UNIVERSITY OF NAIROBI

RAPID ASSESSMENT OF PESTICIDE RESIDUES IN FRUITS AND VEGETABLES USING MACHINE LEARNING ASSISTED DIFFUSE REFLECTANCE SPECTROSCOPY

BY

NDUNG'U NDEGWA CHARLES
B.Sc (Hons)

A Thesis Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of Master of Science in Physics of the University of Nairobi.

DECLARATION

I declare that this thesis is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work, or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.

Signature: .... Date 23/08/2021

Ndung'u Charles Ndegwa
I56/12751/2018
Department of Physics
School of Physical Sciences
University of Nairobi
email:ndungundegwa85@gmail.com

This thesis has been submitted for examination with our approval as research supervisors:

Signature Date

Dr. M. I. Kaniu 23/08/2021
Department of Physics
University of Nairobi
P.O Box 30197-00100
Nairobi, Kenya
ikaniu@uonbi.ac.ke

Dr. J. M. Wanjohi 23/08/2021
Department of Chemistry
University of Nairobi
P.O Box 30197-00100
Nairobi, Kenya
jwanjohi@uonbi.ac.ke

DEDICATION

Whether you can observe a thing or not depends on the theory which you use.
It is the theory which decides what can be observed.

– Albert Einstein

Dedicated to my modest parents, Mr and Mrs Ndegwa and my loving and caring siblings

ACKNOWLEDGMENTS

First and foremost, I wish to sincerely thank my supervisors, Dr. M. I. Kaniu and Dr. J. M. Wanjohi. Your honest, uncensored insights on academic research, spectroscopy, data analytics, machine learning, and writing skills were always a pleasure to receive. They will remain an invaluable aid in whatever future I choose to pursue. I am thankful for the opportunities I had to share everything from scientific results to burdens of conscience. I would not have chosen to endure the rigors of graduate studies if it were not for you. Thank you all for your patient instruction and guidance during my time at the University of Nairobi (UON). Special thanks go to Dr. M. I. Kaniu to generously provide the fruit samples and allow access to his farm for field measurement without any attached cost. I am forever indebted to you.

My heartfelt appreciation to Mr. Dickson Omucheni, the Chief Technologist, Department of Physics, UON, for guiding me on reflectance spectroscopy and being very resourceful. I would be negligent if I did not acknowledge the UON for the MSc. Scholarship through the payment of fees and monthly stipend without which I would not have been able to undertake this research. I wish to also thank Prof. Kenneth Kaduki, also from the Department of Physics, UON, and the International Science Programme, for funding the purchase of the NIR quest 512 spectrometer, which was crucial for this research.

I also extend my heartfelt gratitude to my family, whose encouragement, financial support, and words of wisdom and guidance were arguably the most important contributor to my success in graduate studies. Of course, no mention of family would be complete without my colleagues, whom I had the privilege of working with daily. I am happy that the laser lab brought us all closer than any ordinary coworkers. To Kennedy Odongo and Ikedi Robert, I have undoubtedly learned the most from all of you. Thank you for sharing in that truly sustaining, collaborative, friendly atmosphere that this group thrived on. I challenge you to keep our truly remarkable friendship alive as we embark on another chapter of our lives.

ABSTRACT

There is an urgent need for accurate, non-destructive, rapid, and affordable techniques for screening pesticide residues in fresh fruits to assess whether regulatory standards are met by not surpassing the allowed maximum residue limits (MRL). Although conventional methods such as chromatography and mass spectroscopy are accurate, they are expensive, destructive, and require tedious wet lab sample preparations. This study aimed to assess the utility of machine learning techniques for rapid and non-destructive assessment of residues in fruits based on diffuse reflectance spectroscopy (DRS) measurements in the near-infrared region. Towards this goal, tree tomatoes fruits were spiked with Mancozeb and thiocyclam hydrogen oxalate (THO) in varying concentrations, and near-infrared spectra data (900-2500nm) were collected in DRS geometry. Another dataset was collected from the field in the 200 nm to 1050 nm range on control and treated tree tomatoes for 11 consecutive days for qualitative analysis. All the measurements were transformed into absorbance using $\log_{10}(1/R)$ before preprocessing using the smoothing, normalization, and multiplicative scatter correction techniques. Principal component analysis of the field data showed distinct clusters for the control and treated fruits in the 800 nm to 850 nm third overtone region with PC1 and PC2 accounting for 82% and 5.9% of the variance, respectively. The combination region (1900-2500 nm) was optimal in discriminating the samples with varying pesticide concentrations for the laboratory data, with PC1 and PC2 accounting for 93% and 3.8% of the variance, respectively. The first four PCs, which explained 98% of the cumulative variation of the data, were extracted from the laboratory data and used as inputs to the support vector machine (SVM), artificial neural networks (ANN), and Random forest (RF) machine learning models. All the developed models had R^2 values greater than 92% and RMSEP values of less than 0.06 ppm for Mancozeb models and not more than 0.08 ppm for THO models. Limits of detection and quantification were also determined using a pseudo-univariate approach. The models were tested on a new dataset of samples collected from four local markets. From the results obtained, all predicted values were below the acceptable MRL values (0.5 ppm and 0.3 ppm for Mancozeb and THO, respectively). One-way Tukey ANOVA analysis of the predictions of the market samples showed that ANN and SVR models were more reliable than the RF model. Therefore, it was concluded that the combination of diffuse reflectance spectroscopy with machine learning techniques has potential for rapid, non-destructive, *in-situ* assessment of pesticide residues in fruits and vegetables.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
BHC	Benzene Hexachloride
DDT	Dichloro-Diphenyl-Trichloroethane
DESIR	Dry Extract System Near-Infrared
DRS	Diffuse Reflectance Spectroscopy
EMS	Electromagnetic Spectrum
FOV	Field of View
FT-MIR-DRS	Fourier Transform Mid-Infrared DRS
GC-MS	Gas Chromatography Mass Spectrometry
HPLC	High performance liquid chromatography
KEPHIS	Kenya Plant Health Inspectorate Service
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LOD	Limit of Detection
LOQ	Limit of Quantification
MAD	Mean Absolute Deviation
MIR	Mid Infrared
ML	Machine Learning
MRL	Maximum Residue Limit

MSC	Multiplicative Scatter Correction
MTR	Multi-Target Regression
NFLT	No Free Lunch Theorem
NIR	Near Infrared
PC	Principal Components
PCA	Principal Component Analysis
PLSR	Partial Least Squares Regression
ppb	Parts per Billion
ppm	Parts per Million
RF	Random Forest
RMSE	Root Mean Square Error
RMSEC	Root Mean Square Error of Calibration
RMSEP	Root Mean Square Error of Prediction
ROI	Region of Interest
SD	Standard Deviation
SG	Savitsky Golay
SNR	Signal to Noise Ratio
SNV	Standard Normal Variate
SPA	Successive Projection Algorithm
STM	Single-Target Model
SVM	Support Vector Machines
SVR	Support Vector Regression

THO	Thiocyclam hydrogen oxalate
UV	Ultra Violent
VIS	Visible

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Organic synthetic pesticides (i.e., insecticides, herbicides, fungicides, and nematicides) are widely used in modern agriculture to protect crops from diseases, weeds, and pests such as insects and rodents (Chen *et al.*, 2011). Kenya is heavily reliant on agriculture, accounting for approximately 24 percent of its GDP, and employs an estimated 75 percent of the population. Food security is a top priority in Kenya's Vision 2030 and the agriculture revitalization policy with the government designating it as one of its four deliverables (GOK, 2018). More food production is required due to her growing population which translates into greater use of chemicals in the production of food. In 2018, Kenya imported 17,803 tonnes of pesticides worth 128 million dollars. Insecticides, fungicides, and herbicides accounted for approximately 87 percent in volume and 88 percent of total import costs. As a result, Kenya's pesticides market is relatively big and growing steadily. This has resulted in the environmental pollution in bodies of water, foodstuffs, and soil (Salami *et al.*, 2017).

Regardless of their value in increasing crop yields, the widespread use of pesticides during the production, processing, storage, transportation, or marketing of agricultural commodities can result in increases in residues in foods, making it risky for consumers (Lv *et al.*, 2018). Food is only considered safe if there is a reasonable certainty that eating it will not cause harm (Oloo, 2010). Following public outrage and media attention in Kenya, there has been an increase in concern about chemical residues in food products, particularly fruits and vegetables. Chemical residue traces in food should be kept to a minimum otherwise they can result in the most serious food safety concerns (Lv *et al.*, 2018). These contaminants in foods have carcinogenic, mutagenic, teratogenic, allergic, and sometimes fatal health effects on consumers (Toldrá and Reig, 2006; Tsimbiri *et al.*, 2015; Wang *et al.*, 2009), necessitating regular residue monitoring (Kunyanga *et al.*, 2018).

Concerns about pesticide toxicity, both real and perceived, have pushed for rigorous regulation to safeguard customers leading to the establishment of Maximum Residue Limit (MRL) permitted in food materials. If acceptable agricultural practices are followed (Liu *et al.*, 2015), these residues should be below the MRL. According to many reports, these residues are found more frequently in fruits than in vegetables (Mebdoua, 2019). The high levels of residues in fruits and vegetables can be explained by repeated and high doses of chemical applications to combat the numerous and chronic pests and diseases due to the lack of rapid and early disease diagnosis methods.

Most common used control and residues detection measures use conventional analysis methods such as gas chromatography-mass spectrometry (Anastassiades *et al.*, 2003), high-performance liquid chromatography (Papadopoulou-Mourkidou and Patsias, 1996), enzyme inhibition method (Campanella *et al.*, 2005), enzyme-linked immunosorbent assay (Qian *et al.*, 2009), and electrochemistry (Hart *et al.*, 1997). These methods are time-consuming, costly, complicated, destructive, and require many samples to detect results despite being highly adopted. This compels the adoption of non-destructive, fast, and cheaper detection techniques such as Diffuse Reflectance Spectroscopy (DRS) for qualitative and quantitative analysis of chemical residues on fruits and vegetables.

1.1.1 Diffuse Reflectance Spectroscopy for Residues Assessment

Diffuse reflectance spectroscopy, also known as elastic scattering spectroscopy, deals with diffusely reflected photons whose angular distribution is unrelated to the incident angle. Light encounters scattering and absorption as it travels through tissue. The light that escapes the tissue becomes virtually isotropic after a series of scattering events and is thus classified as diffusely reflected light.

The path taken by diffusely reflected light as it propagates through tissue is determined by the optical characteristics of the tissue. As a result, a diffuse reflectance spectrum contains information about the optical characteristics of the tissue, which are intrinsically linked to the makeup of the tissue.

The fundamental tenet of DRS is that a sample is irradiated with a broadband wavelength light source ranging from UV to NIR. The scattered light is scanned across the sample as a function of wavelength using a spectrometer. The optical properties of the sample can then be derived from the reflected light from tissue using multivariate Machine Learning (ML) model-based techniques.

DRS has been used in the quantitative examination of residues on fruit and vegetables based on Mid Infrared (MIR) spectroscopy, which according to Beć *et al.* (2019) is less sensitive and requires sample preparations or Fourier transform approaches compared to Near Infrared (NIR) spectroscopy. Hiroaki *et al.* (2002) used Fourier Transform Mid-Infrared DRS (FT-MIR-DRS) in the 3571.43 nm to 12500 nm range to measure the concentration of residues on lettuce samples using partial least square regression. Makio *et al.* (2007) also used FT-MIR-DRS in the 5714.29 nm to 10526.32 nm range for detecting fungicide on strawberries and developed a classification model of the spectra using soft independent modeling of class analogy (SIMCA).

This study was centered on the NIR region of the spectrum as it is cheaper compared to MIR, requires little to no sample preparations, provides useful information on molecular structure, interactions, dynamics, and anharmonicity (Beć *et al.*, 2019). However, compared with MIR, NIR spectroscopy suffers from poor chemical specificity, weaker bands, and broad overtones. This is particularly significant in biophysics applications, where complex spectra of biological samples are analyzed. This complexity of NIR spectra is a considerable barrier to practical applications of NIR spectroscopy. Classical methods of data analysis in the NIR region have considerable limitations. Hence, the use of multivariate ML model-based techniques to overcome these challenges (Fernández *et al.*, 2016).

1.1.2 Machine Learning in Spectroscopy

Machine learning algorithms are computational tools used in predictive modeling of complex data such as the spectral data obtained in this work (Michie *et al.*, 1994). ML can overcome challenges associated with classical data analysis techniques. These methods use specific peaks that correspond to the spectrum of particular elements or molecules (Villmann *et al.*, 2008). This entails comparing how one spectrum varies from another using existing spectral libraries. This approach works best for univariate data analysis, using well-defined peaks. However, DRS spectra of fruits samples are broad and overlapping in nature, thereby limiting the use of these methods. The DRS of tree tomatoes treated with varying doses of pesticides formulation was investigated to establish the validity of autonomously identifying residues on vegetables based on the produced spectral data using Random Forest (RF) (Breiman *et al.*, 2017), Artificial Neural Networks (ANN) (Bishop *et al.*, 1995) and Support Vector Machines (SVM) (Boser *et al.*, 1992).

1.2 Statement of the Problem

The critical nature of the consumption of safe foods necessitates the development of rapid, non-destructive, cost-effective, and field-deployable screening techniques. Currently, used methods for detecting residues are laboratory-based, time-consuming, costly, and destructive. Diffuse reflectance spectroscopy can compensate for these shortcomings. However, the spectra are multivariate in nature and cannot be directly decoded for meaningful information. Conventional spectral data analysis techniques are univariate in nature and are therefore unsuitable for analysis of multivariate spectral data sets such as the ones obtained in this study. Additionally, instrumental noise, matrix effects, and multi-dimensionality all contribute to further complicating the analysis. This implies that the DRS technique is insufficient on its own. By utilizing data from the entire wave range with minimal pre-processing, machine learning techniques provide alternatives to the conventional approach. The machine learning assisted diffuse reflectance spectroscopy technique was developed as a substitute or complementary method for screening residues in fruits and vegetables with high accuracy. This novel method is field-deployable and has the potential to simultaneously measure several residues in fruits and vegetables without sample preparations.

1.3 Research Objectives

1.3.1 Main Objective

The goal of this work was to develop a machine learning-assisted diffuse reflectance spectroscopy technique for rapid and reliable pesticide residues detection in fruits and vegetables.

1.3.2 Specific Objectives

(i) To design and optimize a pesticide residues assessment method for rapid DRS measurements in fruits and vegetables.

(ii) To perform pre-processing and exploratory analysis of the DRS measurements obtained from specific objective (i) above using the multiplicative scatter correction, Savitzky-Golay filter, and PCA techniques for spectral noise and dimensionality reduction.

(iii) To develop and test calibration models for quantitative analysis of the data obtained from specific objective (ii) above using the ANN, SVR, and RF machine learning techniques.

(iv) To evaluate the viability of the developed models from specific objective (iii) above using market samples.

1.4 Justification and Significance

Food safety is linked to chemical and physical risks that can result in both short and long-term health complications. Numerous studies have established that pesticide residues are carcinogenic, mutagenic, genotoxic, and have endocrine-disrupting properties. As public awareness of the importance of a healthy diet grows, the demand for rapid residue assessment systems increases. Not only can such systems identify foods above the MRL, but they can also provide information about the food's safety. Due to the inherent limitations of conventional methods for determining residues, rapid, sensitive, non-destructive, and field-deployable analytical techniques are required.

The DRS technique has been widely applied in molecular studies. However, the technique is susceptible to background noise, scattering effects, baseline drifts, and the chemical information hidden in the spectra cannot be decoded directly. Traditionally used data analysis techniques are univariate in nature and work by assigning specific peaks or comparing spectra to identify molecules of interest. Due to the highly correlated and redundant features in the spectra of biological samples such as fruits and vegetables, these approaches are inapplicable for analyzing complex multivariate data. In this context, ML techniques can handle multiple parameters and noisy data, allowing for rapid and accurate residues evaluation.

This work, therefore, aimed at analyzing fruits and vegetables with varying concentrations of pesticide residues using diffuse reflectance spectroscopy to develop a method for rapid, accurate, and reliable screening of residues. The developed method can be of great utility in analyzing sample *in-situ* since its portable, non-destructive, and no sample preparations are required for analysis.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

This section reviews the literature on conventional residue screening methods, the concerns of residues in foods, non-destructive techniques for determining residues in food, and the utility of machine learning in these studies.

2.2 Pesticide Residues in fruit and vegetable samples

There is an increasing concern about chemical residues in food products, particularly fruit and vegetables. A study conducted by Kunyanga *et al.* (2018) sheds light on the prevalence and chemical levels of certain fruits and vegetables commonly consumed in Kenya. Various vegetable samples were tested for residues using standard methods. For each commodity and pesticide, the results were compared to national and global standards for MRL. The study revealed pesticide contamination of fruit and vegetable samples, some of which were above the prescribed limit. Kenya Plant Health Inspectorate Service (KEPHIS) reported residues in vegetable samples across the country in its 2018 annual report. The most severely affected crops were kale (94%), peas (76%), and capsicum (59%). 10% of the samples exceeded the maximum allowable residue levels (Kabano, 2018).

Data on pesticide use or quantity in water, soil, and food, as well as the associated consequences, are not available in Kenya (Abong'o *et al.*, 2018). In Kenya, there are no regular surveillance and safety reports on the levels of these chemical residues. KEPHIS inspects food samples for residue traces but, the specific amounts of residues are undisclosed to the public. This study can close the gap by developing a less expensive screening method and making the data available to the public.

The presence of residues in food is a global issue that is addressed by numerous studies. In Poland, for example, 144 samples of fruits and vegetables were tested for 60 pesticides. Residues of 15 pesticides were found in 32% of the samples. The percentage of residue in samples that exceeded the MRL was 15%, with multi-residues accounting for 9% (Kaczyński *et al.*, 2013). Another study in the UK found residues in 79 percent of fruit and vegetable samples; the most common chemical traces were fungicides (Mebdoua, 2019). Pesticide residues were detected in 160 samples of 13 different types of fresh fruits and vegetables from both domestic and imported sources in Algeria using Gas Chromatography Mass Spectrometry (GC-MS) (Mebdoua *et al.*, 2017).

The proportion of fruit with residues was higher than vegetables in most studies conducted by various countries. Brazilian testing systems discovered that residues were present in 59% of fruit samples but only in 36% of vegetable samples. Similar findings are documented in surveillance programs in the United States and the European Union (Mebdoua, 2018).

2.3 Analytical Methods for Residue Assessment in Fruits and Vegetables

Assessment of pesticide residue levels needs precise analytical methods. The residues, on the other hand, are generally found at low concentrations (Parts per Million (ppm) and Parts per Billion (ppb)), are numerous, and have different chemical structures. The study and quantification of these residues is a scientific challenge that necessitates the use of sensitive and accurate multi-residue analysis methods (LeDoux, 2011). Several analytical methods are available whose choice is determined by the pesticide under investigation or the matrices being tested. The following sections provide a brief overview of some of the most commonly used techniques.

2.3.1 Conventional Techniques of Residue Detection in Fruits and Vegetables

The earliest analytical techniques were established in the 1960s, utilizing an initial acetone extraction followed by a partitioning phase involving non-polar solvents and salts. Sophisticated and solvent-intensive cleaning techniques were required. There was also a lack of selectivity and sensitivity in the equipment used to evaluate the target chemicals. Pressurized liquid and supercritical fluid extraction and were developed in the 1990s as a result of technological advances and a desire to eliminate user interference and speed up sample preparation procedures. Though initially very promising, these techniques have failed in the field of pesticide analysis for a variety of reasons, including the instruments' high cost and unreliability, and the inability to extract different pesticide residues from foods with equal efficiency, frequently necessitating separate optimization for different analytes (Cunha *et al.*, 2011).

Lehotay and coworkers simplified conventional sample preparation procedures by introducing QuEChERS (rapid, easy, cheap, effective, rugged, and safe) (Anastassiades *et al.*, 2003). A simple extraction or partition utilizing acetonitrile and salts, as well as a basic dispersive cleaning procedure, are used in the approach (Cunha *et al.*, 2012). Unfortunately, GC-MS analysis of QuEChERS extracts in acetonitrile is not without complications. Numerous phenomena may occur, including column deterioration by the polar solvent, vapor overload due to high thermal expansion coefficient, system contamination by co-extractives (Rashid *et al.*, 2010), and decreased enrichment factors.

GC-MS and High performance liquid chromatography (HPLC) are the gold standard and the most widely used analytical tools (Stachniuk and Fornal, 2016; Tsagkaris *et al.*, 2019). Food samples provide an immense challenge to analytical wet chemistry methods for residues at trace levels. The diverse range of food matrices, from liquids to solids, necessitates the use of varying sample preparation procedures to get reliable and repeatable findings. Chromatographic methods can enable such analysis at trace levels to meet the MRL required by food safety standards. However, reproducible and reliable results depend on the sample preparation procedures used.

Food sample preparations for GC and HPLC analysis involve homogenization, extraction using solvents, cleanup, extraction of analytes, and concentration and reconstitution of the eluent in a suitable solvent. Sample preparation is the limiting step in achieving acceptable performance parameters accounting for roughly 60% to 70% of the overall analysis time. The use of several sample preparation procedures to account for the variety of pesticides targeted and the type of food matrices are often needed. These procedures must be capable of producing analytically precise findings while being economically viable for routine analysis. Additionally, they must be simplistic and safe.

Conventional pesticide detection techniques are the gold standard and the most widely used analytical tools. Although they produce highly accurate results, they can be time-consuming. For example, Khandekar *et al.* (1982) used HPLC to measure 313 specimens of 14 different vegetables from 5 different markets in India for five residues of organochlorines over three years. They discovered residues individually or in various combinations. Vegetable residues were higher in leafy vegetables than in other types of vegetables. The most common pollutants were Dichloro-Diphenyl-Trichloroethane (DDT), lindane, and Benzene Hexachloride (BHC), with aldrin and endrin being less common.

Mattern *et al.* (1990) detected and quantified 20 pesticide residues in fruits and vegetables using HPLC and mass spectrometry. The sample preparation procedures were time-consuming and involved the use of more than 13 hazardous chemicals and reagents. Stafford and Lin (1992) determined oxamyl and methomyl insecticides in fruits and river water using HPLC. They measured recovery levels ranging from 20 to 1000 ng/mg for raw agricultural commodities and 5 to 50 ng/ml for river water. The method used numerous reagents and wet digestion of the fruits before the time-consuming detection process. Nakamura *et al.* (1994) minced and blend vegetables and fruits samples before the addition of reagents to detect organophosphorus, organochlorines, organo-nitrogen, and pyrethroid pesticides.

GC-MS has also been used extensively in fruits and vegetable residues studies. Podhorniak *et al.* (2001) found organophosphorus pesticide and their metabolites in fruit and vegetable. Yoshii *et al.* (2001) also identified residues of pesticide in tea, broccoli, cod, tomato, eggplant, cucumber, and Japanese radish using liquid chromatography. In 150 orange fruit samples, benzoyl phenylurea insecticide residues, carboxamides, acaricides, and carbamate insecticides were detected and reported in 74.6% of the samples. The residues of the pesticide were below the detection limits of 0.002 to 0.05 mg/kg (Valenzuela *et al.*, 2001). Blasco *et al.* (2002) also reported that 54% of the samples contained 0.005 to 3.34 mg/kg of o-phenyl phenol. The residues exceeded the permissible limit in only 4% of the samples. Michel and Buszewski (2002) determined ten different residues in fruits, vegetables, and cereals. The residue limits varied between 0.02 and 0.25 ng/g. In apples, green beans, and carrots, Lehotay (2002) detected and quantified 89 pesticides in fortified spinach, tomatoes, apples, and strawberries.

Gupta (2004) reported that 15% vegetables in India were contaminated with pesticide residues. Of these, 20% exceeded the recommended MRL. Randhawa *et al.* (2007) detected chlorpyrifos and TCPy traces in vegetables. The maximum quantity of chlorpyrifos residue was established in Spinach at the bare stage (1.87 mg/kg^{-1}), followed by okra (1.41 mg/kg^{-1}) and eggplant (1.25 mg/kg^{-1}). The residues were reduced in samples to 15% from 33% after washing, to 65% from 85% due to post peeling, and to 12% from 48% after Cooking.

González *et al.* (2008) reported 23 residues in 75 leafy vegetables. The highest fungicide concentrations were reported in lettuce (procymidone 12 mg/kg) and the lowest traces in Swisschards (cypermethrin 6 mg/kg). The group also reported that residue levels were higher in lettuce than in the other leafy greens. Hernández-Borges *et al.* (2009) quantified 11 pesticides collected from 57 banana samples from Spain's local markets. Although the amounts of residues were below the MRL, the study revealed that bananas peels and pulp had the highest residues levels.

In 2017, 160 samples of fruit and vegetables were analyzed by Hadian *et al.* (2019) using GC-MS who found that 12% of the samples tested were above recommended MLRs. Moreover, 57.5% of the samples contained at least one residue. In a similar study, Islam *et al.* (2019) determined four major organophosphorus residues in cabbage samples collected from 5 markets in Senegal using GC-MS coupled with flame thermionic detection. They reported that 12% of the samples contained residue above recommended MLRs.

Traditional methods, such as GC-MS and HPLC, produce consistent results. However, they have numerous limitations, including their destructive nature, time-consuming sample preparations, the need to transport and store samples, time-consuming analysis, the need for skilled chemists and advanced laboratories, the use of hazardous, expensive reagents, the lack of real-time detection capabilities, and the technique's non-portability (Hu *et al.*, 2020). non-portability leads to collection, transportation, and storage of samples leading to increased costs Roy *et al.* (1997).

Alternative methods with sufficient detectability, cost-effectiveness, simplicity, non-invasiveness, and portability are thus required. Optical spectroscopy-based methods are safe, non-destructive, quick, reliable, and low-cost alternatives for assessing residues and eliminating operator exposure to harmful compounds.

2.3.2 Non-Destructive Techniques in Residues Monitoring

In research involving pesticide and their derivatives, vibrational spectroscopy is a frequently used tool. It combines high sensitivity, selectivity, structural specificity, and non-destructive sample probing into a single instrument. Not only can molecules be studied in the gas phase, solution, solid-state, or matrices, but they can also be studied in biological samples (Stuart, 2006). This study can shed light on the pesticides' intermolecular interactions, base pairing, and tautomerization.

Optical imaging and spectroscopy involve analysis of spectra or images acquired from samples (dubbed as direct or forward approach) in the reflection or transmittance mode under a particular geometry, such as diffuse, specular, or both. The direct approach is simpler, faster, and easier to execute than conventional methods for online or offline applications.

Conventional NIR spectroscopy uses a direct/forward approach to determine the total amount of light reflected from or transmitted through a sample as a result of photon absorption and scattering by the tissues. Acquired spectral data are processed mathematically to build quantitative or qualitative predictive models. Additionally, chemometric approaches progressed from linear modeling to non-linear modeling. As a result of these advancements, NIR spectroscopy has expanded into a variety of applications, such as food, pharmaceutical, and chemical (Nicola et al., 2007).

Meurens introduced the Dry Extract System Near-Infrared (DESIR) in

1987 (Mattern *et al.*, 1990). In this method, a liquid containing the residue is applied to a low-absorptive substrate, followed by drying to allow analyte concentration and solvent removal. Saranwong and Kawano (2005) was the first to use DESIR to detect residues on fruits. They rinsed 95 tomato samples with acetone before drying the solution on glass fiber filters which were subjected to reflectance measurements. The $RMSECV$ for pure solutions was 6.6 ppm, 7.9 ppm for tomato wash, and 1.6 ppm for Limit of Detection (LOD).

Umesh *et al.* (2012) also implemented DESIR in NIR reflectance spectroscopy in the detection of a contact pesticides residue on various fruits. Partial Least Squares Regression (PLSR) was used for quantification models with an RMSECV of between 0.003 - 0.06 mg. However, they reported that the model performance was poor across different fruits. Chen *et al.* (2011) explains the creation of a DESIR-based PLSR model using pure pesticides solutions and FT-MIR-DRS spectrum acquisition. Pre-processing was done using Multiplicative Scatter Correction (MSC) and first-order derivative of absorbance. The group utilized SVM for a classification model with over-fitting due to the low number of samples in the study. Also, they did not consider issues of matrix and model robustness. DESIR is only relevant for contacts pesticide, which remain on the surface of the samples. Detection of residues scattered throughout the tissue may require sample digestion.

DESIR requires washing samples in acetone/water, and the wash solvent loaded onto a glass filter and allowed to dry. As a result, the technique is unsuitable for *in-situ* measurements and is time-consuming due to sample preparation.

Hyper-spectral imaging is another non-invasive, non-polluting method that requires little or no sample preparation. The novelty of this method is that it obtains both spectral and spatial information from the samples under investigation at the same time. Several authors, as discussed below, have reviewed the versatility of this method in residue detection and analysis. By detecting internal changes in *chlorella pyrenoidosa*, Shao *et al.* (2016) demonstrated the feasibility of using hyperspectral VIS/NIR imaging to detect a variety of pesticides.

Sun *et al.* (2015) detected residues on mulberry using the Adaboost-SVM algorithm in conjunction with hyper-spectral imaging. Furthermore, Sun *et al.* (2015) employed hyper-spectral imaging in conjunction with chlorophyll fluorescence spectra of 150 different lettuce leaf samples containing five different concentrations of dimethoate. The spectral data were pre-processed with the Savitsky Golay (SG) and Standard Normal Variate (SNV) algorithms. Principal Component Analysis (PCA), Successive Projection Algorithm (SPA), and wavelet transforms were used in conjunction with the Mahalanobis distance multi-modeling of the Monte Carlo cross-validation algorithm to identify optimal wavelengths from raw spectra. SVM was used to create a predictive model based on selected wavelengths, with the best model predicting $\boldsymbol{R}2 = 0.987$ and $\boldsymbol{RMSEP} = 0.005$.

Zhan-qi *et al.* (2018) used hyper-spectral imaging and ML algorithms to determine various dimethoate concentrations on spinach leaves. MSC was used for spectral pre-processing. For clustering samples based on concentrations, PCA was used. The selection of chi-square test features combined with SVM, K-Nearest Neighbors (KNN), RF and Linear Discriminant Analysis (LDA) was used for classification. The prediction accuracy was evaluated using ten-fold cross-validation, average, and Standard Deviation (SD). The Chi-square test combined with LDA produced the best results, with an accuracy of 0.997 and a SD of 0.008. Hyperspectral imaging provides both spatial and spectral data. The rich information also causes data processing difficulties, making *in-situ*, field deployable, or online measurement applications challenging.

Among non-destructive analysis techniques, NIR spectroscopy is preferred because of its non-invasive nature, low cost, and high sensitivity (Armenta *et al.*, 2007). Visible and near-infrared spectroscopy in various acquisition modes has proven to be an effective tool for checking and regulating product quality and safety in the food industry (Cortés *et al.*, 2019) in recent years. These methods are faster than traditional detection methods, require little to no sample preparation, and are usable for *in-situ* measurements.

Data can be collected in reflectance mode, transmittance mode, or absorbance mode to provide data linked to C−H, O−H and N−H molecular bonds (Türker-Kaya and Huck, 2017). Table 2.1 displays some studies that used various spectroscopic methods to assess pesticides. All of these studies have demonstrated the possibility of determining residues using the NIR, Visible (VIS) and Ultra Violent (UV) wavelength region of the Electromagnetic Spectrum (EMS).

Table 2.1: Studies using VIS/NIR spectroscopy for pesticide assessment

Method	**Determination attribute**	**Reference**
NIR	Pesticide residues	Shen *et al.* (2009)
NIR	Classification of pesticide in Agriculture products	Makio *et al.* (2007)
NIR	Detecting of Chlorpyrifos content in spinach	LIU *et al.* (2008)
Mid and NIR	Metribuzin in pesticide	Khanmohammadi *et al.* (2008)
NIR	Pesticide determination in commercial formulation	Armenta *et al.* (2007)
NIR	Determination of soil content in chlordecore	Brunet *et al.* (2009)
NIR	Detecting of active ingredients in pesticide	XIONG *et al.* (2010)
Fourier transform IR	Propamocarb in emulsifiable pesticide	Quintás *et al.* (2008)
NIR	Pesticide residues in Peppers	Sánchez *et al.* (2010)
VIS/NIR	Pesticide residues on agricultural produces	Jamshidi *et al.* (2016)
Hyperspectral	Pesticide residue on lettuce	Sun *et al.* (2018)
NIR	Pesticide concentration on fruits	Acharya *et al.* (2012)
NIR	Chemical residues in food	Teye *et al.* (2013)

2.4 Challenges of Diffuse Reflectance Spectral Measurements

Spectral measurements must be precise and accurate representations of samples properties. However, a variety of factors influence measurement performance. They include optical diffusion as well as environmental or experimental problems that may affect the quality of spectral tests (Schaepman *et al.*, 2015). The challenges related to on-site spectral measurements include; atmosphere properties (e.g. wind direction and speed, cloud cover and form, humidity, aerosols, temperature), measuring space, measurement height, measurement orientation, the Field of View (FOV) and calibrations (Shaw and Burke, 2003).

These challenges were taken into account because they affected the precision of spectral measurements. The experimental design minimized inaccurate results due to poor geometry of illumination and transition conditions, the timing of data collection (integration time), and calibration procedures to reduce spectral response variance, such as using a reference standard (Spectralon) (Shepherd and Walsh, 2002).

Diffuse reflectance spectra are prone to background noise, scattering effects, and baseline drifts even after careful experiment design. Molecular overtones and combination bands are expansive in the NIR range for diffuse reflectance measurements, making it arduous to do molecular attributions. Conventional data analysis techniques in spectroscopy are univariate and hence are not satisfactory for multivariate DRS data. ML models were used to solve the limitations of classical data analysis approaches.

2.5 Utility of PCA Scores as Inputs to ML Models

The development of spectroscopic techniques in food safety have necessitated the use of chemometrics and ML techniques in the analysis to help deal with the large amount of data generated in the measurement processes. When the input features are non-linear, more than the predictors, or large, ML techniques are better suited for analysis than statistical methods such as logistic regression and discriminant analysis (Rodriguez-Galiano *et al.*, 2015). The vast number of complex non-linear correlations between features can be identified, modeled, and handled using ML. However, more input features in most cases result in complex models and more model parameters thus requiring more computational resources (Rodriguez-Galiano *et al.*, 2015). Additionally, a significant number of spectral features are duplicated and highly correlated, lowering prediction accuracy. This challenge necessitates the employment of feature selection or dimensionality reduction techniques such as PCA (Wold *et al.*, 1987), which allow for the elimination of superfluous data. However, even after feature selection, more often than not, the resultant data is not orthogonal, implying that the training period will be lengthy, an issue can be resolved using PCA.

PCA allows for data reduction and orthogonalization by transforming features into a new set of uncorrelated features known as Principal Components (PC)s. This is advantageous when handling multidimensional data such as highly redundant and correlated spectral data (Priyadarshini *et al.*, 2019). Direct modeling of such data can be challenging because of over-fitting and high computational resources requirements (Howley *et al.*, 2005).

Using PCs scores as input to the model rather than the original variables have been found to reduce the number of features and accelerate the training period. Kachrimanis *et al.* (2010) reduced the dimensionality of the input space with PCA before modeling a feed-forward back-propagation ANN for quantitative analysis of powder mixtures. He *et al.* (2007) classified various tea samples by integrating wavelet extraction for feature extraction and PCA for visualizing the extracted features. The first eight PCs were used as the inputs to a back-propagation ANN with two hundred samples of eight tea varieties to build the model. The built model was then used to predict 40 new samples with an accuracy of 100%. Additionally, Li and He (2006) used the first nine PCs (85.3% of the data variance) to classify three peach assortments using a back-propagation ANN. The model was used to predict 15 unknown samples with 100% accuracy.

Sigurdsson *et al.* (2004) used ANN for skin cancer diagnosis using Raman data reduced using singular value decomposition (SVD) PCA, which simultaneously extracts the PCs Wu *et al.* (1997), but they did not provide any comparisons of models using raw data. Additionally, Zhou *et al.* (2015) developed three ANNs models for predicting Chlorophyll-a concentrations using three different datasets. Model built on PCs data were found to have the highest accuracy ($R^2 = 0.918$ and RMSE $= 5.88$) when compared to models built on raw data and raw data combined with water quality parameters. Wu and Massart (1996) used back-error propagation ANN to classify nine NIR spectra data sets of drugs to explore the outcome of various data pretreatment methods on the ANN input selection. They selected univariate features (Fisher transformation) and then used PCA to orthogonalize the selected features. This approach reduced the structural complexity of the ANN models. The above-cited studies were limited to the ANN model for prediction. However, the ”no free lunch theorem” (Wolpert and Macready, 1997) connotes that there is no single best model. The performance of a model is determined by the problem at hand (Caruana and Niculescu-Mizil, 2006). Thus, it makes sense to use a variety of models to seek one that matches the information better (Ho and Pepyne, 2002).

Few studies have been undertaken on the effectiveness of using PCA scores as inputs to various machine learning algorithms. The impact of eigenvector decomposition (EVD) PCA, which extracts the PCs simultaneously (Wu *et al.*, 1997), on three ML methods was investigated by Popelínskỳ (2000). In this study, the PCs were combined with the raw data, and the error rate was reduced but the used datasets used were not high dimensional. Howley *et al.* (2005) used non-iterative partial least squares (NIPALS) PCA, which calculates PCs factors sequentially (Wu *et al.*, 1997), to enhance the classification accuracy of several ML methods that utilized Raman spectra. They reported improved models performance and low error rates. Li *et al.* (2016) investigated the spectral characteristics of blood serum from four groups using surface-enhanced Raman spectroscopy. The dimensionality of the spectral data was reduced using PCA, and the PCA scores were used as inputs to develop SVM, linear discriminant analysis, classification, and regression tree with accuracies of 96.5%, 88.8%, and 87.1% respectively.

2.6 Application of Machine Learning in Spectroscopy

For quantitative spectroscopic methods, a variety of analytical techniques have been developed. Despite numerous benefits, there are some drawbacks. These include broad, highly overlapping bands that are typical of multi-component mixtures such as biological samples. Such issues present unique challenges for exploratory research and the study of unknown structures. In this case, chemometrics and closely related methods such as ML may provide a solution. ML can help discover new possibilities in data-intensive fields like spectroscopy, where it can identify patterns or trends in data that the traditional data analysis process cannot. ML enables the algorithm to learn from training data and validate on the test set without having to be explicitly programmed (Liakos *et al.*, 2018).

Román *et al.* (2011) predicted more than 70% of wine fermentation problems and reported that ANN could detect complex nonlinear patterns between inputs and outputs after training with known data. Cheng *et al.* (2010) used FT-MIR-DRS coupled with ML to classify Chinese medicinal products in cancer diagnosis using back-propagation ANN and SVM. Bai *et al.* (2011) also utilized Fourier transform spectroscopy and identified normal gastric tissues, early cancer, and advanced cancer using clustering of C-means and wavelets. Also, Ghosh *et al.* (2019) reported that deep neural networks could learn spectra to 97% accuracy and peak positions to an accuracy of 0.19 eV. The group also demonstrated that ANN could infer the spectra directly from the molecular structure and do not require auxiliary input. Extended-wavelength DRS was used by Dahlstrand *et al.* (2019) to differentiate and identify different skin and tissue types in pigs with a total accuracy of 98.2%. SVM was able to classify the skin type and tissue. Specificity and sensitivity for all skin and tissue forms ranged from 96.4% to 100.0%.

The importance of multivariate calibration methods in spectroscopy is undeniable based on the literature review. These studies have shown that machine learning can be used to classify and predict spectral data, which justifies its importance and applicability in this study. In conclusion, despite accurate and reliable results, conventional methods require time-consuming sample preparation and the use of chemical reagents to prepare standard solutions. According to Holmes *et al.* (2012), there is a high risk of sample swapping or mislabeling in the laboratory setting because once the sample has been processed for analysis with conventional techniques, it can be indistinguishable by eye. Confusion like this can result in the destruction of large amounts of samples or pesticide-polluted samples reaching the consumer.

According to Gritti *et al.* (2010) and Teye *et al.* (2013), conventional methods are time-consuming, expensive, and require competent staff and a specialized laboratory for product monitoring. These difficulties are the driving force behind this research. We used a simple, fast, and dependable method to determine residues in fruits and vegetables using spectroscopy. The analysis was based on the decomposition of NIR spectra acquired in diffuse reflectance geometry from treated and control samples using machine learning tools.

The vast majority of the reviewed literature uses PLSR modeling, which is best suited for linear data analysis. PLSR does not include feature selection, which is used to screen for the best subset of features or to optimize models. Due to the redundant latent variables inherent in a given dataset, such models are prone to over-fitting and may likely model the noise. This harms the predictive capacity of models with future data sets. This study made use of more robust models, such as ANN, SVM, and RF, which can easily model non-linear data.

When dealing with multidimensional data, the calculation of the regression coefficients in convectional statistical analysis methods has the advantage of partially reflecting the relationship between features. For data containing irrelevant and redundant features, feature selection is critical; in this case, the possibility of model optimization and improved regression accuracy is increased. As a result of the multi-collinearity of DRS data, as well as the issue of irrelevant and redundant information, PCA was used to remove obsolete and redundant features from the original dataset to overcome collinearity and aid in the development of more reliable machine learning models.

Many studies have found that using average spectra rather than the full spectrum standard results in more accurate calibration models. In conventional spectroscopy, a mean value representing a homogeneous sample and matched to a single sample spectrum is commonly used. This method is limited because it does not account for sample variation in the calibration model by including all spectra from a variety of test samples. In contrast to univariate approaches, this study presents the use of the entire wavelength from the Region of Interest (ROI) to create more robust prediction models. Furthermore, to account for sample variations, the entire data from the samples were used without averaging. Instead of averaging the redundant information, PCA was used to transform the original DRS data into a new axis, and the PC that explained more than 98 percent of the cumulative variation in the data was extracted and used as input in the development of the ANN, RF and SVM ML models. The novelty of DRS combined with ML is the method's direct, quick, low-cost sample preparation, and field deployability to assess or screen for pesticide residues.

CHAPTER THREE

THEORETICAL BACKGROUND

3.1 Overview

This chapter discusses the theoretical aspects of this work that are necessary for a thorough understanding of the subsequent chapters. The fundamental interactions between light and the plant cuticle are explained first, followed by the theory of the NIR region in molecular vibrations. Finally, data analysis is discussed within the context of multivariate machine learning models.

3.2 Propagation of Light Through Plant Cuticle

The cuticle is the outermost, continuous, heterogeneous, and spatially distributed composite membrane covering cells of leaves, fruits, petals, and non-lignified stems (Figure 3.1). The matrix comprises a long-chain and an insoluble polymer formed by hydroxylated and epoxy-hydroxyl groups called cutin. Other cuticle components are mixtures of homologous series of long-chain aliphatics, such as alkanes, alcohols, aldehydes, fatty acids, esters, and cyclic compounds (Heredia-Guerrero *et al.*, 2014).

The plant cuticle is a dynamic system that interacts with light and pesticides particles and, hence, its properties change. The spectral variability caused by these changes can be characterized using DRS analysis and form the backbone of this study. The measurement of the treated fruit cuticle's chemical information for this study is based on diffuse reflectance. When this cuticle light interaction occurs, 4% of the incident light is scattered by the cuticle as reflected light. The remaining 96% penetrates the cuticle into the cellular components where it is scattered or absorbed. While absorption is due to interactions with chemical components, scattering is due to sudden changes in the refractive index due to irregular cuticle surface or cell wall interfaces (Fernández *et al.*, 2016).

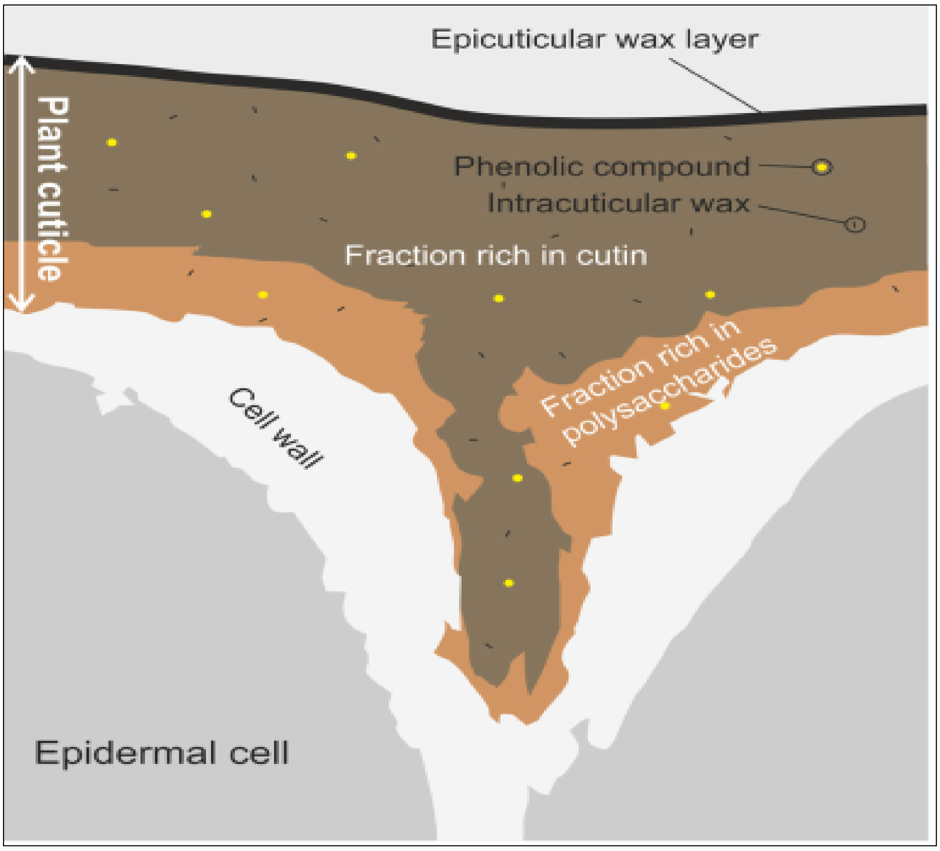


Figure 3.1: This illustration depicts the structure of the plant cuticle between two epidermal cells. The top layer is made of cutin and epicuticular waxes, while the layer below is mainly composed of polysaccharides from the cell wall. The phenolics and intra-cuticular waxes are spread evenly through the cuticle (source: Heredia-Guerrero *et al.* (2014)).

Due to their sophisticated structure, fruits and vegetable cuticles are optically dense, making it hard for light to penetrate. Hence, only a small distance is penetrated by incident radiation before it almost exits immediately near the entry point. However, some light can penetrate a bit (a few millimeters) into the tissue, where its wavelength is altered through the cuticle matrix's interactions. Some of this light manages to find its way and exits through the cuticle. This light contains useful chemical information and is known as diffuse reflectance (Abbott, 1999).

3.3 Diffuse Reflectance Spectroscopy

The radiation reflected from a fruit cuticle is a superposition of two components, Specular and diffuse reflection. Specular reflections, where the incident angle is equal to the reflected angle, occurs at the fruit surface, and hence Fresnel equations can be applied. For a simple case of perpendicular incident radiation, the specular reflectivity is described by equation 3.1.

$$S_f = (n-1)^2 \ + \ \frac{k_o^2}{(n+1)^2} + k_0^2 \tag{3.1}$$

where S_f is the specular reflectance, n is the refractive index of the cuticle and $k_o = n^2k^2$ is the absorption index defined through the Lambert law (equation 3.2).

$$I = I_o \exp \ [\frac{-4\pi k_o d}{\lambda}] = I_o \exp[-\theta d] \tag{3.2}$$

d is the distance traversed in the sample where I_o radiation is reduced to I, and $\theta = -4\pi k_o/\lambda$ is the absorption coefficient.

Diffuse reflection comes from the radiation that penetrates the cuticle and is scattered or absorbed by the cuticle matrix. A small portion of this radiation finds its way to the surface and emerges as diffuse reflection. This radiation contains variable chemical information because it interacts with sample particles and is attenuated along its path (Stenberg *et al.*, 2010). Equation 3.2 can describe the diffuse component where it is termed as the mean absorbance coefficient of the sample, and d is the cuticle thickness penetrated. Fruit samples reflect a composite of specular and diffuse reflectance in unknown proportions, and either can be used to determine and examine the absorbance properties of a medium.

From Equation 3.1, specular reflectivity is directly proportional to the absorption. However, as the radiation penetrates the sample, it gets exponentially attenuated (Equation 3.2). The increased absorbance of the sample lowers the scattering, consequently lowering the diffuse component spectral structure details. Since specular and diffuse reflection always superimposes, depending on the type of spectroscopic analysis, one component must be minimized. The diffuse component was the main focus of this investigation and the specular component was reduced by setting the collection optics at a 45°angle.

3.4 NIR Spectroscopy

NIR is a non-destructive technology that provides a responsive, reliable, and accurate scheme for evaluation of chemical properties of food samples (Ozaki *et al.*, 2006). The technique has matured since Friedrich Wilhelm Herschel discovered it in the year 1800 (Davies, 2000) and has been implemented in fields such as clinical, food, materials, and pharmaceuticals.

When photons with a frequency equal to molecular vibration frequency interact with a molecule, they absorb the radiation and get excited to a higher energy level causing a peak to appear on the NIR spectrum. The atoms may shift in various ways; moving closer or apart from each other, or in a bending motion, or wag symmetrical or anti-symmetrical in and out from the central atom (Zude, 2008). Since each vibrational mode corresponds to a bond within the molecule, its structure can be determined. An essential part of NIR spectroscopy is that a molecule must experience a dipole moment change. If this is not present, the molecule is not NIR active, and thus an NIR spectrum cannot be collected.

Spectrum vibrations are the product of transitions between quantified vibrational energy states. These motions may vary from a diatomic molecule's basic coupled motion to the complicated motion of a poly-functional molecule. Molecules of N atoms have $3N$ degrees of freedom, three of which reflect translational motion in mutually perpendicular directions, i.e., $x-, y-$ and $z-$ axes, and the other three reflect the rotational motion of $x-, y-$ and $z-$ axes. The remaining $3N - 6$ degree of freedom provides the number of potential vibrational modes. Each mode involves the harmonic displacement of atoms from their equilibrium points; for each mode, i, all atoms vibrate with a unique frequency, v_i. The potential energy, $V(r)$ of a harmonic oscillator as a function of the distance between atoms, r, is shown as the dashed line in Figure 3.2.

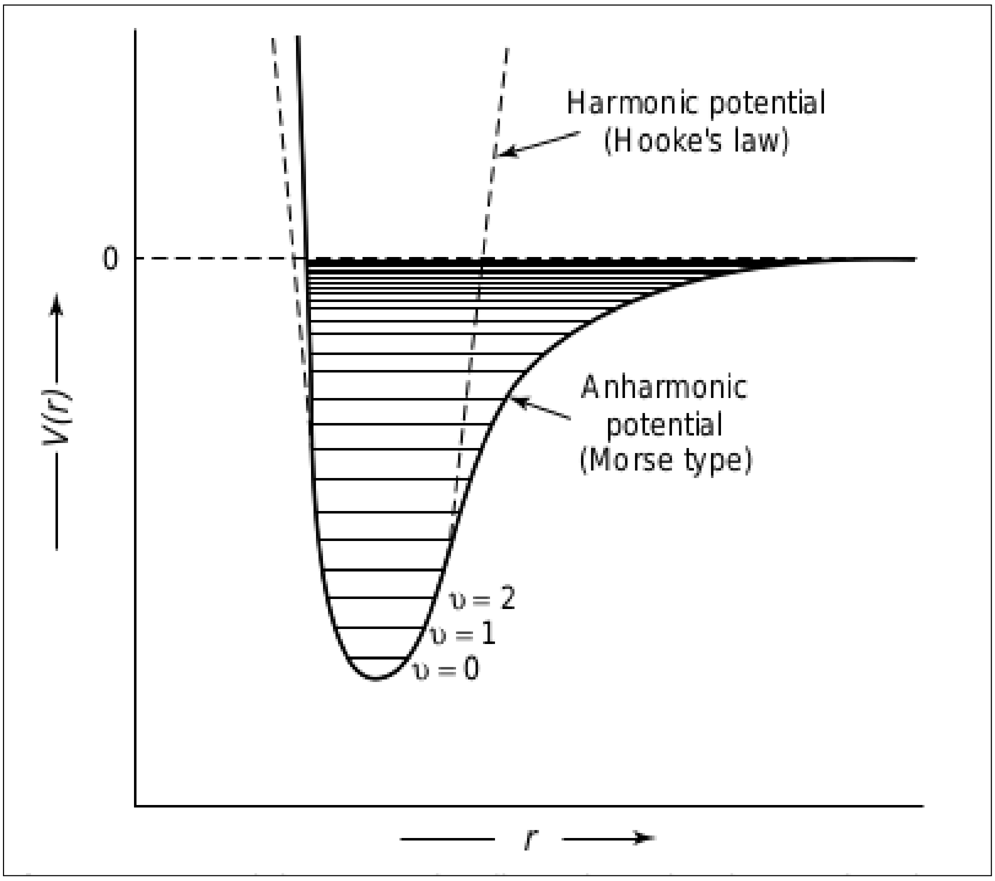


Figure 3.2: Potential energy of a diatomic molecule as a result of atomic displacement during vibration for an anharmonic oscillator (solid line) and anharmonic oscillator (dashed line) (source: Chalmers and Griffiths (2002)).

The motion of atoms is described using the normal coordinate (Φ). During a vibration provided that $\partial\mu/\partial\Phi \neq 0$, a molecule gets excited and changes its dipole moment u. For symmetrical molecules, the degeneracy of vibrations can occur, such that more than one mode has a certain vibrational frequency, while others may be completely forbidden. As a consequence of this degeneracy, the number of fundamental absorption bands that can be detected is sometimes less than $3N - 6$. Rotation of a linear molecule and its bond axis results in the loss of one degree of freedom since there is no displacement of atoms. Thus, a linear molecule has $3N - 5$ modes.

When atoms vibrate harmonically, i.e., in compliance with Hooke's law, they vibrate with the least amount of energy allowed by quantum mechanics. For a diatomic molecule, the vibrational frequency ν can be calculated (equation 3.3) (Afara, 2012), Where κ is the force constant, ν is the vibrational frequency, m_1 m_2 are vibrating atoms masses.

$$\nu = \frac{1}{2\pi}\sqrt{k\frac{(m_1 + m_2)}{(m_1 \times m_2)}} \tag{3.3}$$

According to Blanco and Villarroya (2002), the harmonic oscillator model can only describe molecular vibrations when there is the assumption that the different energy levels E_{vib} are equally spaced and satisfies equation 3.4.

$$E_{vib} = (\nu + \frac{1}{2})\ \frac{h}{2\pi}\ \sqrt{\frac{k}{\mu}} \tag{3.4}$$

ν is the vibrational quantum number, h is the Planck constant, κ is the force constant, and μ is the reduced mass of the bonding atoms. Only those transitions between consecutive energy levels ($\Delta\nu = \pm 1$) that cause a change in dipole moment are possible. Thus, E_{vib} can be rearranged as follow (equation 3.5): where ν is the fundamental vibrational frequency of the bond that yields an absorption band.

$$\Delta E_{vib} = \Delta E_{rad} = h\nu \tag{3.5}$$

Figure 3.2, the shift in the potential energy as a function of the atoms' displacement from their equilibrium positions is shown as a solid line. This curve means that equation 3.3 is valid only for low vibrational quantum values and is not correct for high ν_i values. The harmonic oscillator model accounts for the Coulombic forces, which are attractive at large distances but repulsive at short inter-atomic distances

(Blanco and Villarroya, 2002). Since most molecules do not have equally spaced energy levels, their behavior can be closely modeled by the anharmonic oscillator. In fact, a potential anharmonic (Morse-type) function must be used to characterize Viu (solid line in Figure 3.2). By approximating equation 3.6, potential energy can be calculated where the dimensionless anharmonicity constant is x_i.

$$V_{iu} = hv_i(u_i + \frac{1}{2}) + hv_i x_i(u_i + \frac{1}{2})^2 \tag{3.6}$$

No transformation of v_i by more than ±1 will be allowed if all vibration modes are purely harmonic. The anharmonicity is used to allow bands affected by $\mid \triangle u_i \mid > 1$ to be permitted.

Anharmonicity causes overtones (first and second overtones), which are integral multiples of the fundamental frequencies (Afara, 2012), to be generated between 780-2000 nm, and combination bands which occur between 1900-2500 nm (Blanco and Villarroya, 2002). These frequencies are unique to NIR and are far less probable than fundamental transitions. With hydrogen being the lightest atom, it exhibits the largest vibrations and deviations from harmonic behavior. The main bands typically observed in the NIR region result from weak, broad overtones and combination bands of fundamental vibrations associated with C−H, N−H, O−H, and, S−H functional groups (Tripathi and Mishra, 2009).

Only a few atoms are forcibly displaced in many vibrational modes, and the remaining molecules are almost stationary. The frequency of these modes is unique to a function group where the motion is centered, and the nature of the other atoms in the molecule is minimally affected. Therefore, spectral characteristics in a specific spectral area often indicate a particular functional chemical group in the molecule.

Concerning the principles discussed above, it is clear that residues on fruits can be determined using the chemical information-rich diffuse spectroscopy in the NIR region. Thus, absorption spectra with functional groups' information in fruits and vegetable samples can be obtained using a spectrometer. The NIR spectrum of biological materials contains information primarily from overtones and combinations associated with C−H, N−H, O−H functional groups. Figure 3.3 summarizes the major functional groups from the first, second, and third overtone regions and the combination band region in the NIR region. The position of these bands is also affected by the chemical environment, such as temperature.

The NIR spectrum has very few functional groups; however, its interpretation is not straightforward because the bands are broad and they overlap. Water prevalence (>80%) in raw vegetable samples such as fruit dominates the NIR absorption spectrum, affecting the evaluation of other components. ML methods are thus required to extract useful information from the spectrum.

3.5 Utility of Machine Learning in DRS Analysis

In spectroscopy, conventional data interpretation techniques rely on specific peaks that correspond to specific elements or molecules (Villmann *et al.*, 2008). This entails comparing the differences in spectra using existing spectral libraries. This approach is most effective with data that has well-defined peaks. However, the DRS spectra of fruits samples are broad and overlapping in nature, limiting the use of these univariate data analysis methods.

As previously noted, water present in the cellular matrix dominates fruit spectra, making the spectrum complicated due to tissue heterogeneity, instrument noise, scattering effects, temperature, and other sources of noise (Nicolaï *et al.*, 2008). This work obtained such data sets, which were then processed using machine learning. Exploratory and multivariate calibration of data obtained from treated and control fruit samples were performed using supervised and unsupervised ML algorithms. Before performing multivariate data analysis, various spectral pre-processing techniques were used to ensure a 'clean' dataset.

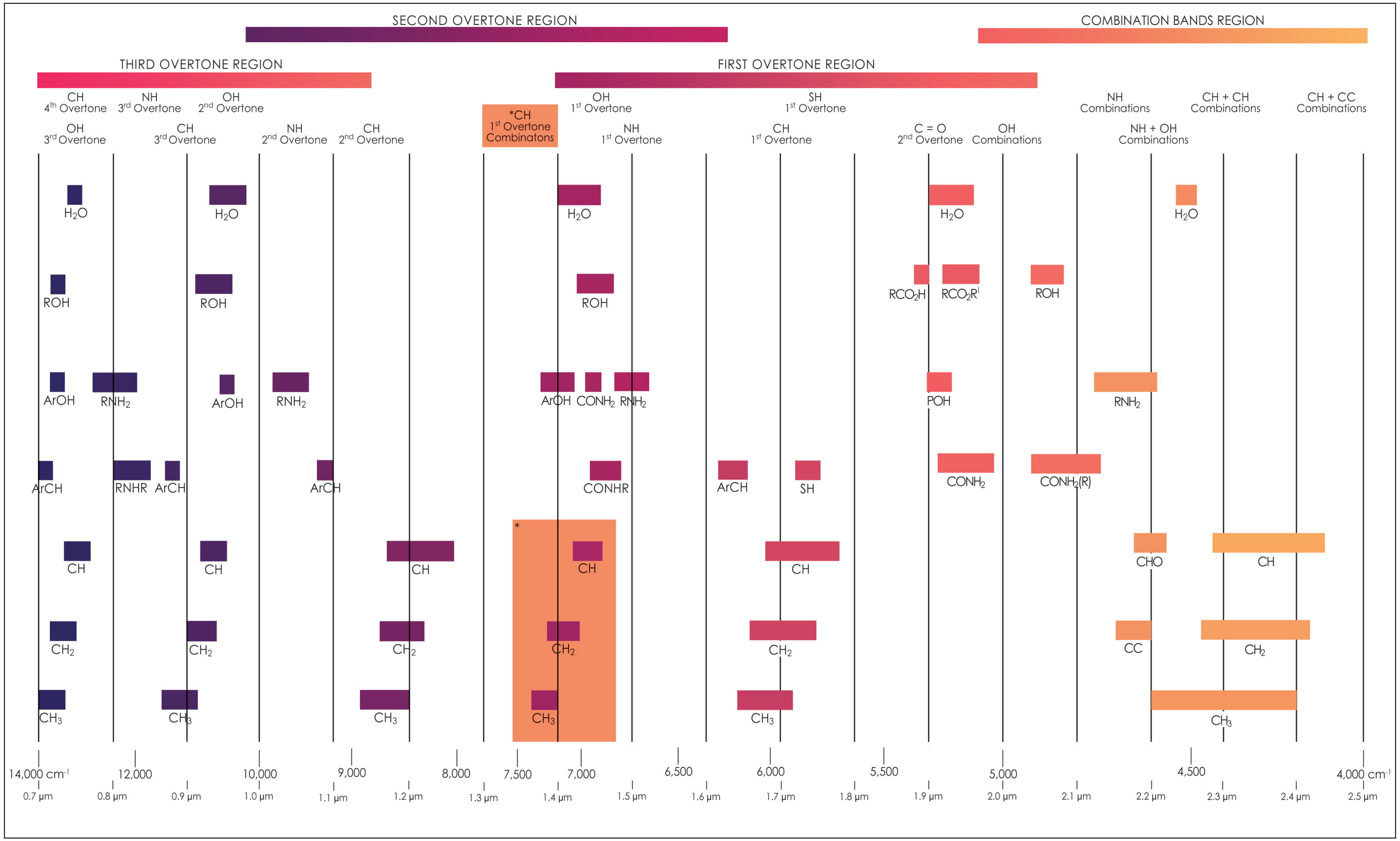


Figure 3.3: NIR spectra bands assignment chart (source: Metrohm (2021)).

3.6 Pre-processing Spectra Using the Multiplicative Scatter Correction

Pre-processing of DRS spectral data was discovered to be critical for producing a clean dataset. It is now widely accepted that quantitative and classification models that utilize pre-processed spectra always outperform models that utilize unprocessed spectra. Preprocessing data aims to eliminate superfluous spectral variations in order to model only the relevant features (Wu *et al.*, 2010). The following section discusses briefly the primary processing techniques used in this study.

For each sample, MSC corrects for light dispersion or path length change, calculated relative to the ideal sample. Theoretically, this measurement should be made on a portion of the spectrum that does not contain any chemical information, i.e., only influenced by light dispersion. However, spectrum areas that do not include any chemical details often have a spectral background where Signal to Noise Ratio (SNR) may be weak. Frequently, the spectrum is used in practice. This can be done as long as the same degree of dispersion exists as the ideal chemical difference between samples. For example, we can use the calibration set average as an approximation of the perfect sample. If the offset correction is done first, MSC will perform best (Sila *et al.*, 2016). For each of the samples:

$$\delta_i = y + a\bar{\delta_j} + \varepsilon \tag{3.7}$$

Where δ_i is the NIR spectrum of the sample, and $\bar{\delta_j}$ is the mean spectrum. For each sample a and y are estimated by ordinary least-squares regression of spectrum δ_i versus $\bar{\delta_j}$ spectrum over the available wavelengths. Each values δ_{ij} of the corrected spectrum δ_i (MSC) is calculated as

$$\delta_{ij}(msc) = \frac{\delta_{ij} - y}{a}; j = 1, 2, 3, 4, \ldots, \kappa \tag{3.8}$$

3.7 Supervised Versus Unsupervised Learning

ML models either use supervised or unsupervised approaches to generate insights from data. The unsupervised model is data-driven and does not need prior training (Seeger, 2000). Unsupervised models are mainly used for exploratory analysis, cluster detection, pattern detection, outliers detection, and qualitative research. On the other hand, for supervised models, the data is split into two sets: one for training and another for validation. The model is first subjected to the training data set with known labels. Afterward, the model is then subjected to the validation/testing data set, which it had not yet seen (Hastie *et al.*, 2009). Its performance is gauged using selected figures of merit to avoid under or overtraining of the models.

3.7.1 Unsupervised Learning Using the Principal Component Analysis

PCA is a statistical approach for reducing a dataset's dimension and collinearity while maintaining as much variability as possible (Jolliffe, 2005; Kjeldahl and Bro, 2010). The general form of a PCA is:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \tag{3.9}$$

where the data matrix $\mathbf{X} \in \mathbb{R}^{N \times K}$ is approximated by the product of two matrices; scores ($\mathbf{T} \in \mathbb{R}^{N \times A}$) and loadings ($\mathbf{P} \in \mathbb{R}^{K \times A}$). The matrix $\mathbf{E} \in \mathbb{R}^{N \times K}$ is regarded to as noise.

3.7.1.1 Number of PCA Components

The variances of the PCA scores are given as a per cent of the total variance (Jolliffe, 2002). The optimal number of PCs can be determined using the cumulative variances of the PCA scores. The Scree plot exhibits a sharp decline after a few PCs since the initial PCs account for the majority of the variance. PCA components with low variances reflect data noise. As a general guideline, the number of PCA components evaluated should account for at least 80%, if not 90%, of the overall variance (Worley *et al.*, 2013; Worley and Powers, 2015).

3.7.1.2 Outliers and Data Distribution

PCA works best for normally distributed data and is not heavily skewed. PCs are biased to the directions of maximum variance which, is increased by outliers in an otherwise uninformative direction. It is common practice to eliminate outlier observations to guarantee that only important information is extracted. To detect outliers in a multivariate dataset, the sample mean vector $\bar{\mathbf{x}} \in \mathbb{R}^K$ and the sample variance-covariance matrix $\mathbf{S} \in \mathbb{R}^{K\times K}$ of the matrix $\mathbf{X} \in \mathbb{R}^{N\times K}$ are computed:

$$\bar{\mathbf{x}} = N^{-1} \sum_{n=1}^{N} \mathbf{x}_n \tag{3.10}$$

$$\mathbf{S} = (N-1)^{-1}\mathbf{X}^T\mathbf{X} \tag{3.11}$$

where $\mathbf{x}_n$ is the n-th observation row vector in $\mathbf{X}$. The set of squared Mahalanobis distances may then be computed as follows (De Maesschalck *et al.*, 2000):

$$d_n^2 = (\mathbf{x}_n - \bar{\mathbf{x}})\mathbf{S}^{-1}(\mathbf{x}_n - \bar{\mathbf{x}})^T \quad \forall n \in 1N \tag{3.12}$$

Each squared distance is compared to a critical value from a T^2-distribution. Because the covariance matrix is highly rank-deficient and consequently non-invertible (i.e. $N \ll K$), the above technique fails in practice. The data matrix can be approximated using PCA (i.e. $\mathbf{X} \approx \mathbf{TP}^T$), to ensure a stable inversion of the covariance matrix, where the number of PCs is smaller than the rank of the data matrix. Squared Mahalanobis distances can be calculated from PCA scores using the orthonormality condition of PCA loadings:

$$d_n^2 = \mathbf{t}_n \left(\frac{\mathbf{T}^T\mathbf{T}}{N-1} \right)^{-1} \mathbf{t}_n^T \quad \forall n \in 1N \tag{3.13}$$

where $\mathbf{t}_n$ is the n-th row of the scores matrix $\mathbf{T}$. Because the matrix $\mathbf{T}^T\mathbf{T}$ is diagonal, it is inverted and calculation of each d_n^2 is greatly simplified. Mahalanobis distances calculated with PCA scores are close approximations of their true values in the original high-dimensional space (De Maesschalck *et al.*, 2000), and can be used to discover outliers. Scatter plots of PCA scores or diagnostics plots of Mahalanobis distances can be used to visually identify outliers (Hotelling, 1931; Worley *et al.*, 2013).

3.7.2 Supervised Machine Learning Models

3.7.2.1 Artificial Neural Networks

ANN (Bishop *et al.*, 1995; Ripley, 2007) replicate the way the human brains work. The prediction result is determined by hidden layers, which are linear combinations of the original predictors. However, this linear combination is typically adjusted with a nonlinear function $\beta(\cdot)$, like the sigmoidal (Kuhn *et al.*, 2013):

$$h_\eta(X) = \beta(\beta o_\kappa + \sum_{i=1}^{P} x_\alpha \beta_{\alpha\eta}), \qquad where \qquad \beta(u) = \frac{1}{1+e^{-u}} \tag{3.14}$$

The influence of the αth predictor on the ηth hidden unit is represented by the coefficient βo_{κ}. An ANN model must use hidden layers to model the outcome. However, no rule specifies how to configure these hidden layers (Mirjalili *et al.*, 2014). Following the determination of the number of hidden units, each one must be related to the outcome in the following manner:

$$f(X) \ = \ \gamma o + \sum_{\eta=1}^{\mu} \gamma_{\eta} h_{\eta} \tag{3.15}$$

There are a total of $\mu(\psi + 1) + \mu + 1$ parameters to estimate for a simple model with ψ predictors, which swiftly grows as ψ increases. Typically, the parameters are set to random values and subsequently solved using specialized methods. The back-propagation algorithm (Rumelhart *et al.*, 1985) is an exceptionally way for determining optimal parameters using derivatives. The solution, however, is not a global solution; the resulting collection of parameters is no better than any other set. Furthermore, due to a large number of coefficients, ANN has a propensity to over-fit (Atkinson and Tatnall, 1997).

Iterative methods that can prematurely halt Wang *et al.* (1994) the optimization when some estimate of the error rate begins to increase (early stopping) are employed to address this issue. The error rate can be overly optimistic, and further partition of the training set is precarious. Because the measured error rate is subject to ambiguity, we cannot tell if it is rising. Another strategy for reducing over-fitting is the employment of weight decay, where high-value coefficients are penalized to considerable tolerance on model errors. The optimization process attempts to minimize the sum of the squared errors for λ:

$$\sum_{i=1}^{n}(y_i - f_i(X))^2 + \lambda \sum_{\eta=1}^{H} \sum_{\alpha=0}^{P} + \lambda \sum_{\eta=0}^{H} \gamma_\eta^2 \tag{3.16}$$

As the regularization parameter increases, the fitted model becomes smoother and less prone to over-fit. The regularization parameter and the number of hidden units must be specified to tune the model. Between 0 and 0.1 is a reasonable λ value. Furthermore, the coefficients must be on the same scale; hence, the predictors must be centered and scaled before modeling.

3.7.2.2 Random Forest

As its name implies, the RF comprises many de-correlated decision-making trees that function as a set. In this tree, every non-leaf node is a decision-maker. These nodes are referred to as decision nodes that conduct a common test to determine where to go next, depending on the result. Either it goes to this node's left branch or right branch. The process continues until a leaf node has been reached. For classification purposes, each leaf node is a class (Hssina *et al.*, 2014). To construct an optimal tree, the Gini index is used as the cost function to assess the splits in the dataset by minimizing its index, as shown in equation 3.17. The split in the dataset involves one input attribute and one value for that attribute and can be used to divide training patterns into two groups of rows.

$$Gini\ (t) \ = \ 1 \ - \sum_{i=0}^{c-1} \ [\ p\ (i \ \mid t\ )\ ]^2 \tag{3.17}$$

p is a probability, t is the dataset, and c is the number of classes in the dataset. A Gini score indicates how well a split is done. A perfect separation results in a

Gini score of 0, whereas the worst-case split results in a score of 1. The Gini score is calculated from every row and splits the data according to the binary tree. The structure of a decision tree is shown in Figure 3.4

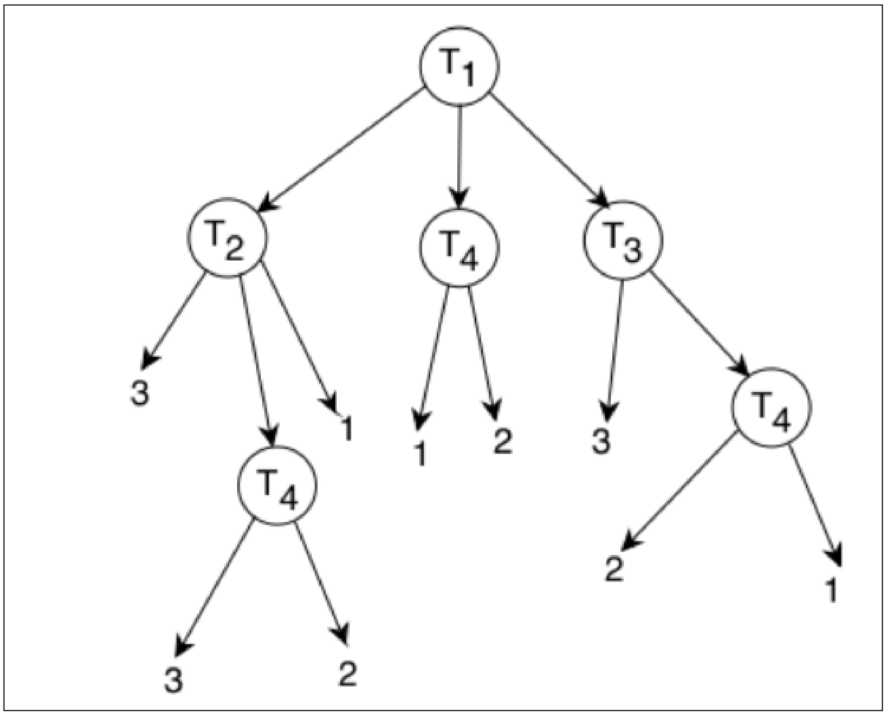


Figure 3.4: A decision tree showing the internal nodes and leaf nodes (source: Rokach and Maimon (2005)).

A decision tree is a predictor, $h : X \longrightarrow Y$, that predicts the label associated with an instance x by traveling from a root node of a tree to a leaf. The focus on the binary classification setting, namely , $Y = 0, 1$. The successor child is chosen based on splitting the input space at each node on the root-to-leaf path. Usually, the splitting is based on one of the features of x or a predefined set of splitting rules. A leaf contains a specific label. In this study, RF was used for regression.

3.7.2.3 Support Vector Machines

SVM is an algorithm that can differentiate between various groups using hyperplanes to maximize the gap between the classes for classification or regression.

Discrimination can be created for two or more data classes by an ideal identifying a hyperplane or a decision boundary that divides all data classes. The hyperplane maximizes the margin or space between the boundary and the points nearest to the decision line (also called support vectors). Support vectors are data points nearest to the hyperplane (Varmuza and Filzmoser, 2009). Support vectors can be described as equation 3.18.

$$fx - k = 0 \tag{3.18}$$

This gap is referred to as the margin and can be seen in Figure 3.5. The margin is the distance of 3.19 and 3.20 for two perpendicular hyperplanes.

$$fx - k = -1 \tag{3.19}$$

$$fx - k = 1 \tag{3.20}$$

And is from $2/||f||$. Minimizing $||f||$ leads to maximizing the margin as shown in equation 3.21. $\boldsymbol{C_i}$ is either -1 or 1, depending on the class label. The SVM model is illustrated on Figure 3.5.

$$c_i(fx_i) \leq 1 \tag{3.21}$$

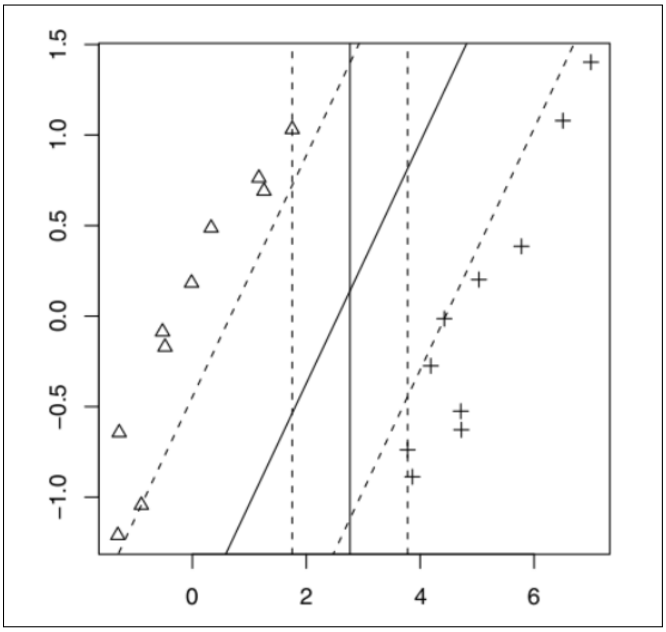


Figure 3.5: The SVM model: the hyperplane serves to maximize the margin. The striped lines are support vectors (source: Papadonikolakis and Bouganis (2012)).

Since there are hundreds or thousands of columns in spectral data, the information can be transformed efficiently into a higher-dimensional space by adding a correct kernel function. Basic segregation can be achieved in this high-dimensional space. Because of the dual representation, the exact transformation is performed using kernel functions such as Gaussian, linear, polynomial, radial kernels.

3.8 Multi Target Model Development

In supervised learning, Single-Target Model (STM) predicts the value of a single response. However, most real-world data has more than one target response. This problem can be solved by predicting these multiple outputs concurrently and is known as Multi-Target Regression (MTR). MTR is a difficult task in which the difficulties stem from capturing and exploiting potential correlations among target responses during training, at the expense of increasing the computational complexity of model training (Melki *et al.*, 2017).

Two general methods have been proposed for solving MTR tasks. The first method is called the local or problem transformation method where the problem is transformed into multiple STM problems where each regression model is solved separately. The second is known as the global or algorithm adaptation methods. Global methods use existing STM to predict all the target responses at the same time (Borchani *et al.*, 2015).

In this study, local problem transformation was adopted because of its simplicity in implementation. Consequentially, two machine learning models were constructed, each predicting a STM response (Mancozeb and THO concentrations). Prediction for validation or new samples was obtained using each separate model and concatenating their results. Conversely, when using global methods for the same two responses, only one model would need to be constructed, which would output all predictions (Melki *et al.*, 2017).

3.9 Figures of Merit in Model performance Evaluation

The predictive performance of the built machine learning models was assessed using statistical parameters. The coefficient of multiple determination (R^2) calculates the proportion of the total variance accounted for by the model, and the errors are due to the remaining variance. The optimum model was based on the calculation of Root Mean Square Error (RMSE). This value is a measure of how well the model fits the known and the predicted value. C_i is the actual value, while $\hat{C}_i$ is the predicted value and *n* number of validation samples.

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{n}(C_i - \hat{C}_i)^2}{n}} \tag{3.22}$$

3.9.1 Cross-validation

External cross-validation is the partition of N observations (i.e. $\mathbf{X}$ and $\mathbf{Y}$) into a training set ($\mathbf{X}_t$ and $\mathbf{Y}_t$) with N_t observations and a validation set ($\mathbf{X}_v$ and $\mathbf{Y}_v$) with N_v observations. The training data set is used independently for model generation, while the validation dataset is for testing. Due to the limited observations in most studies, external cross-validation is uncommon, and training uses all the observations. Internal cross-validation, where N observations are partitioned into G groups, is usually used. Each group is excluded in turn during training. Its response is approximated using the model trained on the remaining data.

3.9.2 Evaluation of Limits of Detection and Limits of Quantification

The LOD and the Limit of Quantification (LOQ) were evaluated to assess the least concentration of residues that could be detected and quantified by the proposed method. The LOD refers to the minimum analyte concentration that is observable with any given analytical technique and can be assumed to be present with any degree of certainty (UhrovÄDÃŋk, 2014). In comparison, LOQ corresponds to the minimal concentration of analytes that can be quantified with acceptable reliability.

The two metrics are easily extracted from uni-variate calibration methods. However, DRS spectra are highly multivariate, and the spectra are highly overlapped and redundant. Hence, uni-variate approaches, which do not adequately consider the different spectral responses for multivariate data, cannot be used. However, from the models calibration curves, if the response is linearly related to the predicted values, and the model can be expressed as a linear equation $y = a + bc$, the LOD and LOQ can be calculated from equations 3.23 and 3.24 respectively (Shrivastava *et al.*, 2011).

$$LOD = \frac{3\sigma}{S} \tag{3.23}$$

$$LOQ = \frac{10\sigma}{S} \tag{3.24}$$

where σ is the standard deviation of the response which can be estimated by the standard deviation the y-intercepts of the regression lines and S is the slope of the calibration curve.

CHAPTER FOUR

MATERIALS AND METHODS

4.1 Overview

This chapter discusses the methodology used. The operating conditions for the NIR system used in the analysis are presented. Additionally, the sample preparation procedure is discussed, as well as the spectral pre-processing techniques and the machine learning models used.

4.2 Samples and Pesticides used

The tree tomatoes (*Cyphomandra betacea*) is a vital fruit of the family Solanaceae (Tomato family) (Thakur *et al.*, 1996). Botanically, the tree tomato is a fruit (Harlan, 1928). Nevertheless, it can be cooked as vegetables making it an essential ingredient in the human diet that provides important nutrients (Klunklin and Savage, 2017). The tree tomato was chosen as case study because it acts as both a vegetable and a fruit. The fruits are low in fat, calories, and cholesterol. Additionally, they are rich in vitamins **A** and **C**, lycopene, *β*-carotene (Mangels *et al.*, 1993) and other antioxidants (Davies *et al.*, 1981). The tree tomato is often used for food directly, either as a raw or cooked vegetable, making it a perfect case study. The fruits were harvested in May 2020 from an organic orchard in Kikuyu (Kenya) and immediately packed in cartons and transported within an hour to the laboratory for analysis.

The two pesticides used in this study were zinc; manganese(2+); N-[2-(sulfidocarbothioylamino)ethyl] carbamodithioate (mancozeb) and N,Ndimethyltrithian-5-amine;oxalic acid (THO). The chemicals were manufactured in 2020 with a shelf life of two years. Their metadata are as shown in Table 4.1. These two pesticides were chosen because they are commonly used by farmers in Kenya and due to their availability.

The pesticides' compatibility was determined by measuring a small amount of water in a glass jar. The pesticides were then added to the water and vigorously agitated for one minute before allowing the solution to stand for fifteen minutes before re-stirring and recording the results. The solutions combined well, forming a smooth mixture with no separation clumps or grainy appearance, demonstrating that the two pesticides can be mixed (Montana State University, 2020).

Table 4.1: Metadata on the pesticides used in this study

Formulation	Category	Active Ingredients	MRL
Wettable granules	Fungicide	Mancozeb 640g/Kg	0.3 ppm
Wettable powder	Insecticide	THO 500g/Kg	0.5 ppm

Pesticide exposure causes cephalea, dizziness, vomiting, skin irritation, and long-term health consequences (Loha *et al.*, 2018). Poisoning has been documented as a result of a lack of protective measures when handling pesticides (Dasgupta *et al.*, 2007). To this end, protective clothing such as gloves and face masks were worn during chemicals handling. Additionally, the experiments were conducted in a well-ventilated room to avoid the risk of inhalation. Personal hygiene practices such as handwashing were followed (Dasgupta *et al.*, 2007).

Due to the empirical nature of residues analysis, it is necessary to establish sample-specific calibrations for practical implementation. The comparative nature of the method also necessitates the quantitative evaluation of treated and control samples.

4.2.1 Samples Preparations

4.2.1.1 Field Samples Preparations and Spectral Measurements

In an organic orchard, three mature tree tomatoes plants bearing ripe fruits were identified and labeled as **A**, **B**, and **C**, respectively. The trees were spaced five meters apart to avoid unintentional pesticide contamination during spraying. Tree **A** was set as the control and was not sprayed with any pesticides. Trees **B** and **C** were sprayed with a mixture of Mancozeb and THO in the ratios recommended by the manufacturer of 2.5g/liter for Mancozeb and 0.75g/liter for THO (Greenlife, 2020; Syngenta, 2020). Adequate coverage was ensured on the fruits and foliage but excessive runoff was avoided during spraying. The fruits were then allowed to dry completely for one hour.

The fruits were then subjected to spectroscopic measurements using Flame USB4000 spectrometer from Ocean Optics (200-1050 nm) for eleven consecutive days with a day interval. The farm was located in Kikuyu (1°15′ S, 36°40′ E) along the equator. When the zenith angle was 90 degrees, the sun was overhead and the irradiance was at its peak. Data were collected around midday. Calibrations with a 99.9% reflectance standard were always performed before each measurement to account for irradiance variability. All measurements were done in February 2020, during the dry season. Pictures are attached in Appendix B and the optimized data collection metrics used are shown in Table 4.2.

Table 4.2: Specifications of the Parameters Used for Data Acquisition in the Field

Parameter	Description	Values
Integration times	Time duration of each measurement	-
Auto-optimize	If active, integration time is ignored and rather adjusted before each measurement to account for illumination conditions	Yes
Acquisition frequency	Measurement repeat interval	30 s
Acquisition period	Window when measurements occurred	Midday
Scans to average	Number of spectra averaged for each recorded spectra	10
Boxscar width	Width of an averaging window that can be used for smoothing spectra to reduce noise	10
Non-linearity correction	Option for correcting for non-linearity response of the detector	Yes
Electric dark correction	Option for subtracting optically dark detectors to remove the baseline noise from spectra	Yes

Flame USB4000 is a miniature spectrometer from Ocean Insight with a 16-bit A/D resolution, an enhanced electric dark-signal correction, and an extended range from 200 nm to 1025 nm. The spectrometer required 250 mill-amperes at 5 volts to operate and was powered from the laptop USB port. The module only weighs around 0.3 kilograms and contains no moving parts making it portable and ruggedized for field measurements. Ocean view software was used for parameter settings and data acquisition. Additional technical specifications are available in Table 4.3.

Table 4.3: Specification of the USB Flame 4000 S-XR1 Utilized in the field Study

USB Flame 4000 S-XR1	
Detector	Linear Silicon CCD array
Weight	0.265 kg
Pixels	2048
Detector Range	200-1025 nm
Entrance Aperture Size	25μm
Dynamic range	3.4×10^6: 1300:1 single acquisition
Optical Resolution	1.69 nm FWHM
SNR At Full Signal	300:1 (at full signal)
Size	$89.1 \times 63.3 \times 34.4$ mm

Reflectance spectroscopy compares the relative level of light (in %) reflected off a sample relative to a reference. A reflectance standard (Spectralon) was used to set the reference level of 100%. Diffuse reflectance uses a fiber optic probe fixed at a 45°angle, a light source and a spectrometer (Figure: 4.1).

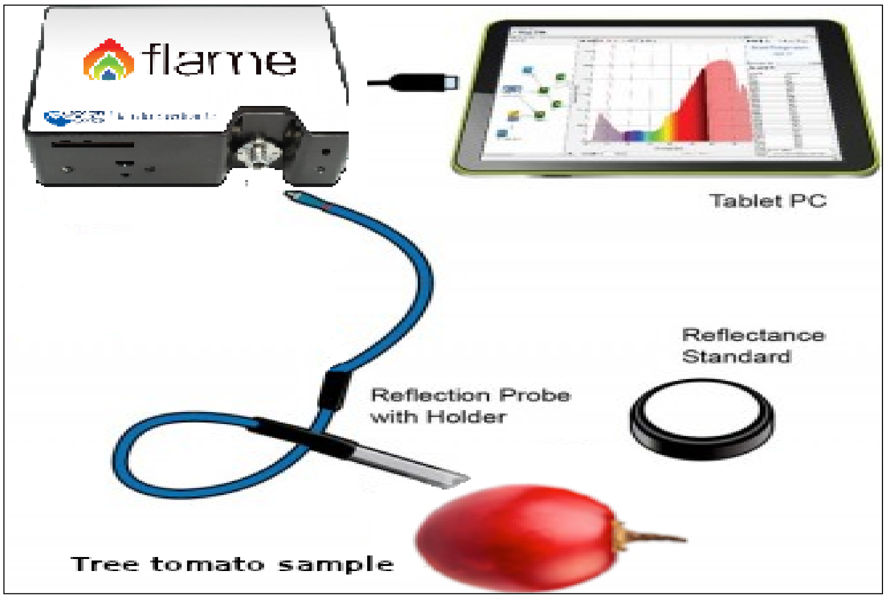


Figure 4.1: Diffuse reflectance set up with fiber optic cable, reflectance standard and fruit samples.

4.2.1.2 Laboratory Samples Preparation Procedures and Spectral Measurements

The range of the prepared calibrations samples concentrations was influenced by the MRL set by the EU for the two pesticides. The MRL for Mancozeb and THO is 0.3 and 0.5 ppm respectively. Randomized concentration ranges within the MRL ranges of the two pesticides were generated using R software. Utilizing the randomized concentrations, the respective volumes required to get the concentrations were calculated using the serial dilution equation (4.1) where C_1 and C_2 are initial and final concentration respectively, and V_1 and V_2 are the initial and final volumes respectively (Diabaté *et al.*, 2014).

$$C_1V_1 = C_2V_2 \tag{4.1}$$

The prepared solutions were vigorously shaken as recommended by the manufacturer to maximize sample homogenization. Each of the samples was then labeled as A, B, C, . . . up to H. A detailed report can be seen in Table 4.4 and pictures in Appendix A.

Table 4.4: Prepared pesticides concentrations that were sprayed on fruits

	THO		Mancozeb	
Label	**Prepared Conc ppm**	**Mass of Solute in Grams in 100ml of Solution**	**Prepared Conc ppm**	**Mass of Solute in Grams in 100ml of Solution**
Stock	500	0.075	2500	0.25
A	Blank	0	Blank	0
B	0.01	1.5×10^{-6}	0.01	1×10^{-6}
C	0.03	4.5×10^{-6}	0.04	4×10^{-6}
D	0.06	9×10^{-6}	0.07	7×10^{-6}
E	0.15	2.25×10^{-5}	0.21	2.1×10^{-5}
F	0.18	2.7×10^{-5}	0.25	2.5×10^{-5}
G	0.5	7.5×10^{-5}	0.3	3×10^{-5}
H	1	1.5×10^{-4}	0.5	5×10^{-5}

NIRQuest 512-2.5 spectrometer from Ocean Insight was used for the laboratory-based data acquisition. The spectrometer is small and light, weighing only 1.2 kg with a diode array technology with no moving parts making the detector rugged even for field applications. Sample excitation was done using DR diffuse reflectance probe from ocean optics with a fixed 45° angle and an integrated halogen tungsten light source. The probe also houses the collection optics. Table 4.5 lists all the specifications of the equipment used for the laboratory-based experiment. Fiber optics cables were used to connect the spectrometer and the probe, and data acquisition was through the Oceanview software.

Table 4.5: Specification of NIRQuest 512-2.5 and the diffuse reflectance probe utilized for the laboratory based study

NIRQuest 512-2.5	
Detector	Hamamatsu G9208-512w in GaAs linear array
Weight	1.2 kg
Pixel Size	$25\mu m \times 250\mu m$
Detector Range	$900 - 2500\ nm$
Entrance Aperture Size	$25\mu m$
Standard Grating	NIR1
Collimating and Focusing Mirrors	Gold Plated for Enhanced NIR Reflectivity
Optical Resolution	$6.3\ nm\ with\ 25\mu m\ slit$
SNR At Full Signal	10000:1 at 100ms integration
Temperature Limits	10-35℃
TEC Range	30-50℃ below ambient
Filter	Second Order filter
TC-DR-Probe	
Collection Spot Size	$\approx$ 12mm
Illumination Spot Size	$\approx$ 15mm
Focal Length	40 mm (Standard)
Light Source	6 Watts Tungsten Halogen
Measurement Angle	45°

Since the diffuse reflectance collection optics had a reasonably small collection field (12mm) and an illumination spot size of approximately 15mm, fruit cuticles were sliced into small sizes. The prepared different concentrations were then sprayed on tree tomatoes cuticles and the pesticides were allowed to dry completely before being subjected to spectroscopic measurements.

Data acquisition for the lab-based approach was much easier due to the use of an external tungsten halogen lamp whose intensity was almost constant. To create more robust models, fruit samples with different ripening degrees were used. Also, to minimize the effects of temperature on NIR measurements, the data acquisition was carried out in a controlled room with an average temperature of 24℃.

The spectra were acquired in diffuse reflectance geometry mode on several spots of the sample's surface (see Figure 4.1). The detector used a 25 μm entrance slit, and three diffraction grating covering the NIR ranges from 900-2500 nm. The TC-DR probe provided measurements at a fixed 45°angle to minimize the specular component from the fruit surface. Spectralon (99% white reflective reference standard) was used for calibration before each measurement. This was done to ensure that the maximum reflectance of the spectralon over the entire spectrum was more than 99% the saturation value. A computer with Ocean view software was used to set the spectra collection parameters for the spectrometer. The best integration time was set automatically by the software. Each spectrum was collected as an average of ten scans on different spots for a more representative spectrum and to minimize the detector's thermal noise (Nicolaï *et al.*, 2008). To smoothen the spectra, a boxcar width of five scans produced the optimal results. The optimized data collection metrics used are shown in ref Table 4.6.

Table 4.6: Specifications of the Parameters Used for Data Acquisition in the Laboratory

Parameter	Description	Values
Integration times	Time duration of each measurement	3s
Auto-optimize	If active, integration time is ignored and rather adjusted before each measurement to account for illumination conditions	Yes
Acquisition frequency	Measurement repeat interval	30 s
Scans to average	Number of spectra averaged for each recorded spectra	10
Boxscar width	Width of an averaging window that can be used for smoothing spectra to reduce noise	5
Non-linearity correction	Option for correcting for non-linearity response of the detector	Yes
Electric dark correction	Option for subtracting optically dark detectors to remove the baseline noise from spectra	Yes

4.2.1.3 Market Samples Preparation Procedures and Spectral Measurements

Tree tomatoes samples (see details in Table 4.7) were purchased from five different local markets to assess the developed models' performance. The samples were measured using DRS, and the resulting spectra were preprocessed in the same manner as the calibration samples before PCA. The four PCs with the most variation in the data were extracted and used for modeling.

Table 4.7: Market Samples Acquired for Testing the Developed Models

Location	Vendor	Number of Samples
Gikomba	A	11
	B	25
	C	13
	D	26
	E	17
Limuru	F	28
	G	25
	H	20
	I	25
Wangige	J	20
	K	22
	L	31
	M	25
Ngara	N	9
	O	10
KL	P	8
	Q	9

4.3 Optimization of the DRS Measurements for rapid pesticide residues assessment

The field-based data acquisition approach was challenging due to uncontrollable factors such as the varying weather conditions on different days as already discussed in chapter two section 2.4. Because the main source of excitation light used in the field approach was from the sun, factors such as cloud cover, humidity, temperature, wind direction, and, speed were taken into consideration. Experimental issues such as measuring space, measuring height, measuring orientation, and, the field of view was also taken into account because they affect the precision of field DRS measurements.

To standardize the experiment, the data was collected at around the same time of the day when the weather conditions were almost similar. Calibration with a 99% reflectance standard was done before every measurement to account for the varying irradiance and weather conditions. The diffuse reflectance measurements were performed on different spots of the selected fruits in a random manner and on fruits with varying degrees of ripeness.

Several factors were taken into account when configuring the system for the operation of the NIRQuest spectrometer out to 2500nm in a laboratory setting. The integration times were kept short to enable the detector to be sensitive up to 2500 nm. This is because the detector bandgap energy must be small, but unfortunately, this also raises the dark detector signal's absolute level. For maximum signal intensity, this study used fluoride fibers from Ocean Insight optics that do not attenuate the signals above 2200 nm.

If the optical spectrometer bench is not well designed, temperature fluctuations increase thermal noise and sensitivity by disrupting the silicon detector's photo-response. This is caused by the thermal expansion of optical components, causing misalignment. This effect manifests itself as a bias offset in the spectra. Another factor that can affect NIR measurements is light source fluctuation due to aging (Acharya *et al.*, 2014). The position of the O−H bond and, therefore, all molecules containing this hydroxyl group are affected by variations in temperatures. This effect is more pronounced in fruits and vegetables, which are high in moisture and manifested as offset or bias. This bias can compromise the prediction performance of the ML model utilizing these wavelengths (Guthrie *et al.*, 2006). Global models were developed to compensate for this undesired effect. Global modeling is the inclusion of samples at different temperatures in the calibration models (Peirs *et al.*, 2003).

Before taking the measurement, another critical consideration was to allow the spectrometer and the light source to warm for 30 minutes (for Lab-based experiment) to stabilize the detector response. The light source employed has a wide spectrum output, with greater output energy in the NIR than in the other EMS regions. The angle was fixed at 45° to collect only the diffuse component and minimize the specular component. A constant focal length of 40 mm was used to reduce measurement errors as the optical geometry remains fixed from sample to sample. Besides, before every measurement, the system was calibrated with a spectralon reflectance standard from Ocean Insight.

Utilizing the relationship in equation 4.2, the mean reflectance values (S) were transformed to relative reflectance (R) in relation to the standard (W) and dark measurements' reflectance (D). The dark measurement was acquired by either covering the entire tip of the reflectance probe or turning off the light source (Cortés López, 2018).

$$R = \frac{S - D}{W - D} \tag{4.2}$$

4.4 Exploratory Analysis and Modelling of DRS Measurements

Machine learning algorithms are computational tools that aid in predictive modeling of complex data such as DRS spectral data that was obtained in this work

(Michie *et al.*, 1994). Multivariate ML techniques were used to overcome problems associated with the classical analysis of spectral data. Supervised and unsupervised methods were used for performing exploratory and multivariate calibration of acquired spectral data. All data analysis and machine learning models were developed and implemented in open-source software R version 3.6.3 (2020-02-29) using the caret package (Kuhn, 2008).

4.4.1 Pre-Processing of DRS Spectral Measurements

DRS analytical results are multivariate. The spectra were influenced by background noise, light scattering, and temperature changes. These variations can negatively impact the creation of the calibration model and result in partisan results. Spectral data pre-processing techniques, namely: PCA for outlier detection, pattern recognition and features reduction, MSC for scatter correction and normalization to de-noise the spectra.

Normalization removes scatter effects from the individual spectra and put all variables on the same scale. Such variables with high and low intensities assume equal significance. Additionally, these spectral pre-processing techniques were useful in removing outliers, reducing variability in the data, and redundant information while retaining the most significant spectra features to be used during subsequent multivariate ML modeling to improve the accuracy and robustness of models.

A total of 329 spectra were collected from the field for the various days from the treated and the control groups. The data was a first-order tensor of $N \times K$ dimensions, where N are the rows that represent the number of samples. The columns (K) are the X variables or predictors which are the spectral signals from the USB4000 detector in the 350 to 900 nm range with 1 nm resolution. Using $\log_{10}(1/R)$, the X variables were converted to apparent absorbance to linearise the correlations (Stenberg *et al.*, 2010). As discussed in chapter three section 3.5, the spectrum of fruits in the NIR is highly complicated due to the dominance of fruit moisture, tissue discrepancies, wavelength-dependent scattering effects, instrumental noise, environmental effects (temperature and light), and, other variability sources (Nicolaï *et al.*, 2008). Thus, various spectral pre-processing techniques were employed to clean data before implementing the unsupervised machine learning technique PCA for qualitative analysis.

A two-degree polynomial and a seven-point window size SG were applied to the normalized spectra to improve the SNR ratio to reduce the effects of sample physiological variability (Cortés López, 2018). Because of the light scattering in the samples cuticle, the light propagation path was not the same before it was captured. This manifested as scattering effects on the spectra and was corrected by the use of MSC. The spectra were finally normalized (Bakeev, 2010) by scaling to a maximum value of one and a minimum of zero to ensure spectral intensities in a data tensor were directly comparable across each observation.

The data collected from the laboratory was also a first-order tensor of $N \times K$ dimensions, where $N = 126$ samples are the rows. The columns (K) are the X and Y variables. X variable or predictors are spectral signals from the NIRQuest detector in the range of 900-2500 nm with 1 nm resolution. In contrast, Y variables or responses were two different columns of concentrations of the pesticides to be discriminated against. After the X variables were converted to apparent absorbance by using a log(1/X) transform, they were corrected for

MSC and smoothed using a SG filter with a second-degree polynomial and a five-point window size. The data was also normalized in a similar way to the field data.

Finally, an unsupervised eigenvector decomposition PCA was done on both the field and lab-based data using chemospec package in R (Hanson, 2016) to reduce dimensionality, detect outliers, visualize the clustering of the data, and, as a data mining technique before ML multivariate analysis. Since each PC consists of a score and a loading vector, PC1 has the highest explained variance, while PC2 which is orthogonal to PC1 has the next highest possible variance and so on for subsequent components (Kara and Dirgenali, 2007). The PCs with the highest explained cumulative variance (99%) were extracted from the PCA results and used as the input dataset for subsequent ML modeling to save processing time, computer memory, and increase model accuracy.

4.4.2 Machine Learning Modeling of DRS Spectra

This section details the use of three ML models; ANN, RF, and SVM to quantify residues on fruits. The inspiration for using multiple models derives from David Wolpert's and William Macready's ”No Free Lunch Theorem (NFLT)” arguing that no model can be said to be consistently better than the other in the absence of any information about the prediction problem. Thus, it makes sense to use a variety of models to seek one that matches the information better (Ho and Pepyne, 2002).

Since the problem was a multi-target regression (MTR), the local problem transformation approach was adopted due to its simplicity in implementation. Consequently, two separate regression models were developed for each response variable (Mancozeb and THO concentration). The reduced PC data (N=126, K(X=4, Y=2)) from the identified ROI was used in this section. The PC data was split into a training set (60 %) for the development of a calibration model and a test set (40 %) for model performance evaluation (Soares *et al.*, 2013). The performance of ML models from transformed spectra was cross-validated and only results with the best prediction performance metrics were selected.

4.4.3 Artificial Neural Network Models

ANN was employed in regression modeling of PCs scores data against the various concentrations of pesticides used in this study. Several models were developed with various cross-validation parameters, activation functions, algorithms, and architectures. Using Root Mean Square Error of Calibration (RMSEC) and $\boldsymbol{R}^2$ as the accuracy metrics, the model with the lowest RMSEC and highest $\boldsymbol{R}^2$ were adopted for this study. The models were tested using a validation data set to determine whether they were under or over-fitting. Since the regression was a multi-target regression, the local transformation problem was used where two different models for each pesticide (Mancozeb and THO) were developed separately.

A list of cross-validation settings was used to develop the ANN model for Mancozeb to optimize for the best models. The comprehensive code is attached to the appendices. The training set had 83 samples and four predictors, which corresponds to the four PCs used. The data were centered and scaled before a five-fold cross-validation re-sampling with sample sizes of 67, 65, 66, 67, and 67 across the tuning parameters. The final architecture had 26 hidden neurons in the first hidden layer, four neurons for the second hidden layer, and finally three hidden neurons for the third hidden layer. This model was able to achieve an RMSEC of 0.0212 ppm and R^2 of 0.9784. The resilient back-propagation with a weight backtracking algorithm with 50 training repetitions produced the best results with 1×10^5 maximum steps for calibration and a 0.01 threshold for the partial derivatives of the error function as stopping criteria. The prediction set with 43 samples achieved the lowest Root Mean Square Error of Prediction (RMSEP) of 0.0451 ppm and an R^2 of 0.9321.

The final THO model was also a fivefold cross-validated model with 83 samples in the calibration set and four predictors. The re-sampling sizes were 67, 65, 66, 67, 67 across several tuning parameters. The data was also scaled and centered. The resilient back-propagation with weight backtracking algorithm with 50 training repetitions produced the best results with 1×10^5 maximum steps for calibration and a 0.01 threshold for the partial derivatives of the error function as stopping criteria. The final values used for the model were layer1 = 40, layer2 = 4 and layer3 = 3. The best calibration model had an RMSEC of 0.0307 and an R^2 value of 0.9846 and an RMSEP of 0.0658 and an R^2 of 0.9645 on the test set.

4.4.4 Random Forest Models

The RF uses a decision tree to model linear or non-linear data. This study utilized the development of regression models for predicting the varying concentrations of the two pesticides. Consequently, independent models were developed; one for Mancozeb and the other for THO. Different cross-validation parameters, number of trees, split (mtry), and, train control parameters were used in model development to prevent overfitting. RMSEC and R^2 were the used figures of merit for model accuracy.

The training data-set was centered and scaled before a leave-one-out cross-validation (LOOCV) re-sampling with sample sizes of 82, 82, 82, 82, 82, and 82 across the tuning parameters. The best Mancozeb model had 83 samples for training with and four predictors corresponding to the uncorrelated PC data used for modeling. The number of variables randomly sampled at each split (mtry) was three for the optimum model with 20 trees. The model had an RMSEC of 0.0229 and an R^2 of 0.9719 on the calibration and RMSEP of 0.0418 and an R^2 of 0.9651 on the test set.

The training data set used for training had 83 samples and four predictors. A five-fold cross-validation re-sampling with a single repeat with sample sizes of 67, 65, 66, 67, 67 were found to produce the best THO model based on the model evaluations metrics. The optimum model had 50 trees and the number of variables tried at each split (mtry) was two. This model had the lowest RMSEC of 0.0499 and R^2 of 0.9780 for calibration while for the test set, an RMSEP of 0.0837 and an R^2 of 0.9763 were achieved.

4.4.5 Support Vector Regression Models

In this study, Support Vector Regression (SVR) was used to quantitatively model for predicting the varying pesticides concentrations for both Mancozeb and THO. The models were cross-validated across a range of parameters such as the cost, gamma, and kernels to determine the optimum number of hyper-planes to use. RMSEC (for both calibration and validation) and R^2 were used as the figures of merit for assessing model accuracy.

Support Vector Machine with a radial basis kernel function was found to give the best performance. The Mancozeb model was trained with a dataset of 83 samples and four predictors. The data were centered and scaled before a three-fold cross-validation re-sampling with sample sizes of 57, 55, and 54 across the tuning parameters. The final values used for the model were sigma of 0.05, cost of 1, epsilon of 0.1, and 51 support vectors. The best model achieved an RMSEC of 0.0434 and R^2 of 0.9194. The model had an RMSEP of 0.0689 and an R^2 0.9167 on the test dataset.

The best THO model also used a radial kernel basis with 83 samples and four predictors. The data were centered and scaled before a three-fold cross-validation re-sampling with sample sizes of 57, 55, 54 across the tuning parameters. The final values used for the model were; sigma equal to 0.5, cost (C) of 16, epsilon of 0.1, and 32 support vectors. The training model achieved the best RMSEC value of 0.0437 and an R^2 of 0.9810. On the test set, the model produced an RMSEP of 0.0646 and R^2 of 0.9809. Figure 4.2 summarizes the data methodology used in this study.

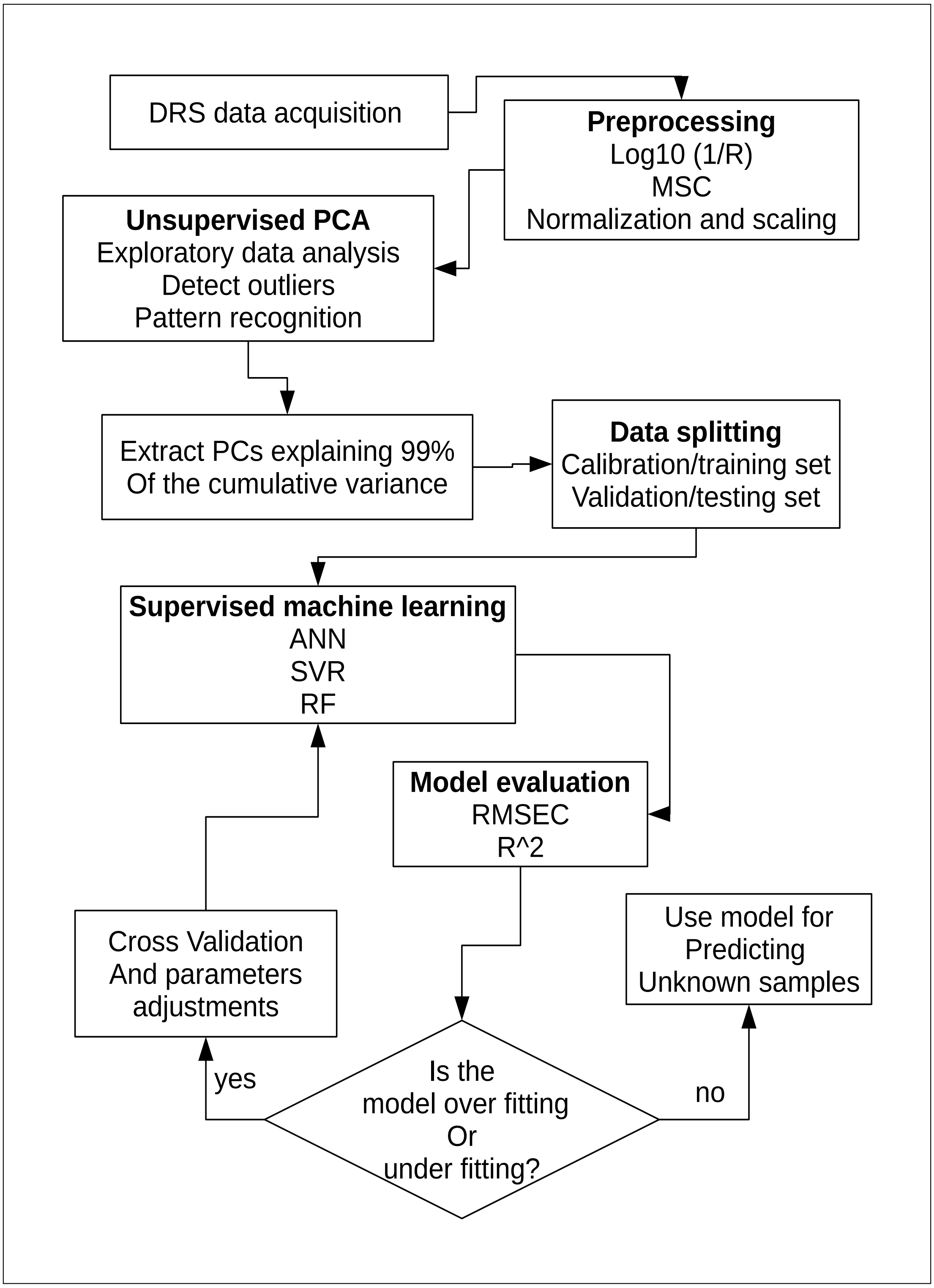


Figure 4.2: Flowchart of the methodology summary employed in this study for supervised and unsupervised machine learning.

CHAPTER FIVE

RESULTS AND DISCUSSION

5.1 Overview

This chapter discusses the results of application of machine learning models to DRS spectral data obtained from residue analyses on tree tomatoes fruits.

5.2 Pre-Processing and PCA of DRS Spectra

5.2.1 Field Based DRS Measurements

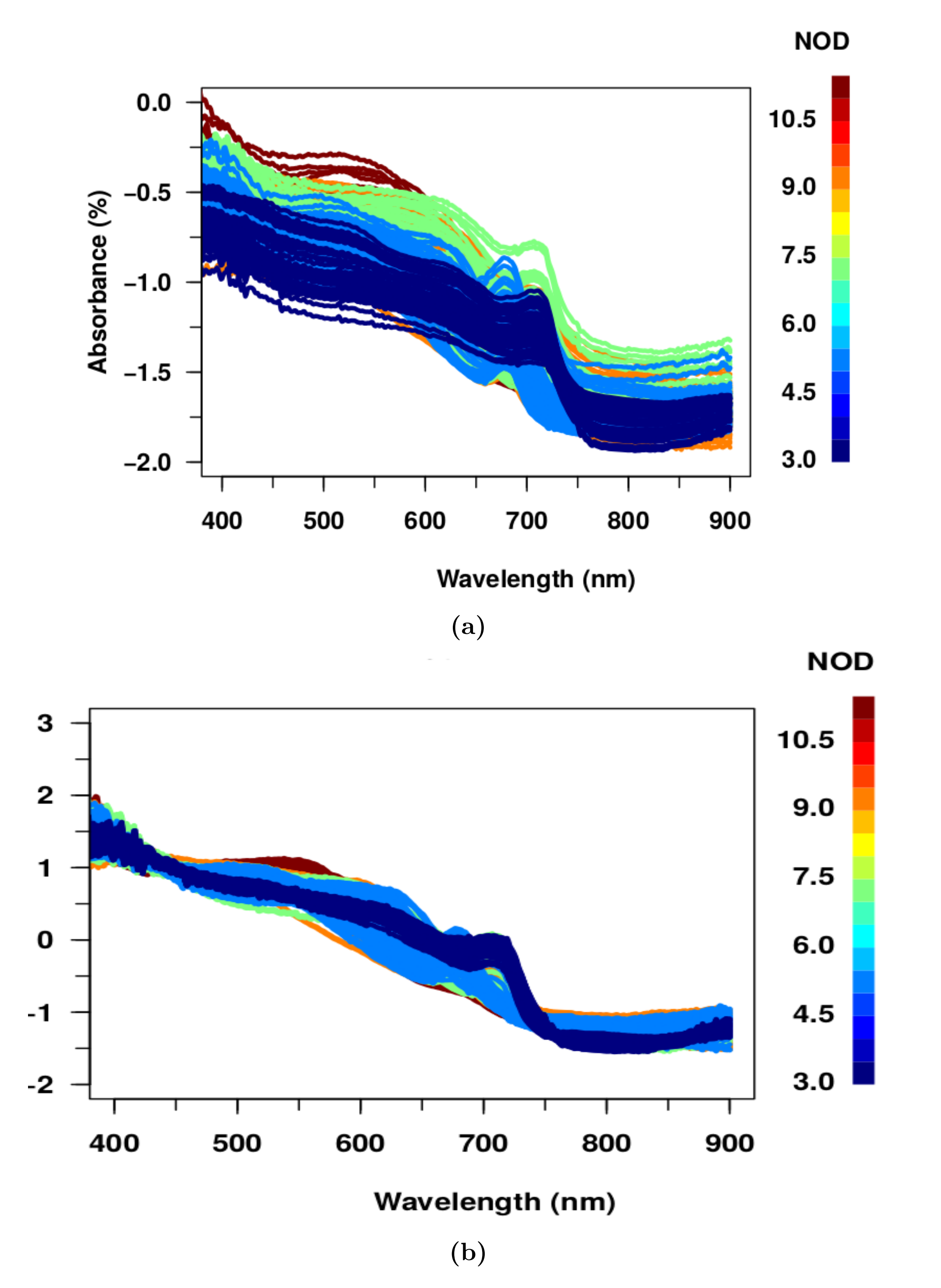
The field data were collected for eleven days with a one-day interval. Figure 5.1a depicts the raw spectra after it was converted to absorbance using log(1/R) and then MSC treated (Figure 5.1b). The number of days is represented by **NOD**. The scatter or offset in the spectra can be attributed to the sun's varying irradiance, as it was the primary source of excitation light in the field.

Because fruits have a high moisture content, spectra above 850 nm were discarded to avoid low SNR due to $2v1 + v3$ ($v1$: symmetric stretching; $v3$: anti-symmetric stretching) water vibrations, which dominate the raw spectra. In the visible region (400 to 700 nm), the spectra show a decreasing trend with some discernible absorption peaks. This absorption in the visible region was primarily related to the color of the fruits, which varied with ripeness. These peaks are due to chlorophyll A (685 nm) and B absorption, respectively.

The short wave NIR region between 700-900 nm shows an increasing trend, which can be attributed to either the third overtone of C−H, second overtone stretching of O−H, or, CH_3 third overtone stretching (Jamshidi *et al.*, 2016). Because it was unaffected by color, the region between 800-850nm produced the best PCA results. The data were divided into two groups for easier, more general comparisons. One for the treated samples over the course of eleven days, and the other for all of the control groups over the course of the eleven days. The data was then plotted with a 0.1 offset, as shown in Figure 5.2.

To evaluate possible classes among the samples, PCA was performed on the data in the frequency range of the pretreated spectra between 800 and 850 nm. The scores plot revealed two distinct clusters that corresponded to the two broad groups (Figure 5.3). PC1 accounted for 82 percent of the observed variance, while PC2 contributed 5.6 percent. PC1 was able to clearly distinguish between the two clusters representing the control and treated groups (Figure 5.3). Positive PC1 can be used to distinguish the control group, whereas the negative portion can be used as a ROI for samples treated with Mancozeb and THO mixture.

Based on the PCA loadings weights, sensitive bands for residues analysis in fruits from field data were proposed (Figure 5.4). Wavelengths ranging from 825 to 850 nm influenced the treated group to cluster on negative PC1, whereas wavelengths ranging from 800 to 825 nm influenced the control group to cluster. The 800-850 nm third overtone region is very narrow and primarily composed of the N−H and C−H stretch regions. The region between 800-825 nm is associated with the primary (RNH_2) and secondary amines (RNHR), whereas the region between 825-850 is associated with the aromatics (ArCH) third overtone C−H stretching vibration (Workman Jr, 2000).



(a)

(b)

Figure 5.1: An illustration of the raw absorbance spectra for treated and control groups from the field data before (Figure 5.1a) and after MSC correction (Figure 5.1b), **NOD** denotes number of days

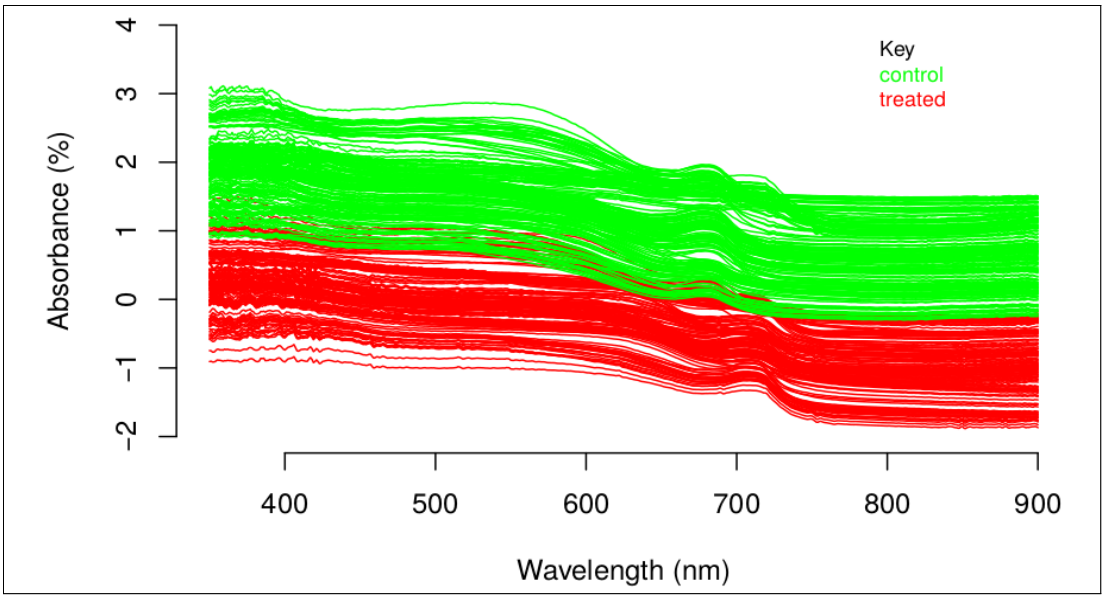


Figure 5.2: Absorbance spectra of the field data for the control and treated groups plotted with a 0.1 offset.

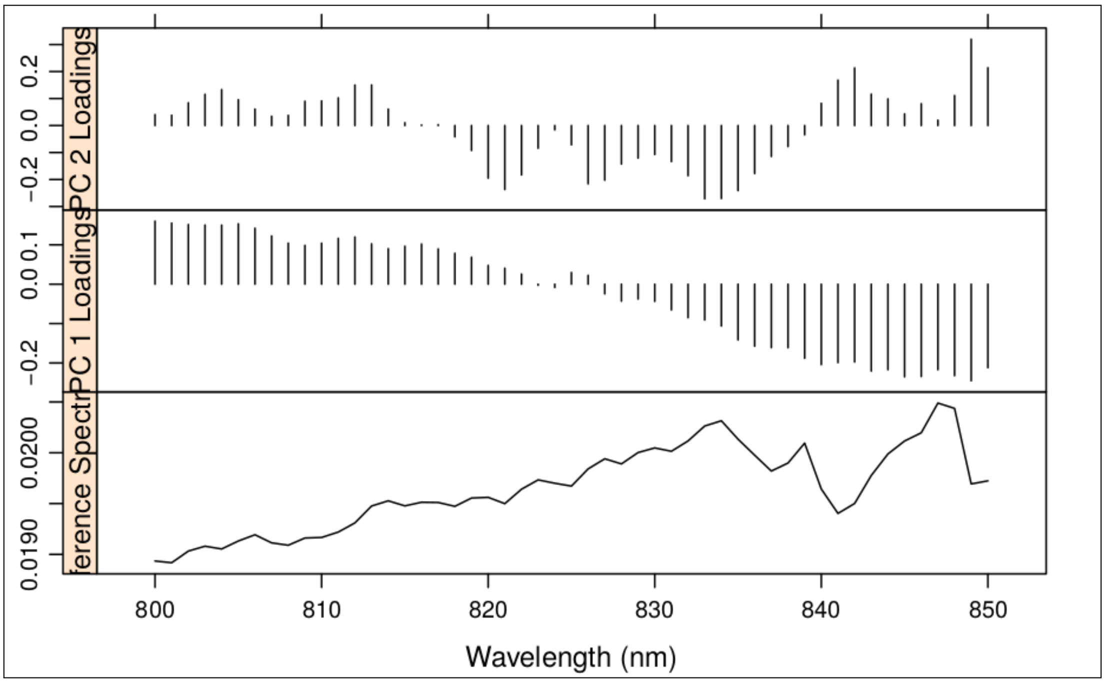


Figure 5.4: Field data PCA loadings plot for the shortwave NIR region between 800-850 nm

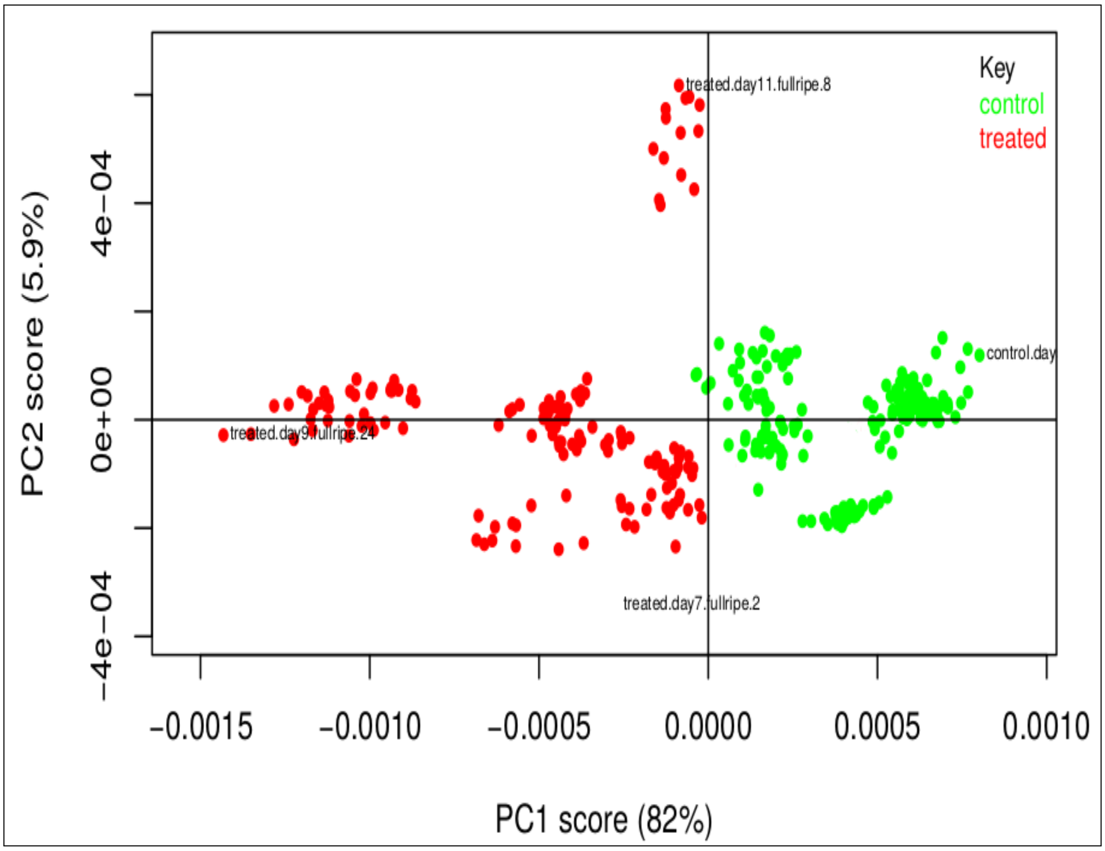
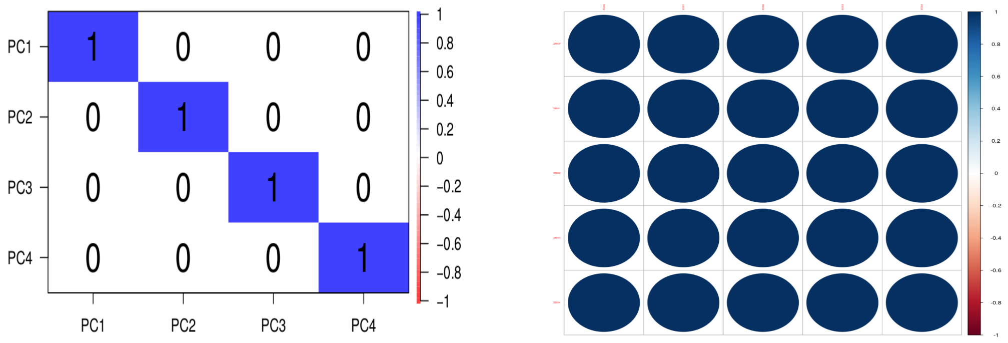


Figure 5.3: The illustration shows that the two groups are easily distinguishable along PC1. The different clusters from each group imply that the method is sensitive enough to distinguish between the different days of the experiment.

5.2.2 Laboratory Based DRS Measurements

The raw and MSC treated spectra (Figure 5.6) show an increasing trend between 1000-1375 nm. The absorption peak between 1500 and 1750 nm could be due to bands in CSNHR structures that can be used to identify Mancozeb. The second overtone absorption band, with a peak between 2000 and 2250 nm, is caused by NH stretching plus amide II. These bands could also be associated with CH groups, which are also found in the Mancozeb formula (Arias *et al.*, 2013; Moros *et al.*, 2007; Osborne, 2006). The combination region between 1900 and 2500 nm (Figure 5.7) shows the spectra of different residue concentrations used.

The PCA analysis was used to remove redundant information and summarize the high-dimensional data into a lower-dimensional dataset with a smaller number of variables known as PCs, which are uncorrelated and relevant (Rodriguez-Campos *et al.*, 2011), as opposed to using the highly correlated wavelengths directly. The correlation plot for the first four PCs is shown in5.5a. The fact that the coefficients for the various components are all zero implies that there is no relationship between them. The same plot is shown in Sub-Figure ***b***, but for the first five wavelengths in the data matrix. A dark blue color on the diagram scale indicates that the wavelengths are highly correlated.

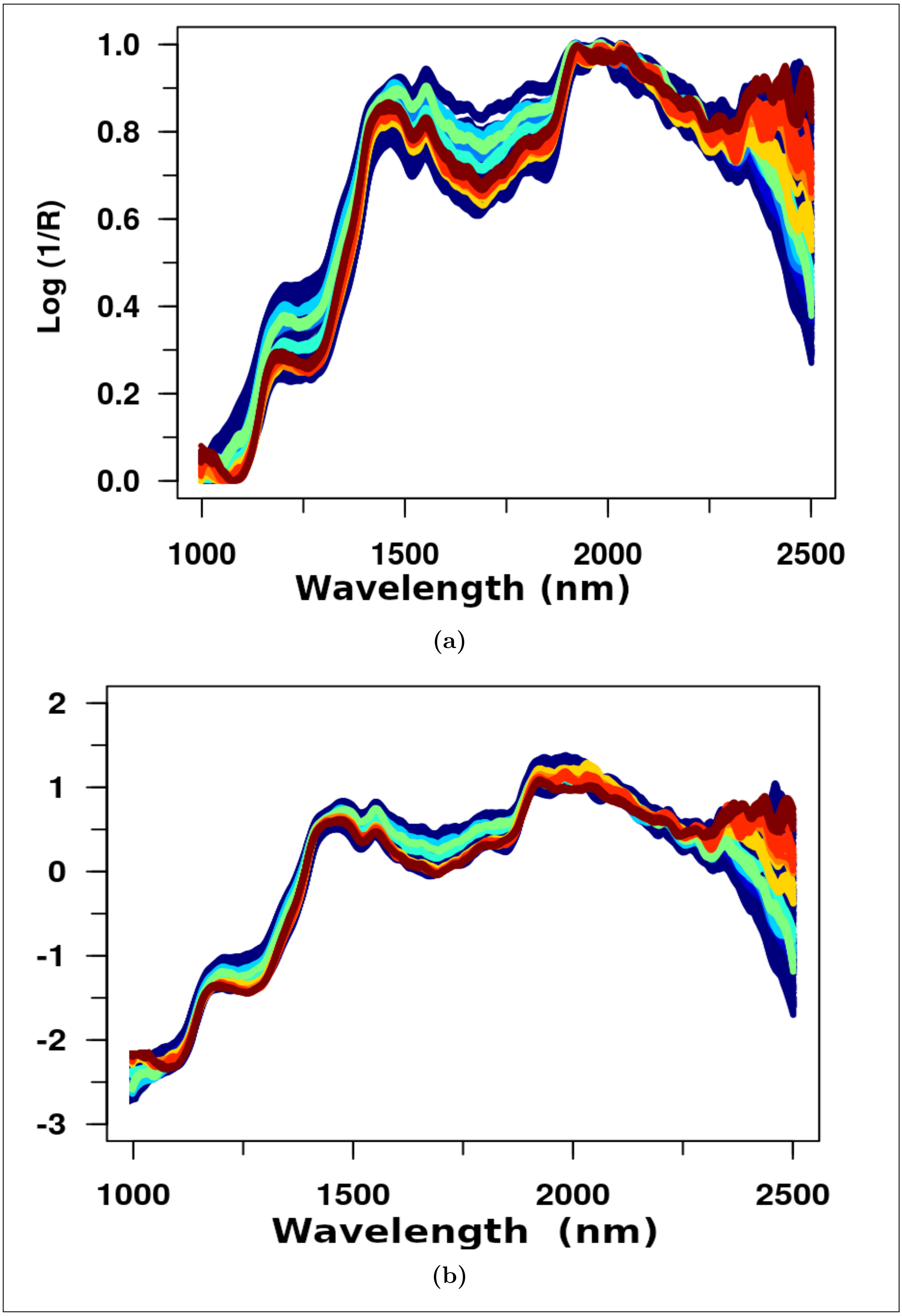


(a)

(b)

Figure 5.5: Using PCA, a completely new data set with uncorrelated features is created. Figure 5.5a depicts the correlation plot of the first four PC. The components are completely unrelated. Figure 5.5b is a correlation plot of the first five wavelengths in the data matrix. Their correlation coefficient is one.

The data set was also unbalanced with 1600 predictors and 126 samples. The number of samples in the training set in ML models must be greater than the number of variables in the model, necessitating a variable selection reduction technique (Cortés López, 2018). Furthermore, only truly relevant variables should be included during modeling, further justifying the use of PCA.



(a)

(b)

Figure 5.6: A depiction of the raw absorbance spectra (Figure 5.6a) before and after MSC treatment (Figure 5.6b)

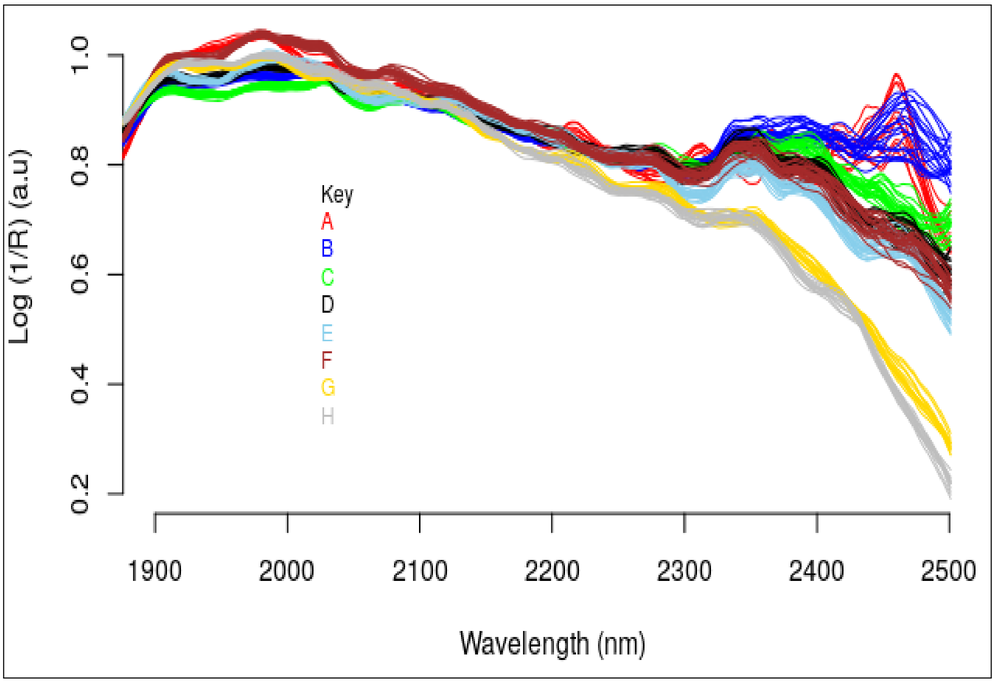


Figure 5.7: The log (1/R) transformed spectra recorded from tree tomatoes fruits samples at various ripening stages treated with two different pesticides, Mancozeb and THO. The absorption features in DRS spectral features are broad and overlapped hence require processing.

The concentrations, as well as the number of samples, are shown in table 5.1. PCA was performed on pre-processed spectra using different wavelength ranges to identify the most optimal ROI for discrimination of pesticide concentrations.

Table 5.1: Table displaying the pesticide concentrations used in this study.

Label	Mancozeb Concentration (ppm)	THO Concentration (ppm)
A	Control	Control
B	0.01	0.01
C	0.03	0.04
D	0.06	0.07
E	0.15	0.21
F	0.18	0.25
G	0.50	0.3
H	1	0.5

ThePCA of 1900-2500 nm range produced the best results, with PC1 accounting for 93% of the variance and PC2 accounting for 3.8%. Figure 5.8 shows eight patterns in ascending order from the lowest (A) to the highest (H) concentrations matching to the concentration used (Table 5.1). Based on the scores plot (Figure 5.8), PC1 explained the fluctuation in concentrations. It's worth noticing that the concentration steadily rose from negative PC1 to positive PC1, or from B to H. PC2 could differentiate between control and treated samples. The loading plot (Figure 5.9) was utilized to better comprehend this grouping.

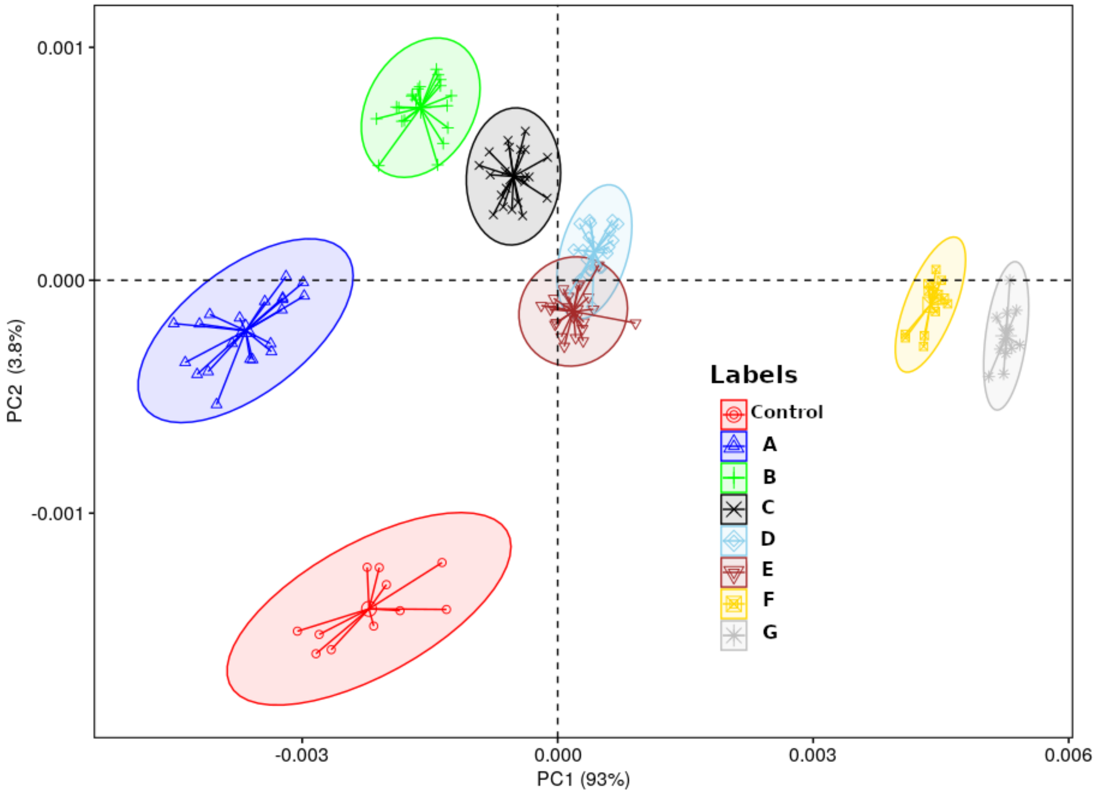


Figure 5.8: Plots of PCA scores for the various pesticide concentrations used in this study. The mean value of the cluster is represented by the center of the lines.

Positive PC1 can be attributed to wavelengths between 1900-2150 nm, while negative PC2 can be attributed to wavelengths between 2200-2500 nm, according to the loadings plot (Figure 5.9). Conversely, for PC2, which accounts for 3.8 percent of the total variance, the positive portion is attributed to vibrations in the region between 2200-2400 nm, while the negative portion is attributed to vibrations in the region between 1900-2150 nm and the peak between 2410-2490 nm.

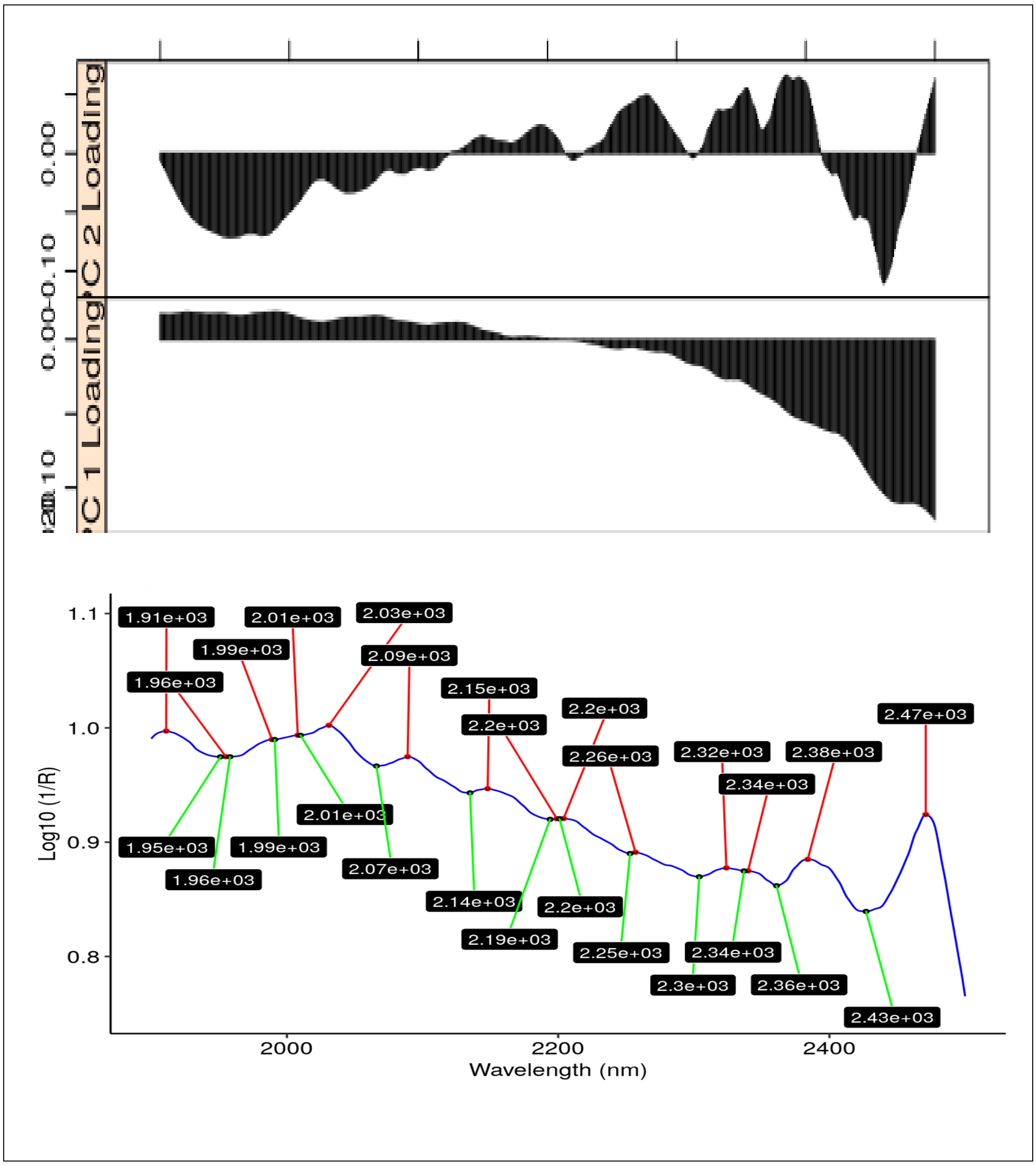


Figure 5.9: The Loadings plot of the region between 1900-2500 nm with all the major peaks observed in the Spectra.

The chemical structure of the two pesticides used was discussed to understand and explain the attribution of the functional groups involved using the loadings plots. THO is a 1,2,3-trithiane hydride with dimethyl (CH_3) amino group replacing one of the oxygen positions. Six aliphatic saturated organosulphur heterocycles with three carbon atoms make up the structure. The oxalate is a dicarboxylic acid obtained by the deprotonation of both carboxy groups of oxalic acid.

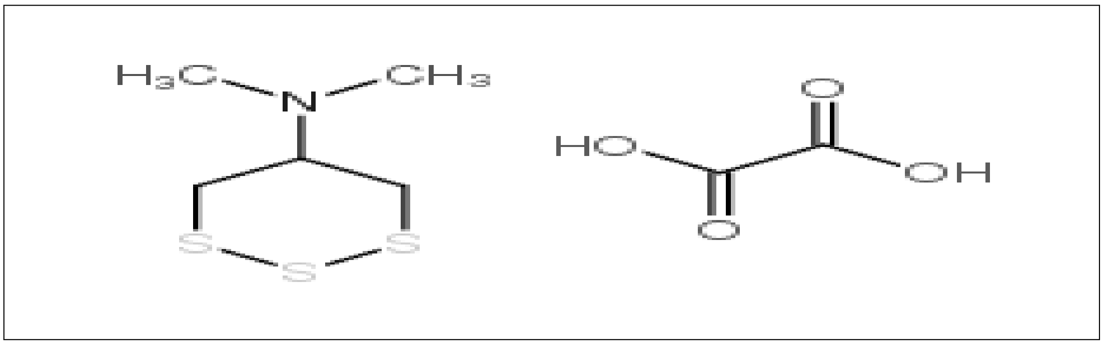


Figure 5.10: Thiocyclam hydrogen oxalate's chemical structure.

Mancozeb ($C_4H_6N_2Mn \cdot C_4H_6N_2S_4Zn$) (Figure **??**) is a Maneb and Zineb mixture; a Manganese (Mn) and Zinc (Zn) mixture with the ethylene ($H_2C{=}CH_2$) bis(dithiocarbamate) anionic ligand. Fruit cuticular waxes, which are made up of long-chain aliphatics (LCA) and pentacyclic triterpenoids, were another important component that was also the base matrix. LCA is made up of fatty acid derivatives of C_{16} and C_{18}, alcohols, aldehydes, and alkanes with long chain lengths ranging from C_{16} and C_{35}.

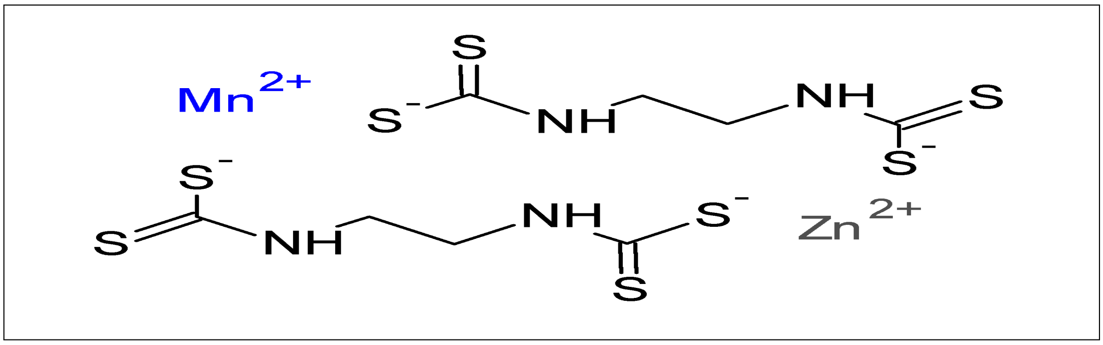


Figure 5.11: Chemical structure of Mancozeb.

The observed peaks (Figure 5.9) were assigned to their respective vibrations (overtones or combination bands). The carboxy group (C=O) second overtone, O−H first overtone stretch, and C−O second stretch were assigned broadband between 1900-1950 nm with a peak around 1910 nm. Cutin was regarded as the band's main contributor. Another influential band at approximately 2010-2070 nm, with a maximum peak at 2030, was assigned to the O−H combinations, N−H asymmetrical stretching plus amide II, second O−H deformation, and C−O deformation. Stretching and combinations of C−O and O−H between 2070-2140 nm, with a peak at 2090 nm, were attributed to alcohols in the cuticle. The NH combination region between 2140-2190 nm with a peak at 2150 nm, the band was caused by C−C , C−O, O−H stretching and combinations. This region was also linked to NH deformation and combination, amide I and amide II

The peak at 2200 nm, located between 2190 and 2250 nm, was in the NH+OH combination region. This region was associated with CH_3, C−C, C−O, O−H stretch and deformation, NH deformations, amide I and amide II. The saturated aliphatic bonds were responsible for the CH_3 combination, CH_2 and CH stretch, and second overtone deformation, which were accompanied by the C−C stretch at around 2320 nm and 2340 nm. The peak at 2380 nm was linked to a

C−H second overtone deformation. Combination stretching vibration of C=C functional groups from phenolic compounds, =C−H, and combinations of NH stretching and bending were assigned to the 2470 nm peak. Table 5.2 summarizes the functional groups assigned from (Workman Jr, 2000).

Table 5.2: Summary of NIR active functional groups in the DRS spectra. The bands were assigned from Workman Jr (2000)

Band Region (nm)	**Peak (nm)**	**Functional Group Assigned**
1900-1950	1910	C=O 2α, OH 1α, C−O α
2010-2070	2030	NH B_α, Amide II, 2αO−H ψ, C−O ψ
2070-2140	2090	ROH, CONH$_2$(R), C−O, O−H α
2140-2190	2150	NH Comb, C−C, C−O, O−H α, NH ψ Comb, 2α Amide I and Amide II
2190-2250	2200	NH + OH Comb, CH$_3$, CC, CHO, RNH$_2$, NH ψ, O−H α, O−H ψ and 2α Amide I, Amide II
2300-2360	2320	CH + CH Comb, CH$_3$, CH$_2$
	2340	CH Comb, C−H + C−C α, C−H 2ψ
2360-2430	2380	C−H 2ψ, CH$_3$, CH$_2$, CH
2430-2500	2470	CH + CC Comb α of C=C, =C−H, N−H α and β

Key: Comb (combination region), 1α (first overtone of stretching fundamental vibration), 2α (second overtone of stretching fundamental vibration), 2β (second overtone of bending fundamental vibration), A_α (antisymmetric stretching), B_α (symmetric stretching), α (stretching), β (bending), ψ (deformation), 2ψ (Second Overtone deformation) (Zapata *et al.*, 2018).

5.2.3 Variable Selection in DRS Spectral Multivariate Modeling

When using data from the entire spectra for multivariate calibration, the practice resulted in ML models with poor prediction ability and general reliability. As a result, only the variables containing the most relevant information for the modeling were used. To that end, PCA was used to remove redundant features as well as noise. The inputs were the results from the region of interest between 1900 and 2500. The first four PCs were required to explain 98 percent of the data's cumulative variance. As a result, they were extracted from the result matrix and fed into machine learning models. The Scree plot (Figure 5.12) was used to obtain the number of PCs to use. The data was modeled using the local transformation method, which solved multiple (STM) separately to model Mancozeb and THO responses.

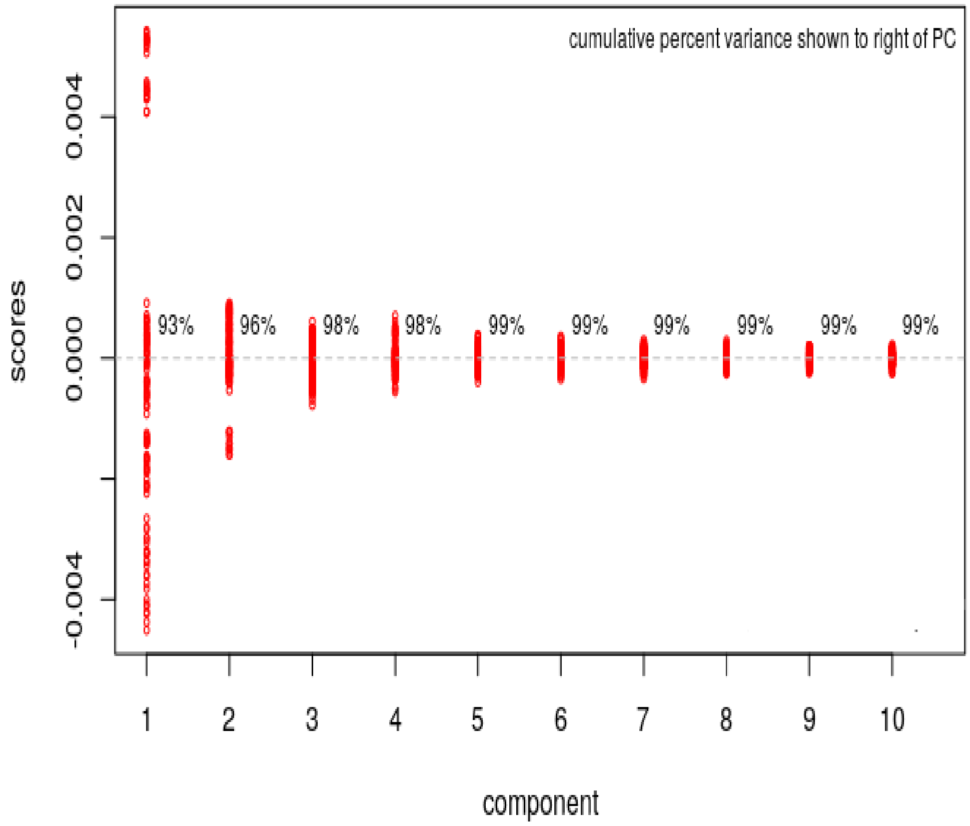


Figure 5.12: The most important components are depicted in a PCA scree plot. The first four PCs, which explained more than 98 percent of the cumulative variance, were extracted and used in the development of all ML models.

5.3 Multivariate Modeling of DRS Spectra

The relationship between pesticide concentrations and PCs data was quantified usingANN, SVM and RF for Mancozeb and THO. The same PCA dataset was used on all models to compare their performance.

5.3.1 Quantification Models for Predicting Mancozeb Residues

The pesticides residues were predicted using the developed ANN, RF and SVR models within the range of experimentally prepared concentrations. The obtained results confirm that ML models can accurately predict values that are nearly identical to those obtained from laboratory prepared concentrations. To validate the predictive ability of the developed multivariate calibration models, the predictions were evaluated using RMSE and R^2. The performance comparison of the three models used in this work based on these parameters is shown in Table 5.3.

Table 5.3: Mancozeb models performance based on RMSE, R^2, LOD and LOQ figures of merit.

	Calibration Set		**Testing Set**	
Model	**RMSEC(ppm)**	R^2	**RMSEP(ppm)**	R^2
ANN	0.02	0.98	0.05	0.93
RF	0.02	0.97	0.04	0.95
SVR	0.04	0.92	0.07	0.92

These findings indicate that employing PCA to remove redundant and noisy data can improve the speed, accuracy, and reliability of the developed models. The goal of developing three ML models was to determine which model performed best during the training and testing phases. The prediction performance of these models is compared in Table 5.3. With a R^2 of 0.98 and a RMSEC value of 0.02 ppm, ANN performed best for the calibration set. With a RMSEP value of 0.04 ppm and a R^2 value of 0.95, the glsrf model was the best on the testing set. This RF performance can be attributed to the fact that it can handle multiple classes well, with the number of trees being the only hyper-parameter to tune. However, these models require more data to learn and may be prone to over-fitting. It is worth noting that, in contrast to RF, ANN and SVR have a plethora of tuning parameters that must be tweaked to achieve the best results.

As the No Free Lunch Theorem implies, there is no single best model.

The problem at hand determines the performance of the models (Caruana and Niculescu-Mizil, 2006). This emphasizes that the three models differ not only mathematically, but also in how they learn from the same dataset. This means that each model captures a different aspect of the underlying complex relationship between the DRS spectra and the various pesticide concentrations used. As a result, they can be viewed as complementary models. The correlation curves of the models for predicting Mancozeb residues in tree tomatoes are depicted in Figure 5.13. On the model calibration plots, the y-intercept standard error and slope were calculated using a pseudo univariate approach. LOD and LOQ were computed using these values, and the results are shown in Table 5.4.

Table 5.4: Mancozeb models LOD and LOQ values calculated from the calibration curves using a pseudo-univariate approach

Model	**Slope**	**Y-intercept error**	**LOD(ppm)**	**LOQ(ppm)**
ANN	0.955	0.036	0.011	0.037
RF	1.008	0.004	0.011	0.037
SVR	1.024	0.004	0.039	0.012

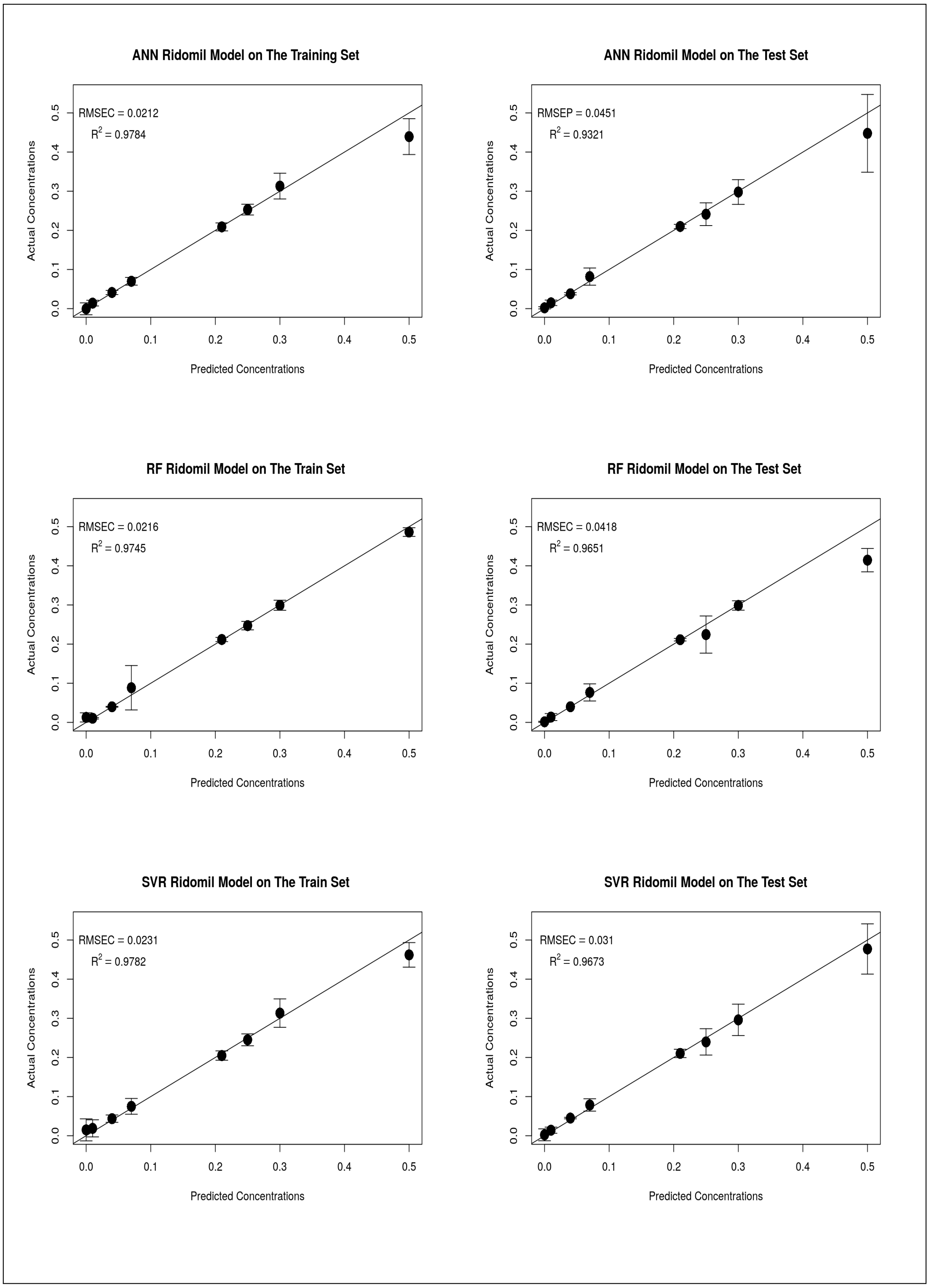


Figure 5.13: ANN, RF and SVR Mancozebs models correlation plots between actual and predicted residues concentrations values for the PCA training and testing data sets from the NIR combination region between 1900-2500 nm.

5.3.2 Quantification Models for Predicting THO Residues

Three ML models (ANN, RF and SVR) were developed to correlate THO concentrations and DRS spectra from tree tomatoes. Figure (5.14) depicts the linear relationships between laboratory prepared concentrations and predicted results for the three models. The predictive performance of the developed multivariate calibration models was evaluated using RMSE and R^2. The results are shown in Table 5.5. On the calibration set, all of the models had a R^2 value of 0.98, with the RMSEC of 0.03 ppm for the ANN model model. On the test set, the SVR model had the lowest RMSEP value of 0.06 ppm, while the RF model had the highest R^2 value of 0.98.

The RMSEC results show that the ANN model for THO outperformed the other two models. This is most likely due to the non-linear activation function used in model development, which allows ANN to resolve linear and non-linear relationships in the DRS spectra (Chen and Ramaswamy, 2000). The LOD and LOQ were calculated from the slopes and y-intercept standard errors of the calibration curves using a pseudo univariate approach. Table 5.6 presents the results.

Table 5.5: THO models performance based on RMSE and R^2

	Calibration Set		**Testing Set**	
Model	**RMSEC(ppm)**	R^2	**RMSEP(ppm)**	R^2
ANN	0.03	0.98	0.07	0.96
RF	0.05	0.98	0.08	0.98
SVR	0.04	0.98	0.06	0.97

Table 5.6: THO models LOD and LOQ values calculated from the calibration curves using a pseudo-univariate approach

Model	Slope	Y-intercept error	LOD(ppm)	LOQ(ppm)
ANN	1.021	0.004	0.013	0.043
RF	1.042	0.004	0.012	0.042
SVR	0.978	0.006	0.019	0.012

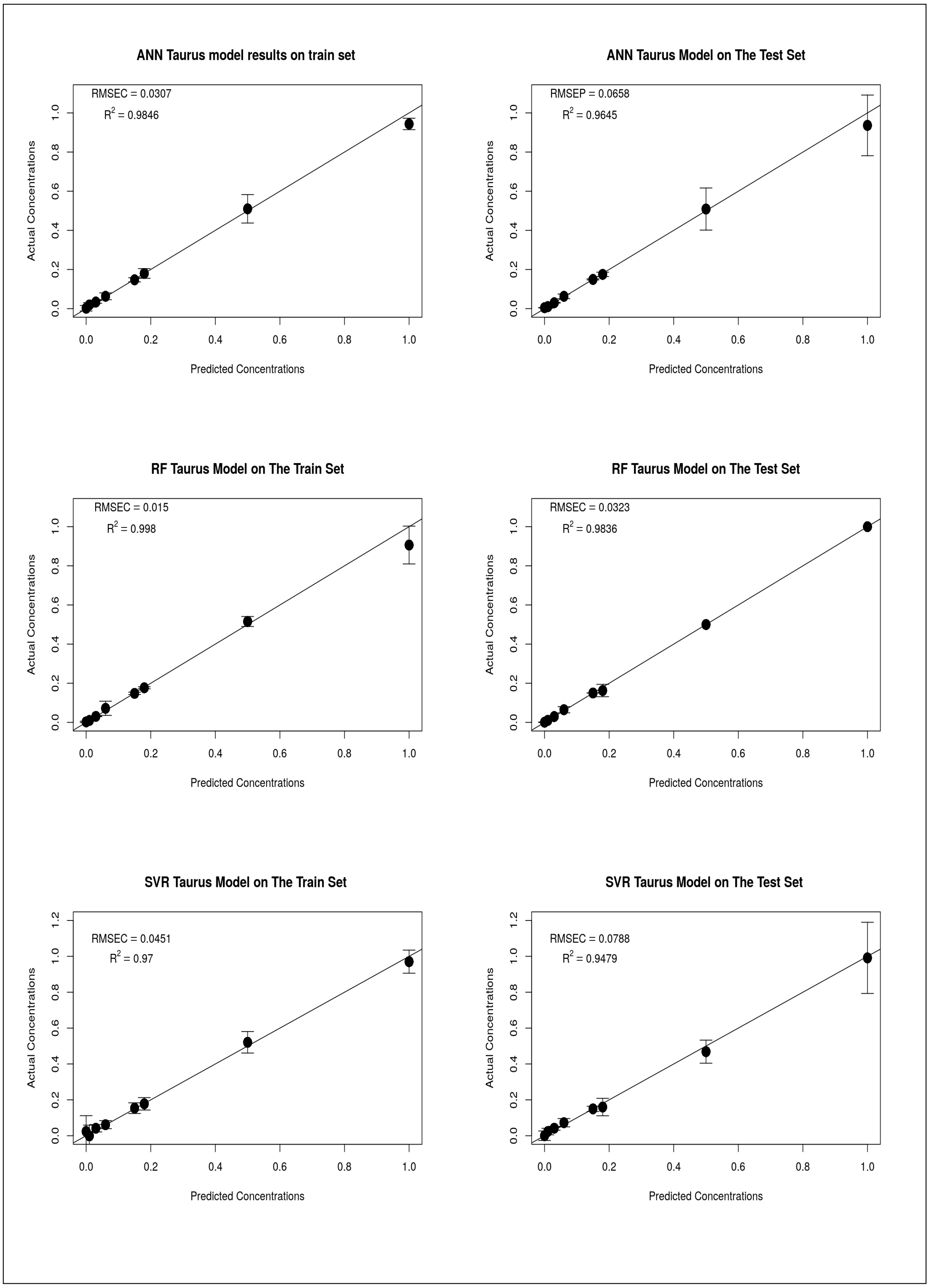


Figure 5.14: ANN, RF and SVR THO models correlation plots between actual and predicted residues concentrations values for the PCA training and testing data sets from the NIR combination region between 1900-2500 nm.

5.4 Performance of the Models Based on Market Samples

One of the study's goals was to see how well the developed models performed on real-world fruit samples from local markets. As a result tree tomatoes fruits were purchased from five different local markets. The market samples were subjected to DRS measurements, and the resulting spectra were preprocessed in the same manner as the calibration samples. The preprocessed spectra were then subjected to a PCA. Four PCs were extracted as they accounted for the greatest amount of variation in the data. Due to the skewed nature of the predicted data, descriptive statistics revealed a discrepancy between the samples' mean and median. As a result, the medium and Mean Absolute Deviation (MAD) metrics were chosen as the best fit for representing the data. Table 5.7 and Figure 5.15 provide a detailed description of the results. The dotted red lines represent the MRL, which is 0.5 and 0.3 ppm for Mancozeb and THO, respectively.

The THO models predicted that the Gikomba samples had residues levels below the MRL. According to the three THO prediction models, Limuru samples, represented by the color green, had THO levels below the MRL. The ANN model predicted samples from Vendor J from Wangige to be 0.38 ppm and the RF model predicted samples to be 0.41 ppm, both of which are above the allowed limits. However, the SVR TM predicted that the levels in sample J would be 0.26 ppm, which is less than the MRL. These model discrepancies can be attributed to the fact that each model had different tuning parameters. Furthermore, each model learns from the data uniquely. The three THO Models predicted that THO concentrations in samples K, L, and M were less than 0.3 ppm. All three models predicted that Ngara samples (N) fell below the red line. According to the RF model, however, sample O had 0.45 ppm.

Two samples (KL) with known residue concentrations were used to evaluate the performance of our models. Sample P contained 0.24 ppm THO and 0.3 ppm Mancozeb, whereas sample Q contained 0.16 ppm THO and 0.10 ppm Mancozeb. Based on this benchmark, the ANN model was the best for THO and Mancozeb prediction, while the RF model was the worst. This is not surprising given that ANN and SVR are more flexible models in terms of parameter tunability than RF. As a result, they should theoretically be better at learning complex data sets like DRS data.

All of the Gikomba fruits were predicted to be below the 0.5 ppm MRL level by the three Mancozeb models. The highest prediction from this location came from vendor C, with the RF model predicting 0.24 ppm. The models predicted that all Limuru samples would be below the red line. The highest prediction (0.24 ppm) came from Limuru vendor F, and the RF model predicted the same way. The SVR models predicted 0.36 ppm for the Wangige samples, closely followed by the ANN model, which predicted 0.35 ppm. The three models predicted the Ngara samples to be within a narrow range of 0.24 ppm to 0.17 ppm. The ANN model was the most accurate for the known samples group KL, with predictions of 0.13 and 0.27 ppm for samples P and Q, respectively.

A one-way analysis of variance (ANOVA) was performed on the market samples' predictions to find the best models for residues prediction on fruits. A null hypothesis (H_0) was assumed, implying that the mean of the ANN model (μ_{ANN}), the mean of the RF model (μ_{RF}), and the mean of the SVR model (μ_{SVR}) were all equal. However, based on the descriptive statistics of the models, non-equality in the means of the groups led to the rejection of the null hypothesis. To avoid type one errors, a posthoc one-way Tukey Kramer ANOVA test, which is only performed after the null hypothesis has been rejected, was used as a pairwise comparison between Mancozeb and THO's models. Given that Tukey's test is a posthoc test, a linear regression model was fitted for each sample data set based on the vendor, followed by ANOVA. For Mancozeb and THO, mean comparisons were performed between $\mu_{ANN}-\mu_{RF}$, $\mu_{ANN}-\mu_{SVR}$ and $\mu_{RF}-\mu_{SVR}$. The results are shown in Table 5.8.

The results were presented using statistical tests, specifically the t-test and the p-test, with a 95 percent confidence interval of the difference between means. The T-test compares the means of independent samples by dividing the difference between groups' means by the standard error of the difference between groups' means. A higher t-test value indicates that the means are unequal, whereas a lower value indicates that the means are similar. The sign of the t-test is determined by whether the difference in means between groups is greater than the standard error of the difference between groups (positive) or vice versa (negative) (negative). A negative t-test value indicates a shift in model directionality and has no bearing or significance on the difference between groups.

The P-value (P) is a metric that indicates how closely an observation agrees with the null hypothesis (H_0). For low P values, the (H_0) is rejected, while for high (P) values, it is accepted. The (P) value has a level of significance (α), which determines whether a value is high or low. When $P < \alpha$, it is assumed that there is statistical significance between groups. $P > \alpha$, on the other hand, indicates that the observed difference between the two groups is not statistically significant. From consensus, the α value is usually 5% (0.05). A statistically significant difference is assumed if $P < 5\%$, whereas there is no statistically significant difference between groups if $P > 5\%$. Based on the results of the post hoc one-way ANOVA Tukey test (Table 5.8), the $\mu_{ANN} - \mu_{SVR}$ had no statistically significant differences in mean in the majority of market sample predictions for both pesticides and are thus best suited for analysis of residues on fruits.

Table 5.7: Predicted values from machine learning models from the market samples

		Mancozeb Models			**THO Models**		
		ANN	**RF**	**SVR**	**ANN**	**RF**	**SVR**
Vendor	**Location**	**Conc (ppm)**	**Conc (ppm)**	**Conc (ppm)**	**Conc (ppm)**	**Conc (ppm)**	**Conc (ppm)**
A	**Gikomba**	0.10 ± 0.06	0.02 ± 0.00	0.12 ± 0.03	0.05 ± 0.04	0.16 ± 0.07	0.13 ± 0.04
B		0.19 ± 0.04	0.03 ± 0.01	0.17 ± 0.04	0.09 ± 0.05	0.10 ± 0.05	0.15 ± 0.04
C		0.18 ± 0.01	0.24 ± 0.03	0.18 ± 0.02	0.13 ± 0.01	0.14 ± 0.01	0.07 ± 0.08
D		0.13 ± 0.07	0.02 ± 0.01	0.13 ± 0.03	0.11 ± 0.06	0.10 ± 0.04	0.10 ± 0.04
E		0.12 ± 0.05	0.02 ± 0.00	0.12 ± 0.02	0.11 ± 0.03	0.07 ± 0.01	0.11 ± 0.01
F	**Limuru**	0.18 ± 0.04	0.24 ± 0.06	0.16 ± 0.03	0.10 ± 0.07	0.16 ± 0.07	0.17 ± 0.07
G		0.05 ± 0.01	0.05 ± 0.03	0.09 ± 0.01	0.05 ± 0.01	0.10 ± 0.04	0.08 ± 0.04
H		0.08 ± 0.07	0.05 ± 0.00	0.09 ± 0.04	0.03 ± 0.04	0.07 ± 0.00	0.13 ± 0.09
I		0.13 ± 0.06	0.02 ± 0.00	0.14 ± 0.03	0.10 ± 0.04	0.05 ± 0.01	0.11 ± 0.03
J	**Wangige**	0.35 ± 0.01	0.24 ± 0.00	0.36 ± 0.01	0.38 ± 0.09	0.41 ± 0.00	0.26 ± 0.04
K		0.07 ± 0.03	0.02 ± 0.01	0.08 ± 0.03	0.01 ± 0.00	0.08 ± 0.04	0.17 ± 0.04
L		0.12 ± 0.01	0.24 ± 0.01	0.12 ± 0.01	0.09 ± 0.03	0.10 ± 0.01	0.16 ± 0.02
M		0.06 ± 0.01	0.06 ± 0.01	0.09 ± 0.03	0.02 ± 0.01	0.06 ± 0.01	0.15 ± 0.04
N	**Ngara**	0.17 ± 0.00	0.24 ± 0.00	0.21 ± 0.01	0.14 ± 0.02	0.15 ± 0.00	0.17 ± 0.03
O		0.20 ± 0.03	0.25 ± 0.00	0.24 ± 0.00	0.18 ± 0.01	0.45 ± 0.01	0.18 ± 0.01
P	**KL**	0.27 ± 0.01	0.39 ± 0.00	0.39 ± 0.01	0.24 ± 0.03	0.61 ± 0.01	0.28 ± 0.05
Q		0.13 ± 0.01	0.24 ± 0.00	0.14 ± 0.01	0.11 ± 0.01	0.10 ± 0.00	0.14 ± 0.01

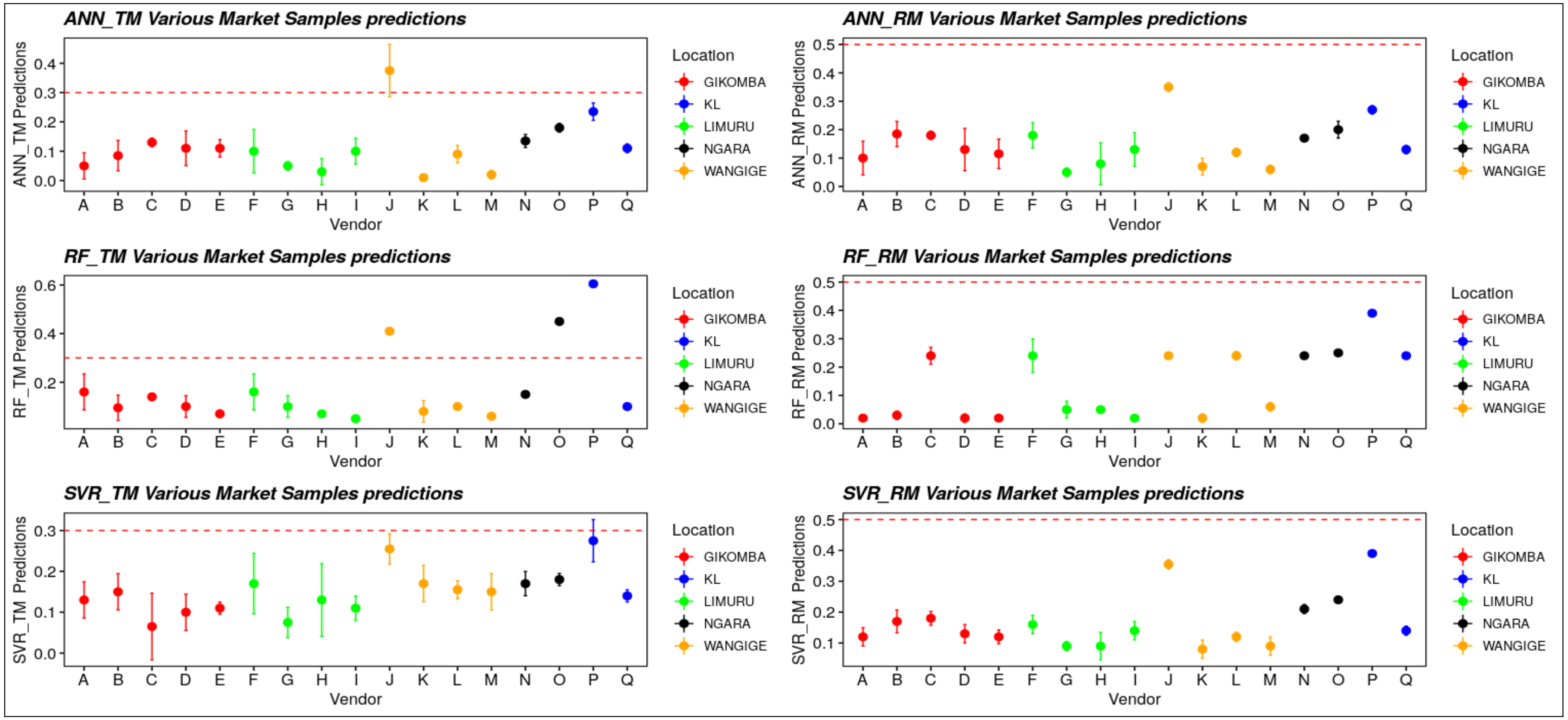


Figure 5.15: The machine learning models that were created were used to predict market samples from five different markets in nearby urban areas. Various fruit batches from various locations were used. Specifically, five came from the Gikomba market, four from Limuru and Wangige, and two from Ngara. To evaluate the model's performance, samples (KL) had known pesticide concentrations. The MRL values of the two pesticides are denoted by the dotted red lines.

Table 5.8: The results of the one-way ANOVA Tukey test for predicted values derived from market samples.

			Mancozeb		**THO**	
Location	**Sample**	**Models**	**T Value**	**P Value**	**T Value**	**P Value**
Gikomba	A	$RF\ -ANN$	−6.6020	< 0.0004 * **	3.5520	0.0036 * *
		$SVR-ANN$	0.890	0.651	4.059	< 0.001 * **
		$SVR-RF$	7.492	< 0.0001 * **	0.507	0.86833
	B	$RF-ANN$	−5.072	< 0.0001 * **	2.084	0.10064
		$SVR-ANN$	−1.005	0.57607	5.264	< 0.0001 * **
		$SVR-RF$	4.067	0.00035 * **	3.180	0.0062 * *
	C	$RF-ANN$	3.604	0.0029 * *	0.646	0.7960
		$SVR-ANN$	0.291	0.9546	−3.644	0.0027 * *
		$SVR-RF$	−3.314	0.00624 * *	−4.290	< 0.001 * **
	D	$RF-ANN$	−5.712	< 0.00001 * **	2.524	0.0363*
		$SVR-ANN$	0.121	0.992	1.133	0.4973
		$SVR-RF$	5.833	< 0.00001 * **	−1.392	0.3506
	E	$RF-ANN$	−8.053	< 0.0001 * **	−1.367	0.3665
		$SVR-ANN$	0.597	0.823	0.894	0.6470
		$SVR-RF$	8.650	< 0.0001 * **	2.261	0.0722
Limuru	F	$RF-ANN$	2.976	0.0107*	3.702	0.0011 * *
		$SVR-ANN$	−0.290	0.9546	4.716	< 0.0001 * **
		$SVR-RF$	−3.266	0.0045 * *	1.014	0.5705
	G	$RF-ANN$	−0.558	0.84281	5.864	< 0.0001 * **
		$SVR-ANN$	3.132	0.0070 * *	3.277	0.0047 * *
		$SVR-RF$	3.690	0.0013 * *	−2.587	0.03119*
	H	$RF-ANN$	−2.867	0.0160*	2.163	0.0868
		$SVR-ANN$	−0.349	0.9352	5.058	< 0.0001 * **
		$SVR-RF$	2.519	0.0386*	2.895	0.0148*
	I	$RF-ANN$	−14.331	0.0001 * **	−3.871	0.0007 * **
		$SVR-ANN$	−0.962	0.603	3.822	0.0008 * **
		$SVR-RF$	13.369	< 0.0001 * **	7.693	< 0.0001 * **
Wangige	J	$RF-ANN$	−7.938	$< 1e-05$ * **	0.671	0.7813
		$SVR-ANN$	0.244	0.968	−4.192	0.0003 * **
		$SVR-RF$	8.182	< 0.00001 * **	−4.863	< 0.0001 * **
	K	$RF-ANN$	−0.692	0.769	5.156	< 0.00001 * **
		$SVR-ANN$	0.7721	0.721	12.141	< 0.00001 * **
		$SVR-RF$	1.465	0.315	6.985	< 0.00001 * **
	L	$RF-ANN$	7.346	< 0.000001 * **	2.709	0.0219*
		$SVR-ANN$	−0.124	0.992	6.757	< 0.001 * **
		$SVR-RF$	−7.470	< 0.000001 * **	4.048	< 0.001 * **
	M	$RF-ANN$	0.914	0.633	2.709	0.0219*
		$SVR-ANN$	1.719	0.205	6.757	< 0.001 * **
		$SVR-RF$	0.805	0.701	4.048	< 0.001 * **
Ngara	N	$RF-ANN$	0.914	0.633	2.709	0.021890*
		$SVR-ANN$	1.719	0.205	6.757	< 0.0001 * **
		$SVR-RF$	0.805	0.701	4.048	0.0003 * **
	O	$RF-ANN$	0.914	0.633	2.709	0.022*
		$SVR-ANN$	1.719	0.205	6.757	< 0.001 * **
		$SVR-RF$	0.805	0.701	4.048	< 0.001 * **
KL	P	$RF-ANN$	18.38	< 0.00001 * **	24.452	< 0.0001 * **
		$SVR-ANN$	18.58	< 0.00001 * **	2.453	0.0571
		$SVR-RF$	0.20	0.978	−21.999	< 0.0001 * **
	Q	$RF-ANN$	31.20	< 0.0001 * **	−0.919	0.634
		$SVR-ANN$	1.93	0.152	0.836	0.685
		$SVR-RF$	−29.27	< 0.0001 * **	1.755	0.206

Signif. codes: 0 ’*’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1**

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6.1 Summary and Conclusions

In summary, this work has encompassed the development of DRS assisted with ML as an alternative or complementary method for screening of pesticides residues in fruits and vegetables. The developed method is rapid, affordable, and non-invasive. Data was acquired from tree tomatoes samples treated with varying Mancozeb and THO concentrations using NIRQuest 512-2.5 and USB4000 spectrometers in diffuse reflectance geometry. The data was de-noised using multiplicative scatter correction and smoothened with a Savitzky-Golay filter. Principal component analysis was performed on the cleaned data and the PCs with more than 98% of the cumulative variation, was used in the development of ML models. The developed ANN, SVR and RF regression models predicted Mancozeb and THO residues with high accuracies ($\boldsymbol{R}^2$ value greater than 92%). The models were tested on a new dataset from market samples. One-way Tukey ANOVA analysis of the predicted values from the market data showed ANN and SVR models to be superior to RF. Therefore, the combination of exploratory techniques such as PCA and machine learning techniques such as PCA, ANN, RF, and SVM with DRS can successfully assess pesticide residues in fruit cuticles. The approach demonstrated high sensitivity and enabled quick analysis.

The method does not require any specific sample preparation procedure resulting in decreased time and costs associated with sample preparation, less time requirement to learn the processes, and reduced errors. These savings in time and money can allow for the simultaneous analysis of several residues with improved performance and adaptability to many classes of pesticides. As a result, the method is relevant for various foods and would be excellent for regulatory measurements.

6.2 Recommendations and Future Prospects

This study was limited to four machine learning techniques, but more can be incorporated into future work to improve the robustness, efficiency, and repeatability of the results. Consideration for the future could be the use of fruits from various geographical locations into the development phase of the models. The development of an open-source database of residues could be implemented using this method to enhance this work's usability. New portable sensor technologies are likely to be used for residue screening, including Raman spectroscopy, Terahertz radiation, X-ray, hyper-spectral imaging, and magnetic resonance. The developed method was used for screening pesticide residues on the cuticles. However, it can have a spin-off to correlate the measurements on the skin and the flesh. Also, conventional techniques can be used to verify the predictions made by the method to compare specificity, sensitivity, and LOD levels.

REFERENCES

Abbott, J. A. Quality measurement of fruits and vegetables. *Postharvest biology and technology*, **15(3)**:207–225 (1999).

Abong'o, D., Wandiga, S., and Jumba, I. Occurrence and distribution of organochlorine pesticide residue levels in water, sediment and aquatic weeds in the nyando river catchment, lake victoria, kenya. *Afr J Aquat Sci*, **43(3)**:255–270 (2018).

Acharya, U. K., Subedi, P. P., and Walsh, K. B. Evaluation of a dry extract system involving nir spectroscopy (desir) for rapid assessment of pesticide contamination of fruit surfaces. *American Journal of Analytical Chemistry*, **3(8)**:524 (2012).

Acharya, U., Subedi, P., and Walsh, K. Spectrophotometer aging and prediction of total soluble solids. In *XXIX International Horticultural Congress on Horticulture: Sustaining Lives, Livelihoods and Landscapes (IHC2014): 1119*, pages 209–212 (2014).

Afara, I. O. *Near infrared spectroscopy for non-destructive evaluation of articular cartilage*. Ph.D. thesis, Queensland University of Technology (2012).

Anastassiades, M., Maštovská, K., and Lehotay, S. J. Evaluation of analyte protectants to improve gas chromatographic analysis of pesticides. *J Chromatogr A*, **1015(1-2)**:163–184 (2003).

Arias, N., Arazuri, S., and Jarén, C. Ability of nirs technology to determine pesticides in liquid samples at maximum residue levels. *Pest Manag Sci*, **69(4)**:471–477 (2013).

Armenta, S., Garrigues, S., and de la Guardia, M. Partial least squares-near infrared determination of pesticides in commercial formulations. *Vib Spectrosc*, **44(2)**:273–278 (2007).

Atkinson, P. M. and Tatnall, A. R. Introduction neural networks in remote sensing. *International Journal of remote sensing*, **18(4)**:699–709 (1997).

Bai, Z., Ye, Y., Liang, B., Xu, F., Zhang, H., Zhang, Y., Peng, J., Shen, D., Cui, Z., Zhang, Z., *et al.* Proteomics-based identification of a group of apoptosis-related proteins and biomarkers in gastric cancer. *Int J Oncol*, **38(2)**:375–383 (2011).

Bakeev, K. *Process Analytical Technology: Spectroscopic Tools and Implementation Strategies for the Chemical and Pharmaceutical Industries, 2nd Edition* (2010). ISBN 978-0-470-72207-7.

Beć, K. B., Grabska, J., Ozaki, Y., Czarnecki, M. A., and Huck, C. W. Simulated nir spectra as sensitive markers of the structure and interactions in nucleobases. *Scientific reports*, **9(1)**:1–13 (2019).

Bishop, C. M. *et al. Neural networks for pattern recognition*. Oxford university press (1995).

Blanco, M. and Villarroya, I. Nir spectroscopy: a rapid-response analytical tool. *TrAC Trends in Analytical Chemistry*, **21(4)**:240–250 (2002).

Blasco, C., Picó, Y., Manes, J., and Font, G. Determination of fungicide residues in fruits and vegetables by liquid chromatography–atmospheric pressure chemical ionization mass spectrometry. *J Chromatogr A*, **947(2)**:227–235 (2002).

Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **5(5)**:216–233 (2015).

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press (1992).

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and regression trees.* Routledge (2017).

Brunet, D., Woignier, T., Lesueur-Jannoyer, M., Achard, R., Rangon, L., and Barthes, B. G. Determination of soil content in chlordecone (organochlorine pesticide) using near infrared reflectance spectroscopy (nirs). *Environ Pollut*, **157(11)**:3120–3125 (2009).

Campanella, L., Bonanni, A., Martini, E., Todini, N., and Tomassetti, M. Determination of triazine pesticides using a new enzyme inhibition tyrosinase opee operating in chloroform. *Sens Actuators, B*, **111**:505–514 (2005).

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168 (2006).

Chalmers, J. and Griffiths, P. *Handbook of Vibrational Spectroscopy, 5 volumes set.* Wiley (2002).

Chen, J., Peng, Y., Li, Y., Wang, W., and Wu, J. A method for determining organophosphorus pesticide concentration based on near-infrared spectroscopy. *Trans ASABE*, **54(3)**:1025–1030 (2011).

Chen, C. and Ramaswamy, H. A neuro-computing approach for modeling of residence time distribution (rtd) of carrot cubes in a vertical scraped surface heat exchanger (sshe). *Food research international*, **33(7)**:549–556 (2000).

Cheng, C., Liu, J., Zhang, C., Cai, M., Wang, H., and Xiong, W. An overview of infrared spectroscopy based on continuous wavelet transform combined with machine learning algorithms: application to chinese medicines, plant classification, and cancer diagnosis. *Appl Spectrosc Rev*, **45(2)**:148–164 (2010).

Cortés, V., Blasco, J., Aleixos, N., Cubero, S., and Talens, P. Monitoring strategies for quality control of agricultural products using visible and near-infrared spectroscopy: A review. *Trends in food science & technology* (2019).

Cortés López, V. *Innovations in non-destructive techniques for fruit quality control applied to manipulation and inspection lines.* Ph.D. thesis (2018).

Cunha, S., Cunha, C., Ferreira, A., and Fernandes, J. Determination of bisphenol a and bisphenol b in canned seafood combining quechers extraction with dispersive liquid–liquid microextraction followed by gas chromatography–mass spectrometry. *Analytical and bioanalytical chemistry*, **404(8)**:2453–2463 (2012).

Cunha, S. C., Fernandes, J., and Oliveira, M. Current trends in liquid-liquid microextraction for analysis of pesticide residues in food and water. *Pesticides-Strategies for pesticides analysis. InTech*, pages p1–26 (2011).

Dahlstrand, U., Sheikh, R., Ansson, C. D., Memarzadeh, K., Reistad, N., and Malmsjö, M. Extended-wavelength diffuse reflectance spectroscopy with a machine-learning method for in vivo tissue classification. *PLoS One*, **14(10)** (2019).

Dasgupta, S., Meisner, C., Wheeler, D., Xuyen, K., and Lam, N. T. Pesticide poisoning of farm workers–implications of blood test results from vietnam. *International journal of hygiene and environmental health*, **210(2)**:121–132 (2007).

Davies, A. William herschel and the discovery of near infrared energy. *NIR news*, **11(2)**:3–5 (2000).

Davies, J. N., Hobson, G. E., and McGlasson, W. The constituents of tomato fruit the influence of environment, nutrition, and genotype. *Critical Reviews in Food Science & Nutrition*, **15(3)**:205–280 (1981).

De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. The Mahalanobis Distance. *Chemom Intell Lab Syst*, **50(1)**:1–18 (2000).

Diabaté, D., Gnago, J. A., and Tano, Y. Toxicity, antifeedant and repellent effect of azadirachta indica (a. juss) and jatropha carcusl. aqueous extracts against plutella xylostella (lepidoptera: Plutellidae). *Journal of Basic and Applied Scientific Research*, **4(11)**:51–60 (2014).

Fernández, V., Guzmán-Delgado, P., Graça, J., Santos, S., and Gil, L. Cuticle structure in relation to chemical composition: re-assessing the prevailing model. *Frontiers in Plant Science*, **7**:427 (2016).

Ghosh, K., Stuke, A., Todorović, M., Jørgensen, P. B., Schmidt, M. N., Vehtari, A., Rinke, P., *et al.* Deep learning spectroscopy (2019).

GOK. *âĂIJBig 4 Will Create Jobs for You, President Kenyatta Assures the Youth on August 12, 2018 In Latest NewsâĂİ(accessed January 1, 2021)* (2018).

González, Rodríguez, R. M., Rial-Otero, R., Cancho-Grande, B., and Simal-Gándara, J. Occurrence of fungicide and insecticide residues in trade samples of leafy vegetables. *Food Chem*, **107(3)**:1342–1347 (2008).

Greenlife. Thiocyclam hydrogen oxalate taurus. `https://www.greenlife.co.ke/taurus-500sp/` (2020). Accessed: 2020-04-04.

Gritti, F., Sanchez, C. A., Farkas, T., and Guiochon, G. Achieving the full performance of highly efficient columns by optimizing conventional benchmark high-performance liquid chromatography instruments. *J Chromatogr A*, **1217(18)**:3000–3012 (2010).

Gupta, P. Pesticide exposure-indian scene. *Toxicology*, **198(1-3)**:83–90 (2004).

Guthrie, J., Liebenberg, C., and Walsh, K. B. Nir model development and robustness in prediction of melon fruit total soluble solids. *Australian Journal of Agricultural Research*, **57(4)**:411–418 (2006).

Hadian, Z., Samira, S., and Yazdanpanah, H. Pesticide residues analysis in iranian fruits and vegetables by gas chromatography-mass spectrometry. *Iranian journal of pharmaceutical research: IJPR*, **18(1)**:275 (2019).

Hanson, B. A. *ChemoSpec: Exploratory Chemometrics for Spectroscopy* (2016). R package version 4.3.17.

Harlan, H. V. The origin of cultivated plants. *J Hered*, **19(4)**:167–168 (1928).

Hart, A., Collier, W., and Janssen, D. The response of screen-printed enzyme electrodes containing cholinesterases to organo-phosphates in solution and from commercial formulations. *Biosens Bioelectron*, **12(7)**:645–654 (1997).

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media (2009).

He, Y., Li, X., and Deng, X. Discrimination of varieties of tea using near infrared spectroscopy by principal component analysis and bp model. *Journal of food Engineering*, **79(4)**:1238–1242 (2007).

Heredia-Guerrero, J. A., Benítez, J. J., Domínguez, E., Bayer, I. S., Cingolani, R., Athanassiou, A., and Heredia, A. Infrared and raman spectroscopic features of plant cuticles: a review. *Front Plant Sci*, **5**:305 (2014).

Hernández-Borges, J., Cabrera, J. C., Rodríguez-Delgado, M. Á., Hernández-Suárez, E. M., and Saúco, V. G. Analysis of pesticide residues in bananas harvested in the canary islands (spain). *Food Chem*, **113(1)**:313–319 (2009).

Hiroaki, I., Toyonori, N., and Eiji, T. Measurement of pesticide residues in food based on diffuse reflectance ir spectroscopy. *IEEE transactions on instrumentation and measurement*, **51(5)**:886–890 (2002).

Ho, Y.-C. and Pepyne, D. L. Simple explanation of the no-free-lunch theorem and its implications. *Journal of optimization theory and applications*, **115(3)**:549–570 (2002).

Holmes, G., Fletcher, D., and Reutemann, P. An application of data mining to fruit and vegetable sample identification using gas chromatography-mass spectrometry. iEMSs (2012).

Hotelling, H. The generalization of Student's ratio. *Annals of Mathematical Statistics*, **2**:360–378 (1931).

Howley, T., Madden, M. G., O'Connell, M.-L., and Ryder, A. G. The effect of principal component analysis on machine learning accuracy with high dimensional spectral data. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 209–222. Springer (2005).

Hssina, B., Merbouha, A., Ezzikouri, H., and Erritali, M. A comparative study of decision tree id3 and c4.5. *International Journal of Advanced Computer Science and Applications*, **4(2)** (2014).

Hu, B., Sun, D.-W., Pu, H., and Wei, Q. Rapid nondestructive detection of mixed pesticides residues on fruit surface using sers combined with self-modeling mixture analysis method. *Talanta*, page 120998 (2020).

Islam, M. A., Ullah, A., Habib, M., Chowdhury, M. T. I., Sirajul, M., Khan, I., Kaium, A., and Prodhan, M. D. H. Determination of major organophosphate insecticide residues in cabbage samples from different markets of dhaka. *Asia Pacific Environmental and Occupational Health Journal*, **5 (2)**:30 – 35 (2019). ISSN 2462 -2214.

Jamshidi, B., Mohajerani, E., and Jamshidi, J. Developing a vis/nir spectroscopic system for fast and non-destructive pesticide residue monitoring in agricultural product. *Measurement*, **89**:1–6 (2016).

Jolliffe, I. T. Choosing a subset of principal components or variables. *Principal component analysis*, pages 111–149 (2002).

Jolliffe, I. Principal component analysis. *Encyclopedia of statistics in behavioral science* (2005).

Kabano, G. *Antecedents of Standards Compliance for the Internationalization of Kenyan Horticulture*. Ph.D. thesis, United States International University-Africa (2018).

Kachrimanis, K., Rontogianni, M., and Malamataris, S. Simultaneous quantitative analysis of mebendazole polymorphs a–c in powder mixtures by drifts spectroscopy and ann modeling. *Journal of pharmaceutical and biomedical analysis*, **51(3)**:512–520 (2010).

Kaczyński, P., Rutkowska, E., Jankowska, M., Hrynko, I., *et al.* Evaluation of pesticide residues in fruit from poland and health risk assessment. *Agricultural Sciences*, **4(05)**:106 (2013).

Kara, S. and Dirgenali, F. A system to diagnose atherosclerosis via wavelet transforms, principal component analysis and artificial neural networks. *Expert Syst Appl*, **32(2)**:632–640 (2007).

Khandekar, S., Noronha, A., and Banerji, S. Organochlorine pesticide residues in vegetables from bombay markets: a three year assessment. *Environmental Pollution Series B, Chemical and Physical*, **4(2)**:127–134 (1982).

Khanmohammadi, M., Armenta, S., Garrigues, S., and de la Guardia, M. Mid- and near-infrared determination of metribuzin in agrochemicals. *Vib Spectrosc*, **46(2)**:82–88 (2008).

Kjeldahl, K. and Bro, R. Some common misunderstandings in chemometrics. *J Chemom*, **24(7-8)**:558–564 (2010).

Klunklin, W. and Savage, G. P. Effect on quality characteristics of tomatoes grown under well-watered and drought stress conditions. *Foods*, **6(8)**:56 (2017).

Kuhn, M. Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, **28(5)**:1–26 (2008). ISSN 1548-7660.

Kuhn, M., Johnson, K., *et al. Applied predictive modeling*, volume 26. Springer (2013).

Kunyanga, C., Amimo, J., Njue, L., and Chemining'wa, G. Consumer risk exposure to chemical and microbial hazards through consumption of fruits and vegetables in kenya (2018).

LeDoux, M. Analytical methods applied to the determination of pesticide residues in foods of animal origin. a review of the past two decades. *J Chromatogr A*, **1218(8)**:1021–1036 (2011).

Lehotay, S. J. Determination of pesticide residues in nonfatty fooda by percritical extraction aqnd gas chromatography/mass spectrometry: Collaborative study. *J AOAC Int*, **85(5)**:1148–1166 (2002).

Li, X. and He, Y. A novel approach to pattern recognition based on pca-ann in spectroscopy. In X. Li, O. R. Zaïane, and Z. Li, editors, *Advanced Data Mining and Applications*, pages 525–532. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). ISBN 978-3-540-37026-0.

Li, X., Yang, T., Li, S., Wang, D., Song, Y., and Yu, K. Different classification algorithms and serum surface enhanced raman spectroscopy for noninvasive discrimination of gastric diseases. *Journal of Raman Spectroscopy*, **47(8)**:917–925 (2016).

Liakos, K., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. Machine learning in agriculture: A review. *Sensors*, **18(8)**:2674 (2018).

LIU, C.-l., SUI, S.-x., SUN, X.-r., and WU, J.-z. Quantitative analysis of chlorpyrifos residue in spinach by nir. *Food Science*, **7** (2008).

Liu, L., Wang, Y., Gao, C., Huan, H., Zhao, B., and Yan, L. Photoacoustic spectroscopy as a non-destructive tool for quantification of pesticide residue in apple cuticle. *Int J Thermophys*, **36(5-6)**:868–872 (2015).

Loha, K. M., Lamoree, M., Weiss, J. M., and de Boer, J. Import, disposal, and health impacts of pesticides in the east africa rift (ear) zone: A review on management and policy analysis. *Crop Protection*, **112**:322–331 (2018).

Lv, G., Du, C., Ma, F., Shen, Y., and Zhou, J. Rapid and nondestructive detection of pesticide residues by depth-profiling fourier transform infrared photoacoustic spectroscopy. *ACS Omega*, **3(3)**:3548–3553 (2018).

Makio, T., Hiroaki, I., Tomohiro, T., Hisaya, Y., Kumiko, N., and Nobuaki, T. Classification of pesticide residues in the agricultural products based on diffuse reflectance ir spectroscopy. In *SICE Annual Conference 2007*, pages 216–219. IEEE (2007).

Mangels, A. R., Holden, J. M., Beecher, G. R., Forman, M. R., and Lanza, E. Carotenoid content of fruits and vegetables: an evaluation of analytic data. *J Am Diet Assoc*, **93(3)**:284–296 (1993).

Mattern, G. C., Singer, G. M., Louis, J., Robson, M., and Rosen, J. D. Determination of several pesticides with a chemical ionization trap detector. *J Agric Food Chem*, **38(2)**:402–407 (1990).

Mebdoua, S. Pesticide residues in fruits and vegetables. *Bioactive Molecules in Food*, pages 1–39 (2018).

Mebdoua, S. *Pesticide Residues in Fruits and Vegetables*, pages 1715–1753. Springer International Publishing, Cham (2019). ISBN 978-3-319-78030-6.

Mebdoua, S., Lazali, M., Ounane, S. M., Tellah, S., Nabi, F., and Ounane, G. Evaluation of pesticide residues in fruits and vegetables from algeria. *Food Additives & Contaminants: Part B*, **10(2)**:91–98 (2017).

Melki, G., Cano, A., Kecman, V., and Ventura, S. Multi-target support vector regression via correlation regressor chains. *Information Sciences*, **415**:53–69 (2017).

Metrohm. A guide to near-infrared spectroscopic analysis nir chart. `https://www.metrohm.com/en/documents/81085026` (2021). Accessed: 2021-01-01.

Michel, M. and Buszewski, B. Determination of carbendazim residues in fruits, vegetables and cereals by hplc with column switching. *Journal of Plant Protection Research (Poland)* (2002).

Michie, D., Spiegelhalter, D. J., Taylor, C., *et al.* Machine learning. *Neural and Statistical Classification*, **13** (1994).

Mirjalili, S., Mirjalili, S. M., and Lewis, A. Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences*, **269**:188–209 (2014).

Montana State University. Pesticides compatibility tests. `https://pesticides.montana.edu/reference/compatibility.html#:~:text=Measure%20one%20pint%20of%20spray,using%20the%20%22D%2DA%2DL%2DE%2DS%22%20plan.` (2020). Accessed: 2020-04-04.

Moros, J., Armenta, S., Garrigues, S., and de la Guardia, M. Comparison of two vibrational procedures for the direct determination of mancozeb in agrochemicals. *Talanta*, **72(1)**:72–79 (2007).

Nakamura, Y., Tonogai, Y., Sekiguchi, Y., Tsumura, Y., Nishida, N., Takakura, K., Isechi, M., Yuasa, K., and Nakamura, M. Multiresidue analysis of 48 pesticides in agricultural products by capillary gas chromatography. *J Agric Food Chem*, **42(11)**:2508–2518 (1994).

Nicolaï, B. M., Verlinden, B. E., Desmet, M., Saevels, S., Saeys, W., Theron, K., Cubeddu, R., Pifferi, A., and Torricelli, A. Time-resolved and continuous wave nir reflectance spectroscopy to predict soluble solids content and firmness of pear. *Postharvest Biology and Technology*, **47(1)**:68–74 (2008).

Oloo, J. Food safety and quality management in kenya: An overview of the roles played by various stakeholders. *African Journal of Food, Agriculture, Nutrition and Development*, **10(11)** (2010).

Osborne, B. G. Near-infrared spectroscopy in food analysis. *Encyclopedia of analytical chemistry: applications, theory and instrumentation* (2006).

Ozaki, Y., McClure, W. F., and Christy, A. A. *Near-infrared spectroscopy in food science and technology.* John Wiley & Sons (2006).

Papadonikolakis, M. and Bouganis, C.-S. Novel cascade fpga accelerator for support vector machines classification. *IEEE transactions on neural networks and learning systems*, **23(7)**:1040–1052 (2012).

Papadopoulou-Mourkidou, E. and Patsias, J. Development of a semi-automated high-performance liquid chromatographic-diode array detection system for screening pesticides at trace levels in aquatic systems of the axios river basin. *J Chromatogr A*, **726(1-2)**:99–113 (1996).

Peirs, A., Scheerlinck, N., and NicolaıÌĹ, B. M. Temperature compensation for near infrared reflectance measurement of apple fruit soluble solids contents. *Postharvest Biology and Technology*, **30(3)**:233–248 (2003).

Podhorniak, L. V., Negron, J. F., and Griffith, F. D. Gas chromatography with pulsed flame photometric detection multiresidue method for organophosphate pesticide and metabolite residues at the parts-per-billion level in representative commodities of fruit and vegetable crop groups. *J AOAC Int*, **84(3)**:873–890 (2001).

Popelínskỳ, L. Combining the principal components method with different learning algorithms. In *In: Proc. of ECML/PKDD IDDM Workshop (Integrating Aspects of Data Mining, Decision Support and Meta-Learning.(2001.* Citeseer (2000).

Priyadarshini, K. N., Sivashankari, V., Shekhar, S., and Balasubramani, K. Comparison and evaluation of dimensionality reduction techniques for hyperspectral data analysis. *Proceedings*, **24(1)** (2019). ISSN 2504-3900.

Qian, G., Wang, L., Wu, Y., Zhang, Q., Sun, Q., Liu, Y., and Liu, F. A monoclonal antibody-based sensitive enzyme-linked immunosorbent assay (elisa) for the analysis of the organophosphorous pesticides chlorpyrifos-methyl in real samples. *Food Chem*, **117(2)**:364–370 (2009).

Quintás, G., Armenta, S., Garrigues, S., and de la Guardia, M. Towards minimization of chlorinated solvents consume in fourier transform infrared spectroscopy determination of propamocarb in pesticide formulations. *Talanta*, **75(2)**:339–343 (2008).

Randhawa, M. A., Anjum, F. M., Ahmed, A., and Randhawa, M. S. Field incurred chlorpyrifos and 3, 5, 6-trichloro-2-pyridinol residues in fresh and processed vegetables. *Food Chem*, **103(3)**:1016–1023 (2007).

Rashid, A., Nawaz, S., Barker, H., Ahmad, I., and Ashraf, M. Development of a simple extraction and clean-up procedure for determination of organochlorine pesticides in soil using gas chromatography–tandem mass spectrometry. *Journal of Chromatography A*, **1217(17)**:2933–2939 (2010).

Ripley, B. D. *Pattern recognition and neural networks.* Cambridge university press (2007).

Rodriguez-Campos, J., Escalona-Buendía, H., Orozco-Avila, I., Lugo-Cervantes, E., and Jaramillo-Flores, M. E. Dynamics of volatile and non-volatile compounds in cocoa (theobroma cacao l.) during fermentation and drying processes using principal components analysis. *Food Research International*, **44(1)**:250–258 (2011).

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., and Chica-Rivas, M. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, **71**:804–818 (2015).

Rokach, L. and Maimon, O. Decision trees. In *Data mining and knowledge discovery handbook*, pages 165–192. Springer (2005).

Román, R. C., Hernández, O. G., and Urtubia, U. A. Prediction of problematic wine fermentations using artificial neural networks. *Bioprocess Biosyst Eng*, **34(9)**:1057–1065 (2011).

Roy, R. R., Wilson, P., Laski, R. R., Roberts, J. I., Weishaar, J. A., Bong, R. L., and Yess, N. J. Monitoring of domestic and imported apples and rice by the us food and drug administration pesticide program. *J AOAC Int*, **80(4)**:883–894 (1997).

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science (1985).

Salami, A., Kamara, A. B., and Brixiova, Z. *Smallholder agriculture in East Africa: Trends, constraints and opportunities.* African Development Bank Tunis, Tunisia (2017).

Sánchez, M.-T., Flores-Rojas, K., Guerrero, J. E., Garrido-Varo, A., and Pérez-Marín, D. Measurement of pesticide residues in peppers by near-infrared reflectance spectroscopy. *Pest Management Science: formerly Pesticide Science*, **66(6)**:580–586 (2010).

Saranwong, I. and Kawano, S. Rapid determination of fungicide contaminated on tomato surfaces using the desir-nir: a system for ppm-order concentration. *J Near Infrared Spectrosc*, **13(3)**:169–175 (2005).

Schaepman, M. E., Jehle, M., Hueni, A., D'Odorico, P., Damm, A., Weyermann, J., Schneider, F. D., Laurent, V., Popp, C., Seidel, F. C., *et al.* Advanced radiometry measurements and earth science applications with the airborne prism experiment (apex). *Remote Sens Environ*, **158**:207–219 (2015).

Seeger, M. Learning with labeled and unlabeled data. Technical report (2000).

Shao, Y., Li, Y., Jiang, L., Pan, J., He, Y., and Dou, X. Identification of pesticide varieties by detecting characteristics of chlorella pyrenoidosa using visible/near infrared hyperspectral imaging and raman microspectroscopy technology. *Water Res*, **104**:432–440 (2016).

Shaw, G. A. and Burke, H. K. Spectral imaging for remote sensing. *Lincoln laboratory journal*, **14(1)**:3–28 (2003).

Shen, F. *et al.* Application of near-infrared spectroscopy to detection of pesticide phoxim residues. *Spectroscopy and Spectral Analysis*, **29(9)**:2421–2424 (2009).

Shepherd, K. D. and Walsh, M. G. Development of reflectance spectral libraries for characterization of soil properties. *Soil Sci Soc Am J*, **66(3)**:988–998 (2002).

Shrivastava, A., Gupta, V. B., *et al.* Methods for the determination of limit of detection and limit of quantitation of the analytical methods. *Chronicles of young scientists*, **2(1)**:21 (2011).

Sigurdsson, S., Philipsen, P., Hansen, L., Larsen, J., Gniadecka, M., and Wulf, H. Detection of skin cancer by classification of raman spectra. *IEEE Transactions on Biomedical Engineering*, **51(10)**:1784–1793 (2004).

Sila, A. M., Shepherd, K. D., and Pokhariyal, G. P. Evaluating the utility of mid-infrared spectral subspaces for predicting soil properties. *Chemometrics and Intelligent Laboratory Systems*, **153**:92–105 (2016).

Soares, S. F. C., Gomes, A. A., Araujo, M. C. U., Galvão Filho, A. R., and Galvão, R. K. H. The successive projections algorithm. *TrAC Trends in Analytical Chemistry*, **42**:84–98 (2013).

Stachniuk, A. and Fornal, E. Liquid chromatography-mass spectrometry in the analysis of pesticide residues in food. *Food Analytical Methods*, **9(6)**:1654–1665 (2016).

Stafford, S. C. and Lin, W. Determination of oxamyl and methomyl by high-performance liquid chromatography using a single-stage postcolumn derivatization reaction and fluorescence detection. *J Agric Food Chem*, **40(6)**:1026–1029 (1992).

Stenberg, B., Viscarra Rossel, R., Mouazen, A., and Wetterlind, J. Chapter five-visible and near infrared spectroscopy. *Soil Science Advances in Agronomy*, **107**:163–215 (2010).

Stuart, B. H. Infrared spectroscopy of biological applications. *Encyclopedia of Analytical Chemistry: Applications, Theory and Instrumentation* (2006).

Sun, J., Cong, S., Mao, H., Wu, X., and Yang, N. Quantitative detection of mixed pesticide residue of lettuce leaves based on hyperspectral technique. *J Food Process Eng*, **41(2)**:e12654 (2018).

Sun, J., Zhang, M., Mao, H., Yang, Z. L., and Wu, X. Identification of pesticide residues on mulberry leaves based on hyperspectral imaging. *Transactions of the Chinese Society for Agricultural Machinery*, **46(6)**:251–256 (2015).

Syngenta. Metalaxyl M Mancozeb taurus. `https://www.syngenta.co.ke/product/crop-protection/ridomil-gold-mz-68-wg` (2020). Accessed: 2020-04-04.

Teye, E., Huang, X.-y., and Afoakwa, N. Review on the potential use of near infrared spectroscopy (nirs) for the measurement of chemical residues in food. *Am. J. Food Sci. Technol*, **1**:1–8 (2013).

Thakur, B. R., Singh, R. K., and Nelson, P. E. Quality attributes of processed tomato products: A review. *Food Reviews International*, **12(3)**:375–401 (1996).

Toldrá, F. and Reig, M. Methods for rapid detection of chemical and veterinary drug residues in animal foods. *Trends in Food Science & Technology*, **17(9)**:482–489 (2006).

Tripathi, S. and Mishra, H. A rapid ft-nir method for estimation of aflatoxin b1 in red chili powder. *Food Control*, **20(9)**:840–846 (2009).

Tsagkaris, A. S., Nelis, J. L., Ross, G., Jafari, S., Guercetti, J., Kopper, K., Zhao, Y., Rafferty, K., Salvador, J. P., Migliorelli, D., *et al.* Critical assessment of recent trends related to screening and confirmatory analytical methods for selected food contaminants and allergens. *TrAC Trends in Analytical Chemistry*, **121**:115688 (2019).

Tsimbiri, P. F., Moturi, W. N., Sawe, J., Henley, P., and Bend, J. R. Health impact of pesticides on residents and horticultural workers in the lake naivasha region, kenya. *Occupational Diseases and Environmental Medicine*, **3(02)**:24 (2015).

Türker-Kaya, S. and Huck, C. A review of mid-infrared and near-infrared imaging: principles, concepts and applications in plant tissue analysis. *Molecules*, **22(1)**:168 (2017).

Umesh, Kumar, A., Phul Prasad, S., and Kerry Brian, W. Evaluation of a dry extract system involving nir spectroscopy (desir) for rapid assessment of pesticide contamination of fruit surfaces. *American Journal of Analytical Chemistry*, **2012** (2012).

Valenzuela, A. I., Pico, Y., and Font, G. Determination of five pesticide residues in oranges by matrix solid-phase dispersion and liquid chromatography to estimate daily intake of consumers. *J AOAC Int*, **84(3)**:901–909 (2001).

Varmuza, K. and Filzmoser, P. *Introduction to Multivariate Statistical Analysis in Chemometrics* (2009).

Villmann, T., MerÃĺnyi, E., and Seiffert, U. Machine learning approches and pattern recognition for spectral data. In *ESANN*, pages 433–444 (2008).

Wang, C., Venkatesh, S. S., and Judd, J. S. Optimal stopping and effective machine complexity in learning. In *Advances in neural information processing systems*, pages 303–310 (1994).

Wang, S., Zhu, H., Ge, Y., and Qiao, H. Current status and management of chemical residues in food and ingredients in china. *Trends in food science & technology*, **20(9)**:425–434 (2009).

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometric Intelligent Laboratory Systems*, **2**:37–52 (1987).

Wolpert, D. H. and Macready, W. G. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, **1(1)**:67–82 (1997).

Workman Jr, J. *The Handbook of Organic Compounds, Three-Volume Set: NIR, IR, R, and UV-Vis Spectra Featuring Polymers and Surfactants.* Elsevier (2000).

Worley, B., Halouska, S., and Powers, R. Utilities for quantifying separation in PCA/PLS-DA scores plots. *Anal Biochem*, **433(2)**:102–104 (2013).

Worley, B. and Powers, R. PCA as a predictor of OPLS-DA model reliability. *Anal Chim Acta* (2015).

Wu, L.-C., Chen, H.-H., Horng, J.-T., Lin, C., Huang, N. E., Cheng, Y.-C., and Cheng, K.-F. A novel preprocessing method using hilbert huang transform for maldi-tof and seldi-tof mass spectrometry data. *PLoS One*, **5(8)**:e12493 (2010).

Wu, W. and Massart, D. Artificial neural networks in classification of nir spectral data: Selection of the input. *Chemometrics and Intelligent Laboratory Systems*, **35(1)**:127–135 (1996).

Wu, W., Massart, D., and De Jong, S. The kernel pca algorithms for wide data. part i: theory and algorithms. *Chemometrics and Intelligent Laboratory Systems*, **36(2)**:165–172 (1997).

XIONG, Y.-m., WANG, D., DUAN, J., *et al.* Study on rapid determination of active ingredient of agrochemicals by near-infrared spectroscopy. *Spectroscopy and Spectral Analysis*, **30(6)**:1488–1492 (2010).

Yoshii, K., Kaihara, A., Tsumura, Y., Ishimitsu, S., and Tonogai, Y. Simultaneous determination of residues of emamectin and its metabolites, and milbemectin, ivermectin, and abamectin in crops by liquid chromatography with fluorescence detection. *J AOAC Int*, **84(3)**:910–917 (2001).

Zapata, F., Ferreiro-González, M., and García-Ruiz, C. Interpreting the near infrared region of explosives. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, **204**:81–87 (2018).

Zhan-qi, R., Zhen-hong, R., and Hai-yan, J. Identification of different concentrations pesticide residues of dimethoate on spinach leaves by hyperspectral image technology. *IFAC-PapersOnLine*, **51(17)**:758–763 (2018).

Zhou, L., Ma, W., Zhang, H., Li, L., and Tang, L. Developing a pca–ann model for predicting chlorophyll a concentration from field hyperspectral measurements in dianshan lake, china. *Water Quality, Exposure and Health*, **7(4)**:591–602 (2015).

Zude, M. *Optical monitoring of fresh and processed agricultural crops.* CRC press (2008).

APPENDIX ONE

Images of Laboratory Based Methodology

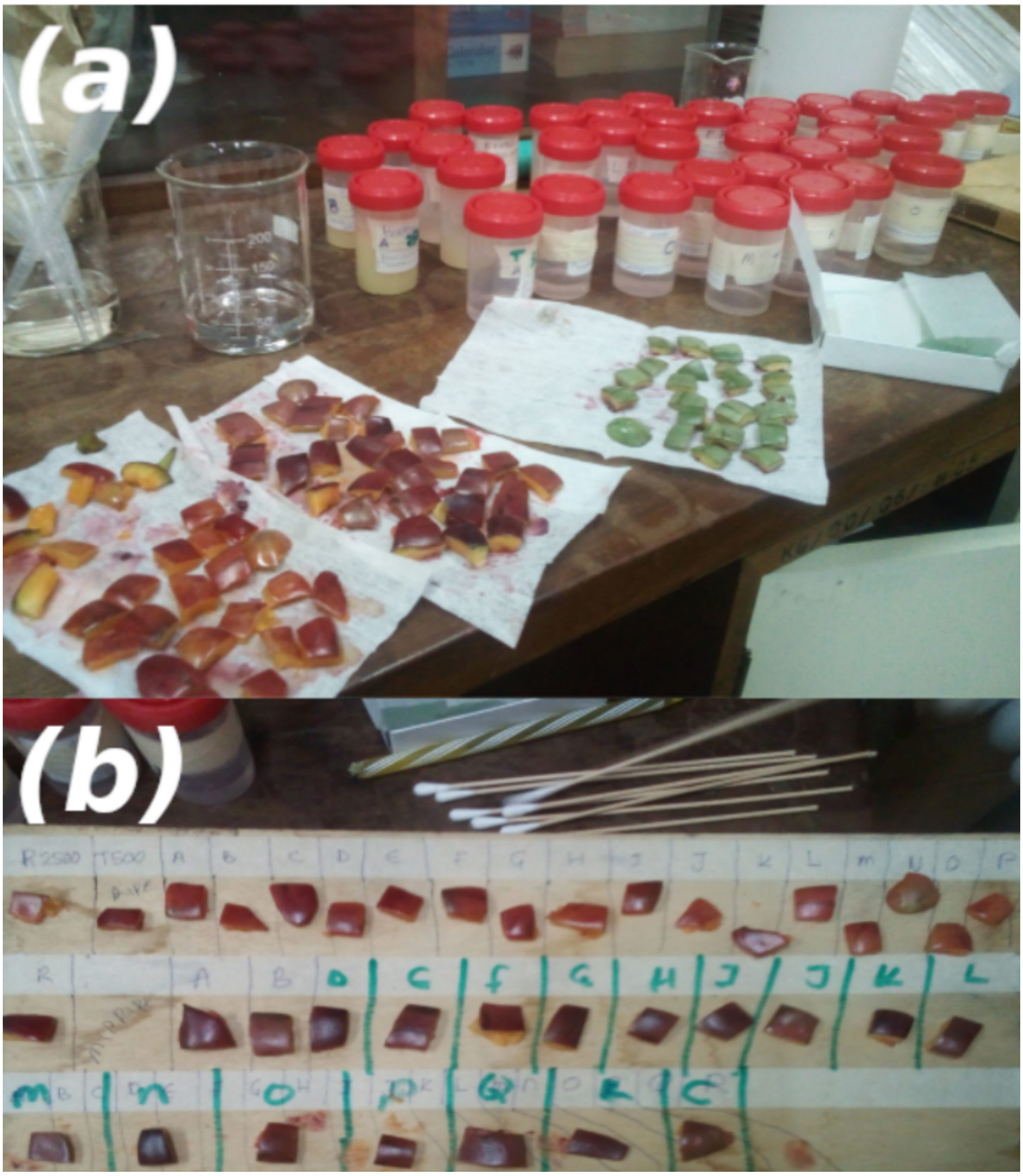


Figure A.1: Picture (a) shows the prepared pesticides concentrations in vials and the tree tomatoes samples. Figure (b) shows the samples labeling before being sprayed with varying concentrations of the pesticides.

APPENDIX TWO
Images of Field Based Methodology

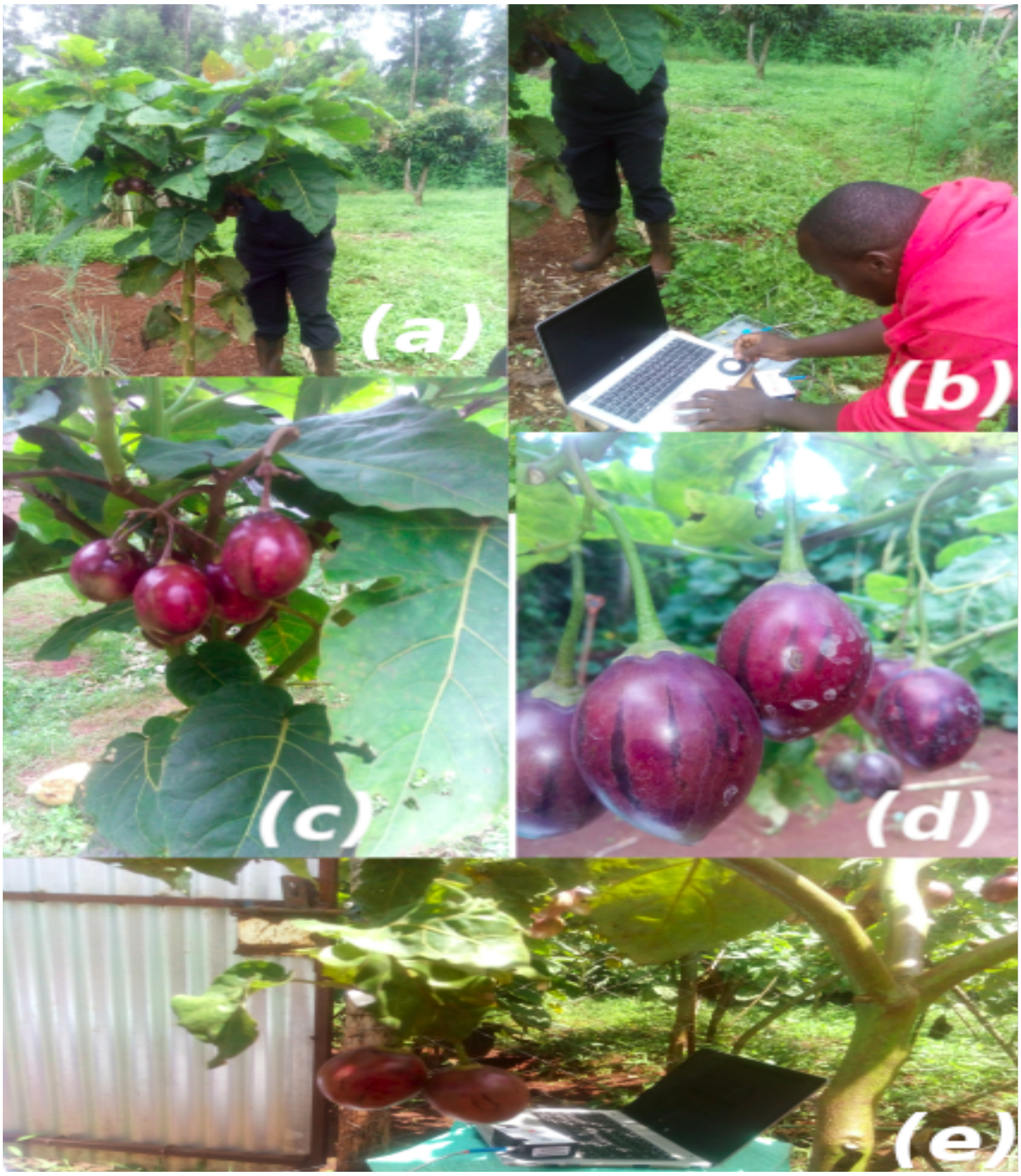
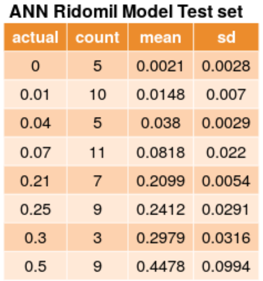


Figure B.1: The tree tomato plants utilized for field analysis are shown in image (a). The system is calibrated using a white reflectance reference, as shown in image (b). Images (c) and (d) show pesticide-sprayed tree tomatoes one hour later. The experimental setup is depicted in the image (e).

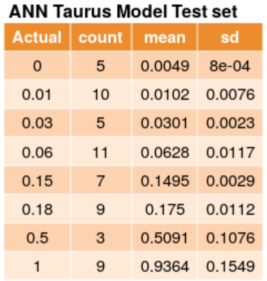
APPENDIX THREE

Machine Learning Models Descriptive Statistics



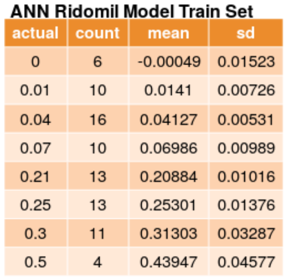
ANN Ridomil Model Test set

actual	count	mean	sd
0	5	0.0021	0.0028
0.01	10	0.0148	0.007
0.04	5	0.038	0.0029
0.07	11	0.0818	0.022
0.21	7	0.2099	0.0054
0.25	9	0.2412	0.0291
0.3	3	0.2979	0.0316
0.5	9	0.4478	0.0994



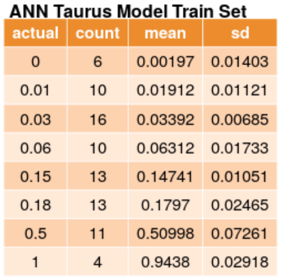
ANN Taurus Model Test set

Actual	count	mean	sd
0	5	0.0049	8e-04
0.01	10	0.0102	0.0076
0.03	5	0.0301	0.0023
0.06	11	0.0628	0.0117
0.15	7	0.1495	0.0029
0.18	9	0.175	0.0112
0.5	3	0.5091	0.1076
1	9	0.9364	0.1549



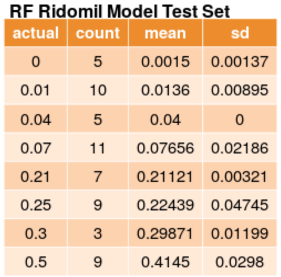
ANN Ridomil Model Train Set

actual	count	mean	sd
0	6	-0.00049	0.01523
0.01	10	0.0141	0.00726
0.04	16	0.04127	0.00531
0.07	10	0.06986	0.00989
0.21	13	0.20884	0.01016
0.25	13	0.25301	0.01376
0.3	11	0.31303	0.03287
0.5	4	0.43947	0.04577



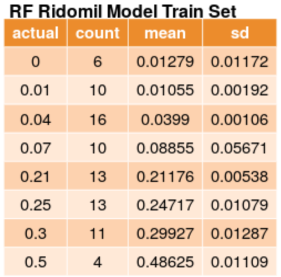
ANN Taurus Model Train Set

actual	count	mean	sd
0	6	0.00197	0.01403
0.01	10	0.01912	0.01121
0.03	16	0.03392	0.00685
0.06	10	0.06312	0.01733
0.15	13	0.14741	0.01051
0.18	13	0.1797	0.02465
0.5	11	0.50998	0.07261
1	4	0.9438	0.02918



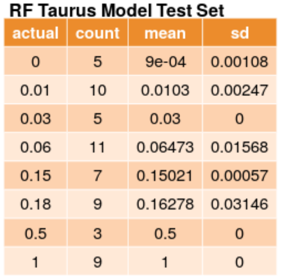
RF Ridomil Model Test Set

actual	count	mean	sd
0	5	0.0015	0.00137
0.01	10	0.0136	0.00895
0.04	5	0.04	0
0.07	11	0.07656	0.02186
0.21	7	0.21121	0.00321
0.25	9	0.22439	0.04745
0.3	3	0.29871	0.01199
0.5	9	0.4145	0.0298



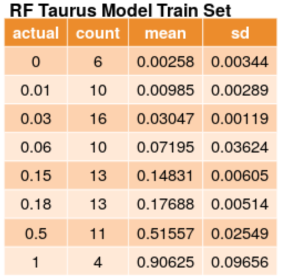
RF Ridomil Model Train Set

actual	count	mean	sd
0	6	0.01279	0.01172
0.01	10	0.01055	0.00192
0.04	16	0.0399	0.00106
0.07	10	0.08855	0.05671
0.21	13	0.21176	0.00538
0.25	13	0.24717	0.01079
0.3	11	0.29927	0.01287
0.5	4	0.48625	0.01109



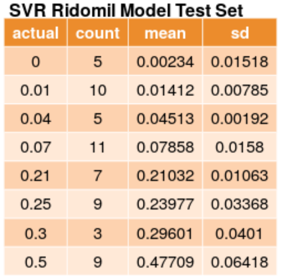
RF Taurus Model Test Set

actual	count	mean	sd
0	5	9e-04	0.00108
0.01	10	0.0103	0.00247
0.03	5	0.03	0
0.06	11	0.06473	0.01568
0.15	7	0.15021	0.00057
0.18	9	0.16278	0.03146
0.5	3	0.5	0
1	9	1	0



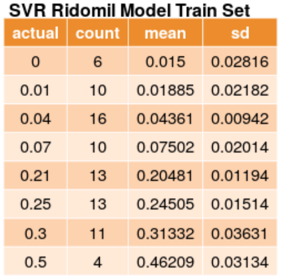
RF Taurus Model Train Set

actual	count	mean	sd
0	6	0.00258	0.00344
0.01	10	0.00985	0.00289
0.03	16	0.03047	0.00119
0.06	10	0.07195	0.03624
0.15	13	0.14831	0.00605
0.18	13	0.17688	0.00514
0.5	11	0.51557	0.02549
1	4	0.90625	0.09656



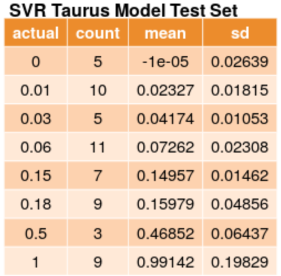
SVR Ridomil Model Test Set

actual	count	mean	sd
0	5	0.00234	0.01518
0.01	10	0.01412	0.00785
0.04	5	0.04513	0.00192
0.07	11	0.07858	0.0158
0.21	7	0.21032	0.01063
0.25	9	0.23977	0.03368
0.3	3	0.29601	0.0401
0.5	9	0.47709	0.06418



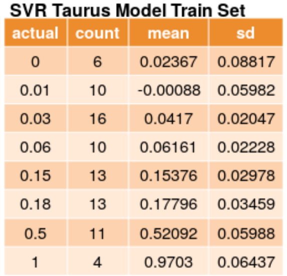
SVR Ridomil Model Train Set

actual	count	mean	sd
0	6	0.015	0.02816
0.01	10	0.01885	0.02182
0.04	16	0.04361	0.00942
0.07	10	0.07502	0.02014
0.21	13	0.20481	0.01194
0.25	13	0.24505	0.01514
0.3	11	0.31332	0.03631
0.5	4	0.46209	0.03134



SVR Taurus Model Test Set

actual	count	mean	sd
0	5	-1e-05	0.02639
0.01	10	0.02327	0.01815
0.03	5	0.04174	0.01053
0.06	11	0.07262	0.02308
0.15	7	0.14957	0.01462
0.18	9	0.15979	0.04856
0.5	3	0.46852	0.06437
1	9	0.99142	0.19829



SVR Taurus Model Train Set

actual	count	mean	sd
0	6	0.02367	0.08817
0.01	10	-0.00088	0.05982
0.03	16	0.0417	0.02047
0.06	10	0.06161	0.02228
0.15	13	0.15376	0.02978
0.18	13	0.17796	0.03459
0.5	11	0.52092	0.05988
1	4	0.9703	0.06437

APPENDIX FOUR
Models Performance on Training Sets

Table D.1: Models performance on training sets

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.0600	0.0714	0.0700	0.0921	0.0000	0.0035	0.0600	0.0714	0.0300	0.0251	0.0000	0.0005
0.0600	0.0621	0.0100	-0.0146	0.0000	0.0030	0.0600	0.0621	0.0600	0.0305	0.0000	0.0030
0.0300	0.0349	0.0000	0.0066	0.0000	0.0015	0.0300	0.0349	0.0000	0.0617	0.0000	0.0030
0.0100	0.0229	0.0000	-0.0015	0.0400	0.0400	0.0100	0.0229	0.0000	0.0540	0.0300	0.0300
0.0100	0.0124	0.0100	0.0336	0.0000	0.0221	0.0100	0.0124	0.0100	-0.1327	0.0300	0.0300
0.1500	0.1539	0.2100	0.1931	0.0400	0.0400	0.1500	0.1539	0.0100	-0.0117	0.0000	0.0000

Continued on Next Page. . .

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.5000	0.4985	0.0400	0.0337	0.0400	0.0400	0.5000	0.4985	0.1500	0.0921	0.0300	0.0300
0.1800	0.1816	0.0400	0.0407	0.0000	0.0175	0.1800	0.1816	0.0300	0.0568	0.0300	0.0300
0.5000	0.4123	0.0400	0.0305	0.0400	0.0400	0.5000	0.4123	0.0300	0.0548	0.0000	0.0000
0.5000	0.4767	0.0400	0.0420	0.0400	0.0400	0.5000	0.4767	0.0300	0.0373	0.0300	0.0300
0.0300	0.0322	0.0400	0.0406	0.0000	0.0292	0.0300	0.0322	0.0300	0.0582	0.0300	0.0300
0.1800	0.1722	0.0400	0.0261	0.0700	0.2495	0.1800	0.1722	0.0300	0.0625	0.0000	0.0090
0.5000	0.5294	0.2500	0.2498	0.0400	0.0415	0.5000	0.5294	0.0600	0.0752	0.0600	0.1750

Continued on Next Page. . .

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.5000	0.5050	0.2500	0.2186	0.0100	0.0100	0.5000	0.5050	0.0600	0.0704	0.0300	0.0315
1.0000	0.9140	0.0700	0.0717	0.0700	0.0815	1.0000	0.9140	1.0000	0.9212	0.0100	0.0100
0.1500	0.1463	0.0700	0.0261	0.0400	0.0400	0.1500	0.1463	0.1500	0.1512	0.0600	0.0600
0.1500	0.1519	0.0700	0.0609	0.0100	0.0100	0.1500	0.1519	0.1500	0.1482	0.0300	0.0315
0.1800	0.1668	0.0700	0.0948	0.0700	0.0700	0.1800	0.1668	0.1500	0.1400	0.0100	0.0165
0.0600	0.0438	0.2100	0.2053	0.0400	0.0400	0.0600	0.0438	0.1500	0.1838	0.0600	0.0600
0.0600	0.0749	0.2100	0.1859	0.0100	0.0100	0.0600	0.0749	0.0000	-0.0506	0.0300	0.0300

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.1500	0.1572	0.2100	0.2244	0.0700	0.0730	0.1500	0.1572	0.0100	-0.0361	0.0100	0.0100
0.1500	0.1440	0.2100	0.2079	0.0400	0.0390	0.1500	0.1440	0.0100	0.0125	0.0600	0.0600
0.1800	0.2135	0.2500	0.2292	0.0100	0.0100	0.1800	0.2135	0.0100	-0.0205	0.0300	0.0300
0.1800	0.2032	0.2500	0.2645	0.0700	0.0670	0.1800	0.2032	0.0100	-0.0261	0.0100	0.0100
0.1800	0.1468	0.2500	0.2523	0.0400	0.0390	0.1800	0.1468	0.0100	0.0348	0.0600	0.0600
0.5000	0.6720	0.3000	0.3804	0.0100	0.0100	0.5000	0.6720	0.0300	0.0369	0.0300	0.0300
1.0000	0.9810	0.3000	0.3127	0.0700	0.0700	1.0000	0.9810	0.0300	0.0142	0.0100	0.0095

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
1.0000	0.9514	0.3000	0.2782	0.0400	0.0400	1.0000	0.9514	0.0300	0.0123	0.0600	0.0600
0.0100	0.0086	0.5000	0.4668	0.0100	0.0100	0.0100	0.0086	0.0300	0.0326	0.0300	0.0300
0.0100	0.0157	0.2100	0.2017	0.0700	0.0675	0.0100	0.0157	0.0300	0.0265	0.0100	0.0100
0.0300	0.0349	0.2100	0.2079	0.0400	0.0408	0.0300	0.0349	0.0600	0.0882	0.0600	0.0645
0.0300	0.0238	0.0000	-0.0158	0.0100	0.0160	0.0300	0.0238	0.0600	0.0458	0.0300	0.0300
0.0300	0.0215	0.0100	-0.0016	0.0700	0.0670	0.0300	0.0215	0.0600	0.0774	0.0100	0.0095
0.0600	0.0899	0.0100	0.0193	0.0400	0.0400	0.0600	0.0899	0.1500	0.1405	0.0600	0.0600

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.0600	0.0365	0.0100	0.0143	0.0100	0.0100	0.0600	0.0365	0.1500	0.1347	0.0300	0.0300
0.1500	0.1554	0.0100	-0.0060	0.0700	0.0700	0.1500	0.1554	0.1500	0.1664	0.0100	0.0045
0.1500	0.1619	0.0100	0.0234	0.0400	0.0420	0.1500	0.1619	0.1500	0.1680	0.0600	0.0600
0.1500	0.1488	0.0400	0.0593	0.0100	0.0095	0.1500	0.1488	0.1800	0.1632	0.0300	0.0345
0.1800	0.1768	0.0400	0.0511	0.0700	0.0700	0.1800	0.1768	0.1800	0.2640	0.0100	0.0085
0.1800	0.1745	0.0400	0.0468	0.0400	0.0380	0.1800	0.1745	0.1800	0.2247	0.0600	0.0600
0.5000	0.4759	0.0400	0.0404	0.0100	0.0100	0.5000	0.4759	0.1800	0.1748	0.0300	0.0300

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.5000	0.4812	0.0400	0.0402	0.2100	0.2100	0.5000	0.4812	0.1800	0.1518	0.0100	0.0100
1.0000	0.9288	0.0700	0.0800	0.0400	0.0380	1.0000	0.9288	0.5000	0.5431	0.1500	0.1500
0.1500	0.1350	0.2100	0.2145	0.2500	0.2170	0.1500	0.1350	0.5000	0.5344	0.5000	0.5000
0.1800	0.1562	0.2500	0.2461	0.2100	0.2100	0.1800	0.1562	0.5000	0.5332	0.1800	0.1655
0.1800	0.2323	0.2500	0.2513	0.3000	0.3100	0.1800	0.2323	0.5000	0.6065	0.1500	0.1500
0.1800	0.1851	0.2500	0.2433	0.2500	0.2500	0.1800	0.1851	1.0000	1.0527	0.5000	0.5300
0.1800	0.1498	0.2500	0.2531	0.2100	0.2120	0.1800	0.1498	0.1800	0.1832	0.1800	0.1800

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.1800	0.1772	0.2500	0.2458	0.3000	0.2975	0.1800	0.1772	0.1800	0.1508	0.1500	0.1500
0.5000	0.4475	0.3000	0.2975	0.2500	0.2500	0.5000	0.4475	0.1800	0.1374	0.5000	0.5750
0.5000	0.5046	0.3000	0.2664	0.2100	0.2173	0.5000	0.5046	0.5000	0.4853	0.1800	0.1800
0.5000	0.6067	0.3000	0.2833	0.3000	0.3000	0.5000	0.6067	0.5000	0.6402	0.1500	0.1545
0.0000	-0.0050	0.5000	0.4541	0.2500	0.2500	0.0000	-0.0050	0.5000	0.5019	0.5000	0.5063
0.0000	-0.0023	0.5000	0.4260	0.2100	0.2120	0.0000	-0.0023	0.5000	0.5002	0.1800	0.1800
0.0000	-0.0031	0.3000	0.3581	0.3000	0.3000	0.0000	-0.0031	0.0000	0.1674	0.1500	0.1440

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.0100	0.0165	0.3000	0.3382	0.2500	0.2500	0.0100	0.0165	0.0000	-0.0656	0.5000	0.5000
0.0300	0.0359	0.3000	0.2988	0.2100	0.2100	0.0300	0.0359	0.0000	-0.0249	0.1800	0.1800
0.0300	0.0445	0.3000	0.3413	0.3000	0.3000	0.0300	0.0445	0.0100	0.0427	0.1500	0.1500
0.0600	0.0721	0.5000	0.5015	0.2500	0.2480	0.0600	0.0721	0.0100	0.0459	0.5000	0.5500
0.0600	0.0460	0.2100	0.2140	0.2100	0.2100	0.0600	0.0460	0.0100	0.0824	0.1800	0.1770
0.1500	0.1467	0.2100	0.1817	0.3000	0.3000	0.1500	0.1467	0.0300	0.0313	0.1500	0.1500
0.1500	0.1551	0.2500	0.2667	0.2500	0.2500	0.1500	0.1551	0.0300	0.0894	0.5000	0.5000

Continued on Next Page. . .

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.1500	0.1355	0.2500	0.2477	0.2100	0.2265	0.1500	0.1355	0.0300	0.0617	0.1800	0.1800
0.1500	0.1246	0.2500	0.2174	0.3000	0.2740	0.1500	0.1246	0.0300	0.0366	0.1500	0.1295
0.0000	0.0264	0.3000	0.2918	0.2500	0.2480	0.0000	0.0264	0.0300	0.0310	0.5000	0.5000
0.0100	0.0243	0.0000	0.0645	0.2100	0.2100	0.0100	0.0243	0.0600	0.0293	0.1800	0.1785
0.0100	0.0052	0.0000	0.0081	0.3000	0.3100	0.0100	0.0052	0.0600	0.0432	0.1500	0.1500
0.0300	0.0353	0.0000	0.0282	0.2500	0.2612	0.0300	0.0353	0.0600	0.0853	0.5000	0.5100
0.0000	0.0093	0.0100	0.0457	0.2100	0.2100	0.0000	0.0093	0.0600	0.0709	0.1800	0.1800

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.0100	0.0451	0.0100	0.0218	0.3000	0.3200	0.0100	0.0451	0.1500	0.2226	0.1500	0.1500
0.0100	0.0154	0.0100	0.0526	0.2500	0.2500	0.0100	0.0154	0.1500	0.1479	0.5000	0.5000
0.0300	0.0403	0.0400	0.0473	0.2100	0.2100	0.0300	0.0403	0.1500	0.1477	0.1800	0.1800
0.0300	0.0412	0.0400	0.0628	0.5000	0.4750	0.0300	0.0412	0.1500	0.1557	0.1500	0.1500
0.0300	0.0357	0.0400	0.0458	0.2500	0.2500	0.0300	0.0357	0.1800	0.2031	1.0000	0.9000
0.0000	-0.0135	0.0400	0.0437	0.2100	0.2030	0.0000	-0.0135	0.1800	0.1739	0.1800	0.1740
0.0100	0.0249	0.0400	0.0471	0.5000	0.4800	0.0100	0.0249	0.1800	0.1561	0.1500	0.1500

Continued on Next Page...

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.0300	0.0310	0.0700	0.0890	0.2500	0.2550	0.0300	0.0310	0.1800	0.1660	1.0000	1.0000
0.0300	0.0411	0.0700	0.0830	0.2100	0.2120	0.0300	0.0411	0.1800	0.1644	0.1800	0.1780
0.0300	0.0393	0.0700	0.0702	0.5000	0.5000	0.0300	0.0393	0.5000	0.4862	0.1500	0.1500
0.0300	0.0260	0.0700	0.0824	0.3000	0.2805	0.0300	0.0260	0.5000	0.4441	1.0000	0.9500
0.0300	0.0253	0.2100	0.2074	0.2500	0.2340	0.0300	0.0253	0.5000	0.4551	0.5000	0.5000
0.0600	0.0792	0.2100	0.2139	0.5000	0.4900	0.0600	0.0792	1.0000	0.9902	0.1800	0.1665

Continued on Next Page. . .

Continuation of Table D.1

Mancozeb Training Results						THO Training Results					
Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted	Actual	ANN Predicted	Actual	SVR Predicted	Actual	RF Predicted
0.0600	0.0552	0.2100	0.2048	0.3000	0.3000	0.0600	0.0552	1.0000	0.9171	1.0000	0.7750

APPENDIX FIVE
Models Performance on Testing Sets

Table E.1: Models performance on testing sets

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.0000	0.0016	0.0146	0.0015	0.0000	0.0045	0.0179	0.0015
0.0000	0.0018	-0.0146	0.0005	0.0000	0.0045	-0.0333	0.0000
0.0000	0.0007	-0.0106	0.0000	0.0000	0.0047	-0.0110	0.0000
0.0000	0.0068	0.0021	0.0020	0.0000	0.0062	-0.0078	0.0005
0.0000	-0.0006	0.0202	0.0035	0.0000	0.0044	0.0342	0.0025
0.0100	0.0093	0.0106	0.0100	0.0100	0.0051	0.0235	0.0095
0.0100	0.0053	0.0133	0.0100	0.0100	0.0055	0.0480	0.0095

Continued on Next Page. . .

Continuation of Table E.1

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.0100	0.0092	0.0114	0.0100	0.0100	0.0062	0.0264	0.0100
0.0100	0.0132	0.0211	0.0100	0.0100	0.0070	0.0419	0.0100
0.0100	0.0269	0.0235	0.0110	0.0100	0.0298	0.0205	0.0075
0.0100	0.0260	0.0246	0.0115	0.0100	0.0127	0.0279	0.0095
0.0100	0.0122	0.0159	0.0390	0.0100	0.0061	0.0354	0.0170
0.0100	0.0159	0.0086	0.0115	0.0100	0.0090	0.0179	0.0100
0.0100	0.0167	-0.0018	0.0115	0.0100	0.0143	-0.0158	0.0100
0.0100	0.0135	0.0141	0.0115	0.0100	0.0063	0.0070	0.0100
0.0400	0.0372	0.0482	0.0400	0.0300	0.0291	0.0501	0.0300

Continued on Next Page...

Continuation of Table E.1

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.0400	0.0362	0.0445	0.0400	0.0300	0.0297	0.0440	0.0300
0.0400	0.0362	0.0456	0.0400	0.0300	0.0273	0.0521	0.0300
0.0400	0.0431	0.0440	0.0400	0.0300	0.0332	0.0267	0.0300
0.0400	0.0373	0.0433	0.0400	0.0300	0.0314	0.0358	0.0300
0.0700	0.0773	0.0917	0.1405	0.0600	0.0627	0.0667	0.1120
0.0700	0.0598	0.0647	0.0700	0.0600	0.0522	0.0717	0.0600
0.0700	0.0674	0.0706	0.0700	0.0600	0.0552	0.0798	0.0600
0.0700	0.0973	0.0937	0.0640	0.0600	0.0503	0.1171	0.0600
0.0700	0.0759	0.0707	0.0815	0.0600	0.0634	0.0439	0.0600

Continued on Next Page...

Continuation of Table E.1

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.0700	0.0651	0.0606	0.0700	0.0600	0.0625	0.0442	0.0600
0.0700	0.0590	0.0583	0.0721	0.0600	0.0493	0.0444	0.0600
0.0700	0.0680	0.0705	0.0700	0.0600	0.0572	0.0726	0.0600
0.0700	0.1195	0.1027	0.0640	0.0600	0.0816	0.0940	0.0600
0.0700	0.0927	0.0828	0.0640	0.0600	0.0763	0.0724	0.0600
0.0700	0.1180	0.0980	0.0760	0.0600	0.0801	0.0919	0.0600
0.2100	0.2194	0.2147	0.2100	0.1500	0.1485	0.1661	0.1500
0.2100	0.2041	0.2150	0.2100	0.1500	0.1514	0.1512	0.1500
0.2100	0.2088	0.2036	0.2100	0.1500	0.1485	0.1413	0.1500

Continued on Next Page. . .

Continuation of Table E.1

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.2100	0.2040	0.1919	0.2100	0.1500	0.1477	0.1210	0.1500
0.2100	0.2091	0.2230	0.2185	0.1500	0.1543	0.1537	0.1515
0.2100	0.2135	0.2185	0.2100	0.1500	0.1508	0.1573	0.1500
0.2100	0.2105	0.2054	0.2100	0.1500	0.1453	0.1564	0.1500
0.2500	0.2292	0.2237	0.2500	0.1800	0.1737	0.1324	0.1680
0.2500	0.2381	0.2474	0.2375	0.1800	0.1745	0.1557	0.1800
0.2500	0.2493	0.2550	0.2500	0.1800	0.1785	0.1750	0.1800
0.2500	0.2519	0.2382	0.2500	0.1800	0.1715	0.1511	0.1800
0.2500	0.2739	0.2700	0.2500	0.1800	0.1933	0.2253	0.1800

Continued on Next Page...

Continuation of Table E.1

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.2500	0.2863	0.2902	0.2500	0.1800	0.1826	0.2359	0.1800
0.2500	0.2411	0.2519	0.2500	0.1800	0.1832	0.1664	0.1800
0.2500	0.2036	0.1980	0.1410	0.1800	0.1590	0.1038	0.1190
0.2500	0.1976	0.1835	0.1410	0.1800	0.1586	0.0925	0.0980
0.3000	0.2881	0.2884	0.3000	0.5000	0.4904	0.4535	0.5000
0.3000	0.3333	0.3394	0.3100	0.5000	0.6247	0.5391	0.5000
0.3000	0.2724	0.2603	0.2861	0.5000	0.4121	0.4130	0.5000
0.5000	0.3314	0.4215	0.4306	1.0000	0.8293	0.8414	1.0000
0.5000	0.2843	0.3716	0.4306	1.0000	0.6124	0.7190	1.0000

Continued on Next Page. . .

Continuation of Table E.1

Mancozeb Testing Results				THO Testing Results			
Actual	ANN Predicted	SVR Predicted	RF Predicted	Actual	ANN Predicted	SVR Predicted	RF Predicted
0.5000	0.5454	0.5713	0.3600	1.0000	1.0481	1.3974	1.0000
0.5000	0.4795	0.4795	0.3896	1.0000	1.0342	0.9789	1.0000
0.5000	0.5326	0.5265	0.4550	1.0000	1.1347	1.0901	1.0000
0.5000	0.5103	0.5011	0.4250	1.0000	0.9898	1.0065	1.0000
0.5000	0.4348	0.4805	0.4250	1.0000	0.9213	0.9835	1.0000
0.5000	0.5477	0.5291	0.4300	1.0000	1.0025	1.0942	1.0000
0.5000	0.3646	0.4128	0.3846	1.0000	0.8549	0.8118	1.0000

APPENDIX SIX
R Machine Learning Code Used

```
# simple function for plotting and pre-processing DRS spectra

# Author: Ndung’u Ndegwa Charles


rm(list = ls())

library(squash)

library(Hmisc)

par(mfrow=c(1,1))


#Create a plot function:

ndungu_plot<-function(mydata, wavelengths, xlim, ylim,main,ylab){

#Create color map:

map <- makecmap(as.numeric(mydata$label),n = 2, breaks = pretty,

symm = FALSE, base = NA,

colFn = colorRampPalette(c(’red’,’green’,’blue’)),

col.na = NA,

right = FALSE, include.lowest = FALSE)

mycol <- cmap(mydata$label, map = map)

par(font=2,las=1,mar = c(5,4,4,10) + 0.1)


#Plot spectra:

matplot(wavelengths,t(mydata$DRS),font.axis=2,

col=mycol,lty=1, xlab="",ylab="",type="l",lwd=3,

xlim=xlim, ylim=ylim,main=main)
minor.tick(nx=2, ny=2,tick.ratio=0.75)
par(mar = c(5,4,4,6) + 0.1)
title(xlab="Wavelength␣(nm)",ylab=ylab,font.lab=2)

#Plot color map:
vkey(map, title = "Key",stretch=2.4, side=2, skip=2,
x=1100,y=min(ylim))
}

#Define plot limits:
xlim<-c(400,900)
ylim<-c(0,1)
main<- ""
ylab<- ""

## Some EDA ON the RAW Spectra

#setwd("~/data ")

## we need to normalise the data to get rid of the negative
   ↪ reflectance
# values before converting to apparent absorbance.

#Flatten or squash a list of lists into a simpler vector
#dealing with negative values and normalizing absorbance spectra
```

```
library(pavo)
set.seed(12345)
spec <- read.csv("~␣data/spectra.csv")
#spec<- t(spec)
#write.csv(spec,"spectra.csv")

### Now we interpolate the data in 1-nm bins and
### convert the data to an rspec object

spec <- as.rspec(spec)
# wavelengths found in column 1
# The spectral data contain 3961 negative value(s),
# which may produce unexpected results if used in models.
# Consider using procspec() to correct them.

is.rspec(spec)
par(mfrow=c(1,1))
plot(spec)
testspecs1 <- spec

###### We need to get rid of the
###### low SNR areas of spectra
par(mfrow=c(1,1))
plot(testspecs1, select = 2, ylim = c(-10, 60))
abline(h = 0, lty = 3)
```

```
## Looks much better now by using the range between
# 400-900 nm

testspecs <- as.rspec(testspecs1, lim = c(400, 900))
plot(testspecs, select = 2, ylim = c(-10, 50))
abline(h = 0, lty = 3)

#testspecs1 <-t(testspecs)
#write.csv(testspecs1,"trans_binned_spectra.csv")


## we can now convert the data to absorbance


spectra<- read.csv("spectra.csv",
sep=",",dec=".",header=TRUE)
spectra[1:10,1:5]
spectra$label<-spectra$wl

mydata <- data.frame(label=I(spectra$label),
#samples=I(spectra$samples),
DRS = I(spectra[2:ncol(spectra)]))


#Retrieve wavelength numbers from colomn names:
wavelengths<-substring(colnames(mydata$DRS),2,7)
```

```
wavelengths<-as.numeric(wavelengths)

#Sort the data by decreasing SN values:
mydata<-mydata[order(mydata$label, decreasing = TRUE),]

#plot the raw data
ndungu_plot(mydata, wavelengths, xlim, ylim =c(0,60),
main = "Raw␣DRS␣spectra",ylab = "Reflectance␣(%)")



# Function to convert reflectance to absorbance using log10(1/R)

Absorb<-function(spectra){
spectra<-as.matrix(spectra)
spect_Absorb <- log10(1/spectra)
return(spect_Absorb)}
#Perform conversion

newspectra<-Absorb(mydata$DRS)


mydataAbsorb <- data.frame(label=I(mydata$label),
DRS = I(newspectra))

#Plot new spectra:
```

```
ndungu_plot(mydataAbsorb, wavelengths, xlim, ylim = c(-2,-0.6),
main = "Absorbance␣from␣reflectance",ylab = "Log10␣(1/R)")

#mydataAbsorb <- as.data.frame(t(mydataAbsorb))
#write.csv(mydataAbsorb,"Absorb_trans_data.csv")

rm(newspectra)


# Start preprocessing the new absorbance spectra

# start by converting it to pavo object
library(pavo)
spec <- read.csv("data.csv")


spec <- as.rspec(spec)
# wavelengths found in column 1
# The spectral data contain 284067 negative value(s),
# which may produce unexpected results if used in models.
# Consider using procspec() to correct them.


is.rspec(spec)
par(mfrow=c(1,1))
plot(spec)
```

```
testspecs <- spec
par(mfrow=c(1,1))

# Now we can Apply two different processing options
testspecs.fix1 <- procspec(testspecs, fixneg = "addmin")
testspecs.fix2 <- procspec(testspecs, fixneg = "zero")


# Plot it
par(mfrow=c(1,1))
par(mar = c(2, 2, 2, 2), oma = c(3, 3, 0, 0))
layout(cbind(c(1, 1), c(2, 3)), widths = c(2, 1, 1))

plot(testspecs, select = 2, ylim = c(-1.7, -1))
abline(h = 0, lty = 3)

plot(testspecs.fix1, select = 2, ylim = c(-0.1, .6))
abline(h = 0, lty = 3)

plot(testspecs.fix2, select = 2, ylim = c(-0.1, .6))
abline(h = 0, lty = 3)

mtext("Wavelength␣(nm)", side = 1, outer = TRUE, line = 1)
mtext("Log10(1/R)", side = 2, outer = TRUE, line = 1)
par(mfrow=c(1,1))
```

```
# Normalizing and Smoothing Spectra



# use the plotsmooth() function to determine a suitable smoothing
   ↪  parameter (span). This function
# allows you to set a minimum and maximum smoothing parameter to
   ↪ try and plots the resulting curves against
# the unsmoothed (raw) data in a convenient multipanel figure.-
sppspec <- testspecs.fix1
plotsmooth(sppspec, minsmooth = 0.01, maxsmooth = 0.07,
curves = 5,specnum = "5",ask = FALSE)

##From the resulting plot, we can see that span = 0.07 is the
   ↪ minimum amount of smoothing to remove spectral noise
# while preserving the original spectral shape. Based on this
   ↪ value, we will now use the opt argument in procspec()
# to smooth data for further plotting and analysis.
par(mfrow=c(1,1))
spec.sm <- procspec(sppspec, opt = "smooth", span = 0.07)

plot(sppspec[, 5] ~ sppspec[, 1],
type = "l",
lwd = 10,
col = "grey",xlab = "Wavelength␣(nm)",ylab = "Log10(1/R)",
main="Smoothing␣with␣0.07␣pan")
```

```
mtext("Wavelength␣(nm)", side = 1, outer = TRUE, line = 1)
mtext("Log10(1/R)", side = 2, outer = TRUE, line = 1)
par(mfrow=c(1,1))
lines(spec.sm[, 5] ~ sppspec[, 1], col = "red",lwd = 2)

##We can also try different normalisations. Options include
   ↪ subtracting the minimum Log (1/R) of a spectrum at
# all wavelengths (effectively making the minimum Log (1/R) equal
   ↪  to zero, opt = "min", left panel, below) and
# making the Log (1/R) at all wavelength proportional to the
   ↪ maximum Log (1/R) (i.e. setting maximum Log (1/R)
# to 1; opt = "max", centre panel, below). Note that the user can
   ↪  specify multiple processing options that will be
# applied sequentially to the spectral data by procspec() (right
   ↪ panel, below).

# Run some different normalisations
specs.max <- procspec(spec.sm, opt = "max")

specs.min <- procspec(spec.sm, opt = "min")

specs.str <- procspec(spec.sm, opt = c("min", "max")) # multiple
   ↪ options
```

```
# Plot results

par(mfrow = c(1, 3), mar = c(2, 2, 2, 2), oma = c(3, 3, 0, 0))

plot(specs.min[, 5] ~ c(400:900), xlab = "", ylab = "", type = "l"
   ↪ )
abline(h = 0, lty = 2)

plot(specs.max[, 5] ~ c(400:900), ylim = c(0, 1),
xlab = "", ylab = "", type = "l")
abline(h = c(0, 1), lty = 2)

plot(specs.str[, 4] ~ c(400:900), type = "l", xlab = "", ylab = ""
   ↪ )
abline(h = c(0, 1), lty = 2)

mtext("Wavelength␣(nm)", side = 1, outer = TRUE, line = 1)
mtext("Normalised␣Log␣(1/R)", side = 2, outer = TRUE, line = 1)
par(mfrow=c(1,1))


# write.csv(specs.str,"cleaned_drs_absorbance_data.csv")
# specs.str1<-t(specs.str)
# write.csv(specs.str1,"cleaned_trans_drs_absorbance_data.csv")
```

```
## more preprocessing of absorbance spectra

spectra<- read.csv("cleaned_trans_drs_absorbance_data.csv",
sep=",",dec=".",header=TRUE)
spectra[1:10,1:5]

spectra$label<-spectra$wl
mydata <- data.frame(label=I(spectra$label),
#samples=I(spectra$samples),
DRS = I(spectra[2:ncol(spectra)]))



#Retrieve wavelength numbers from colomn names:
wavelengths<-substring(colnames(mydata$DRS),2,7)
wavelengths<-as.numeric(wavelengths)

#Sort the data by decreasing SN values:
mydata<-mydata[order(mydata$label, decreasing = FALSE),]

#plot the raw data
ndungu_plot(mydata, wavelengths, xlim, ylim =c(-0.05,1.1),
main = "DRS␣Absorbance␣Spectra",ylab = "Log10␣(1/R)␣␣a.u.")



#Create smoothing function:

```

```
SmoothFast<-function(Spectra,windowsize){

#Create smoothing matrix:
Mat<-matrix(0,length((windowsize+1):(ncol(Spectra)-windowsize)),2*
   ↪ windowsize+1)
for(j in 1:nrow(Mat)){Mat[j,]<-seq(j,j+2*windowsize,1)}

#Smoothing spectra using matrix operations:
newspectra<-matrix(0,nrow(Spectra),
length((windowsize+1):(ncol(Spectra)-windowsize)))
for(i in 1:nrow(Mat)){newspectra[,i]<-apply(Spectra[,Mat[i,]],1,
   ↪ mean)}

#Add front and end tails (not smoothed):
fronttail<-newspectra[,1]
endtail<-newspectra[,ncol(newspectra)]
for(k in 1:(windowsize-1)){fronttail<-data.frame(fronttail,
   ↪ newspectra[,1])
endtail<-data.frame(endtail,newspectra[,ncol(newspectra)])}
newspectra<-data.frame(fronttail,newspectra,endtail)

return(newspectra)}

#Apply smoothing function:
newspectra<-SmoothFast(mydata$DRS,windowsize=3)
```

```
mydataSmooth <- data.frame(label=I(mydata$label),
DRS = I(newspectra))
rm(newspectra)

#Plot smoothed spectra:
ndungu_plot(mydataSmooth, wavelengths, xlim, ylim =c(-0.05,1.1),
main = "Moving␣Average",ylab = "Log␣(1/R)␣␣a.u")



library(signal)

#Apply Savitzky-Golay smoothing to all spectra:
newspectra<-apply(mydata$DRS,1,
FUN=sgolayfilt,
p = 2,
n = 3,
m = 0,
ts = 1)

#Create new data frame:
mydataSmoothSG<-data.frame(label=I(mydata$label),
DRS = I(t(newspectra)))

rm(newspectra)

#Plot spectra:
```

```
ndungu_plot(mydataSmoothSG, wavelengths, xlim, ylim =c(-0.2,1.1),
main = "Savitzky-Golay",ylab = "Log␣(1/R)␣␣a.u")



#Create SNV function:


SNV<-function(spectra){
spectra<-as.matrix(spectra)
spectrat<-t(spectra)
spectrat_snv<-scale(spectrat,center=TRUE,scale=TRUE)
spectra_snv<-t(spectrat_snv)
return(spectra_snv)}

#Perform SNV:
newspectra<-SNV(mydata$DRS)
mydataSNV<-data.frame(label=I(mydata$label), DRS = I(newspectra))
rm(newspectra)
#Plot new spectra:
ndungu_plot(mydataSNV, wavelengths, xlim, ylim = c(-2,2),
main = "SNV␣detrend",ylab = "")




# Baseline removal


library(hyperSpec)
```

```
#Convert mydata to an hyperSpec S4 object:
mydataHS<-new("hyperSpec", spc = as.matrix(mydata$DRS))
#,
#wavelength = wavelengths)

#Compute baselines using order 2 polynomials:
baseline<-spc.fit.poly.below(fit.to = mydataHS, poly.order = 3)

mybaseline<-data.frame(label=I(mydata$label),
DRS = I(baseline@data$spc))

#Plot baseline:
ndungu_plot(mybaseline, wavelengths, xlim, ylim = c(-0.1,1),
main = "Baselines",ylab = "")




#Baseline removal:
newspectra<-mydataHS@data$spc-baseline@data$spc

mydataBSL<-data.frame(label=I(mydata$label),
DRS = I(newspectra))
rm(newspectra)
#Plot new spectra:
ndungu_plot(mydataBSL, wavelengths, xlim, ylim = c(0,.6),
main = "Baseline␣Removal",ylab = "Log␣(1/R)␣␣a.u")



# Perform MSC correction:

library(pls)

newspectra<-msc(as.matrix(mydata$DRS))

mydataMSC<-data.frame(label=I(mydata$label),
DRS = I(newspectra))
rm(newspectra)
#Plot new spectra:
ndungu_plot(mydataMSC, wavelengths, xlim, ylim = c(-0.2,1.1),
main = "MSC",ylab = "Log␣10␣(1/R)␣␣a.u")


#Compute the first derivative for all spectra:
newspectra<-apply(mydata$DRS, 1,
FUN=sgolayfilt,
p = 2,
n = 5,
m = 1,
ts = 1)
```

```
mydataDERIV1<-data.frame(label=I(mydata$label),
DRS = I(t(newspectra)))
rm(newspectra)
#Plot new spectra:
ndungu_plot(mydataDERIV1, wavelengths, xlim, ylim = c(-0.02,.02),
main = "First␣derivative",ylab = "A.U")




#Compute the second derivative for all spectra:
newspectra<-apply(mydata$DRS, 1,
FUN=sgolayfilt,
p = 2,
n = 5,
m = 2,
ts = 1)

mydataDERIV2<-data.frame(label=I(mydata$label),
DRS = I(t(newspectra)))
rm(newspectra)
#Plot new spectra:
ndungu_plot(mydataDERIV2, wavelengths, xlim, ylim = c(-0.002,.002),
   ↪
main = "Second␣derivative",ylab = "")
```

```
### pca using chemospec
rm(list = ls())
suppressMessages(library(ChemoSpec))

#
#MSCdata <- as.data.frame(t(t(mydataMSC)))
#MSCdata<-t(mydataMSC)
#str(MSCdata)
#write.csv(MSCdata,"msc_data.csv")

setwd("pca␣2021")



spec <- matrix2SpectraObject(gr.crit = c("cs","bf","m72"),
gr.cols = c(’green’,’red’,’blue’),
freq.unit = "Wavelength␣[nm]",
int.unit = "Log10␣(1/R)␣a.u",
descrip = "drs␣Spectra",
in.file = "msc_data.csv",
out.file = "",
chk = TRUE, sep = ",", dec = ".")

# Summarizing the data
sumSpectra(spec)
```

```
# Plotting the full EDXRF spectra
plotSpectra(spec,
main = "MSC␣Spectra",
which = c(1:547),
yrange = c(-0.05,1.25),
offset = 0.0001,
lab.pos = 9000,
showGrid = F,
leg.loc = "topright")

# Feature selections
# VIS spectra #
VISspec <- removeFreq(spec,
rem.freq = spec$freq > 600| spec$freq < 500)

# Plotting the spectra
plotSpectra(VISspec,
main = "VIS␣MSC␣Spectra",
which = c(1:547),
yrange = c(0,1.2),
offset = 0,
lab.pos = 5000,
showGrid = F,
leg.loc = "topright")

# NIR spectra #
```

```
NIRspec <- removeFreq(spec,
rem.freq = spec$freq > 900| spec$freq < 750)

# Plotting the spectra
plotSpectra(NIRspec,
main = "NIR␣MSC␣Spectra",
which = c(1:547),
yrange = c(-0.01,0.1),
offset = 0,
lab.pos = 10,
showGrid = F,
leg.loc = "topleft")

# NORMALIZING THE SPECTRA #

VISspec<-normSpectra(VISspec)

NIRspec<-normSpectra(NIRspec)


# PCA analysis of the VIS spectras #

VISpca<-c_pcaSpectra(VISspec,
choice = "autoscale",
cent = T)
```

```
plotScores(VISspec,

VISpca,

main ="VIS␣Spectra",

pcs = c(1,2),

ellipse = "none",

tol = "none",

leg.loc ="right")

# tol = 0.05)

abline(v=0,h=0)

cv_pcaSpectra(VISspec,

pcs = 5)


plotLoadings(VISspec,

VISpca,

main = "VIS␣Spectra",

loads = c(1, 2),

ref = 91,

tol = "none")


plot2Loadings(VISspec,

VISpca,

main = "VIS␣Spectra",

loads = c(1, 2),

ref = 91,

tol = "none")
```

```
sPlotSpectra( VISspec,
VISpca,
pc = 1,
tol = 0.001,
main = "VIS␣Spectra")

# To check pca outliers
diagnostics <- pcaDiag(VISspec,
VISpca,
pcs = 3,
quantile = 0.916,
plot = "SD")

# Scree plot
plot(VISpca, type = "l")

plotScree(VISpca, style = "alt",
main = "VIS␣Spectra")



# PCA analysis of the NIR spectras ##

NIRpca<-c_pcaSpectra(NIRspec,
choice = "autoscale")
#cent = T)
```

```
plotScores(NIRspec,
NIRpca,
main ="NIR␣Spectra",
pcs = c(1,2),
ellipse = "none",
tol = "none",
leg.loc ="topright")
# tol = 0.05)
abline(v=0,h=0)

cv_pcaSpectra(NIRspec,
pcs = 5)

plotLoadings(NIRspec,
NIRpca,
main = "NIR␣Spectra",
loads = c(1, 2),
ref = 91,
tol = "none")

plot2Loadings(NIRspec,
NIRpca,
main = "NIR␣Spectra",
loads = c(1, 2),
ref = 91,
tol = "none")

sPlotSpectra( NIRspec,
NIRpca,
pc = 1,
tol = 0.001,
main = "NIR␣Spectra")

# To check pca outliers
diagnostics <- pcaDiag(NIRspec,
NIRpca,
pcs = 1,
quantile = 0.999,
plot = "SD",
use.sym = F)

diagnostics <- pcaDiag(NIRspec,
NIRpca,
pcs = 1,
quantile = .999,
plot = "OD",
use.sym = F)
#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
########## find outliers based on Sdist##############
```

```
x<- as.data.frame(diagnostics[["SDist"]])
y<-as.data.frame(as.factor(spec[["names"]]))



library(dplyr)
s<-bind_cols(x,y)

#write.csv(s,"/home/ndungu/Desktop/pca journal data analysis/s.
   ↪ csv")



# Scree plot
plot(NIRpca, type = "l")

plotScree(NIRpca, style = "alt",
main = "VIS␣Spectra")

#### Extract the first 10 PC scores for modeling

# PCA_scores1<-as.data.frame(NIRpca[["x"]])
# PCA_scores<-as.data.frame(PCA_scores1[,1:10])
# data_labels <-as.data.frame(spec[["groups"]])
#
#
#
# library(dplyr)
# PCA_data <- bind_cols(groups=data_labels,PCA_scores)
# write.csv(PCA_data,"pcadata.csv")

# #--------------------- machine learning using pca data
   ↪ --------------------------
rm(list = ls())

# import data
pcadata <- read.csv("pcadata.csv")



# use mlbench, caret and DT library, please make sure they are
   ↪ already installed
require(mlbench)
require(caret)
require(DT)



m<-c("nnet","svmLinear","rf")
length(m); m;


# pre-load all packages (does not really work due to other
   ↪ dependencies)
```

```
suppressPackageStartupMessages(ll <-lapply(m, require, character.
   ↪ only = TRUE))

# show which libraries were loaded
sessionInfo()

# load X and Y (this will be transferred to to train function)
X <- pcadata[,2:4]
Y <-pcadata$spec...groups...



# register parallel front-end
library(doParallel); cl <- makeCluster(detectCores());
   ↪ registerDoParallel(cl)

# this is required otherwise the first method is benchmarked
   ↪ wrong
warmup <-train(y=Y, x=X, "rf",
trControl = trainControl(method = "boot632"))

# this setup actually calls the caret::train function, in order
   ↪ to provide
# minimal error handling this type of construct is needed.
trainCall <- function(i)
{
cat("----------------------------------------------------","\n");
set.seed(123); cat(i,"␣<-␣loaded\n");
return(tryCatch(
t2 <- train(y=Y, x=X, (i), trControl = trainControl(method = "
   ↪ boot632")),
error=function(e) NULL))
}


# use lapply/loop to run everything, required for try/catch error
   ↪  function to work
system.time(t2 <- lapply(m, trainCall))


#remove NULL values, we only allow succesful methods, provenance
   ↪ is deleted.
t2 <- t2[!sapply(t2, is.null)]



# extract the neural net model
t2[[1]]
mod1<-t2[[1]]$finalModel
mod1



# print confusion matrix example
caret::confusionMatrix(t2[[1]])

a<-varImp(t2[[1]], scale = FALSE)
```

```
a$importance[1:3,]


## calculate the ROC for each class
nnet_imp <- filterVarImp(x = pcadata[,2:6], y=pcadata$spec...
   ↪ groups...)
head(nnet_imp)


plot(nnet_imp, top = 5,
main = "neural␣net␣model␣-␣Variable␣Importance")



# extract the Support Vector Machines with Linear Kernel
t2[[2]]
mod2<-t2[[2]]$finalModel
mod2




# print confusion matrix example
caret::confusionMatrix(t2[[2]],"none")






a<-varImp(t2[[2]], scale = FALSE)
```

```
a$importance[1:3,]

## calculate the ROC for each class

svm_imp <- varImp(t2[[2]], scale = FALSE)
svm_imp


plot(svm_imp, top = 3,
main = "SVM␣model␣-␣Variable␣Importance")



# extract the Random Forest
t2[[3]]
mod3<-t2[[3]]$finalModel

mod3

# print confusion matrix example
caret::confusionMatrix(t2[[3]],"none")

#### calculate the ROC for each class

rf_imp <- varImp(t2[[3]], scale = FALSE)
rf_imp
```

```
plot(rf_imp, top = 3,
main = "RF␣model␣-␣Variable␣Importance")





# this setup extracts the results with minimal error handling

printCall <- function(i)
{
return(tryCatch(
{
cat(sprintf("%-22s",(m[i])))
cat(round(getTrainPerf(t2[[i]])$TrainAccuracy,4),"\t")
cat(round(getTrainPerf(t2[[i]])$TrainKappa,4),"\t")
cat(t2[[i]]$times$everything[3],"\n")},
error=function(e) NULL))
}

r2 <- lapply(1:length(t2), printCall)

# stop cluster and register sequntial front end
stopCluster(cl); registerDoSEQ();
```

```
# preallocate data types
i = 1; MAX = length(t2);
x1 <- character() # Name
x2 <- numeric() # R2
x3 <- numeric() # RMSE
x4 <- numeric() # time [s]
x5 <- character() # long model name

# fill data and check indexes and NA with loop/lapply
for (i in 1:length(t2)) {
x1[i] <- t2[[i]]$method
x2[i] <- as.numeric(round(getTrainPerf(t2[[i]])$TrainAccuracy,4))
x3[i] <- as.numeric(round(getTrainPerf(t2[[i]])$TrainKappa,4))
x4[i] <- as.numeric(t2[[i]]$times$everything[3])
x5[i] <- t2[[i]]$modelInfo$label
}

# coerce to data frame
df1 <- data.frame(x1,x2,x3,x4,x5, stringsAsFactors=FALSE)

# print all results to R-GUI
df1

# plot models, just as example
# ggplot(t2[[1]])
# ggplot(t2[[1]])
# call web output with correct column names
DT::datatable(df1, options = list(
columnDefs = list(list(className = ’dt-left’, targets = c
   ↪ (0,1,2,3,4,5))),
pageLength = MAX,
order = list(list(3, ’desc’))), # sort according to kappa value
colnames = c(’Num’, ’Name’, ’Accuracy’, ’Kappa’, ’time␣[s]’, ’
   ↪ Model␣name’),
caption = paste(’Classification␣results␣from␣caret␣models’,Sys.
   ↪ time()),
class = ’cell-border␣stripe’) %>%
formatRound(’x2’, 3) %>%
formatRound(’x3’, 3) %>%
formatRound(’x4’, 3) %>%
formatStyle(2,
background = styleColorBar(x2, ’steelblue’),
backgroundSize = ’100%␣90%’,
backgroundRepeat = ’no-repeat’,
backgroundPosition = ’center’
)



# compile models and compare perfomance if we use "ctrl1" or "
   ↪ ctrl2" in "tlabelontrol" parametres
```

```
model_list <- list(ANN = t2[[1]],
SVM =t2[[2]],
RF = t2[[3]])

results <- resamples(model_list)

summary(results)

# boxplot comparing results

bwplot(results,
layout = c(3, 1)) # RMSE, MSE and R-squared

bwplot(results,
metric = "Accuracy",
main = "Comparing␣Algorithms␣accuracy␣")

bwplot(results,
metric = "Accuracy",
main = "Algorithms␣accuracy␣comparing",
xlim = c(0.7,1))

bwplot(results,
metric = "Kappa",
main = "Algorithms␣accuracy␣comparing",
xlim = c(0.75,.93))
```

```
registerDoSEQ()


### END



# ########################################

# # machine learning using raw data
rm(list = ls())

# import data

rawdata<-read.csv("msc_data1.csv")




# use mlbench, caret and DT library, please make sure they are
   ↪ already installed
require(mlbench)
require(caret)
require(DT)
```

```
m<-c("nnet","svmLinear","rf")
length(m); m;



# pre-load all packages (does not really work due to other
   ↪ dependencies)
suppressPackageStartupMessages(ll <-lapply(m, require, character.
   ↪ only = TRUE))

# show which libraries were loaded
sessionInfo()

# load X and Y (this will be transferred to to train function)
X = rawdata[,2:152]
Y = rawdata$label




# register parallel front-end
library(doParallel); cl <- makeCluster(detectCores());
   ↪ registerDoParallel(cl)

# this is required otherwise the first method is benchmarked
   ↪ wrong
```

```
warmup <-train(y=Y,
x=X,
"rf",
trControl = trainControl(method = "boot632"))

# this setup actually calls the caret::train function, in order
   ↪ to provide
# minimal error handling this type of construct is needed.
trainCall <- function(i)
{
cat("-----------------------------------------------------","\n");
set.seed(123); cat(i,"␣<-␣loaded\n");
return(tryCatch(
t2 <- train(y=Y, x=X, (i), trControl = trainControl(method = "
   ↪ boot632")),
error=function(e) NULL))
}

# use lapply/loop to run everything, required for try/catch error
   ↪  function to work
system.time(t2 <- lapply(m, trainCall))

#remove NULL values, we only allow succesful methods, provenance
   ↪ is deleted.
t2 <- t2[!sapply(t2, is.null)]
```

```
# extract the neural net model
t2[[1]]
mod1<-t2[[1]]$finalModel
mod1



# print confusion matrix example
caret::confusionMatrix(t2[[1]])

a<-varImp(t2[[1]], scale = FALSE)
a$importance[1:20,]

## calculate the ROC for each class
nnet_imp <- filterVarImp(x = rawdata[,2:152], y=rawdata$label)
head(nnet_imp)

plot(nnet_imp, top = 10,
main = "neural␣net␣model␣-␣Variable␣Importance")



# extract the Support Vector Machines with Linear Kernel
t2[[2]]
mod2<-t2[[2]]$finalModel
mod2
```

```
# print confusion matrix example
caret::confusionMatrix(t2[[2]],"none")







a<-varImp(t2[[2]], scale = FALSE)
a$importance[1:20,]

## calculate the ROC for each class

svm_imp <- varImp(t2[[2]], scale = FALSE)
svm_imp


plot(svm_imp, top = 10,
main = "SVM␣model␣-␣Variable␣Importance")


# extract the Random Forest
t2[[3]]
mod3<-t2[[3]]$finalModel
```

```
mod3

# print confusion matrix example
caret::confusionMatrix(t2[[3]],"none")

## calculate the ROC for each class

rf_imp <- varImp(t2[[3]], scale = FALSE)
rf_imp


plot(rf_imp, top = 10,
main = "RF␣model␣-␣Variable␣Importance")




# this setup extracts the results with minimal error handling

printCall <- function(i)
{
return(tryCatch(
{
cat(sprintf("%-22s",(m[i])))
cat(round(getTrainPerf(t2[[i]])$TrainAccuracy,4),"\t")
cat(round(getTrainPerf(t2[[i]])$TrainKappa,4),"\t")
cat(t2[[i]]$times$everything[3],"\n")},
error=function(e) NULL))
}

r2 <- lapply(1:length(t2), printCall)

# stop cluster and register sequntial front end
stopCluster(cl); registerDoSEQ();

# preallocate data types
i = 1; MAX = length(t2);
x1 <- character() # Name
x2 <- numeric() # R2
x3 <- numeric() # RMSE
x4 <- numeric() # time [s]
x5 <- character() # long model name

# fill data and check indexes and NA with loop/lapply
for (i in 1:length(t2)) {
x1[i] <- t2[[i]]$method
x2[i] <- as.numeric(round(getTrainPerf(t2[[i]])$TrainAccuracy,4))
x3[i] <- as.numeric(round(getTrainPerf(t2[[i]])$TrainKappa,4))
x4[i] <- as.numeric(t2[[i]]$times$everything[3])
x5[i] <- t2[[i]]$modelInfo$label
}
```

```
# coerce to data frame
df1 <- data.frame(x1,x2,x3,x4,x5, stringsAsFactors=FALSE)

# print all results to R-GUI
df1

# plot models, just as example
# ggplot(t2[[1]])
# ggplot(t2[[1]])

# call web output with correct column names
DT::datatable(df1, options = list(
columnDefs = list(list(className = 'dt-left', targets = c
   ↪ (0,1,2,3,4,5))),
pageLength = MAX,
order = list(list(3, 'desc')), # sort according to kappa value
colnames = c('Num', 'Name', 'Accuracy', 'Kappa', 'time␣[s]', '
   ↪ Model␣name'),
caption = paste('Classification␣results␣from␣caret␣models',Sys.
   ↪ time()),
class = 'cell-border␣stripe') %>%
formatRound('x2', 3) %>%
formatRound('x3', 3) %>%
formatRound('x4', 3) %>%
formatStyle(2,
background = styleColorBar(x2, 'steelblue'),
backgroundSize = '100%␣90%',
backgroundRepeat = 'no-repeat',
backgroundPosition = 'center'
)



# compile models and compare perfomance if we use "ctrl1" or "
   ↪ ctrl2" in "tlabelontrol" parametres

model_list <- list(ANN = t2[[1]],
SVM =t2[[2]],
RF = t2[[3]])

results <- resamples(model_list)

summary(results)

# boxplot comparing results

bwplot(results,
layout = c(3, 1)) # RMSE, MSE and R-squared

bwplot(results,
metric = "Accuracy",
main = "Comparing␣Algorithms␣accuracy␣")
```

```
bwplot(results,
metric = "Accuracy",
main = "Algorithms␣accuracy␣comparing",
xlim = c(0.7,1))

bwplot(results,
metric = "Kappa",
main = "Algorithms␣accuracy␣comparing",
xlim = c(0.75,.93))

registerDoSEQ()




### END

rm(list = ls())



# machine learning using Raw absorbance data without
   ↪ preprocessing

# import data
spec <- read.csv("Absorbance␣spectra.csv")
```

```
# use mlbench, caret and DT library, please make sure they are
   ↪ already installed
require(mlbench)
require(caret)
require(DT)



m<-c("nnet","svmLinear","rf")
length(m); m;


# pre-load all packages (does not really work due to other
   ↪ dependencies)
suppressPackageStartupMessages(ll <-lapply(m, require, character.
   ↪ only = TRUE))

# show which libraries were loaded
sessionInfo()

# load X and Y (this will be transferred to to train function)
X = spec[,2:152]
Y=spec$label
```

```
# register parallel front-end
library(doParallel); cl <- makeCluster(detectCores());
   ↪ registerDoParallel(cl)

# this is required otherwise the first method is benchmarked
   ↪ wrong
warmup <-train(y=Y,
x=X,
"rf",
trControl = trainControl(method = "boot632"))

# this setup actually calls the caret::train function, in order
   ↪ to provide
# minimal error handling this type of construct is needed.
trainCall <- function(i)
{
cat("---------------------------------------------------","\n");
set.seed(123); cat(i,"␣<-␣loaded\n");
return(tryCatch(
t2 <- train(y=Y, x=X, (i), trControl = trainControl(method = "
   ↪ boot632")),
error=function(e) NULL))
}
```

```
# use lapply/loop to run everything, required for try/catch error
   ↪  function to work
system.time(t2 <- lapply(m, trainCall))

#remove NULL values, we only allow succesful methods, provenance
   ↪ is deleted.
t2 <- t2[!sapply(t2, is.null)]
t2



# extract the neural net model
t2[[1]]
mod1<-t2[[1]]$finalModel
mod1



# print confusion matrix example
caret::confusionMatrix(t2[[1]])

a<-varImp(t2[[1]], scale = FALSE)
a$importance[1:20,]

########## calculate the ROC for each class
nnet_imp <- filterVarImp(x = spec[,2:152], y=spec$label)
head(nnet_imp)
```

```
plot(nnet_imp, top = 10,
main = "neural␣net␣model␣-␣Variable␣Importance")



# extract the Support Vector Machines with Linear Kernel
t2[[2]]
mod2<-t2[[2]]$finalModel
mod2





# print confusion matrix example
caret::confusionMatrix(t2[[2]],"none")







a<-varImp(t2[[2]], scale = FALSE)
a$importance[1:20,]

## calculate the ROC for each class

svm_imp <- varImp(t2[[2]], scale = FALSE)
svm_imp
```

```
plot(svm_imp, top = 10,
main = "SVM␣model␣-␣Variable␣Importance")




# extract the Random Forest
t2[[3]]
mod3<-t2[[3]]$finalModel

mod3

# print confusion matrix example
caret::confusionMatrix(t2[[3]],"none")

### calculate the ROC for each class

rf_imp <- varImp(t2[[3]], scale = FALSE)
rf_imp


plot(rf_imp, top = 10,
main = "RF␣model␣-␣Variable␣Importance")
```

```
# this setup extracts the results with minimal error handling

printCall <- function(i)
{
return(tryCatch(
{
cat(sprintf("%-22s",(m[i])))
cat(round(getTrainPerf(t2[[i]])$TrainAccuracy,4),"\t")
cat(round(getTrainPerf(t2[[i]])$TrainKappa,4),"\t")
cat(t2[[i]]$times$everything[3],"\n")},
error=function(e) NULL))
}

r2 <- lapply(1:length(t2), printCall)

# stop cluster and register sequntial front end
stopCluster(cl); registerDoSEQ();

# preallocate data types
i = 1; MAX = length(t2);
x1 <- character() # Name
x2 <- numeric() # R2
x3 <- numeric() # RMSE
x4 <- numeric() # time [s]
```

```
x5 <- character() # long model name

# fill data and check indexes and NA with loop/lapply
for (i in 1:length(t2)) {
x1[i] <- t2[[i]]$method
x2[i] <- as.numeric(round(getTrainPerf(t2[[i]])$TrainAccuracy,4))
x3[i] <- as.numeric(round(getTrainPerf(t2[[i]])$TrainKappa,4))
x4[i] <- as.numeric(t2[[i]]$times$everything[3])
x5[i] <- t2[[i]]$modelInfo$label
}

# coerce to data frame
df1 <- data.frame(x1,x2,x3,x4,x5, stringsAsFactors=FALSE)

# print all results to R-GUI
df1

# plot models, just as example
#ggplot(t2[[1]])
#ggplot(t2[[2]])

# call web output with correct column names
DT::datatable(df1, options = list(
columnDefs = list(list(className = 'dt-left',
targets = c(0,1,2,3,4,5))),
pageLength = MAX,
order = list(list(3, 'desc'))), # sort according to kappa value
colnames = c('Num', 'Name', 'Accuracy', 'Kappa', 'time␣[s]', '
   ↪ Model␣name'),
caption = paste('Classification␣results␣from␣caret␣models',Sys.
   ↪ time()),
class = 'cell-border␣stripe') %>%
formatRound('x2', 3) %>%
formatRound('x3', 3) %>%
formatRound('x4', 3) %>%
formatStyle(2,
background = styleColorBar(x2, 'steelblue'),
backgroundSize = '100%␣90%',
backgroundRepeat = 'no-repeat',
backgroundPosition = 'center'
)

# compile models and compare perfomance if we use "ctrl1" or "
   ↪ ctrl2" in "tlabelontrol" parametres

model_list <- list(ANN = t2[[1]],
SVM =t2[[2]],
RF = t2[[3]])

results <- resamples(model_list)

summary(results)
```

```
# boxplot comparing results

bwplot(results,
layout = c(3, 1)) # RMSE, MSE and R-squared

bwplot(results,
metric = "Accuracy",
main = "Comparing␣Algorithms␣accuracy␣")

bwplot(results,
metric = "Accuracy",
main = "Algorithms␣accuracy␣comparing",
xlim = c(0.7,1))

bwplot(results,
metric = "Kappa",
main = "Algorithms␣accuracy␣comparing",
xlim = c(0.75,.93))

registerDoSEQ()
#
   ↪ -----------------------------------------------------------------------------
   ↪

# require(gridExtra)
# grid.arrange(plot(varImp(object = mod1), # main = "_ - Variable
    ↪  Importance (_ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
# plot(varImp(object = mod7), # main = "_ - Variable Importance (
    ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
#
# plot(varImp(object = mod5), # main = "_ - Variable Importance (
    ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
#
# plot(varImp(object = mod4), # main = "_ - Variable Importance (
    ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
#
# plot(varImp(object = mod6), # main = "_ - Variable Importance (
    ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
# ncol = 5,
# nrow = 1)
#
# png("_.png",
# width = 3200,
# height = 1800,
# units = 'px',
# res = 300)
#
# dev.off()

### END

#
# library(caret)
# library(doParallel)
#
#
# cluster <- makeCluster(detectCores() - 1) # convention to leave
   ↪  1 core for OS
# registerDoParallel(cluster)
# set.seed(12)
# #
# # # compile cross-validation settings
# #
# ctrl <- trainControl(method = "LOOCV",
# returnResamp = "final")
#
# ctrl1 <- trainControl(method = "repeatedcv",
# number = 5,
# repeats = 10,
# allowParallel = TRUE)
#
# ctrl2 <- trainControl(method = "cv",
# number = 5)
#
#
   ↪ #---------------------------------------------------------------------
   ↪
# # train Neural net model model
#
# set.seed(1234)
#
# mod1 <- train(groups˜.,
# data = pcadata,
# method = "nnet",
# metric = "Accuracy",
# tlabelontrol = ctrl1,
# preProcess = c("center", "scale"))
#
```

```
# plot(varImp(object = mod1),
# main = "ANN Variable Importance",
# top = 10,
# ylab = "Variable")
#

# # Ridge or lasso regression
# # note, that if "alpha" is set to 0 this process runs a ridge
    ↪ model,
# # if itâĂŹs set to 1 it runs a LASSO model and an "alpha"
    ↪ between 0 and 1
# # results in an elastic net model
#
# set.seed(1234)
#
# mod4 <- train(groups~.,
# data = pcadata,
# method = "glmnet",
# metric = "Accuracy",
# tlabelontrol = ctrl1,
# preProcess = c("center", "scale"))
#
# plot(varImp(object = mod4),
# main = "Lasso/Ridge - Variable Importance",
# top = 10, ylab = "Variable")
#
```

```
# png(".png", width = 1920,
# height = 1080,
# units = 'px', res = 300)
#

# # RF
#
# rftg <- data.frame(mtry = seq(2, 55, by = 2)) # take a lot of
   ↪ time to compute
#
# # can change parametres or
#
# mtry <- as.integer(sqrt(ncol(pcadata[, 1:10])))
#
# rf.tuneGrid <- expand.grid(.mtry = mtry)
#
# set.seed(12)
#
# mod5 <- train(groups~.,
# data = pcadata,
# method = "rf",
# tuneGrid = rf.tuneGrid, # or rftg
# tlabelontrol = ctrl1,
# importance = TRUE)
#
# plot(varImp(object = mod5),
# main = "Random Forest - Variable Importance",
# top = 10,
# ylab = "Variable")
#
# #------------------------------------------
# # XGBoost
#
# gb.tuneGrid <- expand.grid(eta = c(0.3,0.4,0.5,0.6),
# nrounds = c(5,10,15),
# max_depth = 2:3, gamma = 0,
# colsample_bytree = 0.8,
# min_child_weight = 1,
# subsample = 1)
# set.seed(12)
#
# mod6 <- train(groups~.,
# data = pcadata,
# method = "xgbTree",
# tuneGrid = gb.tuneGrid,
# tlabelontrol = ctrl1)
#
# plot(varImp(object = mod6),
# main = "XGBoost - Variable Importance",
# top = 10,
# ylab = "Variable")
#
```

```
# # SVM
#
# svmRadialTuneGrid <- expand.grid(sigma = c(0.05,0.0456,0.0577),
# C = c(1.5,1.596,1.65,1.89,1.95,2,2.2,2.44))
# set.seed(12)
#
# mod7 <- train(groups~.,
# data = pcadata,
# method = "svmRadial",
# tuneGrid = svmRadialTuneGrid,
# preProcess = c("center", "scale"),
# tlabelontrol = ctrl1)
#
# plot(varImp(object = mod7),
# main = "SVM - Variable Importance",
# top = 10,
# ylab = "Variable")


# # compile models and compare perfomance if we use "ctrl1" or "
   ↪ ctrl2" in "tlabelontrol" parametres
#
# model_list <- list(PLSR = mod1,
# GLMNET = mod4,
# RF = mod5,
# XGBoost = mod6,
# SVM=mod7)
#
# results <- resamples(model_list)
#
# summary(results)
#
# # boxplot comparing results
#
# bwplot(results,
# layout = c(3, 1)) # RMSE, MSE and R-squared
#
# bwplot(results,
# metric = "Accuracy",
# main = "Comparing Algorithms accuracy ")
#
# bwplot(results,
# metric = "Accuracy",
# main = "Algorithms accuracy comparing",
# xlim = c(0,2))
#
# stopCluster(cluster)
# registerDoSEQ()
#
#
# require(gridExtra)
# grid.arrange(plot(varImp(object = mod1), # main = "_ - Variable
   ↪  Importance (_ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
# plot(varImp(object = mod7), # main = "_ - Variable Importance (
   ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
#
# plot(varImp(object = mod5), # main = "_ - Variable Importance (
   ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
#
# plot(varImp(object = mod4), # main = "_ - Variable Importance (
   ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
#
# plot(varImp(object = mod6), # main = "_ - Variable Importance (
   ↪ _ spectra)"
# top = 5,
# ylab = "Variable",
# xlab = "Relevance"),
# ncol = 5,
# nrow = 1)
#
# png("_.png",
# width = 3200,
# height = 1800,
# units = 'px',
# res = 300)
#
# dev.off()


## REgression
# # load all libraries
# library(doParallel)
# library(caret)
# library(dplyr)
# library(DT)
#
# models <- c("glmnet", "knn","nnet","rf",
# "pls", "svmRadial", "svmLinear")
#
#
# cl <- makeCluster(detectCores())
# registerDoParallel(cl)
```

```
#
# # compile cross-validation settings
# ctrl1 <- trainControl(method = "repeatedcv", number = 5,
# repeats = 10, allowParallel = TRUE)
#
# # use lapply/loop to run everything
# l <- lapply(models, function(i)
# {cat("-------------------------------------------------","\n
    ↪ ");
# set.seed(1234); cat(i," <- done\n");
# t <- train(groups~.,
# data = pcadata
# , (i), trControl = ctrl1,
# preProcess = c("center", "scale"),
# metric = "Accuracy")
# }
# )
#
# # use lapply to print the results
# results <- lapply(1:length(l), function(i)
# {cat(sprintf("%-20s",(models[i])));
# cat(round(l[[i]]$results$Rsquared[which.min(l[[i]]$results$RMSE
    ↪ )],4),"\t");
# cat(round(l[[i]]$results$RMSE[which.min(l[[i]]$results$RMSE)
    ↪ ],4),"\t")
# cat(l[[i]]$times$everything[3],"\n")
# }
# )
#
# # stop the parallel processing and register sequential front-
    ↪ end
# stopCluster(cl)
# registerDoSEQ()
#
# # preallocate data types
# i = 1; MAX = length(l);
# x1 <- character() # Name
# x2 <- numeric() # R2
# x3 <- numeric() # RMSE
# x4 <- numeric() # time [s]
# x5 <- character() # long model name
#
# # fill data and check indexes and NA
# for (i in 1:length(l)) {
# x1[i] <- l[[i]]$method
# x2[i] <- as.numeric(l[[i]]$results$Rsquared[which.min(l[[i]]$
    ↪ results$RMSE)])
# x3[i] <- as.numeric(l[[i]]$results$RMSE[which.min(l[[i]]$
    ↪ results$RMSE)])
# x4[i] <- as.numeric(l[[i]]$times$everything[3])
# x5[i] <- l[[i]]$modelInfo$label
# }
```

```
#
# # coerce to data frame
# df <- data.frame(x1,x2,x3,x4,x5, stringsAsFactors = FALSE)
#
# # print all results to R-GUI
# df
#
# # call web browser output with sortable column names
# datatable(df, options = list(
# columnDefs = list(list(className = 'dt-left', targets = c
   ↪ (0,1,2,3,4,5))),
# pageLength = MAX,
# order = list(list(3, 'desc'))),
# colnames = c('âĎŰ', 'Name', 'R2', 'RMSE', 'time [s]', 'Model
   ↪ name'),
# caption = paste('Regression results from "caret" list models'),
# class = 'cell-border stripe') %>%
# formatRound('x2', 3) %>%
# formatRound('x3', 3) %>%
# formatRound('x4', 3) %>%
# formatStyle(3,
# background = styleColorBar(x3, 'steelblue'),
# backgroundSize = '100% 90%',
# backgroundRepeat = 'no-repeat',
# backgroundPosition = 'center',
# fontWeight = 'bold'
# )
# # compile models and compare perfomance
# model_list <- list(GLMNET = l[[1]], KNN = l[[2]], neuralnet = l
   ↪ [[3]],nnet = l[[4]], RF = l[[5]],
# pcaNNet = l[[6]],PCR = l[[7]],pls = l[[8]], svmRadial = l[[9]],
# svmLinear = l[[10]])
# results <- resamples(model_list)
# summary(results)
# # boxplot comparing results
# bwplot(results, layout = c(3, 1)) # RMSE, MSE and R-squared
#-------------------------------------#
# # Run a list cross-validation methods with PCR method
# cl <- makeCluster(detectCores())
# registerDoParallel(cl)
#
# # define all cross-validation methods
# cvMethods <- c("boot", # bootstrap
# "boot632", # 0.632 bootstrap
# "LGOCV", # leave-one-group cross validation, variant of LOOCV
   ↪ for hierarchical data
# "LOOCV", # leave-one-out cross validation, also known as
   ↪ jacknife
# "cv", # cross validation
# "repeatedcv" # repeated n-fold cross validation
# )
#
```

```
# # use R lapply function to loop through all CV methos with qrf
# all <- lapply(cvMethods, function(x)
# {set.seed(1234); print(x); tc <- trainControl(method=(x))
# fit1 <- train(C~., data = RAW.spectra,
# preProcess = c("center", "scale"),
# trControl = tc,
# method ="neuralnet") # may choose any of possible regression
   ↪ models
# #,"nnet","rf","pcaNNet", "svmRadial", "svmLinear"
# }
# )
#
# # stop cluster
# stopCluster(cl)
# registerDoSEQ()
#
# # extract the used cvMethods
# myNames <- lapply(1:6, function(x) all[[x]]$control$method)
#
# # save results
# results <- sapply(all, getTrainPerf)
#
# # change column Names to cv methods
# colnames(results) <- myNames
#
# # get the results
# results
# library(xtable)
#
#
# xtable(results, auto = TRUE)
# xtable(mtcars, auto = TRUE)
#

# # Learning curve plots for R caret classifications and
   ↪ regressions in parallel
# # (ROC vs training size, RMSE vs training size)
# # Source: Max Kuhn (topepo); https://github.com/topepo/caret/
   ↪ issues/278
# # https://github.com/tobigithub/caret-machine-learning
# # Tobias Kind (2015)
#
# #-------------------------------------
# # Library parallel() is a native R library, no CRAN required
# library(parallel)
# nCores <- detectCores(logical = FALSE)
# nThreads <- detectCores(logical = TRUE)
# cat("CPU with",nCores,"cores and",nThreads,"threads detected.\n
   ↪ ")
#
# # load the doParallel/doSNOW library for caret cluster use
# library(doParallel)
# cl <- makeCluster(nThreads)
# registerDoParallel(cl)
#
# #------------------------------------------
# ## function: learning_curve_dat plots training-size vs RMSE or
   ↪ ROC
# ## dat: entire data set used for modling
# ## y: character stirng for the outcome column name
# ## proportion: proportion of data used to train the model
# ## test_prop: proportion of data used initially set aside for
   ↪ testing
# ## verbose: write out a log of training milestones
# ## ...: arguments to pass to 'train'
# #-----------------------------------
# learning_curve_dat <- function(dat,
# outcome = colnames(dat)[1],
# proportion = (1:10)/10, test_prop = 0,
# verbose = TRUE, ...) {
#
# proportion <- sort(unique(proportion))
# n_size <- length(proportion)
#
# if(test_prop > 0) {
# for_model <- createDataPartition(dat[, outcome], p = 1 - test_
   ↪ prop, list = FALSE)
# } else for_model <- 1:nrow(dat)
```

```
#
# n <- length(for_model)
#
# resampled <- vector(mode = "list", length = n_size)
# tested <- if(test_prop > 0) resampled else NULL
# apparent <- resampled
# for(i in seq(along = proportion)) {
# if(verbose) cat("Training for ", round(proportion[i]*100, 1),
# "% (n = ", floor(n*proportion[i]), ")\n", sep = "")
# in_mod <- if(proportion[i] < 1) sample(for_model, size = floor(
   ↪ n*proportion[i])) else for_model
# mod <- train(x = dat[in_mod, colnames(dat) != outcome, drop =
   ↪ FALSE],
# y = dat[in_mod, outcome],
# ...)
# if(i == 1) perf_names <- mod$perfNames
# resampled[[i]] <- merge(mod$resample, mod$bestTune)
# resampled[[i]]$Training_Size <- length(in_mod)
#
# if(test_prop > 0) {
# if(!mod$control$classProbs) {
# test_preds <- extractPrediction(list(model = mod),
# testX = dat[-for_model, colnames(dat) != outcome, drop = FALSE
   ↪ ],
# testY = dat[-for_model, outcome])
# } else {
```

```
# test_preds <- extractProb(list(model = mod),
# testX = dat[-for_model, colnames(dat) != outcome, drop = FALSE
   ↪ ],
# testY = dat[-for_model, outcome])
# }
# test_perf <- mod$control$summaryFunction(test_preds, lev = mod$
   ↪ finalModel$obsLevels)
# test_perf <- as.data.frame(t(test_perf))
# test_perf$Training_Size <- length(in_mod)
# tested[[i]] <- test_perf
# try(rm(test_preds, test_perf), silent = TRUE)
# }
#
# if(!mod$control$classProbs) {
# app_preds <- extractPrediction(list(model = mod),
# testX = dat[in_mod, colnames(dat) != outcome, drop = FALSE],
# testY = dat[in_mod, outcome])
# } else {
# app_preds <- extractProb(list(model = mod),
# testX = dat[in_mod, colnames(dat) != outcome, drop = FALSE],
# testY = dat[in_mod, outcome])
# }
# app_perf <- mod$control$summaryFunction(app_preds, lev = mod$
   ↪ finalModel$obsLevels)
# app_perf <- as.data.frame(t(app_perf))
# app_perf$Training_Size <- length(in_mod)
# apparent[[i]] <- app_perf
#
# try(rm(mod, in_mod, app_preds, app_perf), silent = TRUE)
# }
#
# resampled <- do.call("rbind", resampled)
# resampled <- resampled[, c(perf_names, "Training_Size")]
# resampled$Data <- "Resampling"
# apparent <- do.call("rbind", apparent)
# apparent <- apparent[, c(perf_names, "Training_Size")]
# apparent$Data <- "Training"
# out <- rbind(resampled, apparent)
# if(test_prop > 0) {
# tested <- do.call("rbind", tested)
# tested <- tested[, c(perf_names, "Training_Size")]
# tested$Data <- "Testing"
# out <- rbind(out, tested)
# }
# out
# }
#
# #-------------------------------------------
# # multiplot for plotting multiple ggplots
# # Example: multiplot(p1,p2,p3,p4,p5,p6,cols=3)
# # Source: http://www.peterhaschke.com/r/2013/04/24/MultiPlot.
   ↪ html
```

```
#
# multiplot <- function(..., plotlist = NULL, file, cols = 1,
   ↪ layout = NULL) {
# require(grid)
#
# plots <- c(list(...), plotlist)
#
# numPlots = length(plots)
#
# if (is.null(layout)) {
# layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
# ncol = cols, nrow = ceiling(numPlots/cols))
# }
#
# if (numPlots == 1) {
# print(plots[[1]])
#
# } else {
# grid.newpage()
# pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(
   ↪ layout))))
#
# for (i in 1:numPlots) {
# matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
#
# print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
# layout.pos.col = matchidx$col))
# }
# }
# }
#

# ## Classification example
#
# library(caret)
# library(xgboost)
#
# # set plot to 2x3
# par(mfrow=c(2,3))
#
# set.seed(1412)
# class_dat <- twoClassSim(2000)
#
# set.seed(29510)
# lda_data <- learning_curve_dat(dat = class_dat, outcome = "
   ↪ Class",
# test_prop = 1/4,
# ## ‘train‘ arguments
# method = "lda",
# metric = "ROC",
# trControl = trainControl(classProbs = TRUE,
# method = "boot632",
# summaryFunction = twoClassSummary))
#
# p1 <- ggplot(lda_data, aes(x = Training_Size, y = ROC, color =
    ↪ Data)) +
# geom_smooth(method = loess, span = .8) +
# ggtitle("LDA classification with boot632 CV") +
# theme_bw()
# p1


# set.seed(29510)
# rf_data <- learning_curve_dat(dat = class_dat, outcome = "Class
    ↪ ",
# test_prop = 1/4,
# ## ‘train‘ arguments
# method = "rf",
# metric = "ROC",
# tuneLength = 4,
# trControl = trainControl(classProbs = TRUE,
# method = "boot632",
# summaryFunction = twoClassSummary))
#
# p2 <- ggplot(rf_data, aes(x = Training_Size, y = ROC, color =
    ↪ Data)) +
# geom_smooth(method = loess, span = .8) +
# ggtitle("rf classification with boot632 CV") +
# theme_bw()
# p2
# #----------------------------------
# set.seed(29510)
# rf_data <- learning_curve_dat(dat = class_dat, outcome = "Class
   ↪ ",
# test_prop = 1/4,
# ## ‘train‘ arguments
# method = "parRF",
# metric = "ROC",
# tuneLength = 4,
# trControl = trainControl(classProbs = TRUE,
# method = "boot632",
# summaryFunction = twoClassSummary))
#
# p3 <- ggplot(rf_data, aes(x = Training_Size, y = ROC, color =
   ↪ Data)) +
# geom_smooth(method = loess, span = .8) +
# ggtitle("parRF classification with boot632 CV") +
# theme_bw()
# p3
#
# ## Regression example
#
#
# set.seed(19135)
# reg_dat <- SLC14_1(2000)
#
# set.seed(31535)
# bag_data <- learning_curve_dat(dat = reg_dat, outcome = "y",
# test_prop = 1/4,
# ## ‘train‘ arguments
# method = "treebag",
# trControl = trainControl(method = "boot632"),
# ## ‘bagging‘ arguments
# nbagg = 100)
#
# p4 <- ggplot(bag_data, aes(x = Training_Size, y = RMSE, color =
   ↪  Data)) +
# geom_smooth(method = loess, span = .8) +
# ggtitle("treebag regression with boot632 CV") +
# theme_bw()
# p4
#

# set.seed(31535)
# svm_data <- learning_curve_dat(dat = reg_dat, outcome = "y",
# test_prop = 0,
# ## ‘train‘ arguments
# method = "svmRadial",
# preProc = c("center", "scale"),
# tuneGrid = data.frame(sigma = 0.03, C = 2^10),
# trControl = trainControl(method = "boot632"))
#
# p5 <- ggplot(svm_data, aes(x = Training_Size, y = RMSE, color =
    ↪  Data)) +
# geom_smooth(method = loess, span = .8) +
# ggtitle("svmRadial regression with boot632 CV") +
# theme_bw()
# p5


# set.seed(31535)
# svm_no_test <- learning_curve_dat(dat = reg_dat, outcome = "y",
# test_prop = 1/4,
# ## ‘train‘ arguments
# method = "svmRadial",
# preProc = c("center", "scale"),
# tuneGrid = data.frame(sigma = 0.03, C = 2^10),
# trControl = trainControl(method = "boot632"))
#
# p6 <- ggplot(svm_no_test, aes(x = Training_Size, y = RMSE,
    ↪ color = Data)) +
# geom_smooth(method = loess, span = .8) +
# ggtitle("svmRadial regression with boot632 CV") +
# theme_bw()
# p6
#
#
```

```
# multiplot(p1,p2,p3,p4,p5,p6,cols=3)

#

# stopCluster(cl)

# registerDoSEQ()

# ### END
```