Classifying SARS-CoV-2 and Common respiratory viruses from Genome Assemblies

By

Mohaimen Rahman

A thesis is submitted to the Faculty of Graduate Studies of

The University of Manitoba

In the fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba

Abstract—

Polymerase chain reaction (PCR) testing has widespread use in the systematic identification of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) strains. However, another approach for identifying the SARS-CoV-2 virus is by the machine learning classification of genome sequences, which has shown promising results. While trained clinicians usually perform the classification of genome sequences, a machine learning classifier can be used to complement the process and provide a short list for further analysis. A machine learning approach can provide a unique fingerprint of base pairs and yield a quick classification. To this end, we investigated a k-mer approach in order to classify genome sequences of SARS-CoV-2 and common respiratory viruses, as well as a Human genome sequence. We aim to provide a simplified classification approach that balances validation time while limiting hyperparameter tuning. Our approach achieved F1 scores in excess of 0.99, and perfect scores between the common respiratory viruses. We demonstrated a simple 5-base sub-sequencing scheme which has the power to differentiate over 7.91 million sequences from almost 20 thousand genome assemblies.

Acknowledgments

I would like to thank and praise Almighty ALLAH for my life, unconditional love and, granting me to do M.Sc. at the University of Manitoba.

I would like to express my gratitude to Prof. Dr. Ken Ferens for his supervision in my research and his guidance during my M.Sc. study at the University of Manitoba. I would like to thank him from the core of my heart for being a great supervisor and an awesome person to work with.

I would like to thank my parents Md. Mizanur Rahman and Jasmin Sultana for inspiring me to do M.Sc. abroad and teaching how to keep faith in me. Also, for their numerous support, I am finally at the end of the program.

Finally, I would love to thank from the bottom of my heart to my wife Ayeasha Akhter due to her tremendous support, inspiration and motivation for completing the research program.

Dedicated to

I dedicate this work to my parents and my wife for their love and moral support. I also dedicate this thesis to my supervisor Professor Dr. Ken Ferens for his innovative ideas and supervision during the entire period of the master's program.

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

Respiratory tract infections are defined as the irritation and swelling of the upper airways in the absence of bronchitis or pneumonia [1]. The most common viruses responsible for respiratory tract infections include Rhinovirus, Influenza, Adenovirus, Enterovirus and Respiratory Syncytial Virus. In early 2020, millions of people worldwide were contracting COVID-19, a disease that is caused by Severe Acute Respiratory Syndrome Coronavirus 2 (known as SARS-CoV-2 as per NCBI Taxonomy ID: 2697049). As of April 2022, 488 million people have been infected with SARS-CoV-2 resulting in over 6.1 million deaths worldwide according to John Hopkins University. The heightened need for respiratory virus surveillance help to illustrate that a robust series of diagnostics tests are required.

SARS-CoV-2 belongs to a larger class of coronavirus', which include the Middle East Respiratory Syndrome (MERS) and severe acute respiratory syndrome (SARS), among others. The disease which caused by SARS-CoV-2, also known as COVID-19, is highly transmissible which makes it very difficult to alert to the disease early on [2]. There is also the concern that individuals are

likely to be infected with different pathogens which might have similar syndromes like Sars-CoV-2 because the respiratory system is not only affected by the Sars-CoV-2 but also other

species in Coronaviridae, Influenza, Metapneumovirus, Rhinovirus etc. It has been observed that 19-20% of patients testing positive for SARS-CoV-2 were positive with 1 or more additional pathogens [3], [4], and up to 24% experience superinfection [4] (infection at a later stage of illness). Understanding different respiratory viruses is important because at the early on-stage of the COVID-19 pandemic, antibiotic use was widespread; and unnecessary antibiotic use is associated with increased risk of Chloridoids difficile colitis, adverse drug effects (kidney injury, allergy) and bacterial and fungal resistant infections [5]. It has also been noted that knowing which bacterial infection in respiratory systems is present in a patient, can provide health care practitioners a better scope to guide proper antibiotic therapy [6].

A preliminary SARS diagnosis is based on both clinical and epidemiologic criteria, which is shared among many SARS viruses. A more specific diagnostic test for SARS-CoV-2 through polymerase chain reaction (PCR) assays is required and has been developed [7]. However, these respiratory viruses are mutating every moment which has become difficult in order to identify in which respiratory virus groups they belong. Once new strains emerge, a modification of the procedure is required to alert to the new pathogen. RNA-sequencing is one other approach to characterization which provides a measure of the RNA molecules

present in each sample. One approach to characterization is to encode the nucleotides by frequency, which makes the analysis agnostic to the position in the genome. However, research suggests coding and non-coding regions of the genome have different nucleotide frequency characteristics [10], meaning there is a consideration for which area of the genome is under observation. Understanding which features to look at and in what way requires significant

domain expertise and expert knowledge. Machine Learning tools are uniquely qualified for this because it is impractical to develop a taxonomy of sequence reads to act as a reference database. A more generalized observation of nucleotide frequency and sequencing can provide a systematic identification of SARS-CoV-2 and other common respiratory viruses and many written works have worked to do so.

1.1. Thesis Statement and Overview

This thesis aims to provide a simplified classification approach with a k-mer counting that balances validation time while limiting hyperparameter tuning in order to classify respiratory viruses through genome sequences. The genome sequences are form NCBI (an United States government official site) and the dataset was collected from PACIFIC [17]. For this research, the RNA sequences were adapted though k-mer approach in order to apply machine learning classifiers and then the machine learning classifiers were applied to classify them

with highest accuracy. Two different experiments were performed to achieve the higher accuracy.

1. Application of six different machine learning classifiers to compare the f1 scores, training and validation time.
2. Tuning the hyperparameters of those classifiers which required less training and validation time in order to get highest f1 scores as well as accuracy.

1.2.The contribution of the Thesis

Over 545 million individuals worldwide have contracted acute respiratory tract infections, which rank as the third leading cause of death worldwide and result in nearly 4 million fatalities each year. Various RNA viruses, including coronavirus, influenza, rhinovirus, parainfluenza virus, and metapneumovirus, are among the most common pathogens that cause respiratory infections and illnesses. Coronaviruses and other new respiratory infections have frequently crossed species boundaries. Millions of people have been impacted by COVID-19, an infectious zoonotic disease brought on by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Coronaviruses had also caused the 2002–2003 outbreak of SARS-CoV respiratory disease which has infected around 8,098 people and the outbreaks of Middle East respiratory syndrome coronavirus

(MERS-CoV, 2012–2020), which has infected almost 2,519 people. Accurate, timely and rapid surveillance and highlights should be increased for these recurrent emergences of respiratory viruses.

The respiratory systems are affected by various respiratory viruses like Sars-CoV-2, other species in Coronaviridae, Influenza, Rhinovirus, Metapneumovirus etc. and that’s why the syndromes are also quite similar. Moreover, the respiratory viruses are mutating every moment. For the health care professionals, it has become a must to identify the appropriate respiratory viruses so that proper remedies as well as medications can be provided.

However, many people have worked on different kind of classification approaches [16],[17],[19] for classifying respiratory viruses form genome assemblies but this thesis provides a novel simplified classification approach. The contributions of this research are as follows:

a. This research demonstrates an application of a simple encoding scheme, in this case k-mers, to the classification of a genome sequence. This can be expanded to new applications where sequencing of a data type is important for understanding the underlying process. For example, in the field of Cybersecurity understanding the sequence of API calls of a running process can provide information of the intent behind the process, which can be malicious in nature. Beyond Cyber Security, natural language processing applications also use ngrams in a similar way to capture the semantics of the English language. Applications of ngrams in identifying spam emails has also been well studied, and continues to be studied to this day.

b. This novel method of research requires less training and validation time than the work done by [17].

c. The optimal value of K in this research was 5 which means the classifiers performed with a smaller number of features compared to the reference paper [17] but provided better results. So, it requires much less memory and can perform better on less powerful computers.

d. This research has achieved higher f1 scores in excess of 0.99 which means it is able to correctly classify the Sars-CoV-2 and other respiratory viruses pretty well which can be beneficial for the clinicians to provide the appropriate remedies.

1.3. Outline of the thesis

The structure of this thesis paper is as follows:

In Chapter 1, the idea of categorizing respiratory viruses from genomic assemblies using computational intelligence and machine learning classifiers is introduced. A few literature studies of the techniques currently in use for classifying various viruses through genome assemblies are described in Chapter 2. The background of several machine learning algorithms, including Nearest Neighbor, Linear SVM, RBF SVM, Random Forest, Extra Trees, and MLP, is introduced in Chapter 3. The complete technique for this research, including dataset collections, dataset preprocessing, models and performance metrics, and k-fold cross validation, is described in Chapter 4. The experimental works, analysis of the results, and performance comparison with other machine learning classifiers are all covered in Chapter 5. The thesis's conclusion is presented in Chapter 6, along with a brief list of initiatives to be attempted in the near future to improve the classification method used in this thesis paper.

Chapter 2

2. Background Research

Early analysis of genome sequences examined dinucleotide patterns in the early 1960s [11]. The motivation being that DNA sequences can be largely explained simply be examining the mono-, di- and trinucleotide frequencies [12]. However, frequency-based mappings differ by region, and there is significant loss

in sequence information when considering their frequency of occurrence. As a result, researchers look at subsequences and use k-mers to understand differences and similarities in pathogenicity [13]. For example, in [14] the authors used a k-mer of length 7 while in [15] they looked at varying subsequence lengths from 1 to 10. A full end-to-end implementation was carried out in [16], where they regarded sequences as text data and developed a vocabulary based on 4-mer subsequences. The authors used a Convolutional Neural Network (CNN) coupled with Support Vector Machine (SVM) and achieved 1-4% improved accuracy over existing techniques on 10 benchmark datasets.

Classification has also been of interest in examining patterns in geographical locations where mutations lead to strains that envelop a region. In [18] 837 Indian SARS-CoV-2 strains were classified into 22 different groups based on their common mutations. The authors noted broad

heterogeneity among geographical locations throughout India, with certain strains dominating certain regions. Geographical-based classification of SARS-CoV-2 has also been carried out to distinguish differences in strains among continental regions. In [19] the authors achieved 100% accuracy when distinguishing SARS-CoV-2 from 7 other pathogenic species, however, their analysis was limited to 32 thousand total sequences which suffered from significant class imbalance. They coupled their feature processing with XGBoost and relied on dinucleotide signatures as features. Their approach had significant drawbacks in that relying on too few sequences and solely dinucleotide signatures meant their geographical-based analysis only achieved 67% categorical accuracy.

A more wholistic approach was taken in [17] where the authors utilized a CNN and bi-directional long short-term memory (BiLSTM) network to map the long-range dependencies between nucleotides at any location of the sequence. The work also carried out SARS-CoV-2 and other common respiratory virus classification, similar to the proposed work used in this manuscript. But the approach used in this manuscript is much simpler than them. In this work, simple machine learning classifiers like Random Forest, Extra trees, Linear SVM, RBF SVM, Nearest Neighbor and MLP were used with k-mer approach and hyperparameter tunning was done to get better classification accuracy almost equivalent to them. Their optimum value for k was 9. So larger memory required. The optimal value of k is 5 in this research. So smaller memory required compared to them though in both type of approaches, the f1 score is almost similar (>0.99). They used a machine with an NVIDIA GeForce RTX2080Ti GPU, AMD Ryzen Threadripper 2950 × 16-core processor, and 125.7 GiB RAM running in Ubuntu 18.04.2 LTS for classification and their average processing time for 100,000 samples classification was 108s.

In this research, less powerful machine was used. A machine with Intel core i7 2.5GHz, 8 core, NVIDIA GeForce GT 750M, 16 GiB RAM running macOS version 10.13.6 was used for the research and average processing time was 18.6s for 100,000 samples.

Chapter 3

3. Background of Machine Learning Algorithm

This chapter presents the background of machine learning classifiers and computational intelligence approaches like Linear and RBF SVM, Nearest Neighbor , Random Forest, Extra Trees and MLP.

3.1. Linear and RBF SVM

SVMs, or support vector machines, are frequently employed in supervised machine learning techniques [25]. SVM often examines a variety of samples (data) to process a wide range of classification analyses. When using supervised learning, the algorithm often creates an ideal hyperplane that aids in categorising the new samples (data). This technique develops a model that assigns a category to a new sample by self-learning from a broad variety of training samples, each of which is defined by a certain category (data). Either an RBF or a linear kernel can be used. It can conduct classification, face identification[27], pedestrian detection[28],

handwritten character recognition[26], text categorization, and regression problems using the provided data samples.

If we take a training dataset, $T = \{(x^p, y^p)\}$, where $x^p \in R^n$ represents the SVM's input vector which contains the n-dimensional input features and $y^p \in \{-1, +1\}$ represents the output of the p^{th} training data sample. The output of the p^{th} positive of training samples is indicated by $y^p = 1$ and the output of the p^{th} negative training samples is indicated by $y^p = -1$.

Considering above variables, the decision hyperplane in the form of surface is described as follows:

$$\sum_{i=1}^{p} \boldsymbol{w}^T x^{(i)} + b = 0 \tag{1}$$

w and *b* stand for the weight and bias vectors, respectively. The training procedure establishes the weight and bias term. With the help of these extracted parameters, the decision hyperplane is positioned in the data space at the best possible point. By positioning the decision boundary, SVM optimises the geometric margin of all the training data samples. All training examples are as far away geometrically as feasible from the decision border. The optimization issue is then best explained as follows:

$$\min_{w,b} \frac{1}{2} \| W \|^2 \tag{2}$$

s.t

$$\sum_{i=1}^{p} y^{(i)}(W^T x^{(i)} + b) - 1 \geq 0 \qquad i = 1. \ldots \ldots . n \tag{3}$$

This optimization problem with the constraint border showed in equation (1) can be resolved by the idea of Larange multiplier. The Lagrangian may be written as follows:

$$L(w, b, \alpha) = \frac{1}{2} \parallel W \parallel^2 - \sum_{i=1}^{p} \alpha(y^{(i)}(W^T x^{(i)} + b) - 1) \tag{4}$$

Here, α $(\alpha \geq 0)$ is the multiplication factor. If the Lagrange function is compared with respect to w, b and, α; then the optimization problem mentions in the equation (4) can be formulated as

$$\max_{\alpha} L(\alpha) = \max_{\alpha} (\sum_{i=1}^{p} \alpha_i -$$

$$\frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} x^{(j)}) \tag{5}$$

Subject to

$$\sum_{i=1}^{p} \alpha_i \, y^{(i)} = 0 \tag{6}$$

$$\alpha_i \geq 0, i = 1,2, \ldots \ldots, p$$

The positive and negative training data samples can be distinguished through the optimum decision surface provided by the above solution. However, the samples(data) overlap and are not distinguishable linearly, then to reach a reasonable solution, the kernel can be applied. The kernel parameters C, γ may be required to tune accordingly in order to obtain better solutions. The Gaussian kernel is used to solve such kind of problems. The Gaussian kernel can be expressed as follows

$$K\left(x^{(i)}, x^{(j)}\right) = \exp\left(\frac{-\| x^{(i)} - x^{(j)} \|^2}{2\sigma^2}\right) \qquad (7)$$

A classification approach of SVM has been shown in [6],[8].

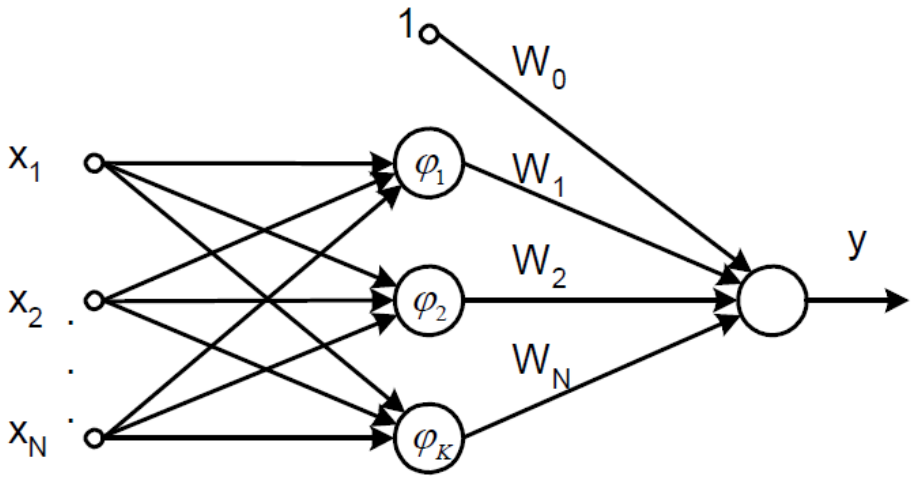


Fig.3.1. Simplified structure of Support Vector machine network.

Networks with radial basis functions RBFs can be used to approximate any function. They consist of a back propagation network with a single hidden layer and a network with a similar

structure. Research findings, which are provided in [30], [31], and [32], support the benefits of RBF neural networks over competing categorization task solutions. A radially symmetric kernel function, K is calculated by M kernel units. In RBF networks, the Gaussian exponential function is frequently utilised.

$$f(\mathrm{x}) = \beta \exp\ (-\sum_i [(x_i - \ c_i)/\sigma_i]^2) \qquad (8)$$

The centroid c_i , constants β and σ_i have to be chosen accordingly to training data set.

In the estimation of a large class of functions, general Gaussian activation functions outperform sigmoid functions [3 3]. A RBF network's basic architecture resembles that of an SVM network (Fig. 1), but instead of using the activation function φ, the exceptional function (8) is used.

3.2.Nearest Neighbor

An example of a supervised machine learning technique is nearest neighbor which is used to resolve classification and regression issues. However, classification approaches are its primary application. The nearest neighbours

algorithm calculates the possibility that a data point will join one group or another based on which group is closest to the data point.

As it doesn't perform any training while providing the training data, it is known as a lazy learning algorithm or lazy learner. Therefore, training takes less time than testing. It doesn't make

any calculations during the training period, just keeps the data. Before a query is run on the dataset, a model is not built. Nearest neighbour is hence perfect for data mining. Because it doesn't make any assumptions about the distribution of the underlying data, it is regarded as a non-parametric method. In a nutshell, closest neighbour seeks to identify the group to which a data point belongs by examining the data points nearby.

Consider the two groups A and B for a moment. To establish whether a data point belongs to group A or group B, the algorithm looks at the states of the neighbouring data points. If the majority of the data points are in group A, then it is quite likely that the particular data point is in group A, and vice versa.

The nearest neighbour method only requires a small number of steps to function. The first stage involves choosing a number (k) of neighbours, after which the Euclidean distance for each of the k neighbours is calculated.

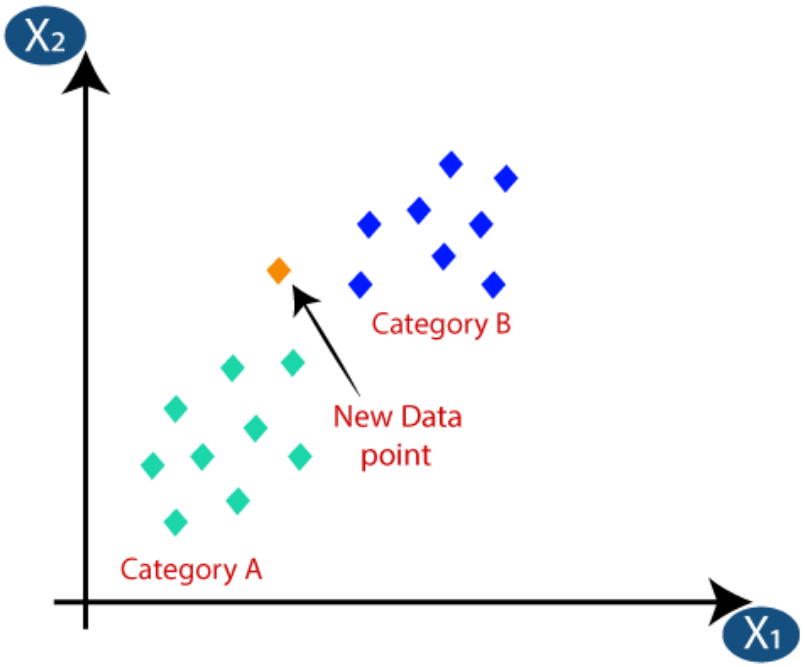


Fig.3.2. Nearest neighbor algorithm data points.

Based on the determined distance, the next k nearest neighbours are chosen. Among these k neighbours, the number of data points in each category is counted, and the new data points are allocated to the category with the maximum number of neighbours.

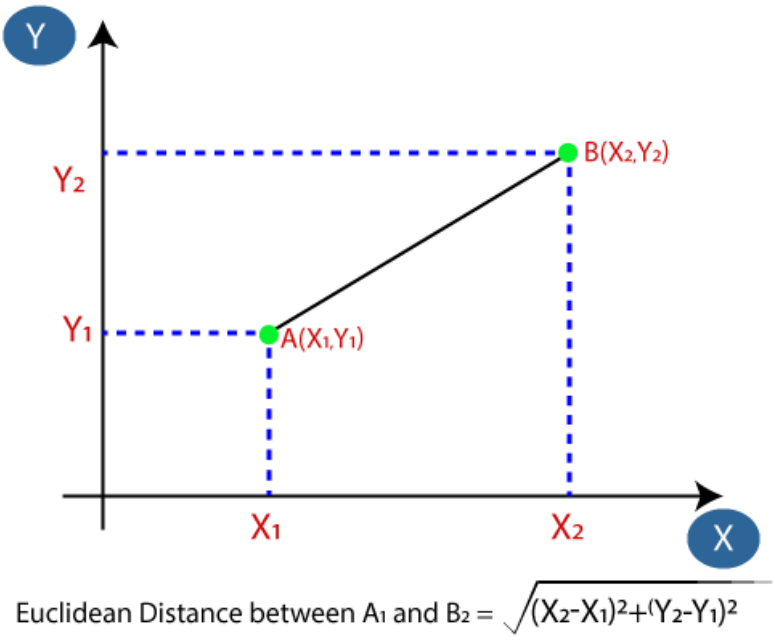


Fig.3.3. Calculation of distance between data points.

3.3.Random Forest

Random Forest is a method that grows many different decision trees and then merges them for a more precise prediction. The Random Forest classifier model is based on the idea that different uncorrelated models work better together than they do separately. Each tree offers a categorization or a "vote" when utilising Random Forest for classification. The classification with the most "votes" is chosen by the forest.

The important thing to note here is that the decision trees that make up the bigger Random Forest model have little to no correlation with one another. The

bulk of the decision trees will be accurate, shifting the overall result in the right direction even while some of them may make mistakes.

The Random Forest classifier primarily operates in two steps. The random forest is produced in the first stage, and in the following stage, predictions are made using the random forest classifier that was created in the first stage.

The Random Forest algorithm has two stages: the first is to generate a random forest, and the second is to produce a prediction using the random forest classifier created in the first step. The entire procedure is illustrated in the figure below, which makes it simple to comprehend.

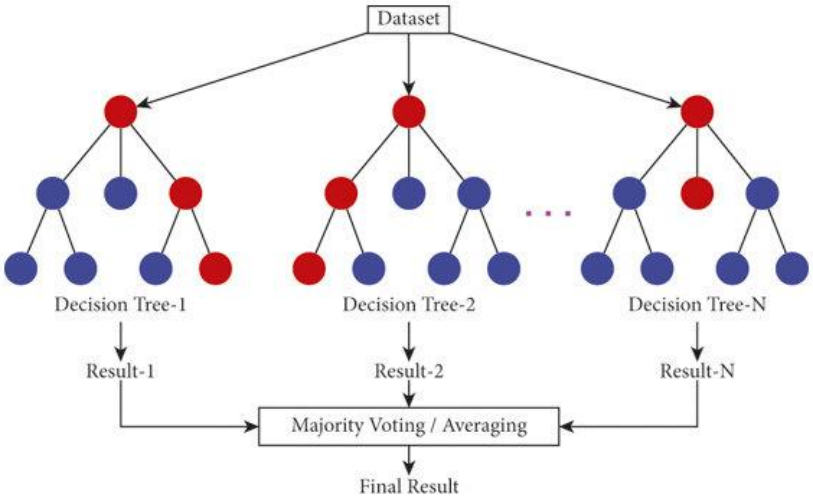


Fig.3.4. Random Forest classifier model.

The random forest classifier model is constructed in the first stage. In the first phase, randomly choosing "K" features out of a total of "m" features is done,

where k << m. The optimal split point is then used to determine the node "d" among the "K" features. The node is then divided into daughter nodes using the best split in the following phase. After then, repeat steps 1 through 3 until "l" nodes have been reached. As there are "n" trees, it is time to grow the forest by repeating these procedures "n" times.

The prediction must be made in the second stage after the creation of the random forest. In a few steps, the random forest prediction is completed. First, test features are utilised to predict the outcome and store the expected outcome (target). Next, votes are generated for each predicted target, and ultimately, the predicted target with the highest number of votes is taken into account as the random forest algorithm's final prediction.

3.4. Extra Trees

Since it constructs numerous trees and divides the nodes using random subsets of features, Extra Trees almost behaves like Random Forest. However, it differs from the random forest in two important ways. It does not bootstrap observations, which entails that it samples without replacing any nodes and splits the nodes at random.

In a nutshell, Extra Trees uses bootstrap = False by default, sampling without replacement, and splitting the nodes based on random splits among a random subset of the features chosen at each node.

Randomness in Extra Trees does not result from data bootstrapping. It originates from the arbitrary divisions of all observations. Because of this, Extremely Randomized Trees is another term for Extra Trees.

3.5.MLP

A multilayer perception (MLP) classifier has a network that feedforward maps the input data sets onto the output data sets. An MLP often has numerous layers of interconnected nodes, with each layer linked to both the one before it and the layer after it. Each node's outputs are weighted

units followed by a nonlinear activation function to separate the data that cannot be separated linearly. The output activation function $a^{(l+1)}$ at layer l+1 is derived by the input activation $a^{(l)}$

$$a^{(l+1)} = \sigma\,(\,w^{(l)}a^{(l)} + \,b^{(l)}) \qquad (9)$$

In the above equation, l corresponds to a specific layer, $w^{(l)}$ and $b^{(l)}$ denotes the weight and bias at layer l and σ represents the nonlinear activation operation function. For an m multiple layer perception, the first input layer is $a^{(l)} = x$ while the last output layer is as follows:

$$h_{w,b}(x) = a^{(m)} \qquad (10)$$

The weights w and bias b in equation (10) are learned by supervised training. The objective function is to minimize the difference between predicted outputs and the desired outputs:

$$J(W, b, x, y) = \frac{1}{2}||h_{w,b}(x) - \mathrm{y}||^2 \qquad (11)$$

Chapter 4

4. Methodology

This chapter describes the overall methodology has been used for this research.

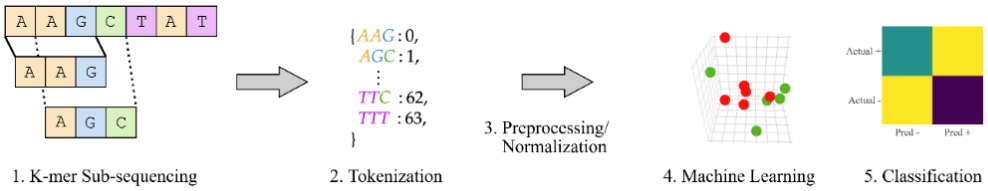


Fig.4.1. Summary of the feature extraction and Machine Learning pipeline used to classify Sars-CoV-2 and other common respiratory viruses.

4.1.Dataset Collection

The dataset used for this research is from National Center for Biotechnology Information (NCBI). Through a variety of mathematical and computational techniques, NCBI carries out fundamental research on biomedical issues at the level of molecules. Additionally, it continues to work in partnership with various NIH institutions, business, academia, and other government-owned organisations. It promotes scientific exchange and aids training initiatives for fundamental and applied computational biology research. For the benefit of the scientific and

medical communities, it creates, distributes, supports, and coordinates access to numerous databases and tools. The NCBI creates and advances the norms for biological nomenclature, data sharing, and databases.

In this work, over 7.91 million viral sequences were extracted from 19,799 genome assemblies from the NCBI Assembly database [20]. Each sequence represents 150nt-long base pairs derived from the source genome assembly. All the DNA samples were independent. DNA of various pathogens including Sars-CoV-2, Coronaviridae, Influenza, Rhinovirus and Metapneumovirus are included in the dataset. In addition to this we included annotated genomes belonging to Humans, which were retrieved from Ensemble 2020 [21] . Including 5 respiratory viruses and the human genome, six total classes were created. The dataset details are shown in Table I. We utilized the preprocessed fasta files available from [17] in our work.

TABLE 1. SUMMARY OF NCBI GENOME SEQUENCES AND ASSEMBLY DATASET.

Class	Number of Bases (millions)	Number of Genome Sequences (millions)	Number of Genome Assemblies	Genus Groups
SARS-CoV-2	0.0	0.86	87	SARS-CoV-2
Coronaviridae	0.0	0.64	12	Alpha, beta, gamma
Influenza	0.0	1.12	1024	H1N1, H2N2, H3N2, H5N1, H7N9, H0N2, Influenza B, others
Metapneumovirus	0.0	0.44	5	Metapneumovirus
Rhinovirus	0.0	1.35	130	A, A1, B, C1, C2, C10, others
Human	525	3.50	18,541	Human transcriptome
Total	**525**	**7.91**	**19,799**	

4.2. Dataset Preprocessing

The dataset has been preprocessed as well as cleaned through 5 different steps. The first step is to remove the file information from each DNA sequence in order to get the raw sequence. Then in the following step, the features have been adapted for applying machine learning classifiers. In the third step, the target column has been labeled by label encoder. Then in forth step, the features are normalized by a scaler function and the dataset was split for training and testing.

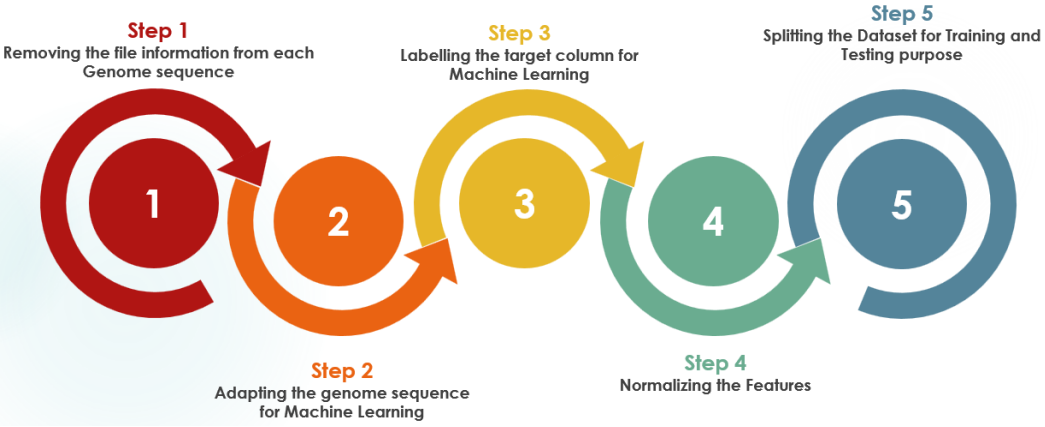


Fig.4.2. Dataset pre-processing steps.

4.2.1. File information removal

There remains a file information in each of the sequences. Each sequence (150nt long) starts with the file along with the location. These information has to be removed in order to get the raw sequence.

>AF144300.1:385:1TTGGGAAAACGACCTCCATGATAACATCTTGTGCT
TCTTTAGCACTGAGATCTGCATGGCCCGGGTTTATATCCACCCGGCG
ACGTATTTTAACTTGGTTTCGGAAATGAACGGGACCGAAGGTTCCA
TGTTTTAACCTTTCAACCTTCT

>FJ445142.1:6004:1CACTAGAAGAGAGTGTGTTTGGTATTGATGGATTA
GAGGCTCTAGATCTAAACACTAGTGCAGGATTTCCTTATGTTTCATT
AGGAATAAAAAAGAAAGACCTTATAGATAAGAAAACAAAAGATATCA
CAAAACTTAAGAAAGCAATTG

Fig.4.3. DNA samples of common respiratory viruses.

If the above DNA samples are investigated then it is found that there are file information Infront of the DNA samples regarding the files and the specific location from where the DNA has been collected from the NCBI database.

4.2.2. Feature Adaption for machine learning application

The features have to be adapted to apply machine learning classifiers. In order to do those two steps has been taken into account. In the first step k-mers has been applied to get the subsequence of the original sequence as well as tokens and then count vectorizer has been applied to get the DNA in numeric values in order to apply machine learning classifiers.

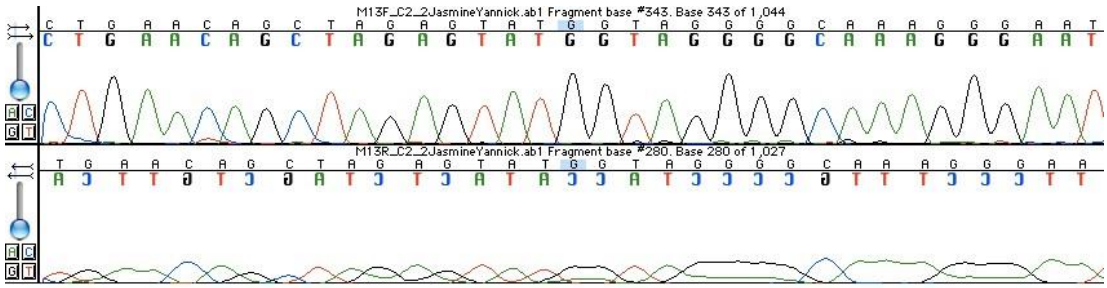


Fig.4.4. Combination of the nucleotides in DNA.

4.2.2.1. k- mer Approach

K-mers are k-length substrings of a biological sequence that are used in bioinformatics. K-mers, which are made up of nucleotides (such as Adenine, Thymine, Guanin, and Cytosine) and are primarily used in the context of computational genomics and sequence analysis which helps to assemble genome sequences, enhance heterologous gene expression, identify species in metagenomic samples, and develop attenuated vaccines.

Typically, the term k-mer refers to all of a sequence's subsequences of length k, such that the sequence AGAT would have four monomers (A, G, A, and T), three 2-mers (AG, GA, AT), two 3-mers (AGA and GAT) and one 4-mer (AGAT). So, a sequence of length L will have $L-k+1$ K-mers and $n\hat{}k$ total possible K-mers where n is the number of possible monomers (for DNA, $n = 4$). If 2-mers is applied where value of K = 2 the below combinations can be found

AAAAAAAAAATTGTAA

(L=16)

AA AA AA AA AA AA AA AA AA AT TT TG GT TA AA

($L-k+1 = 16-2+1= 15$)

Fig.4.5. sub sequencing through k-mers.

Here the sequence length, L is 16 and as the value of K is 2, so 15 subsequences are found.

5.2.2.2. Count vectorization

A fantastic utility offered by the Python scikit-learn module is CountVectorizer. Using the frequency and count of each word that appears in the full text, it is used to convert a given text into a vector. Data is tokenized by CountVectorizer and divided into units called n-grams, the length of which can be specified by giving a tuple to the ngram range argument.

With the application of both K-mer counting and countvectorizer we are able to transform the genome sequences into our desired numeric value for machine learning.

If we apply count vectorizer on the subsequences we got through K-mer, we get the following results.

AA AA AA AA AA AA AA AA AA AT TT TG GT TA AA

{'AA': 0, 'AT': 3, 'TT': 15, 'TG': 14, 'GT': 11, 'TA': 12, 'AC': 1, 'CT': 7, 'TC': 13, 'CA': 4, 'AG': 2, 'GC': 9,'GA': 8, 'GG': 10, 'CC': 5, 'CG': 6} ***(n = 4, k = 2, n^2 = 4^2 = 16)***

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	0	1	0	0	0	0	0	0	0	1	1	0	1	1

Fig.4.6. Countvectorization of the sub sequences found through k-mers.

Here, n=4 for DNA and k=2, so n^k = 4^2 = 16. The original sequence can have a maximum number of 16 subsequences. You can see the maximum possible subsequences here.

4.2.3. Labeling Target Column

There are 5 classes of respiratory virus and a human genome class. In total the whole dataset consists of 6 classes such as Sars-CoV-2, Coronaviridae, Influenza, Rhinovirus, Metapneumovirus and human. A label encoder has been used to adapt the target column.

4.2.4. Feature normalization

The Features have been normalized by the following MinMax scaler function

Class sklearn.preprocessing.MinMaxScaler(feature_range= (0, 1), copy=True)

4.2.5. Splitting dataset

First of all, A dataset of 120k samples were taken where each class contained 20,000 samples. This dataset was divided into 80/20.Where the 80% is the training samples and 20% is the testing samples.

Then the whole dataset was taken into account. Where the number of samples is 7.91 million. In real world, the number of training samples are less

than testing samples. So, 16% samples were taken for training and rest 84% samples for testing. So, ~1.2M samples were used for training and ~6.7M were used for validation.

4.3. Models and Performance metrics

Six different Machine Learning classifiers were utilized from the python library to evaluate the approach of this research in order to classify the dataset found by k-mers and count vectorization such as Nearest Neighbor, Linear SVM, RBF SVM, Extra Trees, Random Forest and MLP.

To assess the model performance precision, recall and F1-score (F1) were evaluated. These metrics consider class imbalance and provide a set of summarized metrics to assess which models are worth fine-tuning. The expressions for precision, recall and F1 scores are shown here. TP, FP, and FN are shorthand notation for True Positive, False Positive and False Negative, respectively.

$$precision = \frac{TP}{TP+FP} \tag{12}$$

$$recall = \frac{TP}{TP+FN} \tag{13}$$

$$F_1 = \frac{2\times precision\times recall}{precision+recall} \tag{14}$$

To assess the model performance precision, recall and F1-score (F1) were evaluated. These metrics consider class imbalance and provide a set of summarized metrics to assess which models are worth fine-tuning. The expressions for precision, recall and F1 scores are shown here.

TP, FP, and FN are shorthand notation for True Positive, False Positive and False Negative, respectively.

4.2. K-fold cross validation

Finally, to ensure the efficacy of the results, a K-fold cross validation was performed over the dataset to ensure the generalizability of the model. The whole dataset was randomly partitioned into 5(k=5) validation sets. In each iteration, one of the 5 validation sets were used as test set and rest of the dataset was used as training set. The process was repeated 5 times as k=5 and the precision, recall and f1-scores were averaged. This way the model has trained and been validated on each subset of the whole dataset, providing robustness to the results found in this research. The expression for the average loss or performance metric L over K iterations, is shown here

$$\mathcal{L} = \frac{1}{K}\sum_{i=1}^{K} \mathcal{L}_i \qquad (15)$$

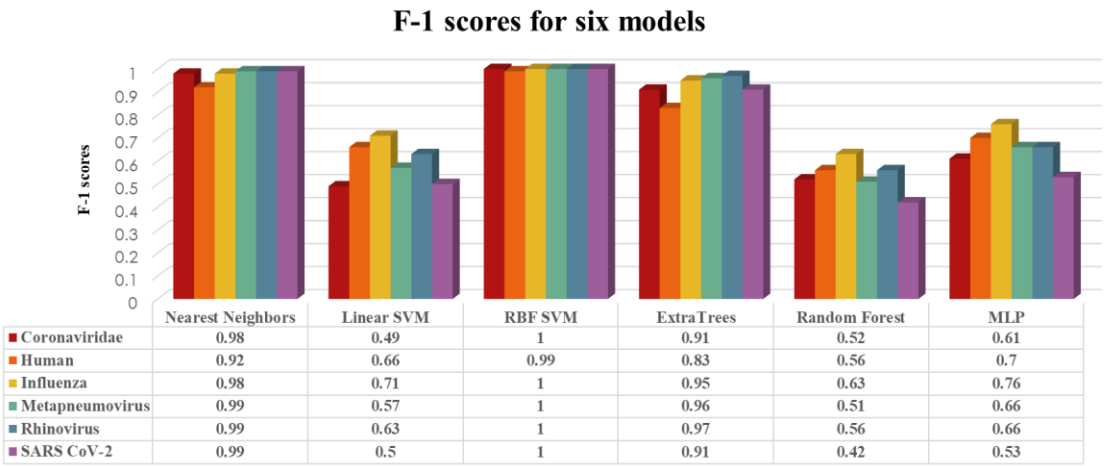
Chapter 5

5. Experiments and results

This section describes the experiments done for this research and the results found through the experiments. This section is divided into the two parts. In the first part, the Model Selection and Performance has been discussed and in the second part, the Fine-Tuned Models and Cross-validation Performance was evaluated.

5.1. Model Selection and performance

Six different machine learning classifiers were applied on 120 thousand samples where each class contained 20 thousand samples. The training set is 80% and test set is 20%. After a 120 of simulations, it was considered that the value of K should be 5 which is still less than the optimal solution of the refence paper from which the dataset was collected where the optimal solution for k was 9. Here in the charts, f1 scores for all the classes are shown in each different model.



	Nearest Neighbors	Linear SVM	RBF SVM	ExtraTrees	Random Forest	MLP
Coronaviridae	0.98	0.49	1	0.91	0.52	0.61
Human	0.92	0.66	0.99	0.83	0.56	0.7
Influenza	0.98	0.71	1	0.95	0.63	0.76
Metapneumovirus	0.99	0.57	1	0.96	0.51	0.66
Rhinovirus	0.99	0.63	1	0.97	0.56	0.66
SARS CoV-2	0.99	0.5	1	0.91	0.42	0.53

Fig.5.1. f1 scores of each class after applying six different Classifiers.

As per the above chart, The Nearest Neighbors, RBF SVM and Extra Trees showed the highest f1 scores and the Linear SVM, Random Forest and MLP classifiers showed lower f1 scores. Nearest Neighbors is parameter less and learns decision boundaries based on the structure of the data itself. Meaning it is likely to aggregate the knowledge of its surroundings to make an informed prediction. It uses all features when determining the distance metric, therefore it encompasses the totality of the feature information when making its prediction on the closest k members. That's why it performed a bit better. Linear SVM (gamma) draws hard decision boundaries between the samples using a hard-margin. The RBF-kernel is able to transform the feature space with the use of the kernel to develop highly non-linear decision boundaries that would have otherwise not been constructed using a linear approach. Random Forest requires a lot of tuning with regards to the size of the tree and how many features are considered at each iteration. For k-mers, excluding some features but not others could have penalized the model performance as considering only some of the 2-mers but not all means the random forest tree is only getting part of the information it requires. Extra trees classifier does not have this problem

as all features are considered for all trees, and therefore outperforms random forest. MLP is easy to overtrain, and the size of the network is likely to have caused overfitting on the training set, leading to poor performance on the validation.

We found that training on 7.91 million samples would have taken a prohibitively long time. To reduce the training time, we randomly chose a smaller training set as follows: from the 7.91 million original dataset, we randomly chose 20k samples for each class, thus forming a data set totalling 120k samples. 120k samples, in which 20k samples were taken from each of the 6 classes, from the 7.91 million and used the 120k subset. The resulting 120k dataset was applied to each classifier under consideration. The following table describes the training and validation time for the six classifiers which were applied on the 120k samples.

Table 2. Comparison between different classifiers based on Training and Validation time.

Model	Training Time (s)	Validation Time (s)
Nearest Neigbors	0.01	73.43
Linear SVM	512.57	99.94
RBF SVM	205.58	51.53
Extra Trees	**0.52**	**< 0.01**
Random Forest	**0.20**	**0.03**
MLP	**23.06**	**0.03**

In the above chart, the training and validation time of six different classifiers are described. Nearest neighbour took lowest training time (0.01s) than others. But the validation time is 73.43s which is much higher than training time. The Linear SVM and RBF SVM took more time in training than validation. The Extra Trees, Random Forest and MLP classifiers also took more training time than validation but still these three classifiers took much less training and validation time than the Nearest Neighbors, Linear SVM and RBF SVM. Accordingly, these three best performers were chosen to train on the entire original dataset of 7.91 million samples. Furthermore, hyperparameter tuning was applied to these best performers to improve the results.

5.2. Fine-tuned model and cross-validation performance

The three classifiers i.e. Random Forest classifier, Extra Trees and MLP classifiers were fined tuned and applied on the whole dataset which is consists of 7.91 million samples. Here the training set was 16% and validation set was 84%. So, about 1.2 million samples were used for training for each of the classifier models and ~6.9 million samples were used for validation. It was chosen in this way because the number of training samples are always much less than the number of test samples in the real world.

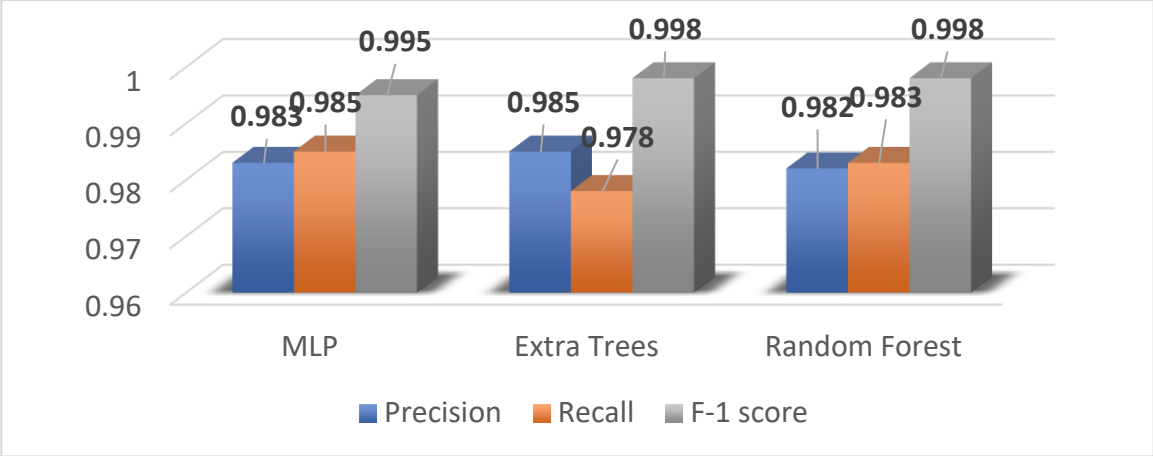


Fig.5.2. Summary of averaged performance metrics over all 6 classes for fine-tuned models.

For the MLP classifier, after applying different parameters several time, a certain set of parameters were passed to get the high f1 score. The hyperparameter

tuning was done through passing some parameters such as initial learning rate was set to 5e-3 which is usually by default set to 0.001. This initial learning rate helps to control the step-size in updating the wights. The hidden layer size has been set to [64,32] which is by default [100,]. The ith element represents the number of neurons in the ith hidden layer. The verbose was set to True instead of false(by default). The early_stopping was set to True which is usually false by Default. If validation score is not improving, then early_ stopping can be used to terminate training. If set to true, it will automatically reserve 10% of training data for validation and stop training if the validation score does not increase by at least l for n_iter_no_change successive epochs. With the exception of a multilabel configuration, the divide is stratified. If early stopping is False, training will cease after n iter no change consecutive trips through the training set if the training loss does not decrease by greater than l. This only works when solver='adam' or'sgd'. And last of all the

validation fraction was set to 0.2 instead of 0.1. By sending the above-mentioned parameter the model reached a high f1 score of 0.995 which is much higher than the one found before doing the hyper parameter tunning.

For Extra Trees classifier, the hyperparameter tuning was done through passing various parameters such as balancing all the class weights through putting the class weight "balanced". The number of trees (n_estimators) in the forest was set to 50 which was by default 100. In order to control the verbosity when fitting and predicting, the verbose was set to 1 instead of default 0. To use all the

processor the number of jobs n_jobs was set to -1 instead of 1.The above hyperparameter tunning was done after several attempts with other parameters and different values and higher f1 0.998 score was found .

For Random Forest classifier, the hyperparameter tuning was done through passing different parameters such as balancing all the class weights through putting the class weight "balanced". The number of trees (n_estimators) in the forest was set to 50 which was by default 100. In order to control the verbosity when fitting and predicting, the verbose was set to 1 instead of default 0. To use all the processor the number of jobs n_jobs was set to -1 instead of 1. The above hyperparameter tunning was done after several attempts with other parameters and different values and higher f1 0.998 score was found .

After all the experiments and analyzing the results it was found that the Fine tuned Random Forest classifier provides higher scores than other. The precision, recall and f1 score for Random Forest classifier is 0.982, 0.983 and 0.998 respectively and it also takes less time to process. For every ~100,000 samples it takes around 18.6 seconds. The classification matrix of a Random Forest classifier are shown below:

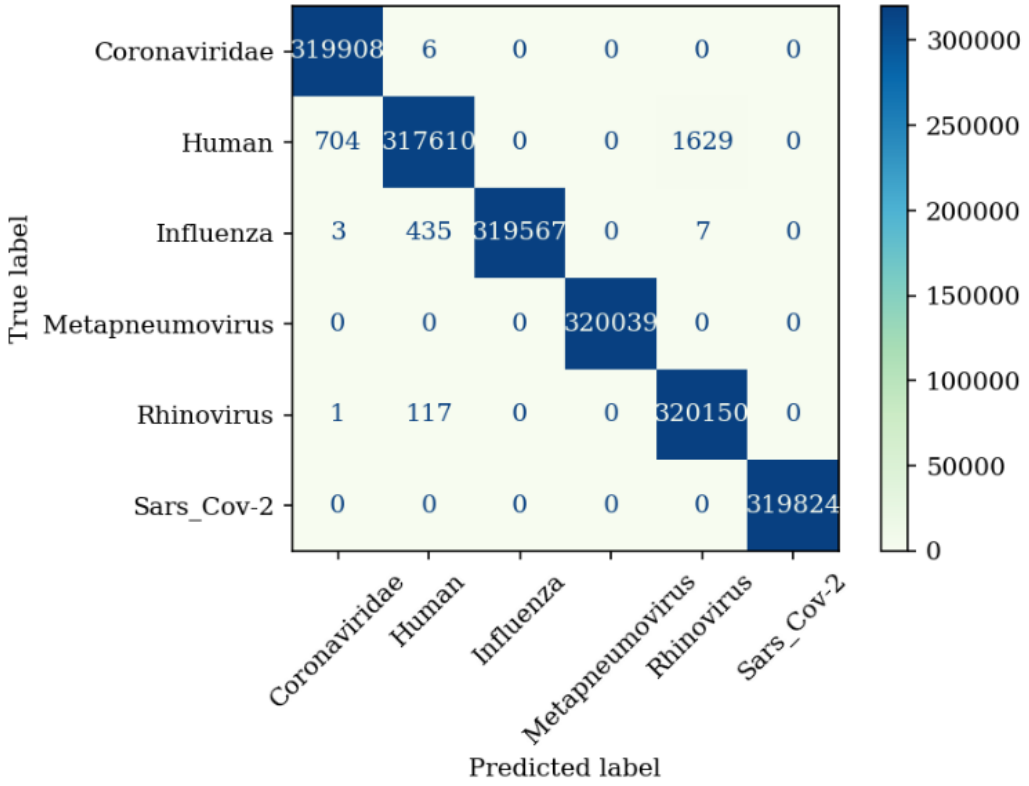


Fig.5.3. Confusion matrix over all 6 classes predicted on a fine-tuned Random Forest Classifier with F1 of 0.998. Validation was carried out on 1.92M examples.

As per the above classification matrix, the highest FP rates for the Rhinovirus being classified as Human (1629) and Coronaviridae being classified as human (704). Overall, the non-Human classes all achieved a near-perfect score with very little FP rates between them. The

classes which achieved perfect F1 score included Influenza, Metapneumovirus and SARS CoV-2; with most of the FP and FN that were not in the single digits belonging to classes involving the Human class. Possibly the large number of unique genome assemblies belonging to the Human class could be one reason for the higher rates of FP and FN, with the other classes being sourced from fewer assemblies, and therefore, providing more homogeneity. Interesting enough Coronaviridae is a large family of single-

stranded RNA viruses that include subfamilies of coronaviruses [24], however, the models did not find consistent overlap in their 5-mers with SARS CoV-2.

Chapter 6

6. Conclusion and Future Work

This chapter describes the conclusion and future work for the further development of this research.

6.1. Conclusion

Machine learning may be used as an aid to help clinicians diagnose and identify respiratory infections from genome sequences. The common respiratory viruses might have same kind of syndromes as all of them attack the respiratory system. In order to properly identify the

exact respiratory virus, DNA classification is one of best approach for the clinicians so that proper vaccines can be applied.

However, this research introduced a k-mer (k=5) approach along with a fined tuned Random Forest Classifier was applied to classify RNA viruses

(Coronaviridae, Influenza, Metapneumovirus, Rhinovirus and Sars_Cov-2), and achieved a classification rate of 99.8 %. Moreover, by doing the hyper parameter tuning in Random Forest classifier by passing parameters such as n_jobs=-1 (helps in using all the processors to run the program), class_weight = "balanced", n_estimators=50 (the number of trees in the forest) and verbose=1 (controls the verbosity when fitting and predicting), high classification accuracy rate (99.8%) was achieved. It was also found that MLP, Extra Trees and Random Forest yielded a much faster processing time than Nearest neighbor, Linear SVM and RBF SVM.

6.2. Future Work

This proposed scheme provides a reasonable solution for classifying the common respiratory viruses through genome assemblies. Some future work can be done for the expansion of the model:

a. The samples used in this research are from independent virus RNA sequence. The future work can be included mutations and substitutions to the validation set to test the robustness of the methods used in this research to slightly mutated sequence.

b. In this research almost 7.91 million genome sequences were examined. This work also limits the samples to sequences 150 bases long, meaning each

example used represents a small snippet of the full genome assembly, requiring less inference time for prediction but less information to learn during training. The future work for this research could be working with a greater number of genome sequences as well as more genome assemblies with longer bases.

c. It was found that the genome sequences, while not random, were somewhat deterministic and may indicate a deeper character when analyzed at several different levels of measure. Therefore, it is proposed to use fractal and chaos theory to create additional features, which may add additional degrees of freedom to classify the data.

References

[1] M. Thomas and P. A. Bomar, “Upper Respiratory Tract Infection,” in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2022. Accessed: Apr. 01, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK532961/

[2] Z. Gao *et al.*, “A systematic review of asymptomatic infections with COVID-19,” *Journal of Microbiology, Immunology and Infection*, vol. 54, no. 1, pp. 12–16, Feb. 2021, doi: 10.1016/j.jmii.2020.05.001.

[3] D. Kim, J. Quinn, B. Pinsky, N. H. Shah, and I. Brown, “Rates of Co-infection Between SARS-CoV-2 and Other Respiratory Pathogens,” *JAMA*, vol. 323, no. 20, pp. 2085–2086, May 2020, doi: 10.1001/jama.2020.6266.

[4] J. S. Musuuza, L. Watson, V. Parmasad, N. Putman-Buehler, L. Christensen, and N. Safdar, “Prevalence and outcomes of co-infection and superinfection with SARS-CoV-2 and other pathogens: A systematic review and meta-analysis,” *PLOS ONE*, vol. 16, no. 5, p. e0251170, May 2021, doi: 10.1371/journal.pone.0251170.

[5] J. Chen, “COVID-19 Scientific Advisory Group Rapid Response Report,” Alberta Health Services, COVID-19 Scientific Advisory Group, Alberta, May 2020. Accessed: Mar. 22, 2022. [Online]. Available: https://www.albertahealthservices.ca/assets/info/ppih/if-ppih-covid-19-sag-anti-microbial-use-for-secondary-infections-rapid-review.pdf

[6] C. Feldman and R. Anderson, “The role of co-infections and secondary infections in patients with COVID-19,” *Pneumonia*, vol. 13, no. 1, p. 5, Apr. 2021, doi: 10.1186/s41479-021-00083-w.

[7] S. L. Emery *et al.*, “Real-Time Reverse Transcription–Polymerase Chain Reaction Assay for SARS-associated Coronavirus,” *Emerg Infect Dis*, vol. 10, no. 2, pp. 311–316, Feb. 2004, doi: 10.3201/eid1002.030759.

[8] F.-E. Chen *et al.*, “Ratiometric Multiplexed PCR Assay on a Portable Platform for Bacterial

Identification from Urine," in *2019 20th International Conference on Solid-State Sensors, Actuators and Microsystems Eurosensors XXXIII (TRANSDUCERS EUROSENSORS XXXIII)*, Jun. 2019, pp. 1116–1119. doi: 10.1109/TRANSDUCERS.2019.8808318.

[9] H. Tombuloglu, H. Sabit, E. Al-Suhaimi, R. A. Jindan, and K. R. Alkharsah, "Development of multiplex real-time RT-PCR assay for the detection of SARS-CoV-2," *PLOS ONE*, vol. 16, no. 4, p. e0250942, Apr. 2021, doi: 10.1371/journal.pone.0250942.

[10] P. D. Cristea, "Conversion of nucleotides sequences into genomic signals," *Journal of Cellular and Molecular Medicine*, vol. 6, no. 2, pp. 279–303, 2002, doi: 10.1111/j.1582-4934.2002.tb00196.x.

[11] J. Josse, A. D. Kaiser, and A. Kornberg, "Enzymatic synthesis of deoxyribonucleic acid. VIII. Frequencies of nearest neighbor base sequences in deoxyribonucleic acid," *J Biol Chem*, vol. 236, pp. 864–875, Mar. 1961.

[12] N. Goldman, "Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences.," *Nucleic Acids Res*, vol. 21, no. 10, pp. 2487–2491, May 1993.

[13] C. F. Mugal, C. C. Weber, and H. Ellegren, "GC-biased gene conversion links the recombination landscape and demography to genomic base composition," *BioEssays*, vol. 37, no. 12, pp. 1317–1326, 2015, doi: 10.1002/bies.201500058.

[14] G. S. Randhawa, M. P. M. Soltysiak, H. E. Roz, C. P. E. de Souza, K. A. Hill, and L. Kari, "Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study," *PLOS ONE*, vol. 15, no. 4, p. e0232391, Apr. 2020, doi: 10.1371/journal.pone.0232391.

[15] S. Solis-Reyes, M. Avino, A. Poon, and L. Kari, "An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes," *PLOS ONE*, vol. 13, no. 11, p. e0206409, Nov. 2018, doi: 10.1371/journal.pone.0206409.

[16] N. G. Nguyen *et al.*, "DNA Sequence Classification by Convolutional Neural Network," *Journal of Biomedical Science and Engineering*, vol. 9, no. 5, Art. no. 5, Apr. 2016, doi: 10.4236/jbise.2016.95021.

[17] P. Acera Mateos, R. F. Balboa, S. Easteal, E. Eyras, and H. R. Patel, "PACIFIC: a lightweight deep-learning classifier of SARS-CoV-2 and co-infecting RNA viruses," *Sci Rep*, vol. 11, no. 1, Art. no. 1, Feb. 2021, doi: 10.1038/s41598-021-82043-4.

[18] R. Sarkar *et al.*, "Comprehensive analysis of genomic diversity of SARS-CoV-2 in different

geographic regions of India: an endeavour to classify Indian SARS-CoV-2 strains on the basis of co-existing mutations," *Arch Virol*, vol. 166, no. 3, pp. 801–812, Mar. 2021, doi: 10.1007/s00705-020-04911-0.

[19] G. S. Dlamini *et al.*, "Classification of COVID-19 and Other Pathogenic Sequences: A Dinucleotide Frequency and Machine Learning Approach," *IEEE Access*, vol. 8, pp. 195263–195273, 2020, doi: 10.1109/ACCESS.2020.3031387.

[20] P. A. Kitts *et al.*, "Assembly: a resource for assembled genomes at NCBI," *Nucleic Acids Res*, vol. 44, no. Database issue, pp. D73–D80, Jan. 2016, doi: 10.1093/nar/gkv1226.

[21] A. D. Yates *et al.*, "Ensembl 2020," *Nucleic Acids Research*, vol. 48, no. D1, pp. D682–D688, Jan. 2020, doi: 10.1093/nar/gkz966.

[22] M. Nishita *et al.*, "Ror2 signaling regulates Golgi structure and transport through IFT20 for tumor invasiveness," *Sci Rep*, vol. 7, no. 1, Art. no. 1, Jan. 2017, doi: 10.1038/s41598-016-0028-x.

[23] S. M. Omohundro, "Five Balltree Construction Algorithms," 1989.

[24] S. Payne, "Family Coronaviridae," *Viruses*, pp. 149–158, 2017, doi: 10.1016/B978-0-12-803109-4.00017-9.

[25] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," Advances in kernel methods, ACM Digital Library, pp. 185-208 , 1999..

[26] Y. LeCun, L. D.Jackel, L. Bottou, A. Brunot, C. Cortes, J.S.Denker, H.Drucker, I. Guyon, U. A. Müller, E. Säckinger, P. Simard and V. Vapnik, "Comparison of learning algorithms for handwritten digit recognition," in International Conf. Artif. Neural Networks, 1995.

[27] E. Osuna, R. Freund and F. Girosit, "Training support vector machines: an application to face detection," in IEEE Computer Society Conference on Computing, Viison and Pattern Recognition.

[28] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna and T. Poggio, "Pedestrian detection using wavelet templates," in Computer Vision & Pattern Recognition, IEEE Computer Society.

[29] M. Chester, Neural Networks. A tutorial, London, Prentice Hall, 1993, pp.50-66

[30] Y. H. Song, Q. X. Xuan, and A. T. John, "Comparison studies of five neural networks based fault classifiers for complex transmission lines, Electric Power System Research, vol. 43, pp. 125-132, 1997.

[31] W.-M Lin, C.-D. Yang, J.-H. Lin and M.-T. Tsay, “A fault classification method by RBF neural network with OLS learning procedure”, IEEE Trans. on Power Delivery, vol. 16, No. 4, pp. 473-477, October 2001.

[32] K. G. Narendra, V. K. Sood, K. Khorasani and R. Patel, “Application of a radial basis function (RBF) neural network for fault diagnosis in a HVDC system”, IEEE Transactions on Power Systems, vol. 13, No. 1, February 1998, pp. 177-183.

[33] S. Geva, J. Sitte, “Networks of Exponential Neurons for Multivariate Function Approximation”, Proc. of IJCNN Singapore, 1991, pp.2305- 2310.