Summer 8-1-2022

Machine Learning Model Comparison And Arma Simulation Of Exhaled Breath Signals Classifying COVID-19 Patients

Aaron Christopher Segura
University of New Mexico - Main Campus

Aaron Christopher Segura

Candidate

Mathematics and Statistics

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

James Degnan, Ph.D., Chairperson

Justin Baca, MD, Ph.D.

Mohammed Motamed, Ph.D.

MACHINE LEARNING MODEL COMPARISON AND ARMA SIMULATION OF EXHALED BREATH SIGNALS CLASSIFYING COVID-19 PATIENTS

by

Aaron Segura

B.A. BIOCHEMISTRY, B.S. BIOLOGY, UNIVERSITY OF NEW MEXICO 2013

B.S. MATHEMATICS, UNIVERSITY OF NEW MEXICO 2019

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

Statistics

The University of New Mexico
Albuquerque, New Mexico

July 2022

DEDICATION

I would like to dedicate this thesis to my parents for their unconditional love and support over the years, my brothers, Ronnie, Andrew, and Adrian, for always believing in me and encouraging me to follow my dreams, to my grandparents who would have been/are so proud of me and my accomplishments. I would also like to thank my friends Marcus Davalos, Fredrick Lee, Jamie Yang, Julie Everett, for many years of companionship and support.

ACKNOWLEDGEMENTS

I heartily acknowledge Dr. James Degnan, Ph.D. my advisor and committee chair, for continuing to encourage me through the years of classroom teachings and the long number of months writing and reviewing these chapters. His guidance and professional style will remain with me as I continue my career.

I also thank my committee members, Dr. Justin Baca, MD, Ph.D., for helping me to find a research project that is medically oriented and pertinent to the current pandemic. Dr. Baca has helped guide my journey into medicine and has been an invaluable resource as a medical student while pursuing graduate education. I enjoyed our discussions during our weekly meetings and have had the opportunity to build my skill set as part of Dr. Baca’s research group.

To Dr. Mohamed Motamed, Ph.D. for his thoughtful feedback. It has tested my knowledge and given me an opportunity to think more critically and consider different perspectives.

MACHINE LEARNING MODEL COMPARISON AND ARMA SIMULATION OF EXHALED BREATH SIGNALS CLASSIFYING COVID-19 PATIENTS

by

Aaron Segura

B.A., Biochemistry, B.S., Biology University of New Mexico, 2013

B.S., Mathematics, University of New Mexico, 2019

M.S., Statistics, University of New Mexico, 2022

ABSTRACT

This study compared the performance of machine learning models in classifying COVID-19 patients using exhaled breath signals and simulated datasets. Ground truth classification was determined by the gold standard Polymerase Chain Reaction (PCR) test results. A residual bootstrapped method generated the simulated datasets by fitting signal data to Autoregressive Moving Average (ARMA) models. Classification models included neural networks, k-nearest neighbors, naïve Bayes, random forest, and support vector machines. A Recursive Feature Elimination (RFE) study was performed to determine if reducing signal features would improve the classification models performance using Gini Importance scoring for the two classes. The top 25% of features determined by Gini Importance scores suggest that profiles from specific Volatile Organic Compounds (VOC) in patient breath may contribute to model performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Symbols

t	wavelength
x_t	signal intensity at wavelength t
$x_{t_{min}}$	minimum signal intensity at wavelength t
$x_{t_{max}}$	maximum signal intensity at wavelength t
x_t'	Min-max normalized signal intensity at wavelength t
$\bar{x}_t$	mean signal intensity at wavelength t
x_{c_t}	mean centered intensity of breath signal at wavelength t
w_{hi}	weight applied to neural network layer at node i of hidden layer h
$\boldsymbol{w}_h$	vector of weights for hidden layer h applied to nodes
y_{hj}	unit value at node j in hidden layer h
$\boldsymbol{y}_h$	vector of unit values for nodes in hidden layer h
z_{hi}	result of applying activation function neural network hidden layer
$\boldsymbol{z}_h$	neural network hidden layer
σ	activation function used in neural network
K	number of classes
p	number of parameters within breath signal (high laser)
I	Gini impurity
$I(\tau)$	Gini impurity at node τ in decision tree
n_i	samples from class for $i = 1, .. K$
N	number of samples
ξ_i	fraction of number of samples from class n_i to number of samples N
S_i	denotes the subset of samples in Recursive Feature Elimination
q	number of lags used to regress against x_{c_t} in Autoregressive process
ϕ_i	coefficient of the i^{th} lag of the series
θ_i	coefficient corresponding to ε_{t-i}
ε_t	represents the white noise error term at time t
Q	denotes the Box-Pierce statistic
Q^*	denotes the Ljung-Box statistic
k	lag being considered for Box-Pierce or Ljung-Box statistic computation
r_k	number of residuals used in the autocorrelation for lag k
l	maximum number of lags being considered Box-Pierce or Ljung-Box statistic
T	denotes the number of observations for r_k residuals
P	number of model parameters used for Box-Pierce or Ljung-Box statistic
r	denotes the order of a Moving Average process or number of past error terms
μ	mean of time series
μ_{C_K}	estimated mean using maximum likelihood estimation for probability distribution
$\sigma^2_{C_K}$	estimated standard deviation using maximum likelihood estimation
$AICc$	Akaike Information Criterion corrected AICc
SSE	residual sum of squares error under the model
C_K	class K used in Naïve Bayes model

CHAPTER 1: INTRODUCTION

In 2020, the SARS-CoV-2 outbreak, also known as COVID-19, has led to a pandemic causing hospitals to overflow with patients resulting in depleted resources—including testing materials and PPE—and more than 5.8 million deaths across the world as of February 2022 (1). Indirect impacts included cancellation of nonemergent and elective surgeries, which in turn negatively affected the quality of provided health care and resulted in loss of hospital revenue in an already stressed healthcare sector (2). Thus, rapid identification of infected individuals and isolating them is essential during this outbreak. Unfortunately, about 40-45% of those who tested positive for the virus are asymptomatic carriers, resulting in many individuals continuing to infect others without realizing they have the disease (3). Currently, nasopharyngeal swabbing to collect viral material for reverse transcriptase PCR (RT-PCR) analysis is the gold standard for COVID-19 testing.

Although typically PCR has high accuracy and specificity, there are many caveats to this method of testing for COVID-19 (4). First, PCR requires adequate sampling to be able to amplify the genetic material of the virus. It is therefore essential to have trained, skilled workers collecting the samples, which can prove challenging for mass testing. The sensitivity of PCR tests can be limited to as low as 60-70% mainly due to incorrect sampling techniques (5). Additionally, one study found that between day 0 and day 10 after infection, the chance of a positive test declined from 94.39% to 67.15% (6). Also, RT-PCR analysis can also take 1-3 days to process and requires an appropriate well-equipped laboratory with skilled technicians, all of which are not always quickly and readily available remotely (7). Furthermore, nasopharyngeal swabbing is an uncomfortable process for many people and can lead to coughing and sneezing, thereby aerosolizing the virus which may lead to further spread (5). Finally, it has been shown that as the disease progresses, the virus multiplies in the lungs rather than the throat (8).

Due to the many disadvantages of using nasopharyngeal swabs and PCR analysis, medical professionals have been looking for more rapid and accurate ways of detecting the virus. Additionally, with limited supplies scientists and engineers have explored novel ways to identify a positive case of COVID-19. One such method that has been studied involves using human exhaled breath as a simple, pain-free, and non-invasive method of screening patients. Here, the interaction between pathogenic viruses in the respiratory tract and the body's microenvironment can produce distinctive volatile organic compounds (VOCs) that the patient exhales in their breath (9, 10). Recently, evaluating the VOCs produced in patient exhaled breath has received an explosion of interest as the analysis of breath constituents as a way of monitoring inflammation and oxidative stress in the lungs (11).

Several studies found that VOCs and their concentration in exhaled breath collected from healthy and diseased human studies, may act as biomarkers of selected diseases or pathophysiological conditions. Of the more than 3000 VOCs present in a patient exhaled breath, the identifiable and potential biologically plausible VOCs include acetone (12), ethanol (13) isoprene (14), methanol (15), methane (16) and aldehydes including acetaldehyde (17), butanal (18) heptanal (19), and propanal (20). Profiles of some of these exhaled breath VOCs reflect the multiple metabolic changes associated with the SARS-CoV-2 viral infection and may be used to rapidly screen for COVID-19 using point-of-care (POC) instruments (21).

Although the gold standard for VOC detection in exhaled breath is gas chromatography, the recent developments in mid-infrared (MIR) laser spectroscopy have led to the promise of compact POC optical instruments enabling single breath diagnostics (22). In this study, one such novel advanced laser-based analytic approach is used as a screening technique: runtime Cavity Ringdown (rtCRD). rtCRD spectroscopy detects trace levels of chemicals in the air including the identification of unique VOCs in patient breath (23). Multiple VOC biomarkers can be observed qualitatively to distinguish the spectrum produced by rtCRD spectroscopy of patients with COVID-19 virus from healthy controls (24). However, current studies of breath analysis of COVID-19 lack sufficient analysis of the multidimensional VOCs data via advanced algorithms such as those in machine learning that may provide better classification performance than visual inspection (21). Therefore, there is a need to assess the efficacy that MIR laser spectroscopy such as rtCRD spectroscopy has in identifying a positive COVID-19 case rapidly and accurately using patient exhaled breath.

In this study, multiple machine learning models were used to classify COVID-19 positive patients versus healthy controls. Using PCR results as ground truth, patient exhaled breath signals generated from an rtCRD spectroscopy device were the only predictors of COVID-19 status. Due to limited sample size, a residual bootstrap procedure was performed from actual patient breath signals to generate simulated samples. The primary outcome of interest of the study was the exhaled breath signals as identifiers of COVID-19 infection in the pre-clinical setting. The objective of this study is to examine the effectiveness of simulating breath signals has on the performance of machine learning classification models with respect to the number of simulated samples. The hypothesis was that as the number of simulated samples increased the average accuracy would increase and the variation of performance would decrease.

CHAPTER 2: SUBJECTS AND MATERIALS USED

Subjects

A preliminary study was conducted at the University of New Mexico Health Science Center in the Department of Emergency Medicine consisting of 18 patients (10 positive; 8 negative cases) from January 1st 2021 through April 30th 2021 with participant information shown in Table 2.1. For classification purposes, a total of 195 patients from Atlanta, Georgia (65 positive; 130 negative cases) at multiple centralized testing locations were enrolled from May 19th, 2021 through June 3rd, 2021; participant information shown in Table 2.2. After giving the informed consent form approved by the respective Institutional Review Boards (IRB), all subjects were deidentified such that subject information cannot be linked to individual participants.

All subjects were non-incarcerated adults, age > 18, and upon enrollment were given unique study IDs. To identify between positive and negative cases of COVID-19, PCR test results were used as the ground truth for binary classification. Each PCR test result was obtained less than 48 hours from the collection of the breath test from an acute care setting or centralized testing location. The patients with invalid or indeterminant PCR results were excluded from the study.

Materials used

To process patient breath samples, a novel advanced laser-based analytical approach was used known as rtCRDS which detects trace levels of chemicals whether in the gas or particle phase (23). The chemical detection by rtCRDS is in the Mid-IR region

(2,500 nm to 25,000 nm) and is commonly used in numerous research areas such as environmental science, exposure assessment and clinical diagnosis (25, 26). The device used is the AG-4000 Breath Test Assembly RingIR® Device shown in Figure 2.1.

	N=10 COVID-19 Positive	**N=8 COVID-19 Negative**
RT-qPCR SARS-COV-2	10	8
% Males	55%	28%
Age		
18-25	0	1
26-40	1	1
41-65	7	7
> 65	2	1
Time Since First Symptom		
Asymptomatic	1	0
1 day	1	3
2-3 days	2	0
4-7 days	4	0
> 1 week	2	0
Comorbidities		
COPD	0	1
Asthma	0	1
Active Malignancy	1	2
Type 2 Diabetes Mellitus	3	2
Smoking	3	4
Symptoms		
Fever	3	0
Cough	5	0
Shortness of Breath	2	2
Recent Loss of Sense of Smell/Taste	2	0
Chills	5	1
Muscle ache	4	0
Headache	5	0
Sore Throat	4	0
Fatigue	5	0
Vomiting/Nausea	2	0
Diarrhea	3	0
Primary Language English	10	7

Table 2.1. UNM Emergency Medicine Department Participant Information (N=18)

	N=65 **COVID-19 Positive**	**N=130** **COVID-19 Negative**
RT-qPCR SARS-COV-2	65	130
Age		
18-25	15	34
26-40	11	43
41-65	26	44
> 65	13	9
Time Since First Symptom		
Asymptomatic	3	0
1 day	10	14
2-3 days	16	31
4-7 days	10	13
> 1 week	15	8
Unknown	11	64
Comorbidities		
Hypertension	25	35
Type 2 Diabetes Mellitus	18	9
Obesity (Excess weight gain)	7	12
Chronic Heart Disease	8	7
Chronic Lung Disease	2	6
Chronic Kidney Disease	2	3
Chronic Liver Disease	1	1
Hemoglobin Disease	2	0
Cancer	4	4
Immunosuppression (From transplant, chemotherapy, medications, or HIV)	2	3
Asthma	6	26
Allergies	6	27
Chronic Sinus Disease	2	3
Other	9	13
None	20	51

Table 2.2. Emory University Testing Site Participant Information (N=195)

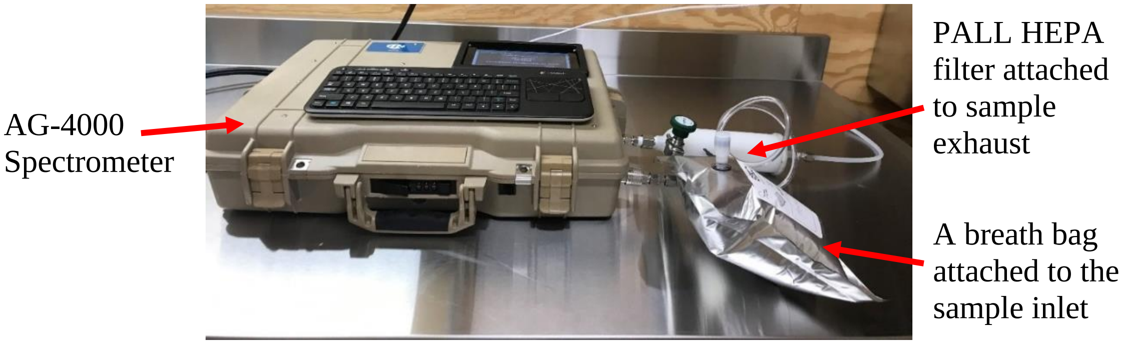

AG-4000
Spectrometer
PALL HEPA filter attached to sample exhaust
A breath bag attached to the sample inlet

Figure 2.1: AG-4000 Breath Test Assembly RingIR Device with Collection Mechanism

This breath test device contains dual lasers of different intensities that are positioned orthogonal to one another and produce the two segments of the spectrum. The wavelength ranges for the low wavelength and high wavelength length lasers are 6800nm to 8600nm and 8600nm to 11,000nm, respectively.

As shown in the Diagram of the Data Collection process, Figure 2.2, the participant exhales into 5 bags of 200 mL in volume each prior to being processed by the device.

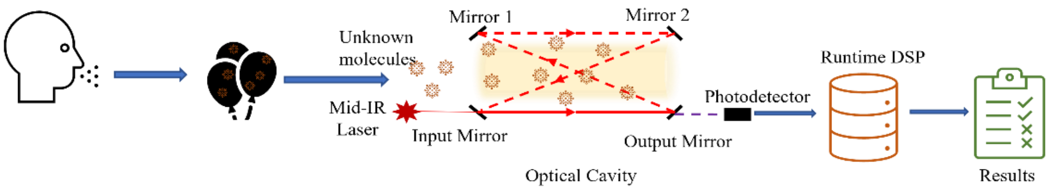


Figure 2.2: Diagram of Data Collection Process

The breath sample from the bag consists of multiple unknown VOC molecules that when excited by the Mid-IR laser within the device produce a fingerprint spectrum unique to the VOCs present. Both Mid-IR lasers emit light that travels through the breath sample multiple times while reflecting off 4 mirrors: an input mirror, mirror 1, mirror 2, and an output mirror. After the light is reflected from the final output mirror, a photodetector is used to collect the light subsequently generating the signal. Finally, a runtime digital signal processing step takes place to produce the signal and this breath signal is then recorded and tabulated with the patients PCR result. The resulting signal spectrum from both lasers spans 6800 nm to 11,000 nm in wavelengths containing at total of 12260 data points. The signal from the low wavelength laser ranges from 6800 nm to 8600 nm wavelengths with 6017 linearly spaced data points. Whereas the signal from the high wavelength laser contains wavelengths between 8600 nm to 11,000 nm data point with 6247 linearly spaced data points.

CHAPTER 3: DATA PREPARATION AND MODELS

Data Preparation

In this study, the dataset is pre-processed to ensure the models are built to effectively classify the signals. An initial model performance comparison is performed to determine if the entire signal should be used for an extensive analysis or subsets containing the low or high wavelength laser data only. Before any model building took place, the dataset was normalized and augmented by reducing features, encoding the classification variable, and split into folds for cross-validation. To reduce signal noise, a background correction of the signals was considered as to further pre-process the signal data. This involves using the clean air samples obtained from the ambient or background air within the hospital/testing center to then normalize the breath samples. However, due to variation in the power of the low wavelength laser, background or ambient signals consisted of various number of data points ranging between 6010-6016 making background correction untenable. Thus, background correction of signals or the utilization of the background spectra for the classification of COVID-19 status was not implemented. For comparison purposes only, a linear interpolation of the background signals (6017 data points) was applied such that, for the high wavelength laser, both the breath and background signals contain 6247 data points.

Min-Max Normalization

Machine learning models trained on scaled data usually have significantly higher performance compared to models trained on unscaled data making rescaling data an essential step for preprocessing data (27). Before model building, a minimum-maximum normalization was applied to the entire signal dataset. This normalization transforms the data linearly by setting the minimum value for each wavelength within the dataset to zero and the maximum value to one. This transformation is shown in the following formula:

$$x_t' = \frac{x_t - x_{t_{min}}}{x_{t_{max}} - x_{t_{min}}}$$

where $x_{t_{min}}$ and $x_{t_{max}}$ denotes the minimum and maximum of a variable in the samples at wavelength t, respectively, and the value x_t is mapped to the normalized value x_t'. This normalization step tends to improve model performance of neural networks and is typical for machine learning models prior to training (28). The disadvantages of the min-max algorithm are that it is sensitive to outliers and if the unseen/testing samples fall outside the training data range of the variable, the scaled values will be outside the bounds of the interval $[0, 1]$. Note that x_t' is not used for signal simulation rather only for model building; x_{c_t} is transformed to x_t before being min-max normalized to x'_t.

One Hot Encoding and Softmax Activation Function

For the classification of the labels, a one-hot encoding scheme was applied to both class labels where $[0, 1]$ represents a positive case and $[1, 0]$ represents a negative case. This step allows for a Softmax function at the final two nodes of neural network models to return probabilities of each class (29). This function normalizes the output of the neural network to a probability distribution over the predicted output classes. The Softmax function or normalized exponential function is a generalization of the logistic function (30) as follows:

$$\sigma(\boldsymbol{z}_h)_i = \frac{e^{z_{hi}}}{\sum_{i=1}^{K} e^{zh_i}}$$

where K is the total number of classes and $\boldsymbol{z}_h = (z_{h1}, \dots, z_{hN}) = w_{h0}y_{h0} + w_{h1}y_{h1} + \cdots + w_{hN}y_{hN} = \sum_{j=1}^{N} w_{hj}y_{hj} = \boldsymbol{w}_h^T\boldsymbol{y}_h \in \mathbb{R}^N$ is the input vector from the last hidden layer h of the neural network for nodes $j = 1, 2, \dots N$ with w_{hj} being the weight that is multiplied by the feature y_{hj} at node j. In other words, the Softmax activation function obtains a class probability from the model by applying the exponential function to each element of $\boldsymbol{z}_h$, then dividing by the sum of all the exponentials such that the sum of all $\sigma(\boldsymbol{z}_h)_i$'s is one for $i = 1, \dots, K$.

Gini Impurity and Gini Feature Importance

During training of the Random Forest classifier, each node within the binary trees must obtain the optimal split through what is known as Gini impurity. Gini impurity $I(\tau)$ is calculated as follows:

$$I(\tau) = 1 - \sum_{i=1}^{K} \xi_i^2$$

Here, $\xi_i = \frac{n_i}{N}$ is the fraction of n_i samples from class $i = \{0, 1\}$ out of the total samples N at node τ. Gini impurity approximates Shannon entropy which measures the quality of a potential split separating the samples of two classes at the node of interest (31). This provides insight into which features may be important for the model to classify data known as Gini Feature Importance.

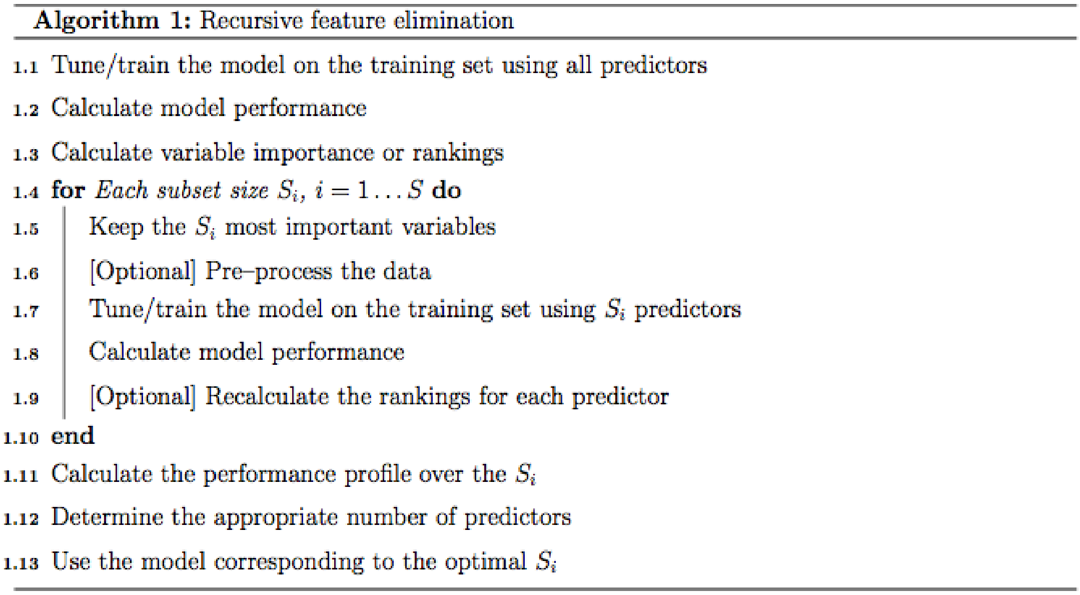
Gini Feature Importance is a feature selection based on the Random Forest classifier and provides multivariate feature importance scores. To compute the Gini Feature Importance for each feature, the accumulated sum of the Gini decrease across every tree of the forest is computed for each time a feature is chosen to split a node. This accumulated sum is then divided by the number of trees in the forest to obtain an average. These averages are representing the Gini importance and are unitless relative values. The feature with the greatest importance being the most influential in classifying the data for the Random Forest model (32).

For learning problems involving spectral data, the high dimensionality of the feature space denoted p may be much greater than that of the number of N samples available for training. Dimension reduction and feature selection of the spectral data help remove multi-collinearity to improve the interpretation of the parameters of the machine learning model. Also, it makes it easier to visualize the data when reduced to relatively low dimensions such as 2D or 3D, and aid in noise reduction (32). An iterative feature reduction was used to illustrate the effect that Gini Feature Importance has on model performance. This involved decreasing the number of features present for training and testing by 1% until only 3 features remained.

Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a backward selection process that aims to reduce the number of uninformative features or variables within a dataset to improve a model’s performance (33). The main goals of feature selection are to determine the important variables related to the outcome variable and obtain a minimal set of variables that give a good predictive model that is not overfitted and able to generalize to new datasets (34). As shown in Figure 3.1, RFE begins by fitting the model with all predictors and subsequently ranks the predictors according to the importance the predictor has for that model.

Let S be the subset size of the candidate predictors to be retained by RFE such that ($S_1 > S_2 \ldots > S_i$) where i denotes the current iteration. After the features have been ranked, RFE will retain the top S_i ranked predictors to then refit the model and access the performance. The goal is to find the S_i which achieves the best model performance. The algorithm may recompute the predictor rankings of the reduced predictor subset during each iteration as well as renormalize the subset S_i before model re-evaluation.



```
Algorithm 1: Recursive feature elimination
Tune/train the model on the training set using all predictors
Calculate model performance
Calculate variable importance or rankings
for Each subset size $S_i$, $i = 1 \ldots S$ do
    Keep the $S_i$ most important variables
    [Optional] Pre–process the data
    Tune/train the model on the training set using $S_i$ predictors
    Calculate model performance
    [Optional] Recalculate the rankings for each predictor
end
Calculate the performance profile over the $S_i$
Determine the appropriate number of predictors
Use the model corresponding to the optimal $S_i$
```

Figure 3.1: Recursive Feature Elimination Algorithm 1 (adapted from 33)

It has been shown that for random forest models, there was a decrease in performance when rankings were recomputed (35). It is not clear if this is the case for other machine learning models. Overfitting may be an issue if the predictor sets focus on features in the training data not found in testing samples, e.g., uninformative predictors or predictors that randomly correlate with the outcome (36). Thus, the RFE algorithm may have a selection bias giving good rankings to variables with the prediction error being lowered yet a different validation set may determine that the predictor was uninformative

(37). In this case, the decrease in gain in Gini Index from the Random Forest model with 3000 estimators was used for a stratified 10-fold average ranking of feature importance

k-fold Cross-Validation and Stratified k-fold Cross-Validation

A k-fold cross-validation involves a resampling procedure applied to the entire dataset to cross-validate the testing of the machine learning models. This cross-validation involves a series of k folds which split the data into training and testing sets for the machine learning model to be trained and evaluated, respectively. After the evaluation of each fold, the data is then randomized and split again for the next fold. This statistical method is used to estimate the skill of models by taking the average performance of the k-folds as a final measure of the quality of the model. Figure 3.2 diagrams the procedure of the k-fold cross-validation below.

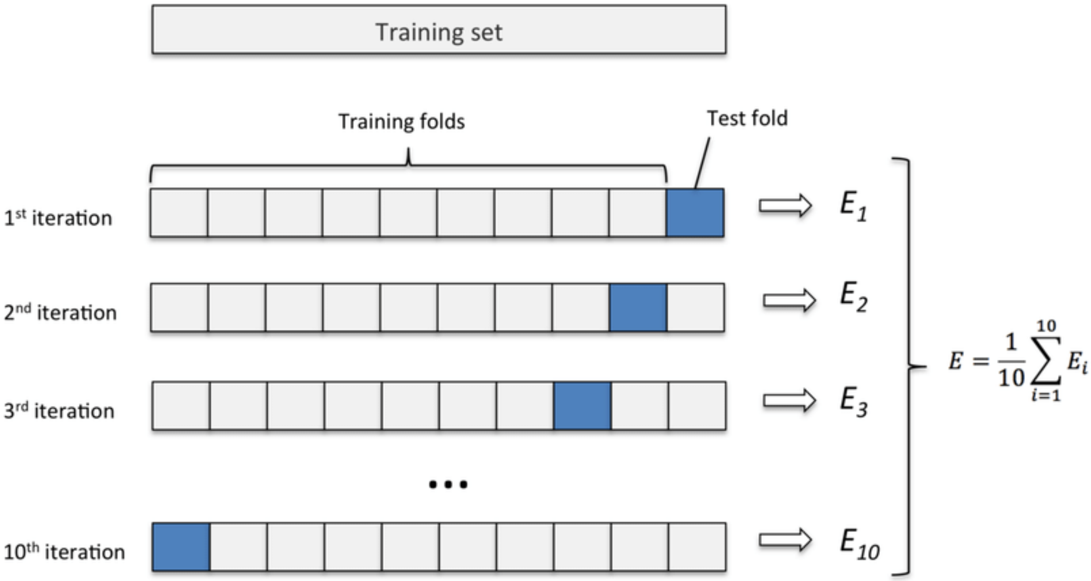


Figure 3.2: Diagram of k-fold Cross-Validation Procedure (adapted from 38)

A caveat of k-fold cross-validation is that when changing the random state, the accuracy of the models can change noticeably. This may suggest that the variation in the distribution of classes selected for the training and testing sets can affect performance and may not be a representative sample.

To address this issue, a stratified 10-fold cross-validation can be used where each fold is stratified such that they are representative of all strata in the data. This reduces variance among the estimates and the average error estimate is reliable (39). In other words, stratified 10-fold cross-validation prevents bias in a classification where each instance is weighted equally without the overrepresented classes being assigned more weight. This ensures that the data is randomly sampled with the distribution of classes remaining relatively constant. In this study, a 2:1 ratio of healthy controls to positive COVID-19 patients is used. The stratified k-fold cross-validation splits the dataset into training and testing datasets to maintain this ratio.

Models

This study utilizes time series models to simulate signal data as well as multiple statistical and machine learning models to classify signals. For the purposes of simulation, signals were mean centered, i.e., intensities at each wavelength were subtracted by the mean intensity of the signals at that wavelength. To fit time series models, the wavelength of the signal was used as the time independent variable t with the mean centered intensity as the dependent variable $x_t - \bar{x} = x_{c_t}$. The models used were limited to a particular case of Autoregressive Integrative Moving Average (ARIMA) models, also called Box-Jenkins models, which does not apply differencing to the data. Without differencing, the models are a combination of Autoregressive (AR) and Moving Average (MA) also known as ARMA models. After the simulated signals were generated, supervised and unsupervised machine learning models classify the signals as either a COVID-19 positive or COVID-19 negative case.

Autoregressive, Moving Average, and Autoregressive Moving Average Models

AR models are used in forecasting when there appears a to be correlation between current values and previous values in the same time series. AR processes can be considered a linear regression of the time series data against one or more of the previous values (40). In other words, the AR process is used to define the current value of a time series, x_{c_t}, as a linear combination of the previous q lags of the series as formalized by:

$$AR(q):\ \ x_{c_t} = c + \sum_{i=1}^{q} \phi_i x_{c_{t-i}} + \varepsilon_t$$

where $AR(q)$ denotes the AR process with q-order, c represents a constant, q is the number of lags that regress against x_{c_t}, $x_{c_{t-i}}$ is the i^{th} lag of the series, ϕ_i is the coefficient of the i^{th} lag of the series and ε_t represents the white noise error term at time t. The error term ε_t is a white noise process by the assumption that the term is uncorrelated with the time series data with mean 0 and constant variance σ^2, i.e., $\varepsilon_t \sim WN(0, \sigma^2)$.

The Box-Pierce or Ljung-Box statistic is used to test the assumption that the residuals do not have any outliers or patterns such as an increasing trend, i.e., resembling white noise. These statistics examine the null hypothesis that there is independence in a given time series and is sometimes known as 'portmanteau' tests since they test for a group of autocorrelations (40). The Box-Pierce test statistic is $Q = T \sum_{k=1}^{l} r_k^2$, where l is the maximum lag being considered, r_k is the autocorrelation for lag k, T is the number of observations for r_k residuals. Values of l tend to be $l = 10$ for non-seasonal data and $l = 2m$ for seasonal data with m being the period of seasonality (40). The Ljung-Box test tends to be the more accurate test than the Box-Pierce with the test statistic $Q^* = T(T+2) \sum_{k=1}^{l} (T-k)^{-1} r_k^2$ where values of Q^* come from a χ^2 distribution with $(l - P)$ degrees of freedom with P being the number of model parameters. Note that for our purposes $P = 0$ since the test is calculated from raw data rather than residuals from the model (40).

Another assumption of the AR model is that an AR process can be included in the model if and only if the time series is a stationary process (41). In the context of time series data, a stationary process describes a stochastic state of the series. This assumption is based on the Wold representation theorem, which states that a linear combination of white noise can represent a stationary process. In this case, the mean and variance of the series do not change over time and the correlation structure of the series including its lags remains the same over time.

In an MA model, the values of the univariate time series, x_{c_t} depend linearly on the current and various past values of a stochastic term ε_t such that ε_t contains some information within the model residuals over time. In other words, by modeling the relationship between x_{c_t} with the error term ε_t and past r error terms of the models, an MA process can capture time series patterns over time. An MA process with r-order is defined in the following:

$$MA(r):\ \ x_{c_t} = \mu + \varepsilon_t + \sum_{i=1}^{r} \theta_i \varepsilon_{t-i}$$

with $MA(r)$ denoting an MA process with r-order, μ represents the mean of the series, $\varepsilon_{t-r}, \ldots, \varepsilon_t$ are white noise error terms, θ_i the coefficient corresponding to ε_{t-i}, and r is the number of past error terms that are used in the equation.

There are two ways that AR models and MA models differ. First, in an $AR(q)$ model, only the ε_t error term is present and not previous error terms to estimate x_{c_t}. In contrast, an $MA(r)$ model, the error term(s) ε_{t-r} are factored into the current estimation of x_{c_t} (40). Additionally, the two models differ in that the AR model, a x_{c_t} value affects values infinitely far into the future since ε_t affects x_{c_t}, which affects $x_{c_{t+1}}$, which affects $x_{c_{t+2}}$, and so on. In the MA model, the value x_{c_t} affects only the r subsequent values in the series (41).

An ARMA model, combines both AR and MA models to handle more complex time series data. For stationary time series, an $ARMA\ (q,r)$ model is used where q denotes the AR parameters and r represents the MA parameters in the following formula:

$$ARMA\ (q,r): \quad x_{c_t} = c + \sum_{i=1}^{q} \phi_i x_{c_{t-i}} + \sum_{i=1}^{r} \theta_i \varepsilon_{i-i} + \varepsilon_t$$

with x_{c_t} being the time series, c is a constant or drift, ϕ_i is the coefficient of the i^{th} lag of the series, $x_{c_{t-i}}$ is the i^{th} lag of the series, q defines the number of lags to regress against x_{c_t}, θ_i corresponds to the coefficient of ε_{t-i}, r is the number of past error terms in the model with white noise error terms $\varepsilon_{t-r}, \dots, \varepsilon_t$.

To fit each 195 mean centered signals separately, appropriate q and r values for the ARMA model were obtained using R's 'auto.arima' function in the 'forecast' package with the differencing 'max.d' set to 0 to allow for only ARMA models to be consider. Model selection criteria included selecting the model with the minimal Akaike Information Criterion corrected (AICc) given by:

$$AICc = \log(\hat{\sigma}_p^2) + \frac{N+p}{N-p-2}$$

where $\hat{\sigma}_p^2 = \frac{SSE(p)}{N}$ with p being the number of parameters in the model, N is the sample size and $SSE(p)$ is the residual sum of squares error under the model. In other words, of the models used to fit for a single signal, the minimum AICc was the selection criteria used. Thus, giving 195 unique ARMA models for the signal dataset. The AICc is a modification of AIC for the small ratio of sample size to number of parameters in the model $\left(\frac{N}{P}\right)$ to prevent overfitting (42).

Residual Bootstrapping

The bootstrap of Efron is a powerful nonparametric tool for approximating the sampling distribution and variance of statistics based on independently identically distributed (iid) observations (43). In residual bootstrapping, a fitted value from a model estimate is obtained along with the model residuals. The residuals are then resampled with replacement before adding them to the fitted value to create a simulated sample (40). This assumes that the residuals are uncorrelated with constant variance meeting the bootstrap criteria that the distribution of residuals are $WN(0, \sigma^2)$. By repeating the residual bootstrapping process, we can replace each of the residuals by sampling from the collection of residuals to create the new simulated observation (40).

In this case, we are treating the spectra as time series data, with the assumption that the subsequent signal intensity errors will be similar to previous intensities errors in the same spectra. It is important to note that each signal has unique fitted values and a unique distribution of residuals from the $ARMA\ (q, r)$ model to be resampled from, i.e., the residuals from one signal are not added to fitted values from another signal. These fitted values $\hat{x}_{c_t}$ from the model estimate with the residuals $\hat{\varepsilon}_t = x_{c_t} - \hat{x}_{c_t}$, for $t = 1, \ldots, p$ can then be used for a residual bootstrapping method generating a simulated breath signal. The random resampling with replacement from the distribution of residuals $\hat{\varepsilon}_t$ for $t = 1, \ldots, p$ created the simulated signals denoted as $x_{c_t}^* = \hat{x}_{c_t} + \hat{\varepsilon}_t^* = c + \sum_{i=1}^{q} \phi_i x_{c_{t-i}} + \sum_{i=1}^{r} \theta_i \varepsilon_{t-i} + \hat{\varepsilon}_t^*$ with $\hat{\varepsilon}_t^*$ being the randomly resampled or residual bootstrapped sample. The simulated dataset retained the class ratio of the original dataset i.e., 2 negative cases for every 1 positive case. Also, the simulated signals samples were randomly sampled with replacement until multiples of the original dataset (i.e., 195, 380, 585, 780, and 975 total samples) were obtained for model fitting.

Neural Networks

Recently, there has been a revival of the neural network model revolutionized the fields of speech recognition (44), computer vision (45), natural language processing (46). Neural networks - also referred to as artificial neural networks (ANN) or multilayer perceptrons (MLP) – are supervised machine learning models that can represent complex nonlinear relationships within input datasets optimizing for classification or regression models (47).

In this case, the MLP model using Scikit-learn library learns a function $f: \mathbb{R}^p \rightarrow \mathbb{R}^K$ by training on a dataset where p is the number of dimensions for the input signal and K is number of classes. The input layer consists of a set of neurons $x_1, x_2, \ldots, x_p$ which represents the input features. Neurons within each of the hidden layers will transform these values by a weighted linear sum of the form $w_1x_1 + w_2x_2 + \cdots + w_px_p$. After assigning these weights, a nonlinear ReLu activation function $\sigma_i: \mathbb{R} \rightarrow \mathbb{R}$, where $\sigma_i(x) = \max{(0, x)}$ and i is the number of hidden layers transforms these values to then be processed by another hidden layer or the terminal neurons. The hidden layer values and respective weights in the network are denoted as described above. To compute the probability of being in either class, the SoftMax function can be applied to the final output layer. This requires that a threshold probability be used to compute the network's error by comparing probability predicted by the network and a specified threshold. The network error is then used for a process known as backpropagation to update the network weights before the next iteration of training. In this case, 66% is the specified threshold since the network must perform better than random chance of selecting all samples as being a negative case in the 2:1 unbalanced dataset.

MLP models are capable of learning non-linear models and models in real-time such as online learning. The disadvantages of the MLP are that it must have a non-convex loss function and if there exists more than one local minimum, then different random weight initialization can lead to different validation accuracies. To track the learning rate and prevent model overfitting, a modifiable neural network using the open-source software library Keras was used. Summarized in Table 3.1 are the hyperparameters of the neural network that when tuned can help improve model performance and prevent overfitting of the dataset.

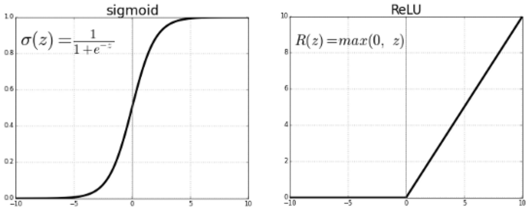
Hyperparameter	**Description**
Number of Neurons and Number of Hidden Layers	Adjusted to the solution complexity where more complex solutions may require more neurons/hidden layers
Learning rate	Adjusts the model in response to the estimated error or loss for each time the neural network model weights are updated
Regularization	Reduces overfitting of the training data by penalizing the coefficients contained within the weight matrices of the nodes
Dropout (%)	Randomly ignores a percentage of neurons during training to prevent overfitting (48)
Callback	Perform actions at various stages of training such as penalizations if the learner does not improve after a specified number of epochs
Activation Functions  **Figure 3.3:** Sigmoid and ReLU Activation Functions	Helps to introduce nonlinearity if there is a nonlinear function such as hyperbolic tangent, arctangent, sigmoid, and exponential linear weighted. Softmax composed of exponential functions produces probabilities and ReLu functions are commonly used on neurons in hidden layers of neural networks

Table 3.1: Summary of Neural Network Hyperparameters Tuned with Descriptions

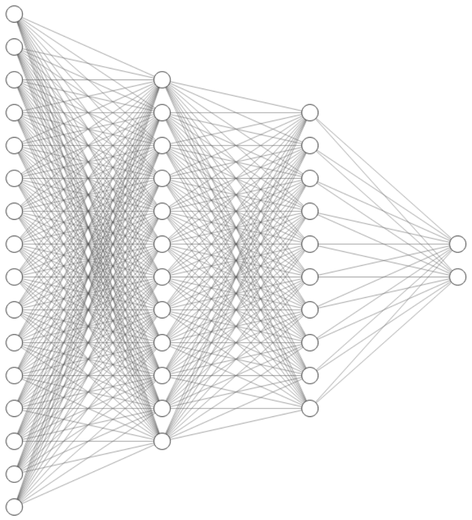


Figure 3.4: Example of 2-Hidden Layer Neural Network with Two Output Neurons

k-Nearest Neighbor

In the k-Nearest Neighbor (k-NN) model, the goal is to predict the label of a class for a new point by using the k number of neighbors around a certain point using number of training samples that are closest in distance to the new point (50). In other words, to classify a new or test case, k-NN computes a majority vote of the k nearest neighbors of each point nearest to the test case. The test case is assigned the data class that has the most representatives of that class. Here, the distance can be any metric measure, but typically Euclidean distance is used.

In contrast to the other models presented here, the neighbors-based classification is a type of instance-based learning. It does not attempt to construct a general internal model but rather it stores instances of the training data to make a classification of the testing data. Note that we cannot use $k = 1$ because if an outlier exists the classification will erroneously classify the point as a class.

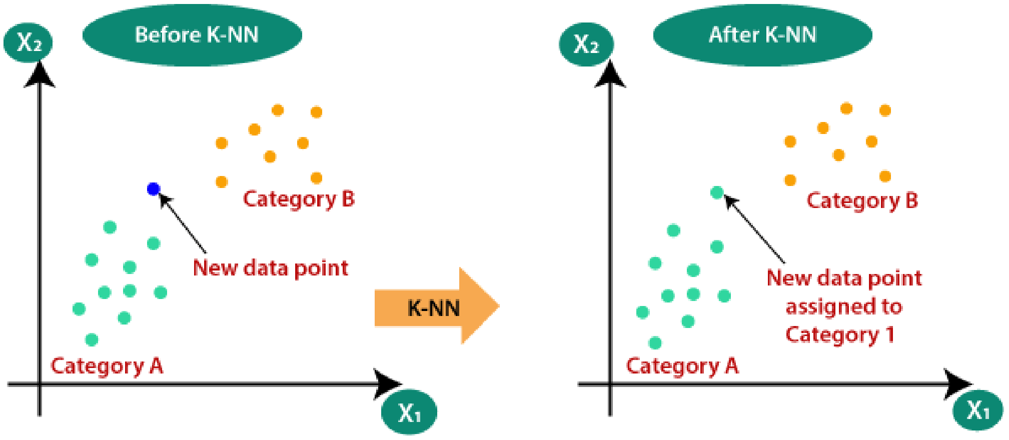


Figure 3.5: k-Nearest Neighbor Classification Plot of new data point (adapted from 51)

Naïve Bayes

The Naïve Bayes (NB) classifier is a probabilistic machine learning model which applies Bayes' theorem to obtain the conditional probability that a sample belongs to a class given a set of predictors. It is mostly used in sentiment analysis, spam filtering, and recommendation systems. Advantages of NB algorithms are that they are fast and easy to implement. The disadvantage is that the classifier has the "naïve" assumption such that predictors are required to be independent given the class (52). In cases where the predictors are dependent, performance is hindered.

To derive the NB model classification, we begin with Bayes' theorem which is written as:

$$\Pr(C_K|\boldsymbol{x}) = \frac{\Pr(C_K)\Pr(\boldsymbol{x}|C_K)}{\Pr(\boldsymbol{x})}$$

where $\boldsymbol{x} = (x_1, \ \dots, \ x_p)$ is the set of predictors, $\Pr(C_K|\boldsymbol{x})$ is the conditional probability that an instance is classified as class C_K for K classes given $\boldsymbol{x}$, $\Pr\ (C_K)$ is the prior probability of observing class C_K, Pr $(\boldsymbol{x}|C_K)$ is the probability of having $\boldsymbol{x}$ predictors given the data is from class C_K, and Pr $(\boldsymbol{x})$ is the probability of observing the data **x** with the p predictors. Since NB assumes that the predictors $\boldsymbol{x}$ are independent variables, we can substitute $\boldsymbol{x} = (x_1,\ \ldots,\ x_p)$ and expand Bayes Theorem to get:

$$\Pr\left(C_K \middle| x_1,\ \ldots,\ x_p\right) = \frac{\Pr(C_K)\Pr(x_1|C_K)\Pr(x_2|C_K)\ldots\Pr(x_p|C_K)}{\Pr(x_1)\Pr(x_2)\ldots\Pr(x_p)}$$

Here, we notice that the denominator is the same for all entries in the dataset and thus we can obtain the following proportionality:

$$\Pr\left(C_K \middle| x_1,\ \ldots,\ x_p\right) \propto \Pr(C_K)\prod_{i=1}^{n}\Pr\,(x_i|C_K)$$

To obtain the NB model classification, we must find the class C_K with the maximum probability:

$$C_K = argmax_{C_K}\Pr(C_K)\prod_{i=1}^{p}\Pr(x_i|C_K)$$

We assume that the distribution of Pr $(x_i|C_K)$ is Gaussian and therefore implement the Gaussian NB algorithm for classification. This means that the likelihood of the features x given C_K are:

$$\Pr(x_i|C_K) = \frac{1}{\sqrt{2\pi\sigma_{C_K}^2}}\exp\left[-\frac{\left(x_i - \mu_{C_K}\right)^2}{2\sigma_{C_K}^2}\right]$$

where the parameters $\sigma_{C_K}^2$ and μ_{C_K} are estimated using the maximum likelihood estimation of the assumed probability distribution. As with NB, the prior probability of each class is required to represent the distribution in terms of its mean and standard deviation (53).

Random Forest

Random forest (RF) is a nonparametric supervised learning technique used for classification and regression. RF models are built from Decision Trees where the Decision Tree classifies the sample based upon the gain in information entropy or gain in Gini index. To train the RF model, the technique of bootstrap aggregating, or bagging, is applied to several Decision Trees. Here, the training set is bagged repeatedly by selecting a random sample with replacement of the training set before fitting the trees to the samples. The subset of the data that is not used for training is known as the out of bag (OOB) sample. The OOB is used for evaluation of the model's performance by a cross-validation method determining an unbiased generalization error (27). The predictions for the OOB samples are then made by either averaging the prediction for all the tree in the case of regression or taking the majority vote in the case of classification trees. This is a strength of the RF model since a single tree is sensitive to noise in the training set and the average of many trees is not sensitive to noise if the trees are not correlated (27). Thus, due to the bootstrapping procedure, the RF model performs better with decreased variance than the Decision Tree model to generate classification predictions.

Support Vector Machines

A Support Vector Machine (SVM) finds a hyperplane which best separates the classes of interest by using what are known as support vectors as shown in Figure 3.5. These are the data points closest to the hyperplane from both classes and help to form a negative and positive hyperplane. A hyperplane is an $(n-1)$-dimensional subset of an n-dimensional Euclidean space dividing it into two disconnected parts. The distance from the support vectors is known as the margin which the SVM algorithm maximizes to obtain the best decision boundary. This decision boundary is the maximum margin hyperplane that is parallel to both the negative and positive hyperplanes.

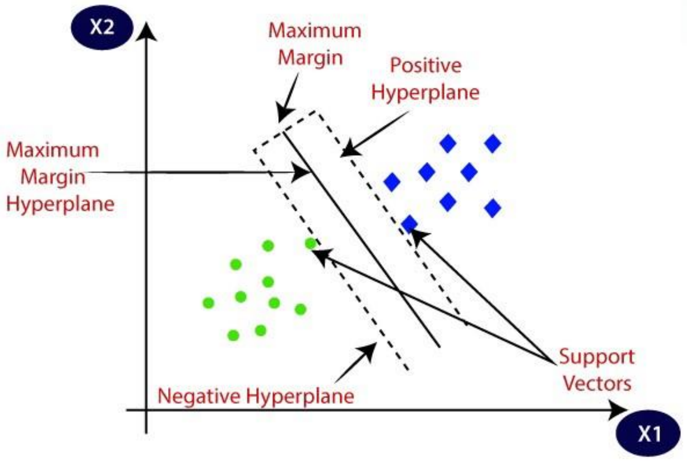


Figure 3.6: Plot Illustrating a 2D SVM (adapted from 43)

In terms of performance, SVMs achieve high accuracies on smaller cleaner datasets in a reasonable amount of time. SVMs can take longer to find the optimal hyperplane on larger noisier datasets with overlapping classes (50). Since non-linearly separable datasets are difficult to separate using a linear hyperplane, the SVM algorithm can utilize the kernel trick (54) to find the best non-linear hyperplane. In Figure 3.6, a SVM using a Gaussian radial basis function separates the data obtaining the maximum margin for the non-linear separation of two classes.

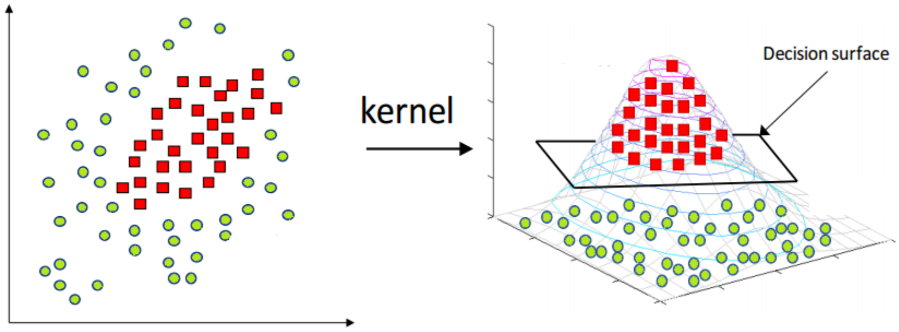


Figure 3.7: Example of the SVM Kernel Trick applied to 2D dataset (adapted from 43)

CHAPTER 4: RESULTS

In this study, the primary objective was to compare the accuracy from each model using the original dataset as well as the simulated dataset generated from the ARMA model residual bootstrap. First, we must check the assumption that COVID-19 breath signals have different spectra from healthy controls, i.e., patients who test positive for COVID-19 differ from those that test negative with respect to their breath signals. In Figure 4.1, we see the results from the preliminary investigation using the AG-4000 device of background corrected exhaled breath signals of these two groups.

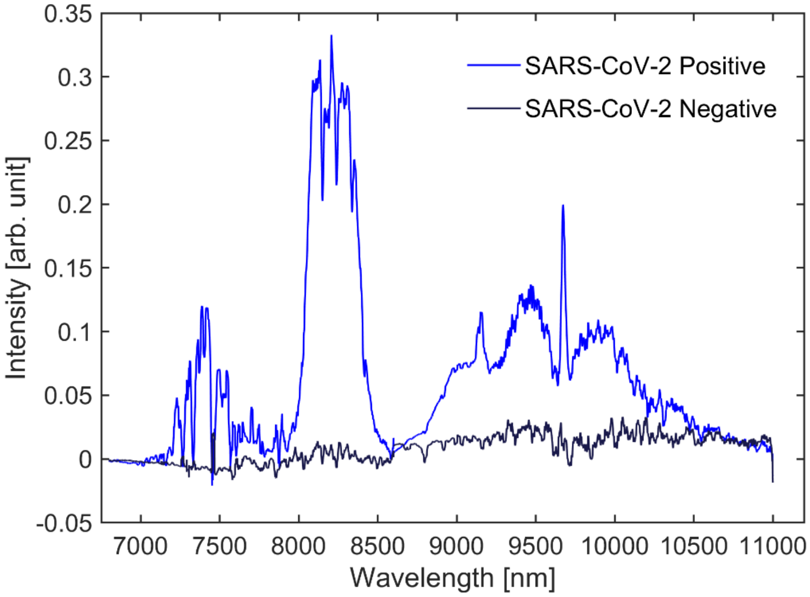
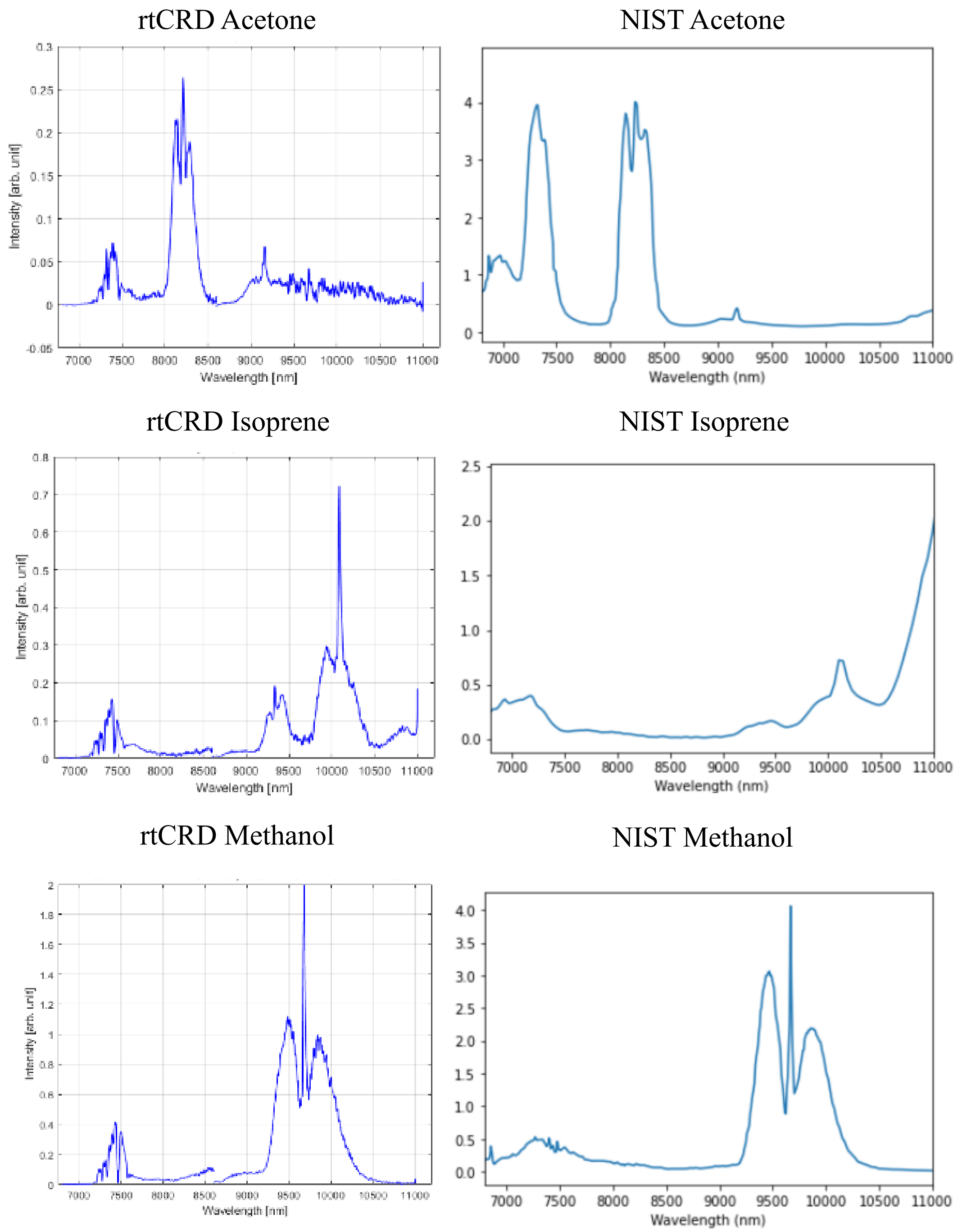


Figure 4.1: Exhaled Breath Signals for SARS-CoV-2 Positive & Negative Subjects

Both low and high wavelength lasers show spectra from SARS-CoV-2 positive patient breath samples that differ in intensity at various wavelengths as compared to the SARS-CoV-2 negative breath sample. Note that these signals are not representative of their respective sample classes and are used only for comparison purposes.

In Figure 4.2, rtCRD and the National Institute of Standards and Technology (NIST) comparable VOC spectra for three VOCs are presented.


rtCRD Acetone

NIST Acetone

rtCRD Isoprene

NIST Isoprene

rtCRD Methanol

NIST Methanol

Figure 4.2: rtCRD and NIST VOC Spectra of Acetone, Isoprene, and Methanol

Both sources produce peaks at similar wavelengths with overlapping waveforms across the mid-IR region. As shown in Figure 4.3, the overlayed plots of Acetone and Methanol VOCs and a SARS-CoV-2 positive breath sample show corresponding peaks at similar wavelengths suggesting the presence of VOCs.

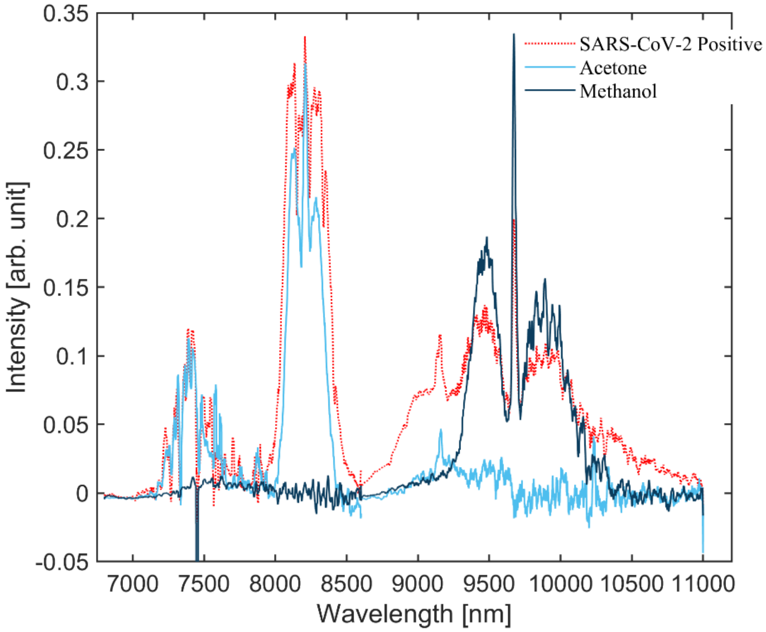
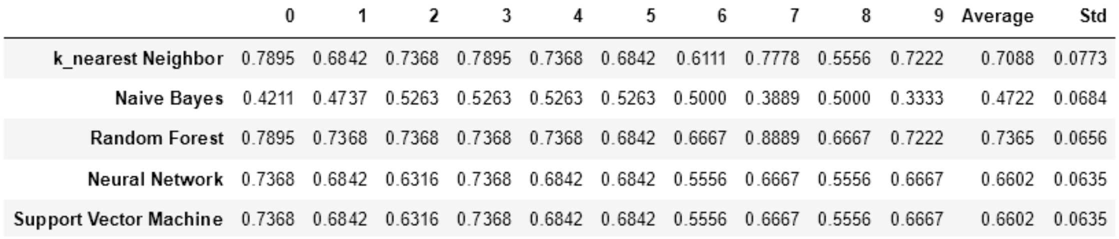


Figure 4.3: Acetone and Methanol VOC spectra with SARS-CoV-2 Positive Signal

Comparison of Performance Between Lasers

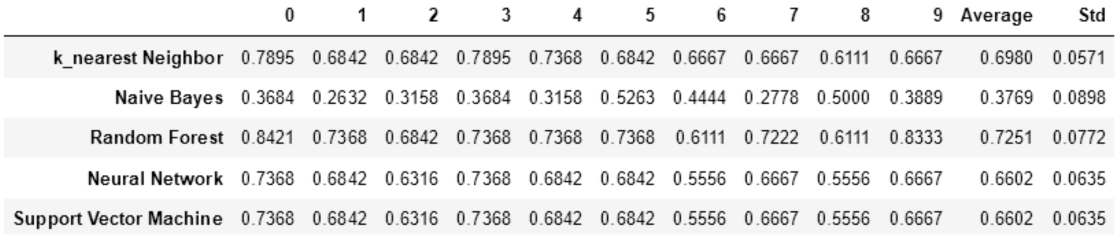
Using the original 195 samples, a 10-fold cross validation of the five models was implemented using the low wavelength laser only, the high wavelength laser only and both the high and low wavelength lasers (Table 4.1). Most models performed best when using the high wavelength laser only for binary classification.

Both High & Low Wavelength Lasers



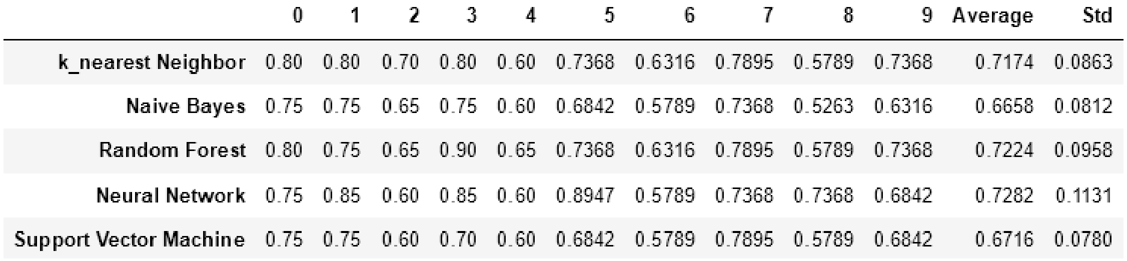
	0	1	2	3	4	5	6	7	8	9	Average	Std
k_nearest Neighbor	0.7895	0.6842	0.7368	0.7895	0.7368	0.6842	0.6111	0.7778	0.5556	0.7222	0.7088	0.0773
Naive Bayes	0.4211	0.4737	0.5263	0.5263	0.5263	0.5263	0.5000	0.3889	0.5000	0.3333	0.4722	0.0684
Random Forest	0.7895	0.7368	0.7368	0.7368	0.7368	0.6842	0.6667	0.8889	0.6667	0.7222	0.7365	0.0656
Neural Network	0.7368	0.6842	0.6316	0.7368	0.6842	0.6842	0.5556	0.6667	0.5556	0.6667	0.6602	0.0635
Support Vector Machine	0.7368	0.6842	0.6316	0.7368	0.6842	0.6842	0.5556	0.6667	0.5556	0.6667	0.6602	0.0635

Low Wavelength Laser Only



	0	1	2	3	4	5	6	7	8	9	Average	Std
k_nearest Neighbor	0.7895	0.6842	0.6842	0.7895	0.7368	0.6842	0.6667	0.6667	0.6111	0.6667	0.6980	0.0571
Naive Bayes	0.3684	0.2632	0.3158	0.3684	0.3158	0.5263	0.4444	0.2778	0.5000	0.3889	0.3769	0.0898
Random Forest	0.8421	0.7368	0.6842	0.7368	0.7368	0.7368	0.6111	0.7222	0.6111	0.8333	0.7251	0.0772
Neural Network	0.7368	0.6842	0.6316	0.7368	0.6842	0.6842	0.5556	0.6667	0.5556	0.6667	0.6602	0.0635
Support Vector Machine	0.7368	0.6842	0.6316	0.7368	0.6842	0.6842	0.5556	0.6667	0.5556	0.6667	0.6602	0.0635

High Wavelength Laser Only



	0	1	2	3	4	5	6	7	8	9	Average	Std
k_nearest Neighbor	0.80	0.80	0.70	0.80	0.60	0.7368	0.6316	0.7895	0.5789	0.7368	0.7174	0.0863
Naive Bayes	0.75	0.75	0.65	0.75	0.60	0.6842	0.5789	0.7368	0.5263	0.6316	0.6658	0.0812
Random Forest	0.80	0.75	0.65	0.90	0.65	0.7368	0.6316	0.7895	0.5789	0.7368	0.7224	0.0958
Neural Network	0.75	0.85	0.60	0.85	0.60	0.8947	0.5789	0.7368	0.7368	0.6842	0.7282	0.1131
Support Vector Machine	0.75	0.75	0.60	0.70	0.60	0.6842	0.5789	0.7895	0.5789	0.6842	0.6716	0.0780

Table 4.1: 10-Fold Model Accuracy and Standard Deviation by Laser

For the high wavelength laser, neural network (89.47%) and random forest (90%) models achieved the highest performance for different folds of data. The greatest 10-fold average accuracy was observed with the neural network (72.82%) model being slightly greater than the random forest (72.24%) model. In the neural network model, the standard deviation in fold accuracy is the greatest among the models using the high wavelength laser only (0.1131%) twice that of dual or other single laser case (0.0635%). Overall, the naïve Bayes model performs poorly using both high & low wavelength lasers (47.22%), low wavelength laser only data (37.69%), and high wavelength laser only (66.58%). The 10-fold average accuracy for k-nearest neighbors (71.74%) and support vector machine models (67.16%) improve slightly when using high wavelength only laser data. Support vector machine and neural network models do not perform better than chance 66.67% for the datasets that were generated using the low wavelength laser. Note: the remainder of the study uses the high wavelength laser data only.

ARMA Residual Bootstrap Dataset

Using R's 'auto.arima' with 'max.d' set to 0 preventing differencing, an ARMA model was fit for each of the 195 sample signals after mean centering the signal by the signal's class mean giving 195 ARMA models.

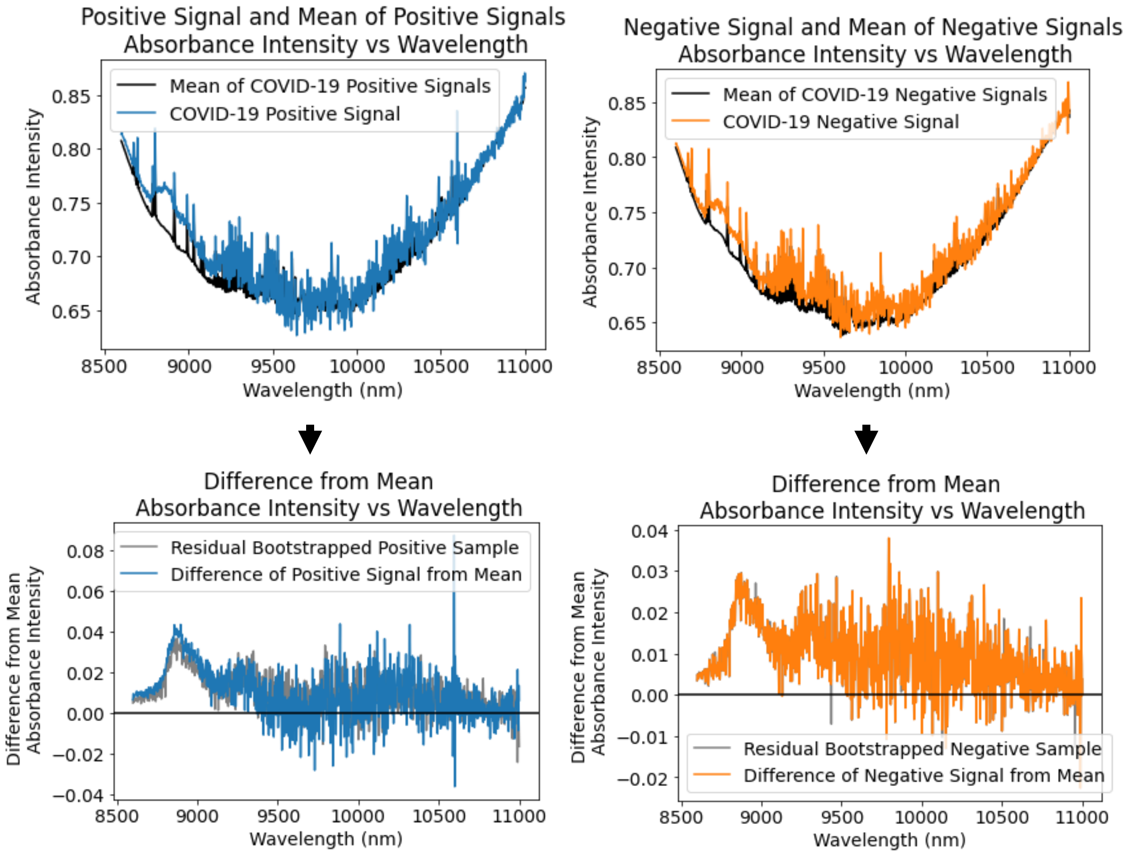


Figure 4.4: Plots of Positive and Negative Signals with Respective Class Means and Corresponding Mean Centered Bootstrapped Residuals

Figure 4.4 shows the plots of a positive and negative signal with the means from both respective classes. Also included in Figure 4.4 are the corresponding plots of the mean centered signals with a residual bootstrapped sample. Figure 4.5 shows the distribution of residuals for COVID-19 positive ($N = 65 \times 6247$) and negative cases ($N = 130 \times 6247$) obtained from the ARMA residual bootstrap. The residuals for both groups have mean near 0, -1.58×10^{-7} and -1.03×10^{-6} for positive and negative bootstrapped residuals, respectively. Ljung-Box tests suggest both residuals are consistent white noise with the minimum test statistic for the set positive case signals $\chi^2 = 44204$ (p-value $< 2.2 \times 10^{-16}$) and the set of negative case signals $\chi^2 = 31200$ (p-value $< 2.2 \times 10^{-16}$).

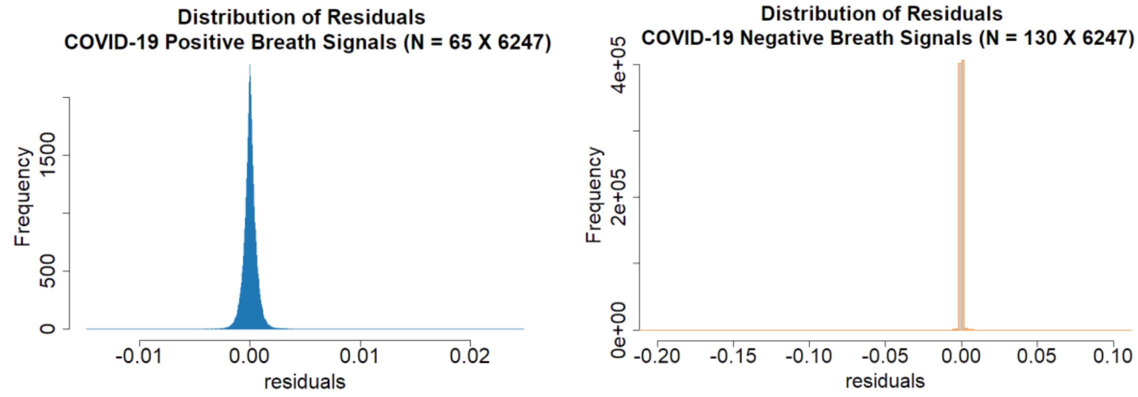


Figure 4.5: Distributions of Positive and Negative Bootstrapped Residual Samples

In Figure 4.6, Principal Component Analysis (PCA) plots show background spectra cluster separately from exhaled breath signals and the bootstrapped simulated signals. In both plots, points in the PC2 vs PC1 plane do not appear to be linearly separatable with multiple potential outliers.

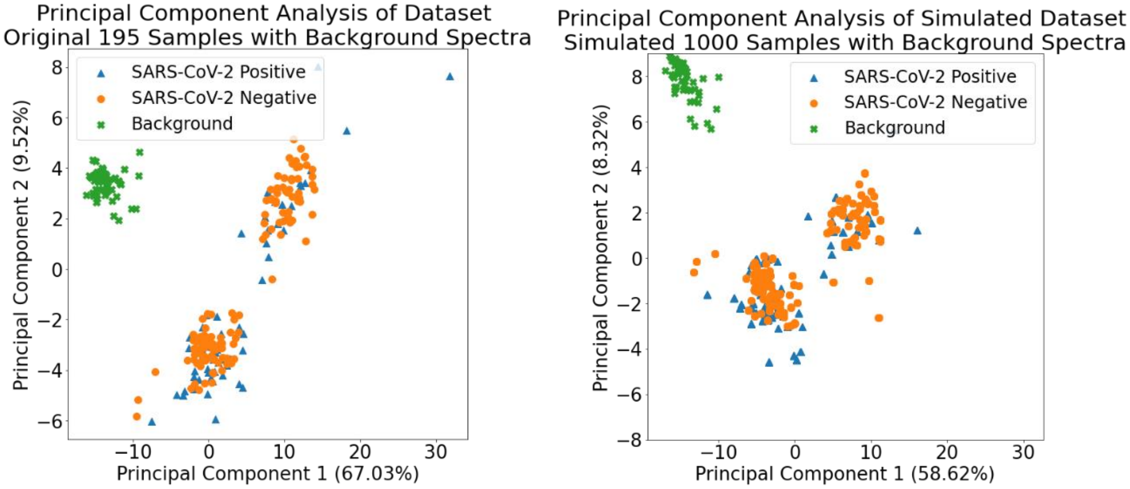


Figure 4.6: PCA Plots of Original 195 and Simulated 1000 Samples with Backgrounds

Figure 4.7 shows the plots of two original unscaled signals and the corresponding min-max normalized signals of the high wavelength laser. There are intermittent peaks present in the COVID-19 positive patients that appear to be absent in the healthy controls. The corresponding min-max normalized signals that are used for modeling appear to be noisier at wavelengths above 10,000 nm relative to lower wavelengths.

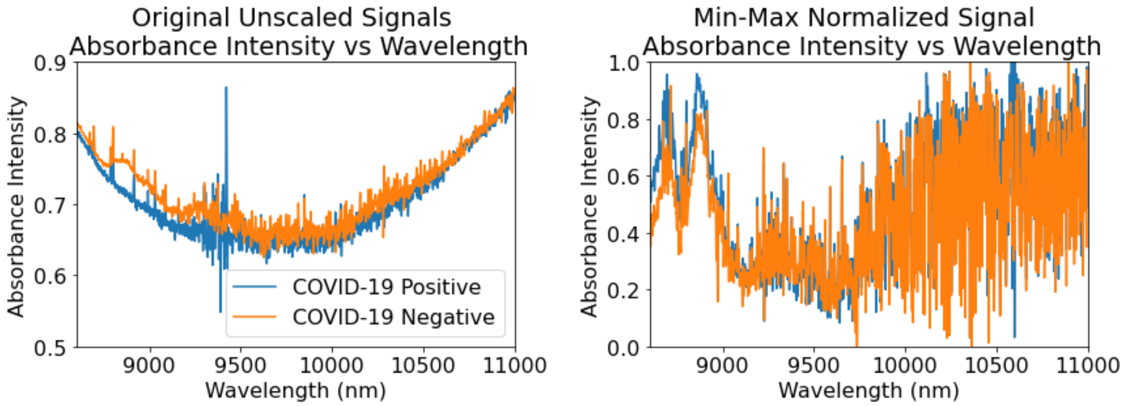


Figure 4.7: Unscaled and Min-Max Normalized Signal Plots

Comparison of Model Performance Using Simulated Dataset

Using the residual bootstrapping method, 100 simulations were generated for the following sample sizes: 195, 380. 585, 780, and 975. For each of the 100 simulations, a 10-fold average accuracy was evaluated for the 5 models as shown in Figure 4.8. The

95% confidence intervals are wider for the smaller simulated sample sizes. The simulated 195 sample size had similar performance as compared to the original 195 samples.

The random forest model improved the most when increasing the simulated sample size followed by k-nearest neighbors, neural network, support vector machine and naïve Bayes. For each of the 5 models, 975 simulated sample set achieved the highest accuracy. The random forest, k-nearest neighbors, and neural network models improved to average accuracy greater than 95% for the 380 simulated sample size and 100% for larger sample sizes. For the 380, 585, 780, and 975 sample sizes, the support vector machine model attaining accuracy of about 85%, 95%, 99%, and 100%, respectively. Naïve Bayes model achieve accuracies slightly below 70% for simulated samples.

In Figure 4.9, the standard deviation of the 10-fold cross validation accuracy at each simulated sample size showed the original 195 samples had greater standard deviation than the simulated samples. As the number of simulated sample size increased, the random forest, k-nearest neighbors, and neural network models approached a minimum with the support sector machine model decreasing more slowly. The naïve Bayes model standard deviation decreased slightly with each increase of the simulated sample size and achieve a minimum standard deviation of about 0.03 at simulated sample size of 975.

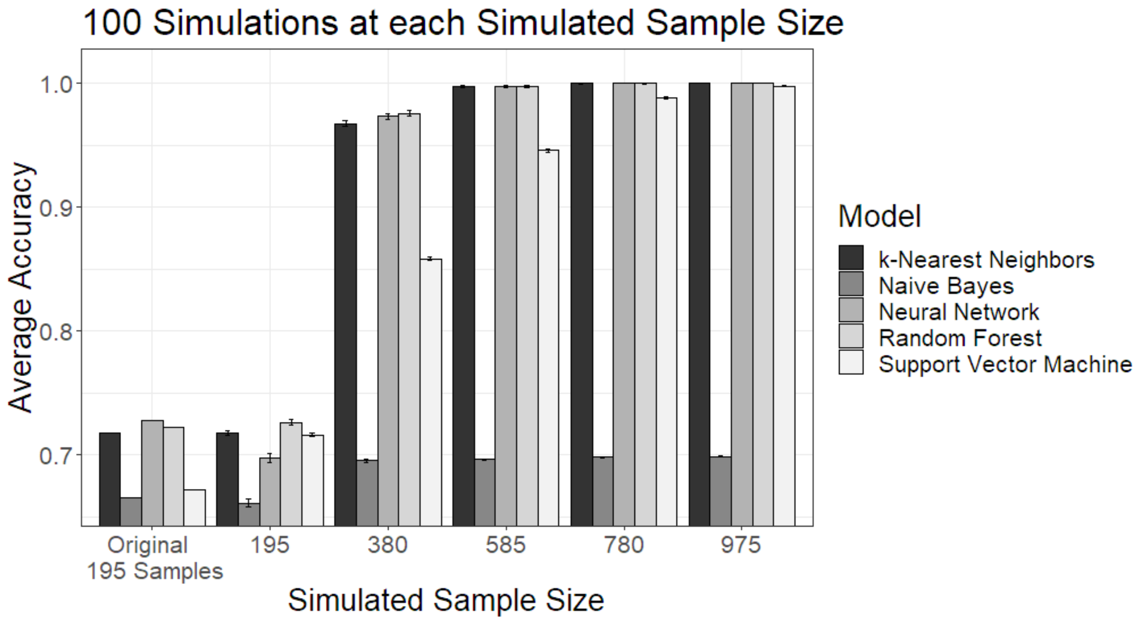


Figure 4.8: 10-fold Average Accuracy of Original and Simulated Datasets by Model

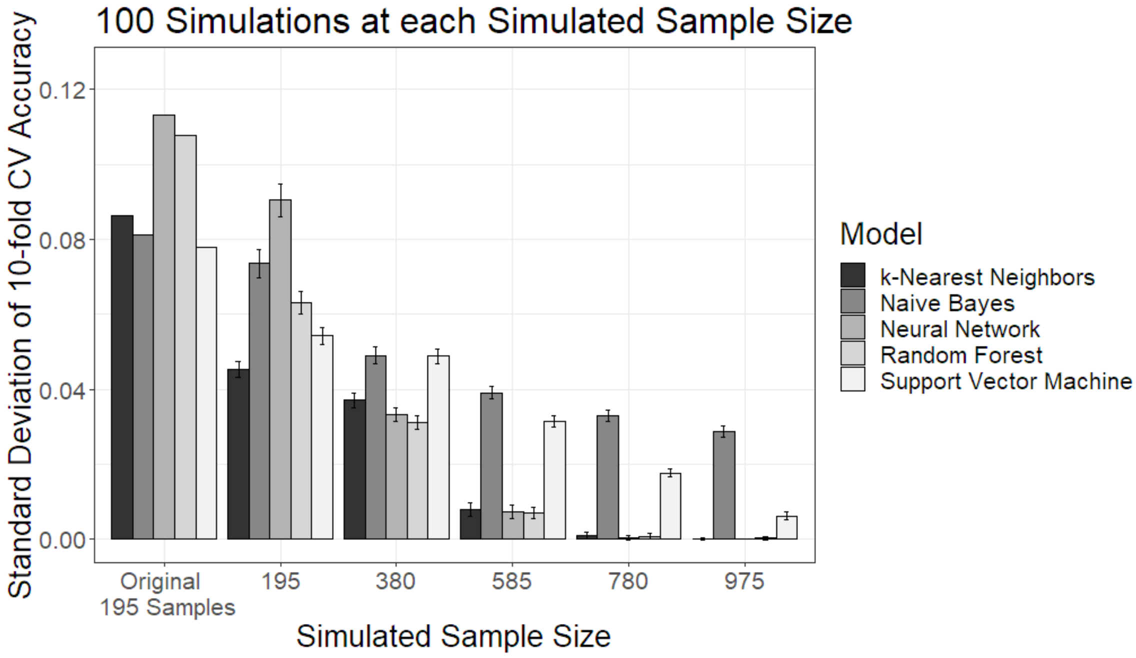


Figure 4.9: Standard Deviation of 10-Fold Accuracy for Original and Simulated Datasets by Model

Multi-Layer Perception and Neural Network Models Using Scikit-Learn and Keras

Using the high wavelength data only, the optimal neural network model was obtained using the ‘MLPClassifier’ function from the Scikit-learn free software machine learning library. This Multi-Layer Perception (MLP) model optimizes the log-loss function using stochastic gradient descent. Tuned hyperparameters include a constant learning rate of 0.001, maximum iterations or epochs of 1000, and ReLu activation functions. This model does not make use of callbacks or dropout layers. The structure of the neural network consists of 7 hidden layers with nodes of (200, 200, 200, 200, 200, 200, 100) for each respective layer. Figure 4.4 shows a caricature of the model.

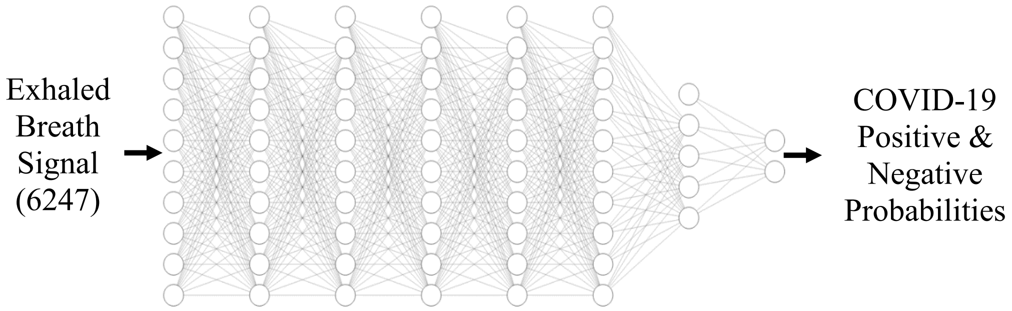
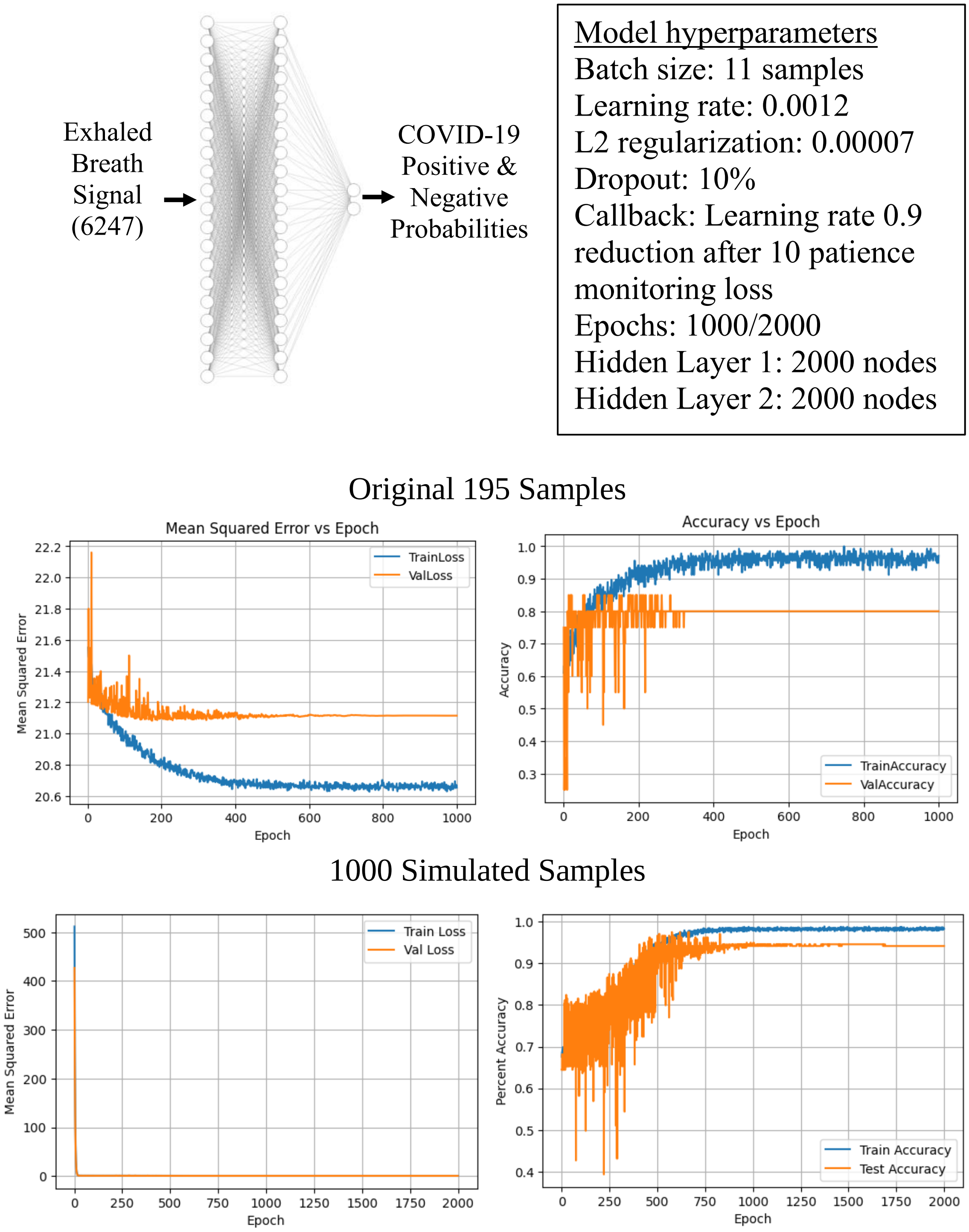


Figure 4.10: Diagram of MLP used for Binary Classification

Figure 4.5 shows the use of callback and dropout hyperparameters tuned using the Keras library model with corresponding loss and learning curves for the original 195 samples and 1000 simulated samples. These curves were obtained by a 70:10:20 split (training:validation:testing) with mean square error loss and accuracy of the training and validation sets at each epoch. The resulting model had an accuracy of 74.36% on the testing set for the 195 original dataset and 93.87% on the testing set for the 1000 simulated dataset.



Model hyperparameters
Batch size: 11 samples
Learning rate: 0.0012
L2 regularization: 0.00007
Dropout: 10%
Callback: Learning rate 0.9 reduction after 10 patience monitoring loss
Epochs: 1000/2000
Hidden Layer 1: 2000 nodes
Hidden Layer 2: 2000 nodes

Original 195 Samples

1000 Simulated Samples

Figure 4.11: Diagram of Neural Network with Hyperparameters tuned using Keras; Plots of Mean Squared Error and Accuracy vs Epoch with Test Accuracy Original 195 and 1000 Simulated Samples

Receiver Operating Characteristic curves and 10-fold Average AUC

In Figures 4.11 and 4.12, Received Operating Characteristic (ROC) curves show the diagnostic ability of the 5 models. The 10-fold average Area Under the Curve (AUC) with standard deviation is given in the legend to the right of the ROC plots. The 195 simulated samples have AUCs for the 5 models of about the same if not better than the original 195 samples. In the 380, 585, 780, and 975 simulated sample sizes the ROC-AUC measure improve with similar to the model accuracies above where random forest model improved the most when increasing the simulated sample size followed by k-nearest neighbors, neural network, support vector machine with naïve Bayes performing the worst.

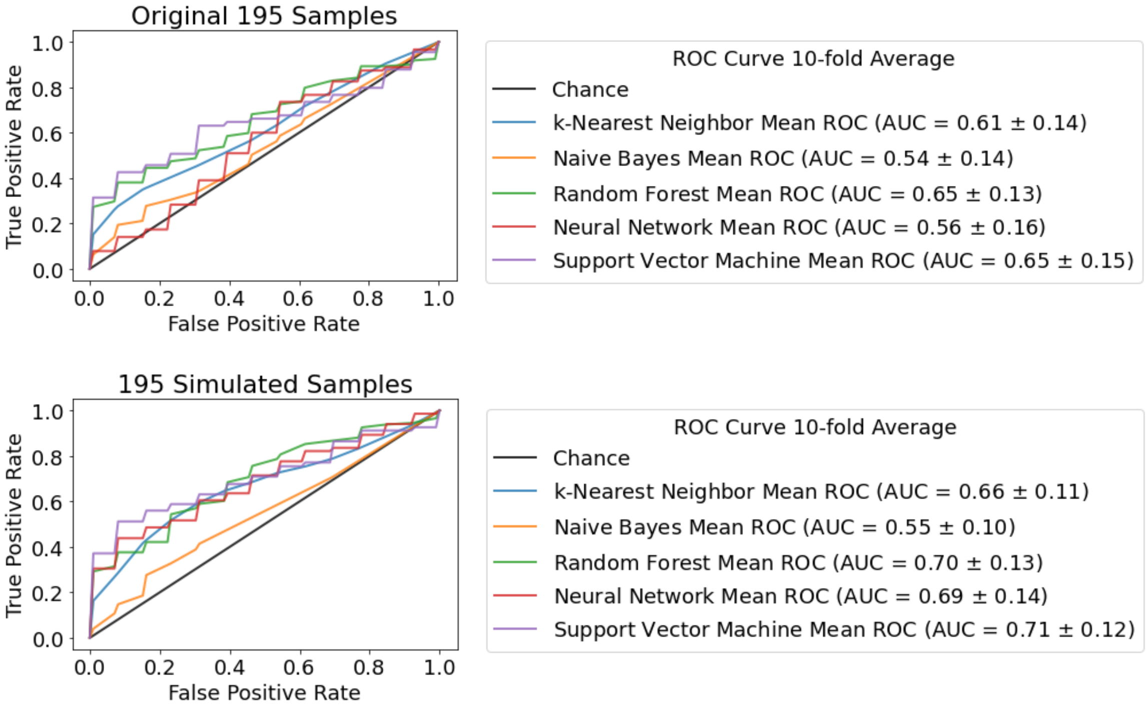


Figure 4.12: 10-Fold Average ROC plot with AUC of Original 195 Samples and 195 Simulated Samples by Model

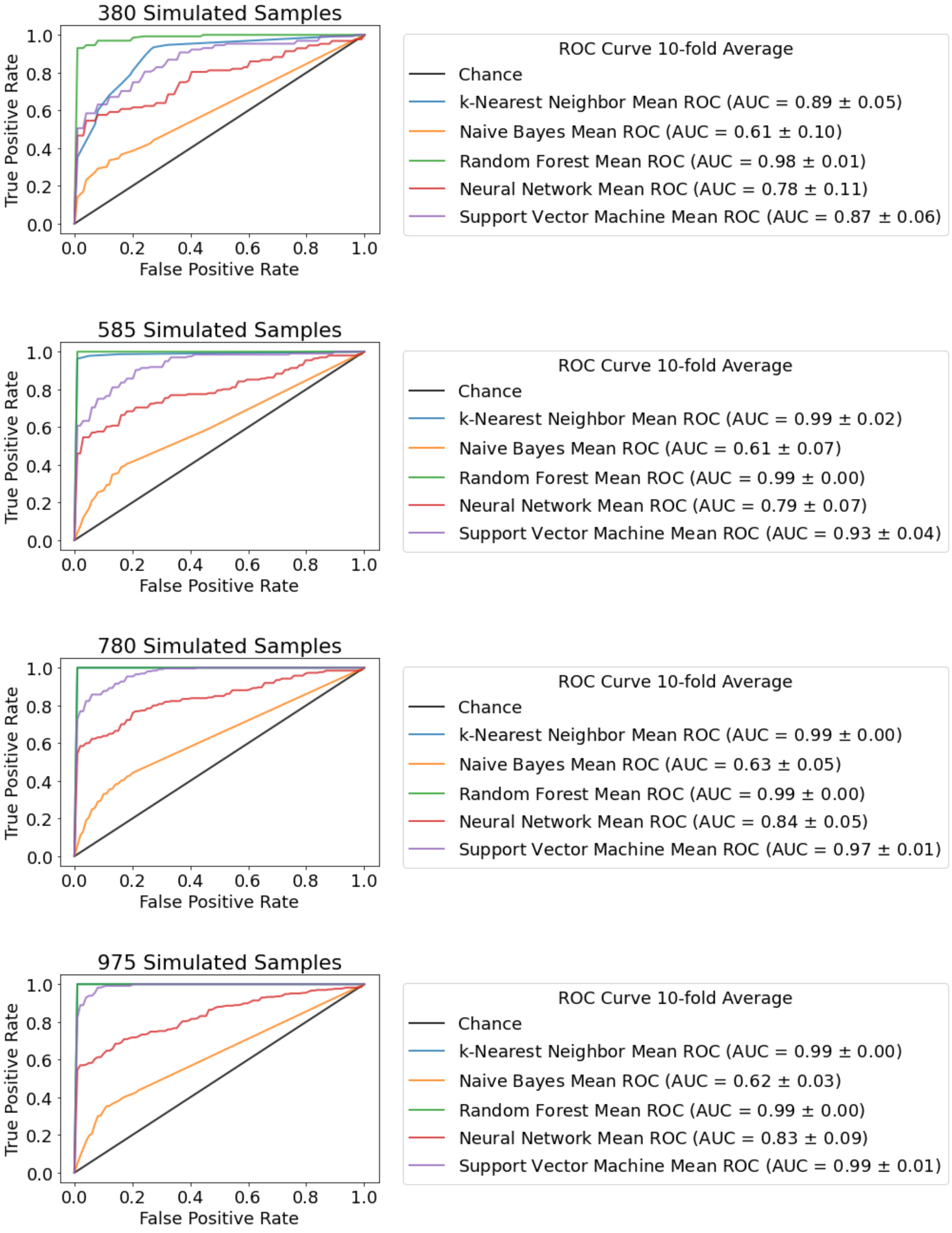


Figure 4.13: 10-Fold Average ROC plots with AUC of Simulated Samples by Model

Recursive Feature Elimination Effects on Model Performance

Reducing the number of features within the dataset improved 10-fold average accuracy as shown in Figure 4.14. In the RFE with re-ranking of the Gini Importance scores, there is an initial decrease in average accuracy for the naïve Bayes model before increasing with an upward trend. The k-nearest neighbor, random, forest, neural network, and support vector machine models tend to increase as the percentage of remaining features of the signal are reduced. The neural network performance seems to vary the most in terms of accuracy. The standard deviation of the 10-fold accuracy trends downwards for most of the models with intermittent peaks for RFE with re-ranking.

In the RFE without re-ranking, the greatest average accuracy observed at about 5% remaining features for the support vector machine and random forest models. The naïve Bayes model appears to increase steadily. With respect to average accuracy, the neural network model varies more in RFE without re-ranking than the RFE with re-ranking. The standard deviation in accuracy appears to vary the least in the RFE without re-ranking.

For RFE without re-normalization or re-ranking, the neural network performance varies substantially more than the other two cases as the features are reduced. The naïve Bayes, k-nearest neighbor, random forest, and support vector machine models appear to have changed slightly with respect to the RFE without Re-Ranking. The standard deviation plot for this RFE shows wide variation in the neural network model with the other 4 models behaving similar to what was observed in the RFE without re-ranking plots.

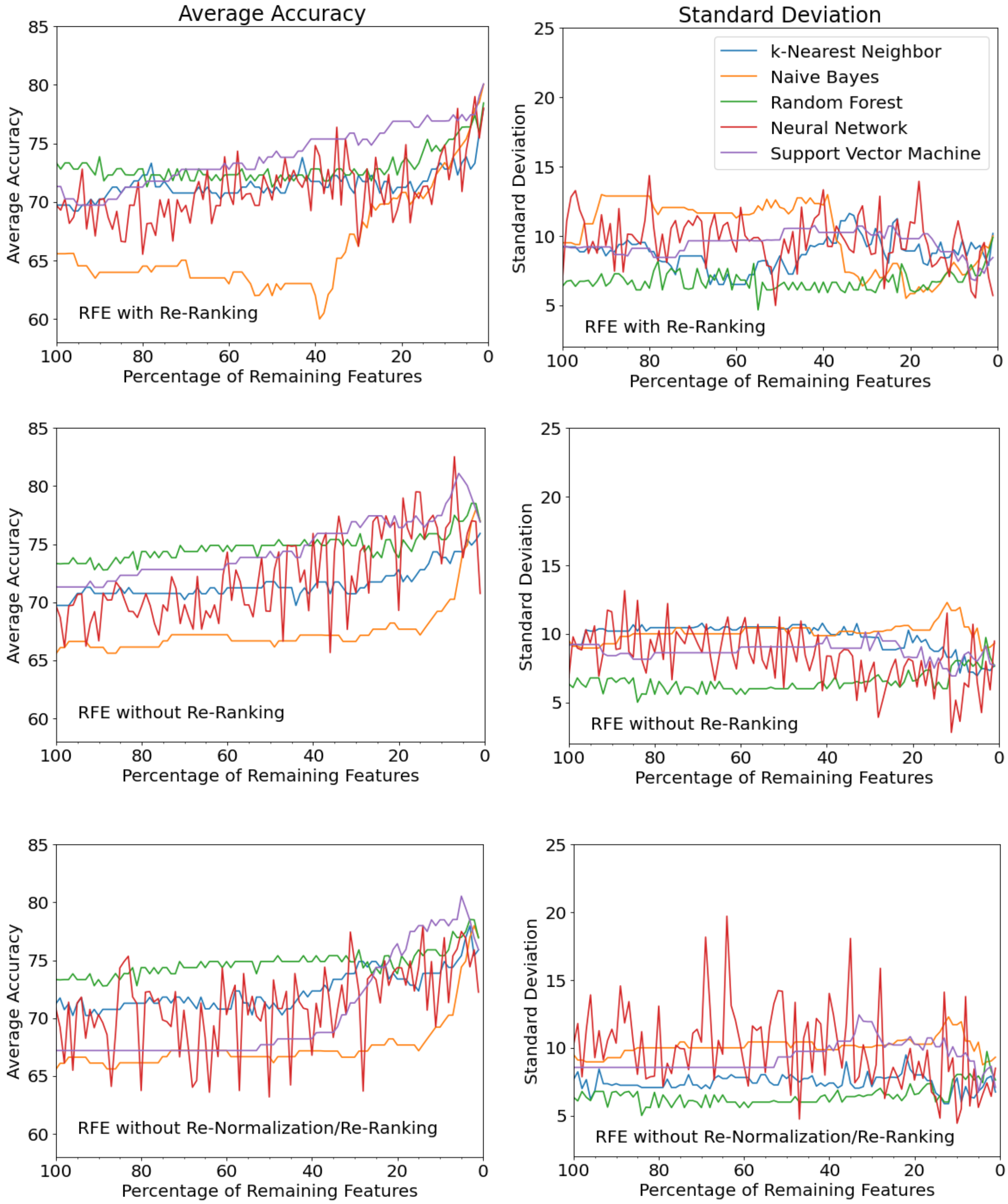


Figure 4.14: 10-fold Average Accuracy and Standard Deviation vs Percentage of Remaining Features by Model

Gini Importance Scoring by Wavelength

The top 25% of the mean of 10-fold average Gini Importance scores were obtained after a 10 nm binning process of the wavelengths resulting in 240 features from the 195 original signals. In Figure 4.15, the Gini Importance frequency with respect to the 10 nm wavelengths are shown in histograms (30 bins) with various VOC signals overlayed. The regions of the spectrum within the gray histogram suggest Gini Importance ranking for COVID-19 classification but do not imply that regions belong to a particular class.

The major peaks of acetone and isoprene in the region of the spectrum occur within 9071 nm to 9219 nm and 10,036 nm to 10,185 nm, respectively. These regions are not found to be important for classification by Gini Importance. Peaks within 9442 nm to 10,036 nm region may suggest an association of the VOC methanol and COVID-19 classification. Ethanol spans regions within the high wavelength laser spectrum that are and are not identified as being important for classification. Heptanal and butanal have multiple peaks corresponding to the regions identified as being important for classification. Acetaldehyde and propanal peaks have some overlap with the regions of the spectrum identified by Gini Importance but relatively less than heptanal and butanal. The VOC Methane does not have significant absorption intensity in this region of the spectrum and is used for comparison purposes only.

The overlay of the min-max normalized COVID-19 positive and negative signals appear to be a combination of some of the VOCs mentioned above with some additional noise relative to the NIST VOC signals but do not appear atypical. The selected COVID-

19 positive signal tends to have greater absorbance intensity that the negative signal at most of the regions within the spectrum.

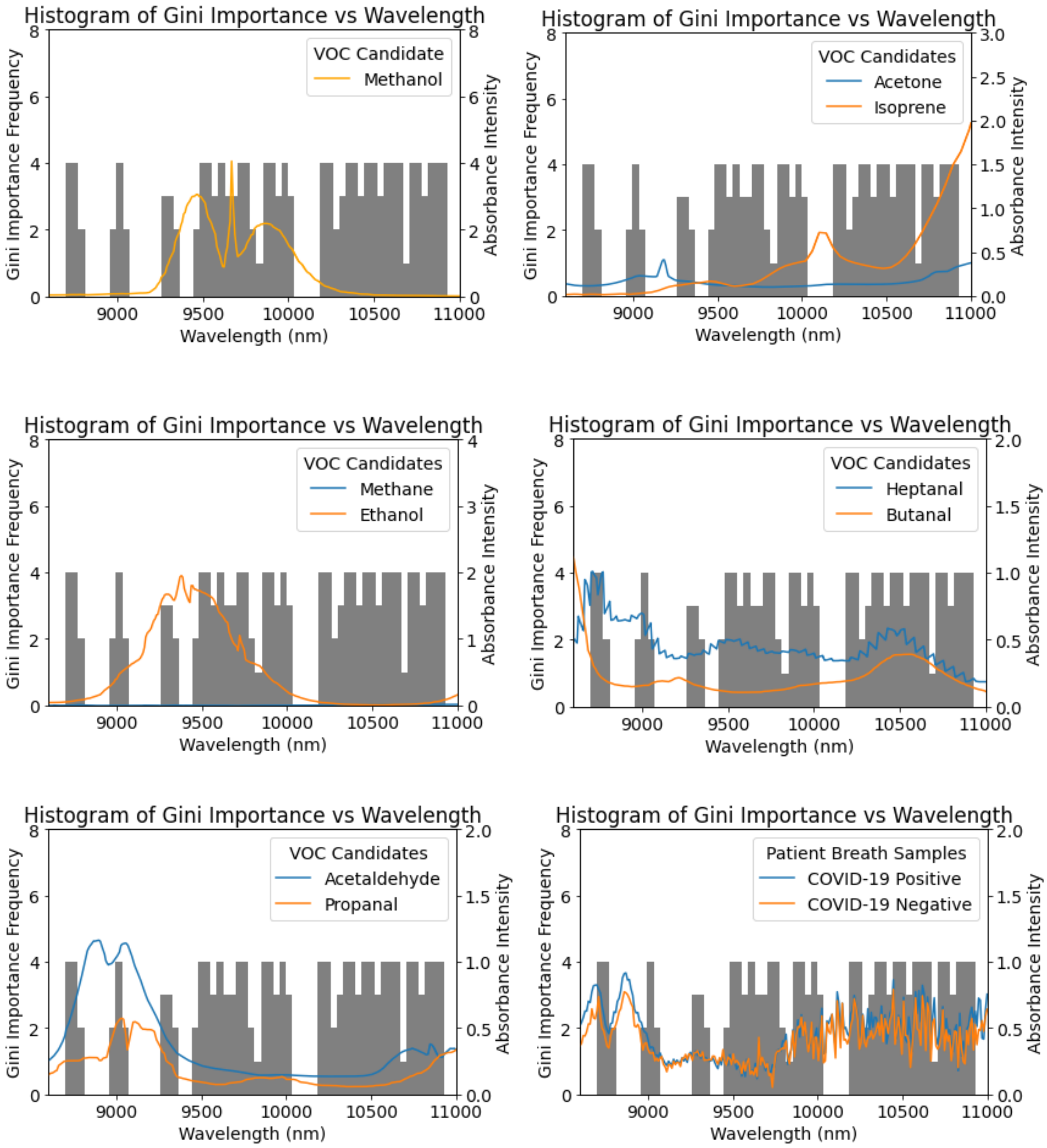


Figure 4.15: Histogram of 10-fold Average Gini Importance Scores vs Wavelength with various VOC Candidates as well as COVID-19 Positive and Negative Signals

CHAPTER 5: DISSCUSSION

In conclusion, using the original dataset of patient exhaled breath signals, machine learning models were able to classify patients as either a COVID-19 positive or negative case with limited accuracy. The neural network, random forest, and k-nearest neighbor models had greater 10-fold average accuracy than the naïve Bayes and support vector machine models. Improved performance was observed by training the models on simulated datasets generated by residual bootstrapping of the mean centered signals from the original dataset. ROC curves and AUC metrics showed that the random forest and k-nearest neighbor models best classified signals as the size of simulated dataset increased. RFE results showed that by reducing the number of lower ranking Gini Importance features improved model performance of the original dataset. The top 25% of features determined by Gini Importance scoring of the exhaled breath signals suggest that regions of the spectrum associated with VOCs may contribute to model classification.

Using PCR test results as the ground truth to distinguish between samples, we can clearly see that COVID-19 breath signals differ from the healthy controls. Additionally, we obtained comparable results using rtCRD and standard NIST for the spectra of pure VOCs suggesting that rtCRD may be a valid method of rapidly capturing the VOC signals within breath samples. The initial investigation suggested that the signals associated with the VOCs acetone and methanol may be similar to the signal of a SARS-CoV-2 positive patient breath sample.

Comparison of models using the original 195 samples showed that the performance using the high wavelength laser data for binary classification had greater 10-

fold accuracy for most models. This may have caused a selection bias since features from the lower wavelength laser may have been important for classification but excluded from further analysis due to the overall low model performance of this region of the data. There is relatively lower standard deviation in the 10-fold accuracy when data from the lower wavelength laser is present which may indicate that this data may has more reliable performance than the high wavelength laser. Further studies may include both regions of the spectrum as there are prominent peaks present at the lower wavelength region such as the peak associated with acetone ranging from 8000 nm to 8400 nm and may contribute to performance reliability.

The residual bootstrap results suggest that resampling the residuals from the ARMA model created signals that each of the machine learning models were able to learn from. The distribution of these resampled residuals for the positive class skew slightly to the right whereas the distribution of residuals from the negative class skew slightly to the left. This suggests that the ARMA models fitted to the two signals selected may tend to under predict the signals from the positive class and over predict signals from the negative class. The Ljung-Box tests suggest that residuals from both classes are consistent with white noise for each signal. Therefore, we do not assume that there are significant biases being introduced in our residual bootstrap method contributing to the improved model performance.

Nonetheless, the residual bootstrap method using the resampled residuals from signal samples may not be the best method for generating simulated samples. Another time series method to consider might be the moving block bootstrap (MBB) method. The MBB resamples data inside overlapping blocks to imitate the autocorrelation in the data

(55). This method would retain neighboring observations within blocks of the signal such that the dependence structure of the random variables at short lag distances would be preserved (56). In this case, there may be fewer instances where the resampled residuals of the ARMA model fitted within a block would be added to a fitted value uncharacteristic of that region of the spectrum. Whereas with the residual bootstrap method, there is no mechanism in place to prevent residuals of distinct regions of the signal from being used to generate a signal that may not be observed in nature.

After performing 100 simulations for each simulated sample size, 95% confidence intervals were obtained showing that the 10-fold average accuracy increased as the simulated sample size increased. However, the support vector machine lags the random forest, k-nearest neighbor, and neural network models with respect to performance. From the RFE, we see that reducing the features that are less important by the Gini Feature Importance scoring result in an improvement in accuracy. This suggests that there might be noise or uninformative features within the signal that contribute to the lower performance of the support vector machine model as this model is more sensitive to noise.

Additionally, the naïve Bayes model performs only slightly better than random chance of selecting a negative case. Since naïve Bayes treats all absorbance intensities independently, it ignores the physical properties associated with the mid-IR wavelengths. In other words, regardless of where the signal intensity is on the spectrum the naïve Bayes classifier would classify this value of the intensity according to the distribution of intensities associated with each class. Thus, naïve Bayes model is a poor classifier since there is specificity associated with the peaks and troughs in the spectrum that help to distinguish samples from positive and negative cases. The random chance performance of the naïve Bayes model may be a result of the COVID-19 positive signals containing a greater concentration of VOCs that contribute to the signals than the healthy controls.

An RFE study was conducted to better understand which regions of the spectrum may be important for classification by the random forest model and to provide some model interpretability. In addition to RFE, another technique that could be used to help explain model classification is Permutation Feature Importance (PFI). PFI involves randomly shuffling the data one feature at a time for the entire dataset and calculating how much the performance metric of interest decreases with the greater the change suggesting the more important that feature is for that model (31). In future studies, RFE and PFI may aid in determining which VOCs contribute to various model classification with a larger dataset.

Additional factors could be included in future studies to account for variability in the breath signals as well as improve model performance. The VOCs provided in this study is not an exhaustive list of possible compounds. There may be other VOCs that have activity within the mid-IR region that may help explain differences between the two groups. Also, testing VOC combinations at different concentrations with the rtCRD device may help provide insight to how much and which VOCs the patient is exhaling. Investigating the effect that time since symptom onset has on VOC production could also provide clinicians with useful information pertaining to disease progression. In intensive or critical care cases, continuous sampling of patient breath may be a useful indicator to signal when a patient’s condition is improving or deteriorating prompting clinicians to act.

As mentioned above, background correction can be used to reduce the noise within the signal samples by normalizing the signals by their background spectra. This may aid in better model performance since the noise contributing to the signal intensity may be dependent on the molecules present in the air at the time of sampling and are not necessarily the same each day in the hospital. Another contributor of noise in the sample may be oral hygiene as microorganisms such as bacteria may produce substances detected by the device. Undiagnosed or not disclosed conditions such as respiratory diseases, COPD, asthma, cancer, sleep apnea or other viral or bacterial infections can also affect VOC profiles (57). These conditions were not excluded and could be pursued further using a device similar to the one used in this study.

In future studies, a larger sample size could be used to validate the use of the rtCRD device using InspectIR COVID-19 Breathalyzer for comparison. The InspectIR COVID-19 Breathalyzer was recently approved by the US Food and Drug Administration (FDA) in April 2022 and was validated using over 2400 symptomatic and asymptomatic patients. The InspectIR COVID-19 Breathalyzer uses a technique called gas chromatography gas mass-spectrometry (GC-MS) to separate and identify chemical mixtures and rapidly detect five Volatile Organic Compounds (VOCs) associated with SARS-CoV-2 infection in exhaled breath (58). In this case, the InspectIR COVID-19 Breathalyzer detects VOC returning a presumptive (unconfirmed) positive test result for COVID-19 and is then confirmed with a molecular test. The use of this device and the rtCRD with clinical data could help provide further insight with respect to rtCRD sensitivity and specificity performance measures.

Lastly, machine learning models are often difficult to interpret and sometimes described as black boxes simply taking inputs and generating outputs (59). As the prevalence of machine learning tools rises, it is important for researchers and clinicians alike to better understand these models and know what factors contribute to the decision process. The intermediate steps in the modeling process may require careful design and consultation with subject matter experts such that the models are interpretable. Also, whenever possible incorporate a scientific basis for explainability of the machine learning model.

In summary, using a novel method such as the non-invasive rtCRD exhaled breath samples spectroscopy device in combination with machine learning may be an effective means of quickly screening patients for SARS-CoV-2 viral infections. With the current level of understanding of the COVID-19 pathophysiology, the exhaled breath VOC pattern may be detectable. However, for the machine learning models to be used for classification, the simulation study suggests that there may need to be a larger sample size. As indicated by the top 25% of Gini Importance ranking of features, there are regions within the spectrum that are associated with numerous VOCs spectra some of which have biological plausibility.

REFERENCES

1. World Health Organization (n.d.). WHO Coronavirus (COVID-19) Dashboard. World Health Organization. Retrieved February 12, 2022, from https://covid19.who.int.

2. Kaye, Alan D., Chikezie N. Okeagu, Alex D. Pham, Rayce A. Silva, Joshua J. Hurley, Brett L. Arron, Noeen Sarfraz, et al. "Economic Impact of COVID-19 Pandemic on Healthcare Facilities and Systems: International Perspectives." *Best Practice and Research: Clinical Anaesthesiology*, 2020. https://doi.org/10.1016/j.bpa.2020.11.009.

3. Oran, Daniel P., and Eric J. Topol. "Prevalence of Asymptomatic SARS-CoV-2 Infection." *Annals of Internal Medicine* 173, no. 5 (September 1, 2020): 362–67. https://doi.org/10.7326/M20-3012.

4. Davis, Cristina E., Michael Schivo, and Nicholas J. Kenyon. "A Breath of Fresh Air - the Potential for COVID-19 Breath Diagnostics." *EBioMedicine* 63 (January 2021): 103183. https://doi.org/10.1016/j.ebiom.2020.103183.

5. Sawano, Makoto, Kyousuke Takeshita, Hideaki Ohno, and Hideaki Oka. "A Short Perspective on a COVID-19 Clinical Study: 'Diagnosis of COVID-19 by RT-PCR Using Exhale Breath Condensate Samples.'" *Journal of Breath Research* 14, no. 4 (October 6, 2020): 042003. https://doi.org/10.1088/1752-7163/abb99b.

6. Wikramaratna, Paul, Robert S. Paton, Mahan Ghafari, and José Lourenço. "Estimating False-Negative Detection Rate of SARS-CoV-2 by RT-PCR." MedRxiv, April 14, 2020, 2020.04.05.20053355. https://doi.org/10.1101/2020.04.05.20053355.

7. Gould, Oliver, Norman Ratcliffe, Ewelina Król, and Ben de Lacy Costello. "Breath Analysis for Detection of Viral Infection, the Current Position of the Field." *Journal of*

Breath Research 14, no. 4 (July 21, 2020): 041001. https://doi.org/10.1088/1752-7163/ab9c32.

8. Walker, Heather J., and Michael M. Burrell. "Could Breath Analysis by MS Could Be a Solution to Rapid, Non-Invasive Testing for COVID-19?" *Bioanalysis* 12, no. 17 (September 2020): 1213–17. https://doi.org/10.4155/bio-2020-0125.
9. Broza YY, Mochalski P, Ruzsanyi V, Amann A, Haick H. Hybrid Volatolomics and Disease Detection. Angew Chemie - Int Ed. 2015;54(38):11036–11048. doi: 10.1002/anie.201500153.
10. Chen H, Qi X, Ma J, Zhang C, Feng H, Yao M. Breath-borne VOC Biomarkers for COVID-19. MedRxiv. 2020. Accessed 20 June 2022. https://www.medrxiv.org/content/10.1101/2020.06.21.20136523v1.
11. Kharitonov, S. A., & Barnes, P. J. (2001). Exhaled markers of pulmonary disease. *American journal of respiratory and critical care medicine*, 163(7), 1693–1722. https://doi.org/10.1164/ajrccm.163.7.2009041.
12. Wang, Z., &amp; Wang, C. (2013). Is breath acetone a biomarker of diabetes? a historical review on breath acetone measurements. Journal of Breath Research, 7(3), 037109–037109.
13. Kaisdotter, A. A., Kron, J., Castren, M., Muntlin, A. A., Hok, B., &amp; Wiklund, L. (2015). Assessment of the breath alcohol concentration in emergency care patients with different level of consciousness. Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine, 23, 11–11. https://doi-org.libproxy.unm.edu/10.1186/s13049-014-0082-y

14. Salerno-Kennedy, R., &amp; Cashman, K. D. (2005). Potential applications of breath isoprene as a biomarker in modern medicine: a concise overview. Wiener Klinische Wochenschrift : The Middle European Journal of Medicine, 117(5-6), 180–186. https://doi-org.libproxy.unm.edu/10.1007/s00508-005-0336-9

15. Patrik Španěl, Kseniya, D., Petra Vicherková, &amp; David, S. (2015). Increase of methanol in exhaled breath quantified by sift-ms following aspartame ingestion. Journal of Breath Research, 9(4). https://doi-org.libproxy.unm.edu/10.1088/1752-7155/9/4/047104

16. Mollar, A., Villanueva, M. P., NÚÑez Eduardo, CarratalÁ Arturo, Mora, F., BayÉs-GenÍs Antoni, MÍnguez Miguel, Marrachelli, V. G., Monleon, D., Navarro, D., Sanchis, J., &amp; NÚÑez Julio. (2019). Hydrogen- and methane-based breath testing and outcomes in patients with heart failure. Journal of Cardiac Failure, 25(5), 319–327. https://doi-org.libproxy.unm.edu/10.1016/j.cardfail.2018.10.004

17. Seeman, J. I., Dixon, M., &amp; Haussmann, H.-J. (2002). Acetaldehyde in mainstream tobacco smoke: formation and occurrence in smoke and bioavailability in the smoker. Chemical Research in Toxicology, 15(11), 1331–50.

18. Fuchs, P., Loeseken, C., Schubert, J. K., &amp; Miekisch, W. (2010). Breath gas aldehydes as biomarkers of lung cancer. International Journal of Cancer, 126(11), 2663–70. https://doi-org.libproxy.unm.edu/10.1002/ijc.24970

19. Corradi, M., Pignatti, P., Manini, P., Andreoli, R., Goldoni, M., Poppa, M., Moscato, G., Balbi, B., &amp; Mutti, A. (2004). Comparison between exhaled and sputum oxidative stress biomarkers in chronic airway inflammation. The European Respiratory Journal, 24(6), 1011–7.

20. Jung, Y. J., Seo, H. S., Kim, J. H., Song, K. Y., Park, C. H., &amp; Lee, H. H. (2021). Advanced diagnostic technology of volatile organic compounds real time analysis analysis from exhaled breath of gastric cancer patients using proton-transfer-reaction time-of-flight mass spectrometry. Frontiers in Oncology, 11, 560591–560591. https://doi-org.libproxy.unm.edu/10.3389/fonc.2021.560591

21. Chen, H., Qi, X., Zhang, L., Li, X., Ma, J., Zhang, C., Feng, H., & Yao, M. (2021). COVID-19 screening using breath-borne volatile organic compounds. *Journal of breath research*, *15*(4), 10.1088/1752-7163/ac2e57. https://doi.org/10.1088/1752-7163/ac2e57

22. Selvaraj, R., Vasa, N. J., Nagendra, S., & Mizaikoff, B. (2020). Advances in Mid-Infrared Spectroscopy-Based Sensing Techniques for Exhaled Breath Diagnostics. Molecules (Basel, Switzerland), 25(9), 2227. https://doi.org/10.3390/molecules25092227

23. Rajapaksha, R. D., Tehrani, M. W., Rule, A. M., & Harb, C. C. (2021). A Rapid and Sensitive Chemical Screening Method for E-Cigarette Aerosols Based on Runtime Cavity Ringdown Spectroscopy. *Environmental science & technology*, 55(12), 8090–8096. https://doi.org/10.1021/acs.est.0c07325

24. Candelaria, L., Segura, A., Smith-Fassler, A., Taylor R., Baca, J., (2022). Exhaled Breath Analysis for COVID-19 Detection Using runtime Cavity Ringdown (rtCRD) Spectroscopy - A Feasibility Study. (In press).

25. Abe, H., Lisak, D., Cygan, A., & Ciuryło, R. (2015). Note: Reliable, robust measurement system for trace moisture in gas at parts-per-trillion levels using cavity

ring-down spectroscopy. *The Review of scientific instruments*, *86*(10), 106110. https://doi.org/10.1063/1.4934976

26. Stacewicz, T., Wojtas, J., Bielecki, Z., Nowakowski, M., Mikołajczyk, J., Mędrzycki, R. & Rutecka, B. (2012). Cavity ring down spectroscopy: detection of trace amounts of substance. *Opto-Electronics Review*, *20*(1), 53-60. https://doi.org/10.2478/s11772-012-0006-1
27. Cao, X. H., Stojkovic, I., & Obradovic, Z. (2016). A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC bioinformatics*, *17*(1), 359. https://doi.org/10.1186/s12859-016-1236-x
28. Haykin, S. (2010). *Neural networks and learning machines, 3/E*. Pearson Education India.
29. Nwankpa, C. E., Ijomah, W., Gachagan, A., &amp; Marshall, S. (2018). Activation functions: comparison of trends in practice and research for deep learning. Arxiv, (2018 11 08).
30. I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning.” MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
31. Breiman, L. Random Forests. *Machine Learning* **45,** 5–32 (2001). https://doi.org/10.1023/A:1010933404324
32. Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., Hamprecht, F. A., &amp; SpringerLink (Online service). (2009). A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data, Volume:10. https://doi-org.libproxy.unm.edu/10.1186/1471-2105-10-213

33. Kuhn, Max (2019). The caret package. Chapter 20: Recursive Feature Elimination. Algorithm 1: Recursive Feature Elinimation. Date published: 3/27/2019. Date retrieved: 3/18/22 from Website.https://topepo.github.io/caret/recursive-feature-elimination.html.

34. Bahl, A., Hellack, B., Balas, M., Dinischiotu, A., Wiemann, M., Brinkmann, J., Luch, A., Renard, B. Y., &amp; Haase, A. (2019). Recursive feature elimination in random forest classification supports nanomaterial grouping, 15. https://doi-org.libproxy.unm.edu/10.1016/j.impact.2019.100179

35. Svetnik, V., A. Liaw, C. Tong, and T. Wang. (2004). “Application of Breiman's Random Forest to Modeling Structure-Activity Relationships of Pharmaceutical Molecules.” Lecture Notes in Computer Science 3077:334–43.

36. Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 93–104. https://doi-org.libproxy.unm.edu/10.1016/j.isprsjprs.2011.11.002

37. Ambroise, C., &amp; McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. Proceedings of the National Academy of Sciences of the United States of America, 99(10), 6562–6566.

38. Rosaen, Karl (2016) K-fold cross-validation figure. Website scikit-learn Pipeline gotchas, k-fold cross-validation, hyperparameter tuning and improving my score on Kaggle's Forest Cover Type Competition. Accessed on: 3/26/2022. Retrieved from http://karlrosaen.com/ml/learning-log/2016-06-20/

39. ObCom 2011 (2011: Vellore Institute of Technology), Venkata Krishna, P., Rajasekhara Babu, M., &amp; Ariwa, E. (2012). Global trends in information systems and software applications: 4th international conference, obcom 2011, vellore, tn, india, december 9-11, 2011. proceedings (Vol. Part ii /, Ser. Communications in computer and information science, 270). Springer.

40. Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice* (Third print). OTexts. https://doi-org.libproxy.unm.edu/10.1007/978-3-642-29216-3

41. Krispin, R. (2019). Hands-On Time Series Analysis with R: Perform Time Series Analysis and Forecasting Using R. Birmingham: Packt Publishing, Limited.

42. Shumway, R. H., &amp; Stoffer, D. S. (2017). Time series analysis and its applications: with r examples (Fourth, Ser. Springer texts in statistics). Springer. https://doi-org.libproxy.unm.edu/10.1007/978-3-319-52452-8

43. Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26.

44. D Yu, L Deng. (2011) Deep learning and its applications to signal and information processing. IEEE Signal Proc Mag 28(1), 145–154.

45. Krizhevsky, A., Sutskever, I., &amp; Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the Acm, 60(6), 84–90. https://doi.org/10.1145/3065386

46. Torfi, A., Keneshloo, Y., Fox, E. A., Tavaf, N., &amp; Shirvani, R. A. (2020). Natural language processing advancements by deep learning: a survey. *Arxiv*, (2020 03 02).

47. Qiu, J., Wu, Q., Ding, G., Xu, Y., &amp; Feng, S. (2016). A survey of machine learning for big data processing. Eurasip Journal on Advances in Signal Processing, 2016(1), 1–16. https://doi-org.libproxy.unm.edu/10.1186/s13634-016-0355-x.

48. Srivastava, N., Hinton, G., Krizhevsky, A., (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 15(56):1929−1958.

49. Goyal, C., (2021) Top 15 Questions to Test your Data Science Skills on SVM. Analytics Vidhya. Website last updated: 5/20/21. Website visited: 3/27/2022. https://www.analyticsvidhya.com/blog/2021/05/top-15-questions-to-test-your-data-science-skills-on-svm/

50. Hastie, T., Tibshirani, R., &amp; Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Second, Ser. Springer series in statistics). Springer.

51. Java T Point. (2021). K-Nearest Neighbor (KNN) Algorithm for Machine Learning. Picture: Plot of k-Nearest Neighbor Classification of new data point Accessed on 4/15/2022 from https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

52. Domingos, P., &amp; Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning, 29(2-3), 103–130. https://doi-org.libproxy.unm.edu/10.1023/A:1007413511361

53. Sucar, L. E. (2021). Probabilistic graphical models: principles and applications (Second, Ser. Advances in computer vision and pattern recognition). Springer. https://doi-org.libproxy.unm.edu/10.1007/978-3-030-61943-5

54. Guyon, I, J Weston, S Barnhill, and V Vapnik. 2002. “Gene Selection for Cancer Classification Using Support Vector Machines.” *Machine Learning* 46 (1): 389–422.

55. Kreiss, J., Wolfgang, H., Horowitz, J., (2001). Bootstrap methods for time series. Humboldt-Universitat Berlin. http://hdl.handle.net/10419/62726

56. Lahiri, S.N., 1992. Edgeworth correction by moving block bootstrap for stationary and nonstationary data. In: LePage, R., Billard, L. (Eds.), Exploring the Limits of Bootstrap. Wiley, New York, pp. 183–214.

57. Bruderer, T., Gaisl, T., Gaugg, M. T., Nowak, N., Streckenbach, B., Müller, S., Moeller, A., Kohler, M., &amp; Zenobi, R. (2019). On-line analysis of exhaled breath: focus review. Chemical Reviews, 119(19), 10803–10828. https://doi-org.libproxy.unm.edu/10.1021/acs.chemrev.9b00005

58. McKinney, J. 2022. Coronavirus (COVID-19) Update: FDA Authorizes First COVID-19 Diagnostic Test Using Breath Samples. Website Accessed on 4/22/2022 from https://www.fda.gov/news-events/press-announcements/coronavirus-covid-19-update-fda-authorizes-first-covid-19-diagnostic-test-using-breath-samples

59. Quintanilla, L., (2021) Interpret model predictions using Permutation Feature Importance. Website accessed 3/30/2022. Retrieved from: https://docs.microsoft.com/en-us/dotnet/machine-learning/how-to-guides/explain-machine-learning-model-permutation-feature-importance-ml-net