance are calculated through the training process of the classifier [108, 109, 120]. Table 3.1 shows the main advantages and disadvantages and gives few examples of each technique.

Table 3.1: Main advantages and disadvantages of the feature selection techniques [108]

Method	Advantages	Disadvantages	Examples
Filter	Independent of the classifier Lower computational cost than wrapper Fast Good generalization ability	No interaction with the classifier	Chi-Squared, Information Gain, and Relief
Embedded	Interaction with the classifier Lower computational cost than wrapper Captures feature's dependencies	Classifier-dependent selection	SVM-RFE, Feature Selection-Perceptron
Wrapper	Interaction with the classifier Captures features dependencies	Computationally expensive Risk of overfitting Classifier-dependent selection	Heuristic Selection Algorithms, and Meta-Heuristic Search Algorithms

The embedded FS technique combines the advantages of both the filter and the wrapper and avoid their disadvantages. As a result, we use the SVM-RFE algorithm as a FS technique in our work. SVM-RFE is a widely used technique for the feature selection that is used with both linear and non-linear data based on the training model and the kernel function used [70, 109, 120–122]. Also, the RFE is an example of backward selection, where the process starts with the complete feature set, and then a feature is removed per iteration based on its significance. In addition, it has several advantages such as its robustness, having lower risk of overfitting, lower complexity, and better generalization performance to the new data. Originally, the RFE technique was used in the cancer classification process, where the number of training samples is less than 100 and the number of features is several

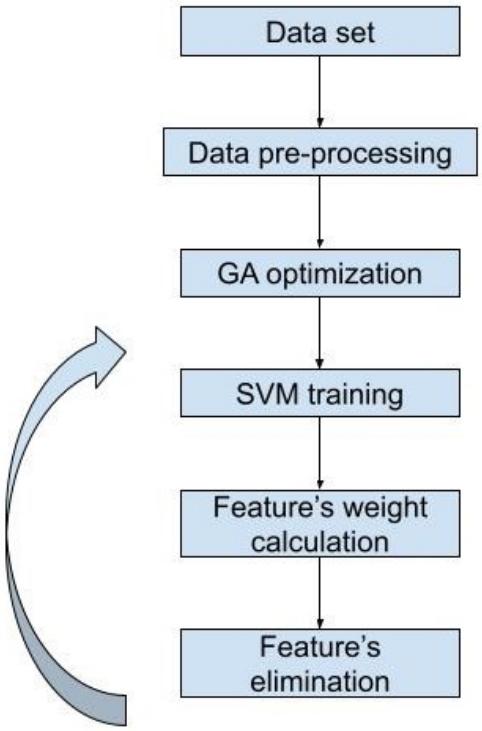


Figure 3.16: Main steps of the SVM-RFE feature selection technique.

thousands [122].

In our work, SVM-RFE is used as a non-linear data classifier. It reduces the number of features by selecting the most important features that affect the classification accuracy based on their calculated weight or feature score using the SVM training model [70]. Furthermore, the SVM-RFE is an iterative process, where in each iteration the least important feature that has the minimum calculated feature score is removed. We developed the MATLAB code that is used to implement this algorithm, which is based on the algorithm proposed in [70]. The main steps of the process are summarized in the flow-chart shown in Figure 3.16. and explained as follows:

1. Process the data through the pre-processing stage.

2. Calculate the optimal value for the hyper-parameters of the kernel function σ, β, and

l used in the training model through the genetic algorithm (GA) stage.

3. Train the ML algorithm (SVM), calculate the training coefficients α, and calculate the RQK function $k(x_i, x_j)$ using Eq. (3.7). These parameters are needed to calculate the feature score or weight.

4. Calculate the feature score (c_p) for each feature through an iterative process using the following equations [70]:

$$c_p = \frac{1}{2}|\alpha^T \mathbf{H}\alpha - \alpha^T \mathbf{H}^{(-p)}\alpha|, p = 1, 2, ..m \tag{3.20}$$

$$H_{ij} = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, 2, ...N \tag{3.21}$$

$$H_{ij}^{(-p)} = y_i y_j k(\mathbf{x}_i^{(-p)}, \mathbf{x}_j^{(-p)}) \tag{3.22}$$

where m is the total number of features, N is the total number of samples, $(-p)$ indicates the feature being removed, α are the coefficients calculated in step 3, and y_i and y_j are the corresponding labels for samples x_i and x_j respectively .

5. At the end of the iterations, the feature with the minimum weight is removed.

6. Repeat steps 3-5 until all the features are removed.

7. Finally, the outputs of this process are presented in two vectors: the weight vector of the features and the position of the removed feature in each iteration.

The above algorithm can be used with both binary classification and multiclass classification. First, in case of binary SVM-RFE, there is only one binary model with two classes of labels $y = [+1, -1]$ [123]. Second, in case of multiclass SVM-RFE, there are multiple techniques to perform the SVM-RFE feature selection in the literature, such as MSVM-RFE-OVA which is based on the OVA technique [124, 125], MSVM-RFE-WW method by Weston and Watkins, MSVM-RFE method by Crammer and Singer, and MSVM-RFE method by Lee, Lin and Wahba [123, 125].

In our work, we use the MSVM-OVA-RFE technique, as explained in the following steps and summarized in Figure 3.17:

1. As there are three classes, three binary models are built equal to the number of classes. In each model, there is one class with a label of $+1$, which is called the main class, while the other two are combined and have the -1 label.

2. The SVM-RFE is applied on each binary model. In each binary model, the GA and SVM training are applied. Then, the features of the main class are ranked based on their score/weight through the RFE.

3. Finally, the feature weight vectors are used to calculate the reduced feature set based on a sweep of the number of removed features and finding the corresponding classification accuracy. Then, the reduced feature set that gives an acceptable accuracy with a reduced complexity is selected. The classification results are discussed in the next chapter.

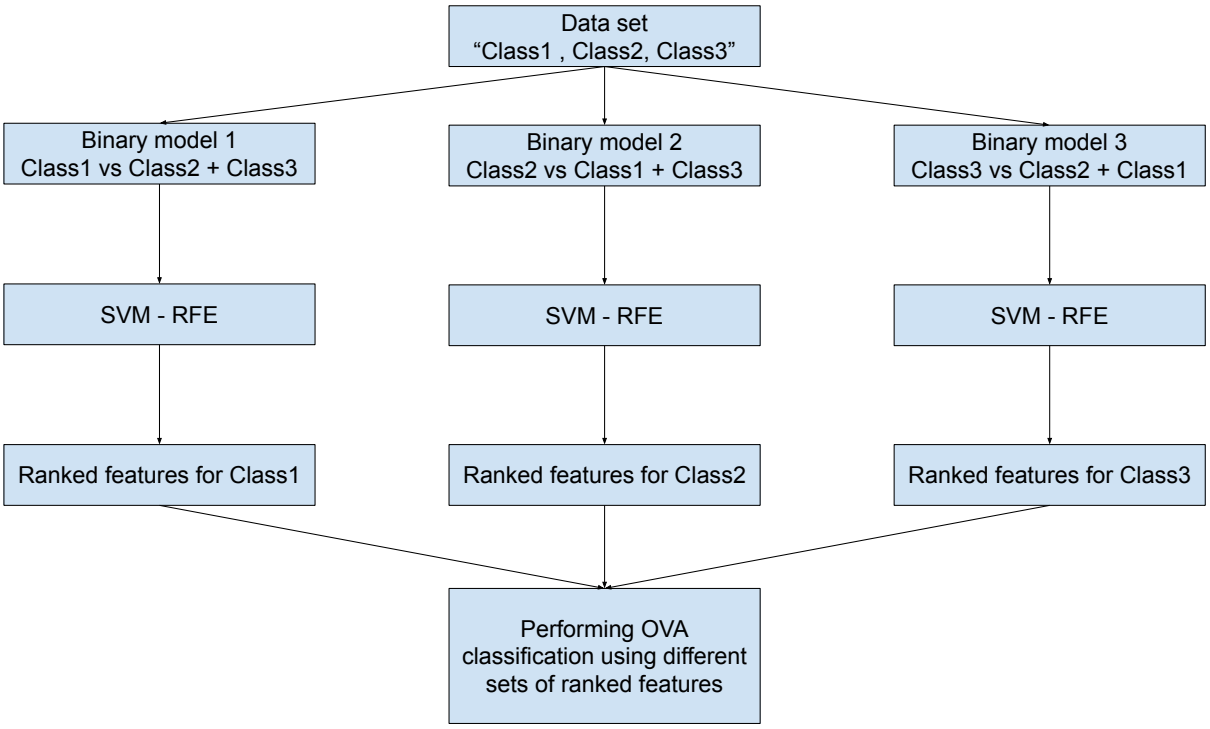


Figure 3.17: Main steps of the mutliclass SVM-RFE feature selection technique.

3.6 Conclusion

This chapter presented the steps and details of the classification process, including data preprocessing, dimension reduction, the SVM ML algorithm, dimension reduction techniques and classification types. Chapter 4 presents the bacteria acquisition, sample preparation, and spectrum capturing steps, followed by the classification accuracy based on different classification types for three bacteria types.

Chapter 4

Classification results

4.1 Introduction

The previous chapters described Raman spectroscopy for bacteria identification and the different algorithms used for classification and complexity reduction. This chapter presents the performance of different algorithms and classification classes based on Raman spectra captured experimentally for different bacteria types. First, the three steps followed to capture the data set are explained: bacteria acquisition, sample preparation, and Raman spectra capturing using the experimental setup. These steps were performed by other members of my research team.

The rest of this chapter is organized as follows. Section 4.2 explains the process of bacteria acquisition. The sample preparation is discussed in Section 4.3. The experimental setup and data collection are explained in Section 4.4. The classification results versus different parameters and different reduction techniques are presented in Section 4.5. Finally, Section 4.6 sums up the classification results.

4.2 Bacteria acquisition and sample preparation

In our classification setup, we use four bacteria types: *Escherichia coli* (EC), *Cutibacterium acnes* (CA), *methicillin-resistant Staphylococcus aureus* (MRSA), and *methicillin-sensitive Staphylococcus aureus* (MSSA). Because the classification process is based on the species and not on the strain, MSSA and MRSA are joined as S. aureus bacteria and called MS in the classification.

In addition, all the bacteria cells used have gold nanoparticles attached to them to enhance the Raman signal through SERS [126–128]. Also, a polyvinylpyrrolidone (PVP) capping agent is used to stabilize the nanoparticles, and control the interaction between them and the bacteria cells [129]. All the materials used in this experiment were provided by the Ottawa Hospital, and ethical approval was obtained from the Ottawa Health Science Network Research Ethics Board to use the CA bacterial strains collected from peri-prosthetic joint infection patients for this study.

4.3 Sample preparation

After the bacteria acquisition step, the sample is prepared. This step is divided into two phases: creating the bacterial culture and preparing the slides.

4.3.1 Creating bacterial culture

The process of creating a bacterial culture can be summarized in the following steps [128]:

1. A brain heart infusion (BHI) agar plate is streaked with the cells of EC, and the plates are incubated overnight aerobically.

2. A blood agar (BA) plate is streaked with the cells of CA, and the plates are left for 72 hours under anaerobic conditions.

3. A fresh tryptic soy agar (TSA) plate is streaked with the cells of MS, and the plates are incubated overnight aerobically.

4. Then, single colonies are suspended in 5 mL of BHI for EC and in 5 mL of TSA for MS. Then, they are incubated overnight with agitation at 200 rpm. CA suspensions are prepared in BHI and incubated under anaerobic and static conditions for 72 hours.

5. Bacterial concentrations are verified by plating and counting each bacterium. Finally, all bacterial cultures are incubated at 37°C.

4.3.2 Slides preparation

The nanoparticle solution is mixed with each of the bacteria samples in a microfuge tube in the ratio of 1:9 and shacked slightly. Then, some drops of the mixture are spread onto a glass slide and then allowed to settle. After 1 hour, the bacteria are heat fixed to the slide and the liquid is aspirated. Then, the slide is dried under a stream of clean filtered air. Finally, the slide is rinsed in a gentle stream of deionized water to remove un-fixed cells and loosely bound debris, then dry again with air. To ensure enough bacteria adhere to the slide for measurement, we repeat this process 3 times.

4.4 Experimental setup and data collection

After the samples are prepared, the next step is to collect the Raman spectrum from the cells under the test. The data collection is based on a dark-field microscope system. Figure 4.1 shows the experimental setup used to capture the Raman spectrum of a bacteria sample [128]. In this setup, a 10 mw laser beam is generated from a continuous wave (CW) distributed Bragg reflector (DBR) laser with a wavelength of 785 nm, then it is collimated and reflected through a notch filter. Then, the light is reflected by the dichroic mirror towards the dark-field setup, where there is an objective lens that focuses the light on the sample. The sample is placed on a glass slide and mounted on a Thorlabs xy-direction

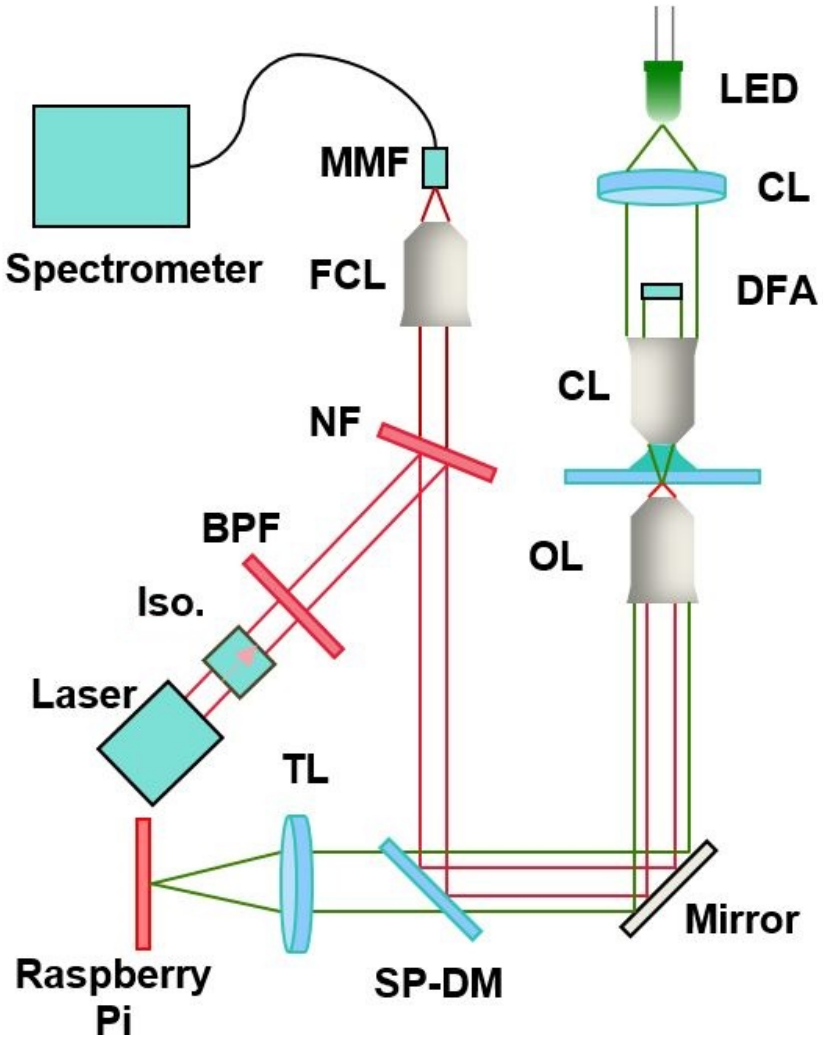


Figure 4.1: Experimental setup used to capture the Raman spectrum of the bacteria sample. Iso.: isolator, BPF: bandpass filter, NF: notch filter, FCL: fiber coupling lens, MMF: multi-mode fiber, SP-DM: short pass dichroic mirror, TL: tube lens, R_{Pi}: Raspberry Pi, MM: microscope mirror, OL: Objective lens, CL: condenser lens, DFA: dark-field aperture, CL: collector lens

movable stage. A green LED is used as a source of light from the top of the sample to create the dark field. The scattered signal from the sample travels back through the objective lens towards the dichroic mirror. Then, the images of the bacteria cells with either nanoparticles or their clusters on a black background are captured. These images are produced through the dark-field setup and viewed on the Raspberry Pi through a complementary metal oxide semiconductor (CMOS) camera. Also, the reflected signal from the dichroic mirror passes

through the notch filter, where it allows the signal to pass through to a long pass filter and block the laser wavelength. The long pass filter further blocks the Rayleigh scattered signal with wavelengths near the infrared range. Finally, scattered Raman spectrum is captured with the Kaiser f/18i apectrograph with a TE-cooled Andor CCD camera which comes with Andor SOLIS software in the spectrometer. The CCD consists of a large number of pixels, where it converts the scattered photons received by each pixel into electrons and saves them in a sequence of pixels, which is then related to the Raman wavenumbers or features. It should be noted that the number of saved electrons is proportional to the number of photons received or the light intensity by each pixel [115]. Before running any samples, a test sample of dried rhodamine 6G (R6G) samples is used to calibrate and align all the components of the system. Most of the recorded spectrum of the samples contain 10 accumulations or iterations where they are considered as a single sample file. In the testing stage, the file sample is considered to be classified correctly if 5 or more samples in the same file are true and vice versa. The number of sample files and the total number of samples are shown in the following table.

	CA	EC	MS
Files	39	30	82
Total samples	390	300	825

Table 4.1: Data sample size used in the classification results.

4.5 Classification results

This section presents the classification results for the captured bacteria samples using both the binary and the multiclass classification processes. In order to provide more accurate results, the accuracy is averaged over 100 iterations, where the data is randomly divided between the training set and the testing set in each iteration. In this random division, we ensure that the samples corresponding to the same file can only contribute to either

the training set or the testing set to avoid training and testing the SVM algorithm with correlated samples.

The basic data set was previously used to perform the classification process using the GA-SVM, but in a different way than the suggested algorithm in this thesis, where the OVO multiclass classification technique was used instead of the OVA technique used for this thesis [128]. Using the OVO technique, the overall classification accuracy was approximately 77%, lower than the classification accuracy achieved in this thesis, as shown in Section 4.5.2. The higher classification accuracy achieved in the work for this thesis can be attributed to using different pre-processing steps and parameters on the data set, using different multiclass classification, and reducing the data set. The significant increase achieved in our work in addition to the reduction in the complexity will pave the way for a more practical biosensor for rapid and high accuracy bacteria identification.

4.5.1 Binary classification

This section presents the classification results using binary classification. First, we optimize the value of λ which controls the smoothness of the fitted curve in the airPLS as explained in Section 3.3.2. We sweep the value of λ in the range of $\lambda = [10^3 - 10^6]$, and calculate the classification accuracy of the three bacteria models, i.e., EC vs CA, EC vs MS, and MS vs CA.

Figure 4.2 shows the classification accuracy versus different λ values. It can be observed that the optimal value of λ is determined to be 10^5. Also, the classification accuracy is above 90% for all binary classes due to the large data set, quality of captures, and training of the SVM classifier.

After choosing the optimum value of $\lambda = 10^5$, we focus on the quantitative analysis step. As discussed in the previous chapter, this step focuses on choosing the reduced samples set that can achieve high classification accuracy and low complexity compared with the complete data set.

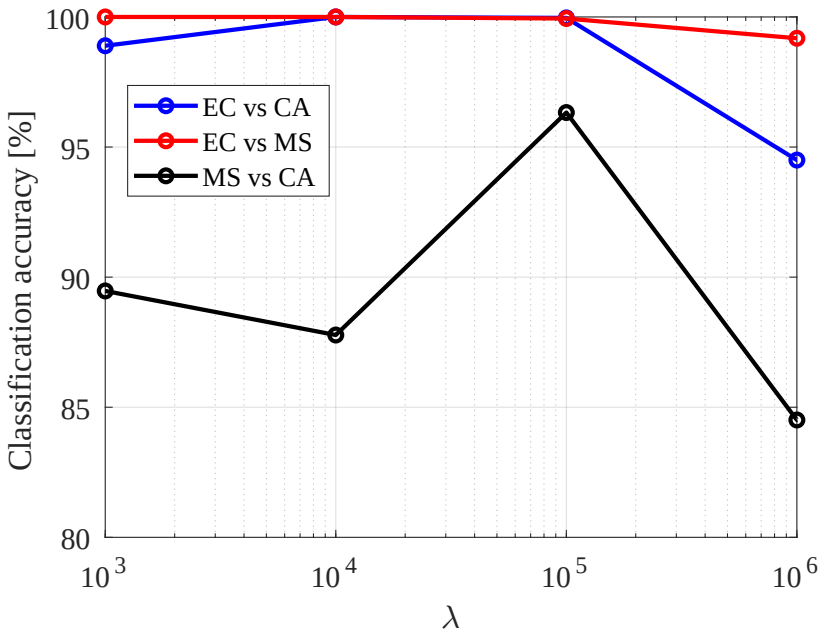


Figure 4.2: Classification accuracy versus different λ values within the range 10^3 to 10^6.

First, the quality factors (QFs) are calculated for each sample after calculating the SNR of each sample per feature as per Eq. (3.19). The distribution of the calculated QFs for the three bacteria types is shown in Figure 4.3. The figure shows that few samples have a significantly high QF compared with the rest of the samples for the EC and MS types. On the other hand, the CA has a more spread out or uniform distribution of the QF across the samples. As a result, we expect a significant reduction in complexity can be achieved by removing low QF samples while maintaining a relatively high classification accuracy, especially for the CA case.

After calculating the QF vector per bacteria type, they are sorted in an ascending order. Then, we sweep the QF threshold by sweeping the percentage of the removed samples from 0 to 70%. At each threshold value, the samples achieving a higher QF compared with the threshold are passed to the classifier, and the lower QF samples are discarded. For example, at 20% sample reduction, we remove 20% of the original data set samples with the lowest QF. Then, the remaining samples are divided according to 60% and 40% of their size for

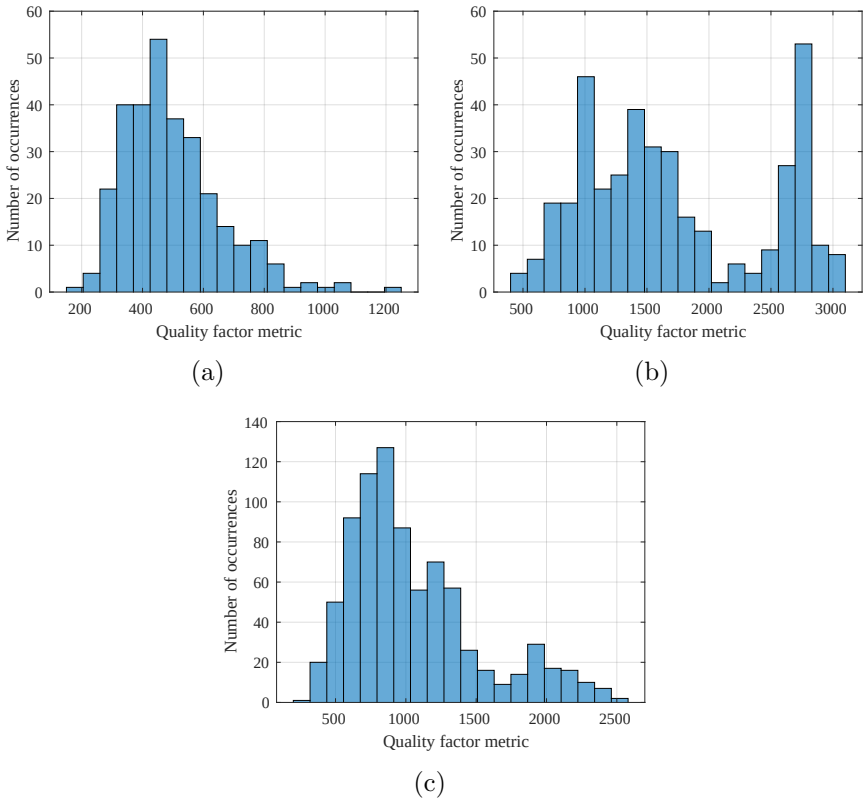


Figure 4.3: Histogram for the quality factor metric for the three bacteria types (a) EC, (b) CA, and (c) MS.

training the SVM algorithm and testing, respectively. Finally, the classification accuracy is calculated over 100 iterations.

The classification accuracy versus the percentage of samples removed is shown in Figure 4.4. Due to the instability of the results when significant reduction is applied and the sample size used is relatively small, we limit the maximum reduction to 60%, 60%, and 70% for the the EC vs CA, EC vs MS, and MS vs CA, respectively. Without any sample reduction, the classification accuracy is higher than 95%, which can be attributed to high SNR samples and the SVM algorithm being well trained. By removing samples, we observe a gradual decrease in the classification accuracy as expected. There is clearly a trade-off between the accuracy and the number of samples removed. So, we choose to target a classification

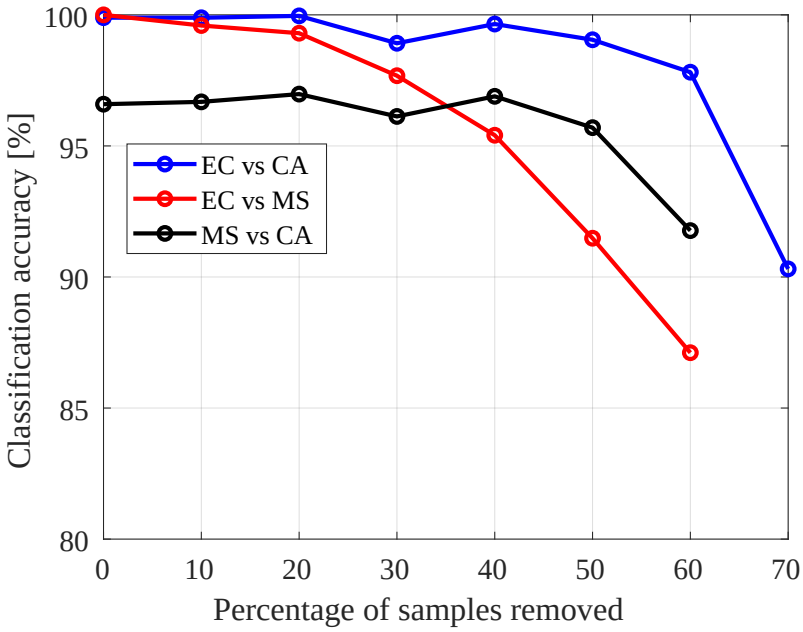


Figure 4.4: Classification accuracy versus percentage of samples removed based on the sample SNR.

accuracy value of $> 90\%$, which is relatively high, and a smaller number of samples is used which will reflect on the speed and the complexity of the ML algorithm. Hence, we select the reduced sample set size achieved by removing 70%, 50%, and 60% of the samples for the EC vs CA, EC vs MS, and MS vs CA, respectively.

Then, we focus on the reduction in the feature dimension. First, both the position and the weight of each feature are calculated using the SVM-RFE algorithm as explained in Section 3.5.2. Figure 4.5 shows the feature weights or scores versus the wavenumber for each of the binary models. It can be noted that not all features have equal weight. Also, fewer than 15 features from the entire 885 features have high a feature score. After calculating the feature scores, we sort the feature vector in an ascending order with respect to the feature score. Similar to the SNR reduction, we sweep the feature reduction percentage and calculate the classification accuracy of the three binary models.

Figure 4.6 presents the classification accuracy versus the percentage of the removed features. It should be noted that the features are sorted based on their score. In other

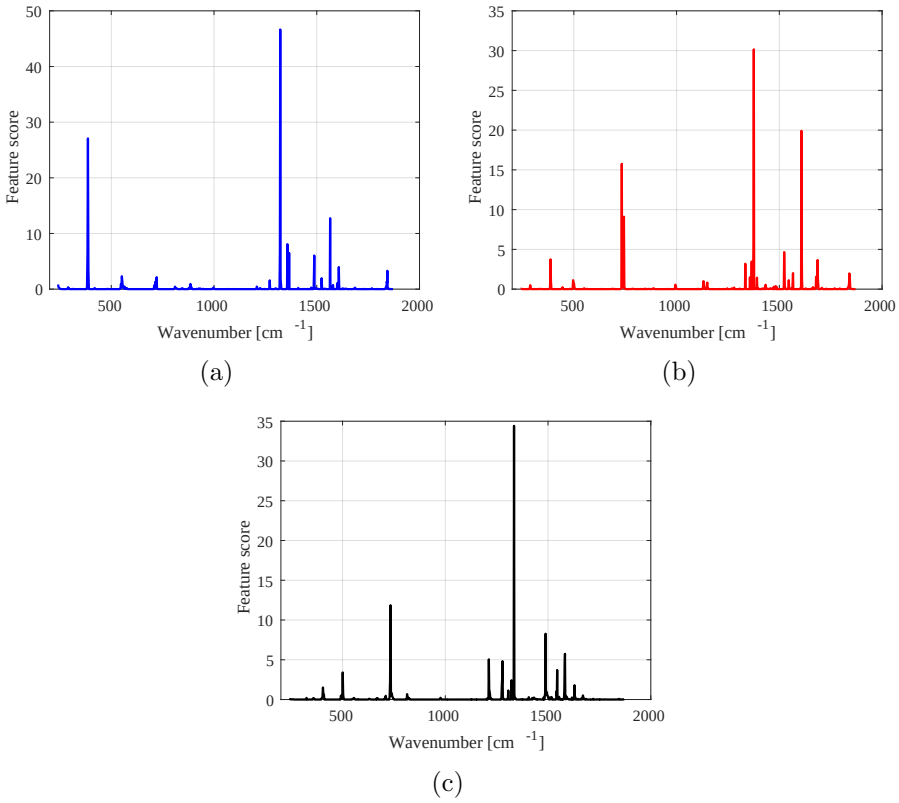
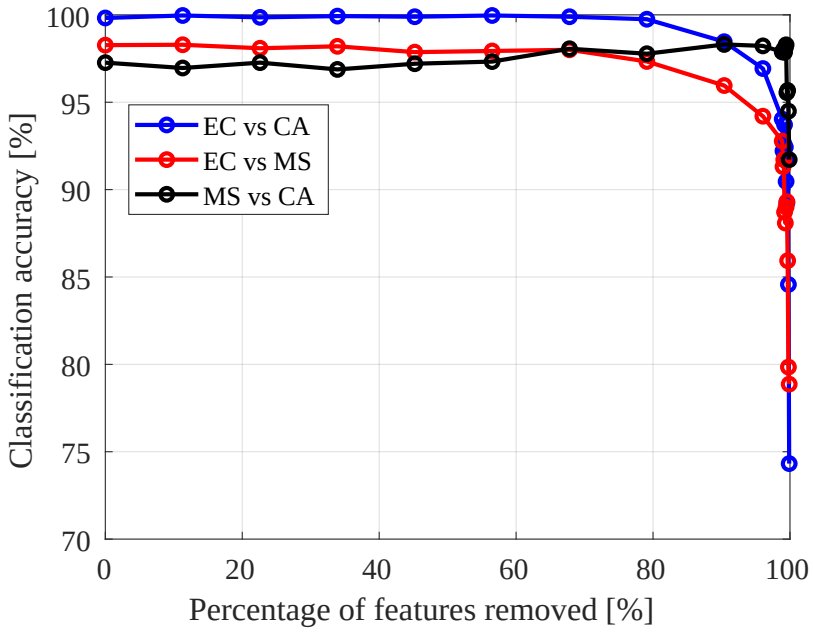


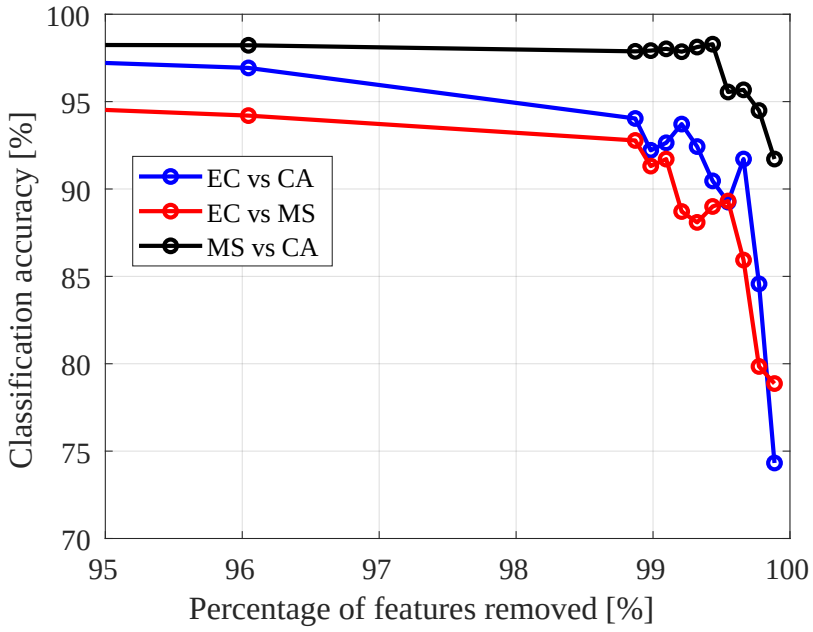
Figure 4.5: Feature score for the binary models: (a) EC vs CA, (b) EC vs MS, and (c) MS vs CA.

words, the removal process starts with removing the features with the minimum score. As a result, the accuracy remains high until a certain percentage of features is reached, then it starts to decrease gradually. That percentage differs from one model to the other. For example, we can achieve more than 95% classification accuracy for the EC vs CA model when up to 96% of the features are removed. Hence, we can use few features for the EC vs CA classification without a penalty in the classification accuracy. Similarly, it can be observed that at least 10% and 0.5% of the features are required to maintain more than 95% classification accuracy for the EC vs MS and MS vs CA binary models, respectively.

As a final result for the binary classification, we perform a classification process for the three binary models after reducing on both the samples and features to find the reduced data set. The reduced data set is selected for each binary model with respect to the reduced



(a)



(b)

Figure 4.6: Classification accuracy vs features being removed for binary models EC vs CA, EC vs MS, and MS vs CA. (a) Entire range, and (b) details of the sweep range.

number of features and the reduced number of samples achieving higher than 90% accuracy. The classification accuracy for the complete and reduced data set is shown in Table 4.2. By comparing the classification accuracy of the complete data set and the reduced data set, it can be observed that we maintained higher than 90% classification accuracy while achieving a significant reduction in the complexity and run time for the classification process. Based on our computation capabilities, we calculated the processing time over multiple servers and used the average value. The full data set and reduced data set finished processing 100 iterations in approximately 120 mins and 2.6 mins, respectively. This corresponds to more than 98% reduction in the processing time. Moreover, there is a significant reduction in the memory requirements for the reduced data set where the full data set and reduced data set require 20.1 MB and 0.2 MB of memory, respectively. This reduction can be beneficial for embedded applications. Hence, depending on the application requirements for accuracy and complexity, the reduced data set can be chosen.

	EC vs CA	EC vs MS	MS vs CA
Removed features	875	850	875
Removed samples %	70	50	60
Classification accuracy %	96.35	91.85	97.67
Classification accuracy of complete data set%	99.8	99.8	96.5

Table 4.2: Reduced data sets and the corresponding classification accuracy for the binary models.

4.5.2 Multiclass classification

In this section, I present the classification results using multiclass classification based on the same data set used for binary classification in the previous section. In our results, we consider using the OVA multiclass classification since it is generally less complex compared with the OVO. Hence, three classes are used in calculating the classification accuracy: EC vs CA + MS, CA vs EC + MS, and MS vs EC + CA.

First, we find the optimum value of λ, similar to the binary classification. Figure 4.7

shows the classification accuracy versus the λ parameter: the optimum performance of the OVA is achieved at $\lambda = 10^5$, which is the same value found to be optimum in the binary classification case.

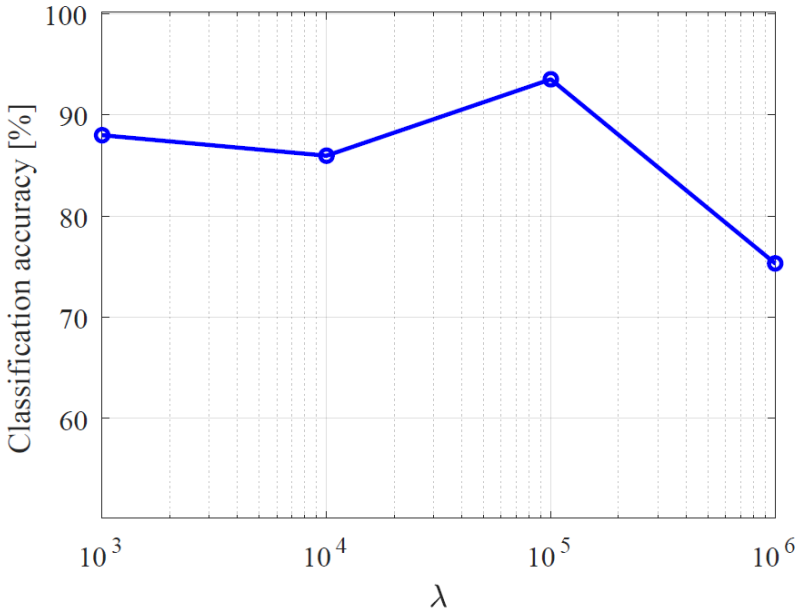


Figure 4.7: OVA classification accuracy over λ sweep.

After choosing the optimum value of λ, we focus on the dimensionality reduction. The results of the reduction in the sample dimension are presented first. Similar to the binary classification, the QF or SNR metric presented in the previous section where the QF for the three bacteria types are shown in Figure 4.3 is used for the reduction.

Figure 4.8 presents the classification accuracy versus the percentage of the samples removed (the blue curve). The black curve represents using a single capture/iteration per sample where the mean of the 10 captures is used per each file sample as one sample. It can be observed that the classification accuracy is reduced by approximately 10%. This can be attributed to using a smaller sample set by averaging the 10 captures. Also, the classification decision is based on a single sample, which causes more errors in the classification. This is in comparison to the errors in the majority of 10 samples, which occur only

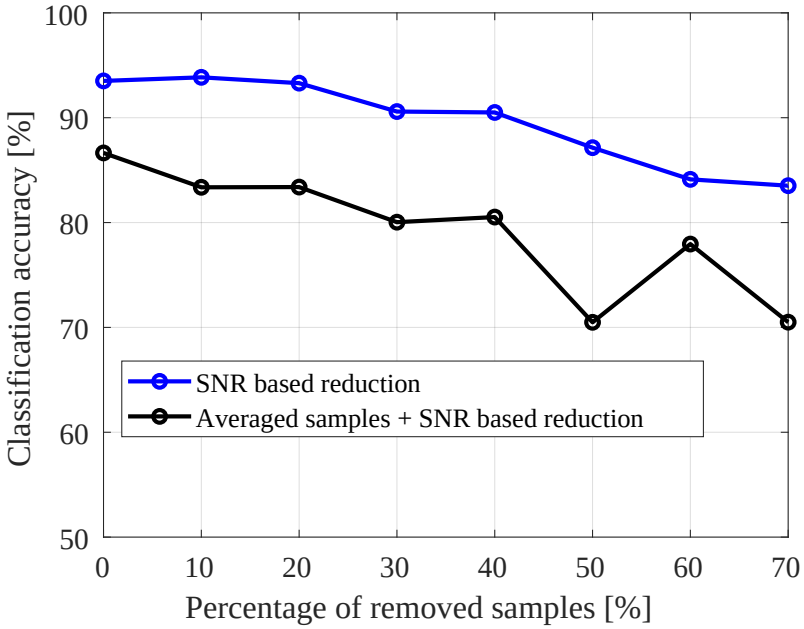


Figure 4.8: Multiclass classification accuracy versus percentage of samples removed based on SNR reduction.

when more than 5 samples in the same file are classified incorrectly.

Similar to the binary classification results, we can significantly reduce the number of samples while achieving a high classification accuracy. For example, we can achieve more than 80% and 90% classification accuracy after a reduction of the sample size by more than 70% and 40%, respectively.

For the feature reduction, we perform similar steps to the binary classification steps. First, we calculate the feature score of each bacteria class versus all. Figure 4.9 shows the feature scores of the three models used for the classification. Similar to the binary results discussed in the previous section, the features scores are significant among a small number of features, which means the rest of the features have a detrimental or negligible contribution to the classification accuracy.

Figure 4.10 presents the OVA classification accuracy versus the percentage of the removed features. The top figure shows the classification accuracy over the entire range of the removed features. It can be observed that the classification accuracy improves when

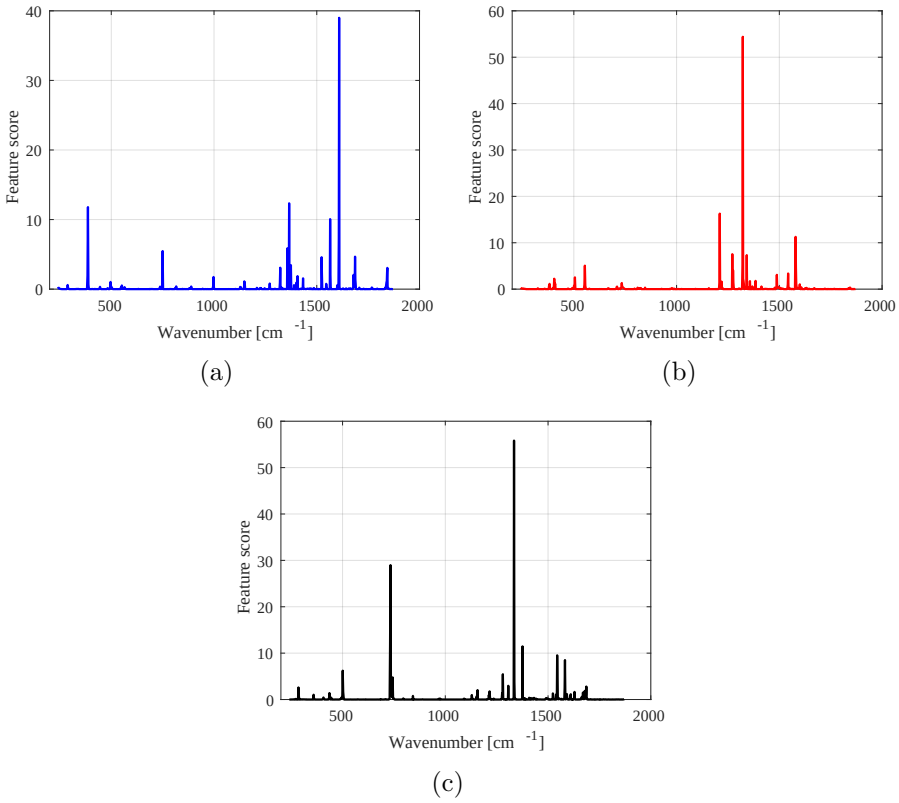
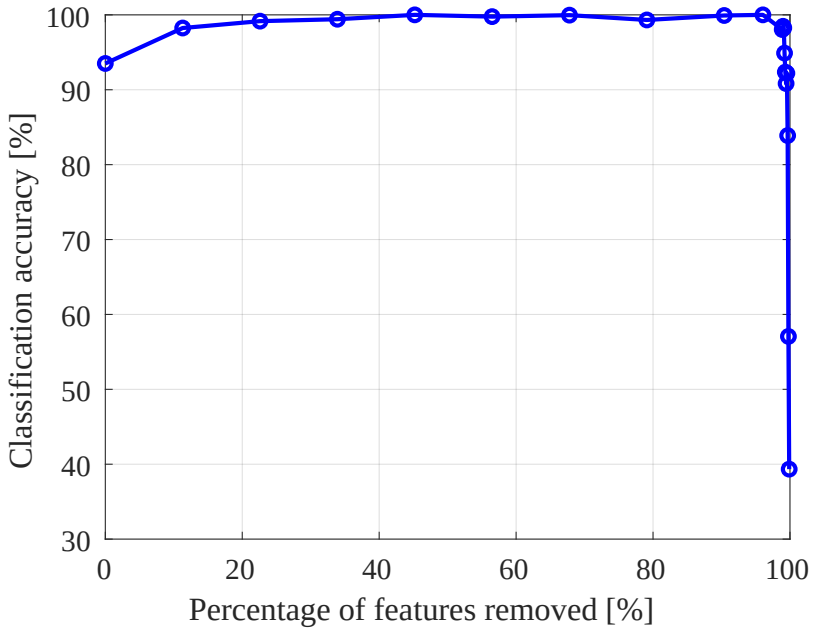


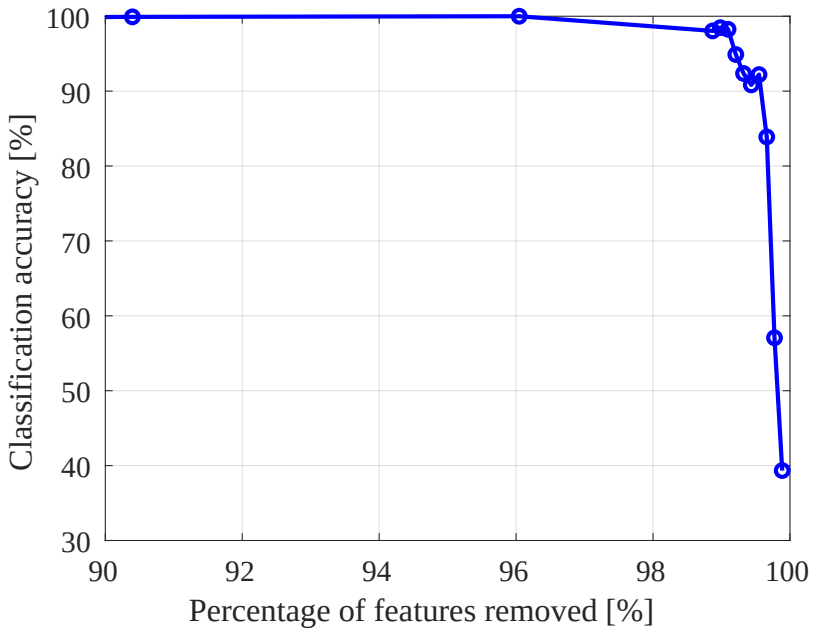
Figure 4.9: Feature score for the OVA models: (a) EC vs all, (b) CA vs all, and (c) MS vs all.

low-weighted featured scores are removed. After 40% of the features are removed, the classification accuracy is approximately unchanged until 96% of the features are removed. Figure 4.10(b) shows the classification accuracy for the range $\geq$ 95% features removed. The classification accuracy decreases with further removal of features due to the removal of highly weighted features. Hence, the few highly weighted features should be preserved to achieve a high classification accuracy and significantly decrease the complexity.

Finally, the classification accuracy is calculated for the reduced data set with respect of the number of samples and number of features. We consider a case for the reduced data set where we target $>$ 90% classification accuracy, which will result in a trade-off between classification accuracy and complexity which can be beneficial in certain applications where a short processing time is important. The percentage of removed samples is chosen to be



(a)



(b)

Figure 4.10: Multiclass classification accuracy versus percentage of removed features. (a) Entire range, and (b) last 10% range.

60% of the samples, and 880 features of the lower weighted features are removed. The classification accuracy of the reduced data set is calculated and found to be 92.9%. Table 4.3 compares the reduced data set to the full data set case. Also, the processing time based on our computer is shown in the table where the full data set and reduced data set needs approximately 246 mins and 5 mins, respectively to process 100 iterations of the accuracy, This corresponds to approximately 98% reduction in the processing time per sample. Similarly, we achieve more than 98% reduction in the memory requirements by the reduced data set compared to the full data set as shown in table 4.3.

	Complete data set	reduced data set
Removed features	0	880
Removed samples %	0	60
Classification accuracy %	93.50	92.9
Processing time	246 mins	5 mins
Memory requirements	$\sim$ 10 MB	$\sim$ 30 KB

Table 4.3: Reduced data set and the corresponding classification accuracy for the OVA model.

4.6 Conclusion

This Chapter presented the classification results for three different bacteria types using two classification methods.

First, using binary classification, I presented the simulation results for three binary models: EC vs CA, EC vs MS, and MS vs CA. Using the full data set, a classification accuracy of more than 95% was achieved for the three models. The complexity of the samples data set was then reduced based on the sample SNR. Using less than 60% of the original data set size, the classification accuracy was maintained at more than 95% for the three models. The RFE-SVM algorithm was then used to reduce complexity in the feature dimension. It was noted that a small number of features were heavy weighted

compared with the rest of the features. Hence, we could significantly reduce the number of features used in the classification and maintain a high classification accuracy. This shows that bacteria identification between two different types of bacteria can be more rapid and accurate.

Second, I presented the classification accuracy using the multiclass OVA method. Using the complete data set, the OVA method showed more than 90% classification accuracy. Similar to the binary model, dimension reduction was applied to the input samples. Using the SNR reduction to reduce the input samples by more than 60% resulted in a classification accuracy higher than 80%. Furthermore, using the RFE-algorithm to reduce the complexity on the features, and using only the 5% top-weighted features, resulted in a classification accuracy of approximately 92%. Finally, by combining both reduction dimensions and the classification accuracy for the reduced data set, a classification accuracy of more than 92% was achieved using a significantly reduced data set. Moreover, we achieve more than 98% reduction in the processing time and memory requirements using the reduced data set compared to the full data set. Similar to the binary classification, these results show the potential for fast and accurate classification between multiple types of bacteria in a sample.

Chapter 5

Conclusion and future work

5.1 Overview

Recently, Raman spectroscopy has been widely used in different fields due to its effective and rapid results. The captured Raman spectrum contains detailed information about a sample's structure, chemical components, and molecular interactions. As a result, Raman based biosensors have gained significant attention in the biological field. They are used in the classification and identification of different types of bacteria based on the fact that every bacteria has its unique Raman fingerprint.

However, using Raman spectroscopy to identify bacteria poses a few challenges. The captured Raman spectrum is contaminated with different noise sources. Also, the signal is multi-variate and non-linear due to the correlation between the Raman features or peaks. As well, the Raman signal is very weak compared with Rayleigh scattering. Using ML and SERS, we can overcome such challenges and achieve high classification accuracy between different bacteria types. Moreover, the processing times can be significantly reduced compared with the days needed using the bacterial culture method. Hence, rapid, accurate, and non-invasive bacteria identification can be achieved in a way that could have a significant impact on a patient's health.

5.2 Summary of original contributions

In this thesis, I presented different results for bacteria classifications based on captured Raman spectra. The main contributions of this thesis are as follows:

- Using binary classification, I presented the results for three binary models: EC vs CA, EC vs MS, and MS vs CA. Using the full dataset, classification accuracy of more than 95% was achieved for all three models. Then, I showed results based on reducing the complexity of both the sample size and its feature. First, using less than 60% of the original data set sample size, classification accuracy was maintained at more than 95% for all three models. Then, the RFE-SVM algorithm was applied to calculate the feature weights and reduce the complexity in the feature dimension. Given that a small number of features were more heavily weighted than the rest of the features, the number of features used was significantly reduced to 10 for the classification. Using both SNR and RFE reduction, the classification accuracy was 96.35% for the EC vs CA model, 91.85% for the EC vs MS model, and 97.67% for the MS vs CA model. Moreover, the full data set and reduced data set needed approximately 120 mins and 2.6 mins, respectively to train and process 100 iterations excluding the feature score calculation. This corresponds to more than 98% reduction in the processing time. Similarly, we achieve more than 98% reduction in the memory requirements which will be beneficial for embedded applications.

- I also presented the classification accuracy results of using the multiclass OVA method. Using the complete data set, the OVA method showed more than 90% classification accuracy. Similar to the binary model, dimension reduction was then applied to the input samples. Using the SNR reduction, the input samples were reduced by more than 60% while maintaining a classification accuracy higher than 80%. By using the RFE-algorithm to reduce the complexity of the features, and using only the 5% top-weighted features, classification accuracy of approximately 92% was achieved.

Finally, by combining both reduction dimensions and the classification accuracy for the reduced data set, classification accuracy of more than 92% can be achieved using a significantly reduced data set. Moreover, we achieve more than 98% reduction in the processing time and memory requirements using the reduced data set compared to the full data set.

5.3 Future work

In this thesis, I presented ways to improve the accuracy and speed of bacteria identification using Raman scattering and machine learning, based on binary and multiclass models. Opportunities for future research exist in further investigating modifications to the ML algorithm and the feature reduction techniques. The aim of these suggested ideas is to increase accuracy and reduce processing time.

For the ML algorithm, different types of kernel functions can be used with the SVM algorithm. The performance of such kernel functions can be compared to the RQK used in our work. It is expected that the trade-off between complexity and accuracy can be further optimized based on this application. For example, the exponential radial basis function (RBF) can be used, as expressed in the following equation [60]:

$$k_{gauss}(x_i, x_j) = exp(-\frac{||x_i - x_j||}{2\sigma^2}) \tag{5.1}$$

where this kernel function has a lower number of hyperparameters compared with the RQK, which means less complexity in terms of the training and optimization process.

For the feature reduction techniques, two approaches based on using the SVM-RFE technique with a few modifications could be tested.

1. Using the enhanced RFE algorithm (EnRFE): This algorithm is similar to the operation of the SVM-RFE in ranking the features based on their weight. Also, similar steps are followed in the weight calculation. The main difference is in the reduction.

For the SVM-RFE, the minimum weight feature is directly removed. On the other hand, the EnRFE calculates the effect of removing the least-weighted feature on the classification accuracy. If removing such a feature causes a reduction in the classification accuracy, it will not be removed from the feature set even if it has a minimum weight. Otherwise, it will be removed from the feature set. This process is then repeated on all EnRFE [110]. It is expected that using such a method would achieve a better classification accuracy than using the SVM-RFE; however, this would be at the expense of higher complexity in the RFE process, since it involves calculating the classification accuracy before deciding on removing any weight.

2. Using a minimum-redundancy maximum-relevancy (MRMR) filter along with the SVM-RFE to get more accurate classification accuracy with a smaller number of features: This modified algorithm combines two feature selection techniques, the filter (MRMR) and the embedded method (SVM-RFE). The SVM-RFE selects the features based on their weight and their relevance without taking the redundancy into consideration which can affect the classification accuracy. As a result, the MRMR filter is used to overcome this issue by reducing the redundancy in the features that affects the classification accuracy [130].

Moreover, we can include the performance of the system in the presence of an unknown class in the validation process. Using the current model used in our results. If a different bacteria type is used in the validation process, it will be mapped to one of the three known classes used in the training of the MLA. Hence, it will be classified incorrectly. To overcome such disadvantage, we can add a threshold on the empirical loss where the unknown sample can be discarded from the classification.

Finally, a comparison could be made between the classification performance of different types of ML algorithms. Such comparison would be based on the supervised and the unsupervised ML algorithms by using our implemented algorithm SVM-RFE and the deep neural network (DNN) [131], respectively.

Appendix

Training system accuracy

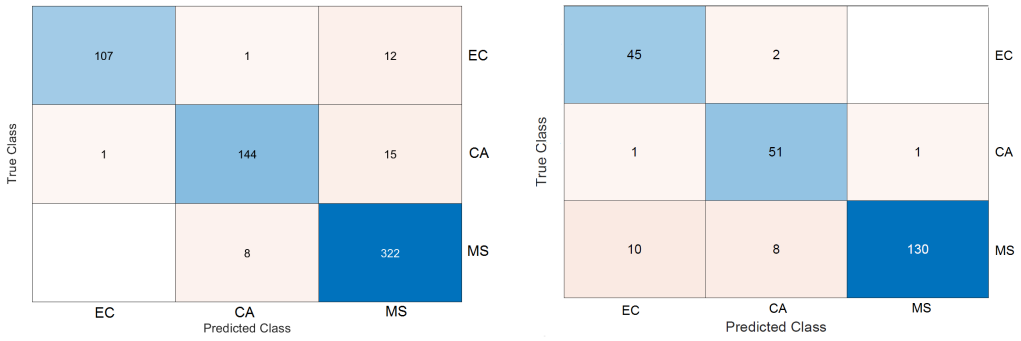
The training performance expresses the accuracy of the training of the MLA used in the classification process. It can be calculated by the empirical loss which is expressed in Eq. (3.2). The MLA is trained by 60 % of the data set, and 40 % is used for for calculating the empirical loss for the binary models. The following table shows the calculated empirical loss where it follows the performance of the binary models described in Chapter 4. In other words, the loss is high for the MS vs CA where the classification accuracy is lower than the other two models.

	EC vs CA	EC vs MS	MS vs CA
Empirical loss	55.83	47.73	107.34

Empirical loss for the binary models.

Confusion matrices

The multiclass classification accuracy can be detailed for each bacteria type using the confusion matrix, where the correct and incorrect class mapping can be calculated. Figure 5.1 present the confusion matrices for the full and reduced data sets for one iteration. It can be observed that the accuracy values in figure 5.1 and figure 4.3 are not identical due to the randomness in selecting the training and the testing sets.



(a) (b)

Figure 5.1: Confusion matrix (a) complete data set, and (b) reduced data set.

References

[1] P. Mehrotra, "Biosensors and their applications-A review," *Journal of oral biology and craniofacial research*, vol. 6, no. 2, pp. 153–159, 2016.

[2] B. Nikhil, J. Pawan, F. Nello, and E. Pedro, "Introduction to biosensors," *Essays Biochem*, vol. 60, no. 1, pp. 1–8, 2016.

[3] M. Bloomfield, D. Andrews, P. Loeffen, C. Tombling, T. York, and P. Matousek, "Non-invasive identification of incoming raw pharmaceutical materials using spatially offset Raman spectroscopy," *Journal of pharmaceutical and biomedical analysis*, vol. 76, pp. 65–69, 2013.

[4] X. Cao, Z.-q. Wen, T. Wang, D. Meriage, L. A. Craig, K. Parks, and C. Undey, "Development considerations of adapting Raman spectroscopy for raw material fingerprinting," *Raman Spectroscopy: Tools, Techniques, and Applications*, p. 2, 2018.

[5] T. E. Matthews, C. Coffman, D. Kolwyck, D. Hill, and J. E. Dickens, "Enabling robust and rapid raw material identification and release by handheld Raman spectroscopy," *PDA Journal of Pharmaceutical Science and Technology*, vol. 73, no. 4, pp. 356–372, 2019.

[6] E. Hanlon, R. Manoharan, T. Koo, K. Shafer, J. Motz, M. Fitzmaurice, J. Kramer, I. Itzkan, R. Dasari, and M. Feld, "Prospects for in vivo Raman spectroscopy," *Physics in Medicine & Biology*, vol. 45, no. 2, p. R1, 2000.

[7] R. Baker, P. Matousek, K. L. Ronayne, A. W. Parker, K. Rogers, and N. Stone, "Depth profiling of calcifications in breast tissue using picosecond Kerr-gated Raman spectroscopy," *Analyst*, vol. 132, no. 1, pp. 48–53, 2007.

[8] M. E. Darvin, C.-S. Choe, J. Schleusener, and J. Lademann, "Non-invasive depth profiling of the stratum corneum in vivo using confocal Raman microscopy considering the non-homogeneous distribution of keratin," *Biomedical optics express*, vol. 10, no. 6, pp. 3092–3103, 2019.

[9] W. W. Chiu, J. Travaš-Sejdić, R. P. Cooney, and G. A. Bowmaker, "Studies of dopant effects in poly (3, 4-ethylenedi-oxythiophene) using Raman spectroscopy," *Journal of Raman Spectroscopy*, vol. 37, no. 12, pp. 1354–1361, 2006.

[10] S. Wang, B. Cheng, C. Wang, S. Dai, K. Jin, Y. Zhou, H. Lu, Z. Chen, and G. Yang, "Raman spectroscopy studies of Ce-doping effects on Ba 0.5 Sr 0.5 Tio 3 thin films," *Journal of applied physics*, vol. 99, no. 1, p. 013504, 2006.

[11] N. S. Mueller, S. Heeg, M. P. Alvarez, P. Kusch, S. Wasserroth, N. Clark, F. Schedin, J. Parthenios, K. Papagelis, C. Galiotis *et al.*, "Evaluating arbitrary strain configurations and doping in graphene with Raman spectroscopy," *2D Materials*, vol. 5, no. 1, p. 015016, 2017.

[12] B. Sharma, R. R. Frontiera, A.-I. Henry, E. Ringe, and R. P. Van Duyne, "SERS: Materials, applications, and the future," *Materials today*, vol. 15, no. 1-2, pp. 16–25, 2012.

[13] J.-C. Lagier, S. Edouard, I. Pagnier, O. Mediannikov, M. Drancourt, and D. Raoult, "Current and past strategies for bacterial culture in clinical microbiology," *Clinical microbiology reviews*, vol. 28, no. 1, pp. 208–236, 2015.

[14] I. Barman, C.-R. Kong, N. C. Dingari, R. R. Dasari, and M. S. Feld, "Development of robust calibration models using support vector machines for spectroscopic monitoring of blood glucose," *Analytical chemistry*, vol. 82, no. 23, pp. 9719–9726, 2010.

[15] M. F. Akay, "Support vector machines combined with feature selection for breast cancer diagnosis," *Expert systems with applications*, vol. 36, no. 2, pp. 3240–3247, 2009.

[16] P. Rösch, M. Harz, M. Schmitt, K.-D. Peschke, O. Ronneberger, H. Burkhardt, H.-W. Motzkus, M. Lankers, S. Hofer, H. Thiele *et al.*, "Chemotaxonomic identification of single bacteria by micro-Raman spectroscopy: application to clean-room-relevant biological contaminations," *Applied and environmental microbiology*, vol. 71, no. 3, pp. 1626–1637, 2005.

[17] U. Neugebauer, J. H. Clement, T. Bocklitz, C. Krafft, and J. Popp, "Identification and differentiation of single cells from peripheral blood by Raman spectroscopic imaging," *Journal of biophotonics*, vol. 3, no. 8-9, pp. 579–587, 2010.

[18] L. J. Jacob and H.-P. Deigner, "Nanoparticles and nanosized structures in diagnostics and therapy," in *Precision Medicine*. Elsevier, 2018, pp. 229–252.

[19] https://www.horiba.com/en_en/raman-imaging-and-spectroscopy/#:~:text=Raman.

[20] T. Burke, D. Smith, and A. Nielsen, "The molecular structure of Mof6, WF6, and UF6 from infrared and Raman spectra," *The Journal of Chemical Physics*, vol. 20, no. 3, pp. 447–454, 1952.

[21] D. W. Shipp, F. Sinjab, and I. Notingher, "Raman spectroscopy: techniques and applications in the life sciences," *Advances in Optics and Photonics*, vol. 9, no. 2, pp. 315–428, 2017.

[22] R. R. Jones, D. C. Hooper, L. Zhang, D. Wolverson, and V. K. Valev, "Raman techniques: fundamentals and frontiers," *Nanoscale research letters*, vol. 14, no. 1, pp. 1–34, 2019.

[23] D. A. Long, "Raman spectroscopy," *New York*, vol. 1, 1977.

[24] W. M. Tolles, J. W. Nibler, J. McDonald, and A. B. Harvey, "A review of the theory and application of coherent anti-Stokes Raman spectroscopy (CARS)," *Applied Spectroscopy*, vol. 31, no. 4, pp. 253–271, 1977.

[25] D. Cialla, A. März, R. Böhme, F. Theil, K. Weber, M. Schmitt, and J. Popp, "Surface-enhanced Raman spectroscopy (SERS): progress and trends," *Analytical and bioanalytical chemistry*, vol. 403, no. 1, pp. 27–54, 2012.

[26] R. Zhang, Y. Zhang, Z. Dong, S. Jiang, C. Zhang, L. Chen, L. Zhang, Y. Liao, J. Aizpurua, Y. e. Luo *et al.*, "Chemical mapping of a single molecule by plasmon-enhanced Raman scattering," *Nature*, vol. 498, no. 7452, pp. 82–86, 2013.

[27] M. Moskovits, "Surface-enhanced Raman spectroscopy: a brief perspective," in *Surface-Enhanced Raman Scattering*. Springer, 2006, pp. 1–17.

[28] A. Campion and P. Kambhampati, "Surface-enhanced Raman scattering," *Chemical society reviews*, vol. 27, no. 4, pp. 241–250, 1998.

[29] R. A. Tripp, R. A. Dluhy, and Y. Zhao, "Novel nanostructures for SERS biosensing," *Nano Today*, vol. 3, no. 3-4, pp. 31–37, 2008.

[30] H. T. Beier, C. B. Cowan, I.-H. Chou, J. Pallikal, J. E. Henry, M. E. Benford, J. B. Jackson, T. A. Good, G. L. Coté *et al.*, "Application of surface-enhanced Raman spectroscopy for detection of beta amyloid using nanoshells," *Plasmonics*, vol. 2, no. 2, pp. 55–64, 2007.

[31] M. E. Benford, I.-H. Chou, H. T. Beier, M. Wang, J. Kameoka, T. A. Good, and G. L. Coté, "In vitro detection of beta amyloid exploiting surface enhanced Raman scattering (SERS) using a nanofluidic biosensor," in *Plasmonics in Biology and Medicine V*, vol. 6869. International Society for Optics and Photonics, 2008, p. 68690W.

[32] D. S. Grubisha, R. J. Lipert, H.-Y. Park, J. Driskell, and M. D. Porter, "Femtomolar detection of prostate-specific antigen: an immunoassay based on surface-enhanced Raman scattering and immunogold labels," *Analytical chemistry*, vol. 75, no. 21, pp. 5936–5943, 2003.

[33] A. M. Mohs, M. C. Mancini, S. Singhal, J. M. Provenzale, B. Leyland-Jones, M. D. Wang, and S. Nie, "Hand-held spectroscopic device for in vivo and intraoperative tumor detection: contrast enhancement, detection sensitivity, and tissue penetration," *Analytical chemistry*, vol. 82, no. 21, pp. 9058–9065, 2010.

[34] A. Mobed, S. Razavi, A. Ahmadalipour, S. K. Shakouri, and G. Koohkan, "Biosensors in Parkinson's disease," *Clinica Chimica Acta*, vol. 518, pp. 51–58, 2021.

[35] X. Yang, H. Li, X. Zhao, W. Liao, C. X. Zhang, and Z. Yang, "A novel, label-free liquid crystal biosensor for Parkinson's disease related alpha-synuclein," *Chemical Communications*, vol. 56, no. 40, pp. 5441–5444, 2020.

[36] M. Fatima, M. Pasha *et al.*, "Survey of machine learning algorithms for disease diagnostic," *Journal of Intelligent Learning Systems and Applications*, vol. 9, no. 01, p. 1, 2017.

[37] G. Battineni, G. G. Sagaro, N. Chinatalapudi, and F. Amenta, "Applications of machine learning predictive models in the chronic disease diagnosis," *Journal of personalized medicine*, vol. 10, no. 2, p. 21, 2020.

[38] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE access*, vol. 7, pp. 19 143–19 165, 2019.

[39] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.

[40] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–6.

[41] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1085–1088.

[42] J. Rzeszótko and S. H. Nguyen, "Machine learning for traffic prediction," *Fundamenta Informaticae*, vol. 119, no. 3-4, pp. 407–420, 2012.

[43] A. Boukerche and J. Wang, "Machine learning-based traffic prediction models for intelligent transportation systems," *Computer Networks*, vol. 181, p. 107530, 2020.

[44] S. Yan, F. N. Khan, A. Mavromatis, D. Gkounis, Q. Fan, F. Ntavou, K. Nikolovgenis, F. Meng, E. H. Salas, C. Guo *et al.*, "Field trial of machine-learning-assisted and SDN-based optical network planning with network-scale monitoring database," in *2017 European Conference on Optical Communication (ECOC)*. IEEE, 2017, pp. 1–3.

[45] D. Wang, M. Zhang, Z. Li, J. Li, M. Fu, Y. Cui, and X. Chen, "Modulation format recognition and OSNR estimation using CNN-based deep learning," *IEEE Photonics Technology Letters*, vol. 29, no. 19, pp. 1667–1670, 2017.

[46] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," in *2011 IEEE control and system graduate research colloquium*. IEEE, 2011, pp. 37–42.

[47] T. O. Ayodele, "Types of machine learning algorithms," *New advances in machine learning*, vol. 3, pp. 19–48, 2010.

[48] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[49] S. Ray, "A quick review of machine learning algorithms," in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 35–39.

[50] J. Kirchner, A. Heberle, and W. Löwe, "Classification vs. regression-machine learning approaches for service recommendation based on measured consumer experiences," in *2015 IEEE World Congress on Services*. IEEE, 2015, pp. 278–285.

[51] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[52] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[53] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

[54] Y. Ma and G. Guo, *Support vector machines applications*. Springer, 2014, vol. 649.

[55] A. Tzotsos and D. Argialas, "Support vector machine classification for object-based image analysis," in *Object-Based Image Analysis*. Springer, 2008, pp. 663–677.

[56] A. Mondal, S. Kundu, S. K. Chandniha, R. Shukla, and P. Mishra, "Comparison of support vector machine and maximum likelihood classification technique using satellite imagery," *International Journal of Remote Sensing and GIS*, vol. 1, no. 2, pp. 116–123, 2012.

[57] H. S. Jang, K. Y. Bae, H.-S. Park, and D. K. Sung, "Solar power prediction based on satellite images and support vector machine," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1255–1263, 2016.

[58] B. Schölkopf, A. J. Smola, F. Bach *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[59] K.-B. Duan, J. C. Rajapakse, and M. N. Nguyen, "One-versus-one and one-versus-all multiclass SVM-RFE for gene selection in cancer classification," in *European conference on evolutionary computation, machine learning and data mining in bioinformatics*. Springer, 2007, pp. 47–56.

[60] M. Somvanshi, P. Chavan, S. Tambade, and S. Shinde, "A review of machine learning techniques using decision tree and support vector machine," in *2016 international conference on computing communication control and automation (ICCUBEA)*. IEEE, 2016, pp. 1–7.

[61] V. Jakkula, "Tutorial on support vector machine (SVM)," *School of EECS, Washington State University*, vol. 37, no. 2.5, p. 3, 2006.

[62] J. Shawe-Taylor, N. Cristianini *et al.*, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[63] Y. Tajiri, R. Yabuwaki, T. Kitamura, and S. Abe, "Feature extraction using support vector machines," in *International Conference on Neural Information Processing*. Springer, 2010, pp. 108–115.

[64] E. Alpaydin, "Introduction to machine learning ethem alpaydin," *Introd. to Mach. Learn*, 2014.

[65] J. L. Rojo-Álvarez, M. Martínez-Ramón, J. Muñoz-Marí, and G. Camps-Valls, "Support vector machine and kernel classification algorithms," 2018.

[66] M. Hofmann, "Support vector machines-kernels and the kernel trick," *Notes*, vol. 26, no. 3, pp. 1–16, 2006.

[67] A.-H. Karimi, "A summary of the kernel matrix, and how to learn it effectively using semidefinite programming," *arXiv preprint arXiv:1709.06557*, 2017.

[68] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5g cloud radio access networks," in *2020 international symposium on networks, computers and communications (ISNCC)*. IEEE, 2020, pp. 1–5.

[69] R. Hunter and H. Anis, "Genetic support vector machines as powerful tools for the analysis of biomedical Raman spectra," *Journal of Raman Spectroscopy*, vol. 49, no. 9, pp. 1435–1444, 2018.

[70] Y. Xue, L. Zhang, B. Wang, Z. Zhang, and F. Li, "Nonlinear feature selection using Gaussian kernel SVM-RFE for fault diagnosis," *Applied Intelligence*, vol. 48, no. 10, pp. 3306–3331, 2018.

[71] M. Achirul Nanda, K. Boro Seminar, D. Nandika, and A. Maddu, "A comparison study of kernel functions in the support vector machine and its application for termite detection," *Information*, vol. 9, no. 1, p. 5, 2018.

[72] R. Hunter, A. N. Sohi, Z. Khatoon, V. R. Berthiaume, E. I. Alarcon, M. Godin, and H. Anis, "Optofluidic label-free SERS platform for rapid bacteria detection in serum," *Sensors and Actuators B: Chemical*, vol. 300, p. 126907, 2019.

[73] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM parameter optimization using grid search and genetic algorithm to improve classification performance," *Telkomnika*, vol. 14, no. 4, p. 1502, 2016.

[74] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[75] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural computation*, vol. 13, no. 3, pp. 637–649, 2001.

[76] R. L. McCreery, *Raman spectroscopy for chemical analysis*. John Wiley & Sons, 2005, vol. 225.

[77] J. Smulko, M. S. Wróbel, and I. Barman, "Noise in biological Raman spectroscopy," in *2015 International Conference on Noise and Fluctuations (ICNF)*. IEEE, 2015, pp. 1–6.

[78] T. Bocklitz, A. Walter, K. Hartmann, P. Rösch, and J. Popp, "How to pre-process Raman spectra for reliable and stable models?" *Analytica chimica acta*, vol. 704, no. 1-2, pp. 47–56, 2011.

[79] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[80] T. W. Randolph, "Scale-based normalization of spectral data," *Cancer Biomarkers*, vol. 2, no. 3-4, pp. 135–144, 2006.

[81] R. Gautam, S. Vanga, F. Ariese, and S. Umapathy, "Review of multidimensional data processing approaches for Raman and infrared spectroscopy," *EPJ Techniques and Instrumentation*, vol. 2, no. 1, pp. 1–38, 2015.

[82] F. Gan, G. Ruan, and J. Mo, "Baseline correction by improved iterative polynomial fitting with automatic threshold," *Chemometrics and Intelligent Laboratory Systems*, vol. 82, no. 1-2, pp. 59–65, 2006.

[83] A. Jirasek, G. Schulze, M. Yu, M. Blades, and R. Turner, "Accuracy and precision of manual baseline determination," *Applied spectroscopy*, vol. 58, no. 12, pp. 1488–1499, 2004.

[84] C. A. Lieber and A. Mahadevan-Jansen, "Automated method for subtraction of fluorescence from biological Raman spectra," *Applied spectroscopy*, vol. 57, no. 11, pp. 1363–1367, 2003.

[85] J. Zhao, H. Lui, D. I. McLean, and H. Zeng, "Automated autofluorescence background subtraction algorithm for biomedical Raman spectroscopy," *Applied spectroscopy*, vol. 61, no. 11, pp. 1225–1232, 2007.

[86] L. Shao and P. R. Griffiths, "Automatic baseline correction by wavelet transform for quantitative open-path fourier transform infrared spectroscopy," *Environmental science & technology*, vol. 41, no. 20, pp. 7054–7059, 2007.

[87] Z.-M. Zhang, S. Chen, Y.-Z. Liang, Z.-X. Liu, Q.-M. Zhang, L.-X. Ding, F. Ye, and H. Zhou, "An intelligent background-correction algorithm for highly fluorescent samples in Raman spectroscopy," *Journal of Raman spectroscopy*, vol. 41, no. 6, pp. 659–669, 2010.

[88] X.-G. Shao, A. K.-M. Leung, and F.-T. Chau, "Wavelet: a new trend in chemistry," *Accounts of Chemical Research*, vol. 36, no. 4, pp. 276–283, 2003.

[89] X. Shao, W. Cai, and Z. Pan, "Wavelet transform and its applications in high performance liquid chromatography (HPLC) analysis," *Chemometrics and intelligent laboratory systems*, vol. 45, no. 1-2, pp. 249–256, 1999.

[90] F. Zhang, X. Tang, A. Tong, B. Wang, and J. Wang, "An automatic baseline correction method based on the penalized least squares method," *Sensors*, vol. 20, no. 7, p. 2015, 2020.

[91] Y.-Z. Liang, A. K.-M. Leung, and F.-T. Chau, "A roughness penalty approach and its application to noisy hyphenated chromatographic two-way data," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 13, no. 5, pp. 511–524, 1999.

[92] H. F. Boelens, R. J. Dijkstra, P. H. Eilers, F. Fitzpatrick, and J. A. Westerhuis, "New background correction method for liquid chromatography with diode array detection, infrared spectroscopic detection and Raman spectroscopic detection," *Journal of chromatography A*, vol. 1057, no. 1-2, pp. 21–30, 2004.

[93] Y. Cai, C. Yang, D. Xu, and W. Gui, "Baseline correction for Raman spectra using penalized spline smoothing based on vector transformation," *Analytical methods*, vol. 10, no. 28, pp. 3525–3533, 2018.

[94] S. He, S. Fang, X. Liu, W. Zhang, W. Xie, H. Zhang, D. Wei, W. Fu, and D. Pei, "Investigation of a genetic algorithm based cubic spline smoothing for baseline correction of Raman spectra," *Chemometrics and Intelligent Laboratory Systems*, vol. 152, pp. 1–9, 2016.

[95] P. J. Green and B. W. Silverman, *Nonparametric regression and generalized linear models: a roughness penalty approach*. Crc Press, 1993.

[96] J. Peng, S. Peng, A. Jiang, J. Wei, C. Li, and J. Tan, "Asymmetric least squares for multiple spectra baseline correction," *Analytica chimica acta*, vol. 683, no. 1, pp. 63–68, 2010.

[97] P. H. Eilers and H. F. Boelens, "Baseline correction with asymmetric least squares smoothing," *Leiden University Medical Centre Report*, vol. 1, no. 1, p. 5, 2005.

[98] Z.-M. Zhang, S. Chen, and Y.-Z. Liang, "Baseline correction using adaptive iteratively reweighted penalized least squares," *Analyst*, vol. 135, no. 5, pp. 1138–1146, 2010.

[99] Y. Xie, L. Yang, X. Sun, D. Wu, Q. Chen, Y. Zeng, and G. Liu, "An auto-adaptive background subtraction method for Raman spectra," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 161, pp. 58–63, 2016.

[100] P. H. Eilers, "A perfect smoother," *Analytical chemistry*, vol. 75, no. 14, pp. 3631–3636, 2003.

[101] S. He, W. Zhang, L. Liu, Y. Huang, J. He, W. Xie, P. Wu, and C. Du, "Baseline correction for raman spectra using an improved asymmetric least squares method," *Analytical Methods*, vol. 6, no. 12, pp. 4402–4407, 2014.

[102] "airpls opensource matlab code." https://code.google.com/archive/p/airpls/.

[103] J. C. Cobas, M. A. Bernstein, M. Martín-Pastor, and P. G. Tahoces, "A new general-purpose fully automatic baseline-correction procedure for 1D and 2D NMR data," *Journal of Magnetic Resonance*, vol. 183, no. 1, pp. 145–151, 2006.

[104] N. Tarcea, J. Popp, J. Dubessy, M. Caumon, and F. Rull, "Raman data analysis," *Raman spectroscopy applied to earth sciences and cultural heritage, PPT*, 2012.

[105] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, no. 1-9, p. 2, 2005.

[106] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

[107] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[108] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature selection for high-dimensional data*. Springer, 2015.

[109] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in bioinformatics*, vol. 2015, 2015.

[110] X.-w. Chen and J. C. Jeong, "Enhanced recursive feature elimination," in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. IEEE, 2007, pp. 429–435.

[111] G. Manikandan, E. Susi, and S. Abirami, "Feature selection on high dimensional data using wrapper based subset selection," in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*. IEEE, 2017, pp. 320–325.

[112] X. Zeng, Y.-W. Chen, and C. Tao, "Feature selection using recursive feature elimination for handwritten digit recognition," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2009, pp. 1205–1208.

[113] F. Dallaire, F. Picot, J.-P. Tremblay, G. Sheehy, É. Lemoine, R. Agarwal, S. Kadoury, D. Trudel, F. Lesage, K. Petrecca *et al.*, "Quantitative spectral quality assessment technique validated using intraoperative in vivo Raman spectroscopy measurements," *Journal of Biomedical Optics*, vol. 25, no. 4, p. 040501, 2020.

[114] M. Člupek, P. Matějka, and K. Volka, "Noise reduction in Raman spectra: Finite impulse response filtration versus Savitzky-Golay smoothing," *Journal of Raman Spectroscop*, vol. 38, no. 9, pp. 1174–1179, 2007.

[115] S. Barton, "Methods for improving Signal to Noise Ratio in Raman spectra," Ph.D. dissertation, National University of Ireland Maynooth, 2019.

[116] A. Wang and E. A. Gehan, "Gene selection for microarray data analysis using principal component analysis," *Statistics in medicine*, vol. 24, no. 13, pp. 2069–2087, 2005.

[117] S. Raychaudhuri, J. M. Stuart, and R. B. Altman, "Principal components analysis to summarize microarray experiments: application to sporulation time series," in *Biocomputing 2000*. World Scientific, 1999, pp. 455–466.

[118] S. Jonnalagadda and R. Srinivasan, "Principal components analysis based methodology to identify differentially expressed genes in time-course microarray data," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–16, 2008.

[119] P. Cunningham, "Dimension reduction," in *Machine learning techniques for multimedia*. Springer, 2008, pp. 91–112.

[120] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[121] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 science and information conference*. IEEE, 2014, pp. 372–378.

[122] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[123] L. Zhang and X. Huang, "Multiple SVM-RFE for multi-class gene selection on DNA microarray data," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–6.

[124] P. M. Granitto and A. Burgos, "Feature selection on wide multiclass problems using OVA-RFE," *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, vol. 13, no. 44, pp. 27–34, 2009.

[125] X. Zhou and D. P. Tuck, "MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data," *Bioinformatics*, vol. 23, no. 9, pp. 1106–1114, 2007.

[126] P. A. Mosier-Boss, "Review on SERS of bacteria," *Biosensors*, vol. 7, no. 4, p. 51, 2017.

[127] F. Porcaro, L. Carlini, A. Ugolini, D. Visaggio, P. Visca, I. Fratoddi, I. Venditti, C. Meneghini, L. Simonelli, C. Marini *et al.*, "Synthesis and structural characterization of silver nanoparticles stabilized with 3-mercapto-1-propansulfonate and 1-thioglucose mixed thiols for antibacterial applications," *Materials*, vol. 9, no. 12, p. 1028, 2016.

[128] M. T. H. A. M. Deb, R. Hunter and H. Anis, "Rapid detection of bacteria using gold nanoparticles in SERS with three dierent capping agents: thioglucose, polyvinylpyrrolidone, and citrate," *Submitted and under review*, 2022.

[129] C. M. Phan and H. M. Nguyen, "Role of capping agent in wet synthesis of nanoparticles," *The Journal of Physical Chemistry A*, vol. 121, no. 17, pp. 3213–3219, 2017.

[130] P. A. Mundra and J. C. Rajapakse, "SVM-RFE with MRMR filter for gene selection," *IEEE transactions on nanobioscience*, vol. 9, no. 1, pp. 31–37, 2009.

[131] F. U. Ciloglu, A. Caliskan, A. M. Saridag, I. H. Kilic, M. Tokmakci, M. Kahraman, and O. Aydin, "Drug-resistant Staphylococcus aureus bacteria detection by combining surface-enhanced Raman spectroscopy (SERS) and deep learning techniques," *Scientific reports*, vol. 11, no. 1, pp. 1–12, 2021.