International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Innovative Methods for Classifying COVID-19 using Amino Acid Encoding Combined with Recursive Feature Elimination for XGBoost Classification

Mr. Anurag Golwalkar *[1], Dr. Abhay Kothari [2]

Submitted: 26/11/2023 **Revised**: 28/12/2023 **Accepted**: 10/01/2024

Abstract: Introduces a groundbreaking method for identifying COVID-19 by analyzing amino acid sequences. This method employs a two-fold approach: firstly, encoding the amino acids to transform biological data into a format suitable for computational analysis, and secondly, utilizing Recursive Feature Elimination (RFE) to refine the dataset for enhanced classification accuracy with the XGBoost algorithm. The study's core lies in its innovative use of RFE, a technique that iteratively evaluates and discards the least significant features, thereby streamlining the classification process. When combined with the robust, gradient-boosting framework of XGBoost, this approach not only simplifies the complex amino acid sequences but also significantly improves the classification performance. The results are compelling: the method achieved an impressive accuracy of 99.89%, with a sensitivity of 99.87% and specificity of 99.75%. These metrics were obtained using just 7 features, a notable reduction compared to other methods, which not only underscores the efficiency of the approach but also reduces computational time to just 2.43 seconds. This research contributes significantly to the field of bioinformatics and epidemiology by offering a fast, accurate, and efficient method for COVID-19 classification. The approach's simplicity and high accuracy make it a promising tool for rapid screening and early detection of COVID-19, which is crucial in managing and controlling outbreaks.

Keywords: *COVID-19, Amino Acid Sequences, Recursive Feature Elimination, XGBoost, NGDC dataset.*

1. Introduction

In 2019, the world encountered a highly infectious pathogen, the severe acute respiratory syndrome Coronavirus 2 (SARS-CoV-2), responsible for the respiratory illness known as coronavirus disease 2019 (COVID-19). This virus rapidly spread across the globe, leading to an extensive number of infections and numerous deaths. Consequently, in March 2020, the World Health Organization (WHO) declared COVID-19 a global pandemic due to its severe impact on human health.

This research introduces an innovative methodological framework that combines amino acids encoding with the efficiency of Recursive Feature Elimination (RFE) to improve the classification performance of the XGBoost algorithm. The framework is based on the understanding that the genetic structure of the SARS-CoV-2 virus, primarily consisting of proteins and amino acids, is key to its identification and classification. Amino acids encoding transforms these biological sequences into a numerical format suitable for computational analysis. However, the complexity and high dimensionality of this data present significant challenges.

[1,2]*Department of Computer Science and Engineering*
[1] *Research Scholar, SAGE University, Indore*
[2] *Research Supervisor, SAGE University, Indore, (M.P.), India.*
E-mail Id: [1] *agolwelkar@gmail.com,*
[2]*abhaykothari333@gmail.com*
** Corresponding Author: Mr. Anurag Golwalkar*
Email: agolwelkar@gmail.com

To address these challenges, the study employs RFE, a method that effectively selects essential features by systematically eliminating less significant variables. This streamlines the dataset for more precise and efficient analysis. The study further integrates RFE with the XGBoost classifier, a renowned machine learning algorithm appreciated for its speed and accuracy in handling large, complex datasets. By merging these advanced techniques, the research aims to enhance the accuracy and efficiency of COVID-19 classification while also alleviating the computational demands typically associated with processing extensive genomic data. This approach represents a significant step forward in the rapid and precise classification of COVID-19, contributing to the broader efforts of managing and understanding the pandemic.

The goals of this study are outlined below:

- To develop a new and precise model for COVID-19 capable of classifying different types of coronaviruses and distinguishing SARS-CoV-2 from other variants.
- To employ machine learning methods to assess the model's performance in metrics such as accuracy, precision, sensitivity, and specificity.
- To minimize the feature set of the new model to improve its overall efficiency and effectiveness.

This paper is organized in the following manner: Section 1 introduces the topic. Section 2 explores previous research

and studies in a literature review. Section 3 outlines the methodology used and describes the proposed model. Section 4 examines the application of the model and discusses the NGDC database. Section 5 is dedicated to presenting the experimental results. Finally, Section 6 concludes the paper with key findings and observations.

2. Literature Review

Matsuki, Yoshio et al. (2022): This study sought to trace the origin of the envelope protein in Covid-19, hypothesizing its similarity to human liver enzymes. A specific amino acid sequence, shared by both, was identified, leading to further investigation using quantum mechanics. This approach aimed to understand electron capture probabilities in the Covid-19 envelope protein and how it might differ from human liver enzymes, potentially revealing other proteins and shedding light on the virus's origin. [1]

Setthapramote, Chayanee et al. (2023): Research focused on the genetic diversity of SARS-CoV-2 variants in Thailand during three COVID-19 waves. Whole-genome sequencing of 33 samples revealed distinct dominant variants in each wave, with particular mutations linked to disease severity and transmission. This study emphasized the importance of genome analysis in tracking virus evolution and formulating pandemic responses. [2]

Nagata, Naoyoshi et al. (2023): Investigating the relationship between gut microbes, metabolites, cytokines, and COVID-19 complications, this study employed shotgun metagenomic sequencing and metabolomics. It found significant correlations, particularly in severe cases, between COVID-19-related microbes, gut metabolites, and inflammatory cytokines. This research offers potential for microbial-based diagnostics and insights into the biological mechanisms of the disease. [3]

Zhang, Lizhou et al. (2023): Examining the Pfizer-BioNTech and Moderna mRNA-LNP vaccines, this study compared their components. Despite identical spike proteins, differences were found in ionizable lipids, untranslated regions (UTRs), and nucleotide composition. Notably, Moderna's lipid SM-102 showed better performance in mRNA delivery and antibody production than Pfizer-BioNTech's ALC-0315, offering insights for future vaccine enhancements. [4]

Bi, DeWu et al. (2023): This research undertook genomic characterization of SARS-CoV-2 variants using next-generation sequencing. Analyzing 80 genomes from 39 patients, it identified a new variant with significant mutations in the spike protein. These mutations were found to alter the flexibility of the S1 subunit, impacting ACE2 receptor interaction, underscoring the virus's ongoing evolution. [5]

Gama-Almeida, Marcos C. et al. (2023): In a study based in Rio de Janeiro, Brazil, NMR and MS-based metabolomics were used to identify metabolic markers related to COVID-19 severity and outcomes. Findings showed altered metabolite levels linked to severe disease, with non-survivors exhibiting signs of liver and kidney dysfunction. The study also highlighted sex-based metabolic differences, emphasizing their importance in pandemic response. [6]

Zhang, Xiaoxiao et al. (2023): Investigating the impact of exogenous factors like high valine and glycine in SARS-CoV-2 proteins, this research explored how these proteins influence calcium buildup and aggregation in cells, potentially causing cellular stress and long-term health effects. This may explain some post-recovery sequelae in COVID-19 patients. [7]

Sreejith, S. et al. (2023): This review highlighted the growing role of biosensors, particularly Graphene Field Effect Transistor (Gr-FET) based biosensors, in sensitive and selective biomarker detection, including SARS-CoV-2. The review focused on the performance of various Gr-FET biosensors, underscoring their potential in rapid COVID-19 detection and broader healthcare applications. [8]

Zhou, Shilin et al. (2023): Addressing the need for SARS-CoV-2 therapeutics, this study focused on the envelope (E) protein as a drug target. It compared the E protein's amino acid sequence with other human coronaviruses, examining its role in viral pathogenesis and potential drug targets. The study also discussed the implications of E protein mutations for the virus and host. [9]

Benazraf, Amit et al. (2023): Conducted a comprehensive study to understand the SARS-CoV-2 ORF3a viroporin by creating a library of bacteria with various mutations of the protein. Genetic selection and deep sequencing identified mutations that affected ORF3a's influence on bacterial growth, especially in conserved residues. This research offers insights into ORF3a's functionality and suggests a method for analyzing viroporins, with implications for potential COVID-19 therapeutics. [10]

Bodaghi, Ali et al. (2023): This review provides an extensive overview of biomarkers, discussing their history, definitions, classifications, and characteristics. It delves into their transformative impact on society, focusing on their use in diagnosing, prognosing, and treating diseases over the past decade. The review aims to inspire new research and development in biomarkers, emphasizing their growing role in early disease detection and risk evaluation. [11]

Oliveira Andrade et al. (2023): Explored the possible etiology of post-COVID-19 type 1 diabetes (DM1), specifically examining the development of anti-Zinc Transporter 8 antibodies (ZnT8A) through molecular mimicry. The study assessed similarities between ZnT8 protein and COVID-19 proteins, suggesting how COVID-

19 infection might trigger anti-ZnT8A production and lead to DM1. [12]

Choi, Gihoon et al. (2023): Reviewed the expanded application of Reverse Transcription Loop-mediated isothermal Amplification (RT-LAMP) during the SARS-CoV-2 pandemic. The review compares RT-LAMP with RT-qPCR and rapid antigen tests, discusses advancements in sample preparation, challenges in primer and assay design, and highlights its potential in future diagnostic applications for various conditions. [13]

Ana Paula, C. et al. (2023): Investigated COVID-19's pathophysiology in Brazil, especially Rio de Janeiro, using NMR and MS-based metabolomics. The study compared severe COVID-19 cases with controls, identifying metabolic changes that suggest methyl donor dysregulation and liver and kidney dysfunction. The findings, particularly more pronounced metabolic changes in women, underscore the importance of gender considerations in pandemic response. [14]

Siebert, Hans-Christian et al. (2023): Utilized the global SARS-CoV-2 outbreak to test new medical strategies in glycobiology, nanopharmacology, and nanomedicine. The research linked clinical data with structural biology and evaluated new diagnostic and therapeutic approaches, including incretin mimetics for long-COVID symptoms. It emphasized the role of biophysical factors in understanding and treating long COVID. [15]

Zhang, Jingjing et al. (2023): Focused on the Cytochrome P450s (CYPs) enzyme superfamily, analyzing their diverse sequences and conserved structural features. The study aimed to uncover common functional traits among human CYPs that might be related to enzyme malfunction and disorders, contributing to the understanding of metabolic disorders and abnormal drug metabolism. [16]

Ferreira, Luís Marcos Cerdeira et al. (2023): Reviewed advancements in electrochemical biosensors for detecting COVID-19 biomarkers. The article discusses various biorecognition strategies, their specificity, sensitivity, and the challenges and progress in the field. It addresses the need for rapid, accurate, and cost-effective diagnostic tools and considerations for their clinical application. [17]

Chen, Ke-Lin et al. (2023): Developed a cell-specific, constraint-based modeling technique for the alpha variant of SARS-CoV-2 in infected lungs. The study constructed a viral biomass reaction (VBR) using gene sequences and lipid stoichiometry between the virus and host cell. Integrating VBR with gene expression data and the Recon3D human metabolic network, they created a genome-scale metabolic model and an antiviral target discovery (AVTD) platform. This platform identified dihydroorotate dehydrogenase (DHODH) as a potential therapeutic target to inhibit viral replication with minimal side effects, showcasing the potential of computational systems biology in COVID-19 antiviral research. [18]

Wilner, Ofer I. et al. (2023): Discussed the rising demand for quick, reliable, and affordable Point-of-Care (POC) diagnostic tests during the COVID-19 pandemic. The paper reviews DNA Polymerase Chain Reaction (PCR) and isothermal amplification reactions and current POC nucleic acid diagnostic devices, bridging academic research with industry developments. It highlights advancements in portable, cost-effective, and automatic POC nucleic acid diagnostic tools, emphasizing their crucial role in managing COVID-19 and future infectious diseases. [19]

Hossain, Kazi Amirul et al. (2023): Analyzed the role of acidic residues (Asp/Glu) in DNA binding specificity, particularly their preference for cytosine, through classical and ab initio simulations. The study found that Asp/Glu act as negative selectors at non-cytosine sites and require multiple cytosines for favorable interaction, providing insights into sequence-specific DNA binding mechanisms. [20]

Taysi, Seyithan et al. (2023): Explored the properties and potential of Caffeic acid phenethyl ester (CAPE), a flavonoid with anti-cancer and anti-inflammatory effects. Molecular docking studies suggest CAPE's binding capability with cancer cell replication enzymes and its inhibitory effects against the main protease of SARS-CoV-2, indicating potential applications in COVID-19 treatment. The review discusses CAPE's role in alternative medicine and its pharmacological value. [21]

Nakhaie, Mohsen et al. (2023): Investigated the impact of mutations in the NSP2 gene of SARS-CoV-2 on immune evasion, pathogenicity, and transmission speed. Analyzing RNA from COVID-19 patients using RT-PCR, gel electrophoresis, and Sanger sequencing, the study identified significant mutations in NSP2 and assessed their effects on protein structure and stability, offering insights into potential therapeutic targets and the virus's evolutionary path. [22]

Jerca, Florica Adriana et al. (2023): Studied the use of Poly(2-isopropenyl-2-oxazoline) (PiPOx) in creating smart biomaterials for siRNA transfection. By modifying PiPOx with amino acid side chains, the study developed cationic polymethacrylamides demonstrating efficient siRNA complexation and high in vitro transfection efficiency, potentially advancing siRNA-based therapies. [23]

Rtayli, Naoufal et al. (2020): Addressed Credit Card Fraud (CCF) by proposing a new hybrid model for Credit Card Fraud Detection (CCFD) using Machine Learning (ML) methods. Combining Recursive Feature Elimination (RFE), GridSearchCV, and Synthetic Minority Oversampling (SMOTE), the model showed effectiveness in identifying

fraudulent transactions, surpassing previous studies in robustness and efficiency. [24]

S. Lefkovits, L. Lefkovits (2017): Focused on the challenges in object detection in machine vision, introducing a method using Gabor filters for face and eye feature extraction. The study proposed using entropy measurements and information gain from weak classifiers for feature selection due to high data dimensionality, comparing this approach with other learning methods. [25]

F. Ardelean (2017): Analyzed a part manufactured in three shifts using the Design for Six Sigma (DFSS) and Analysis of Variance (ANOVA) method. The study applied one-way ANOVA to determine if the manufacturing shift affected the realization of a significant characteristic dimension, providing insights for production quality consistency. [26]

E. Benhamou, V. Melot (2010): Revisited the Pearson Chi-squared independence test, offering a modern perspective and intuitive graphical presentation. The paper aimed to enhance the understanding and application of the Chi-squared test in statistical analysis. [27]

Berezhnoy, Georgy et al. (2023): Investigated metabolic, proteomic, and immunologic phenotypes in patients with acute and long-term COVID-19 syndrome (LTCS). Using 1H-NMR-based metabolomics and cytokine quantification, the study identified distinct metabolic and immunologic profiles in LTCS, suggesting key roles of immune dysregulation and inflammation. [28]

Fopase, Rushikesh et al. (2023): Reviewed the potential of siRNA-mediated mRNA degradation against genetic disorders and COVID-19. Discussing the challenges of siRNA stability and delivery, the review highlighted nanotechnology solutions and detailed various nano formulations and surface functionalization techniques, aiming to improve siRNA-based therapies. [29]

3. Proposed Methodology

3.1 Proposed Method:

The presented model employs feature selection algorithms to identify and extract relevant features, subsequently reducing the feature set by eliminating those deemed irrelevant. It incorporates four distinct methods of feature selection to achieve this:

- Recursive Feature Elimination (RFE) [24]
- Information gain (IG) [25]
- Analysis of variance (ANOVA) test [26]
- Chi-square ($\chi 2$) statistic test [27]

The model reduces the number of extracted features by discarding those that are irrelevant. In the context of protein sequence analysis, it involves transforming sequence characters into numerical values using amino acid encoding for feature extraction. Finally, the model classifies and predicts the type of coronavirus using six different machine learning algorithms:

- Bagging ensemble (BE)
- Decision trees (DT)
- Gradient boosting (GB),
- k-nearest neighbors (KNN)
- RF
- SVM.
- XGBoost (Extreme Gradient Boosting).

In summary, as shown in Fig. 1, the proposed model consists of three phases:

- Feature extraction,
- Feature reduction,
- Classification.

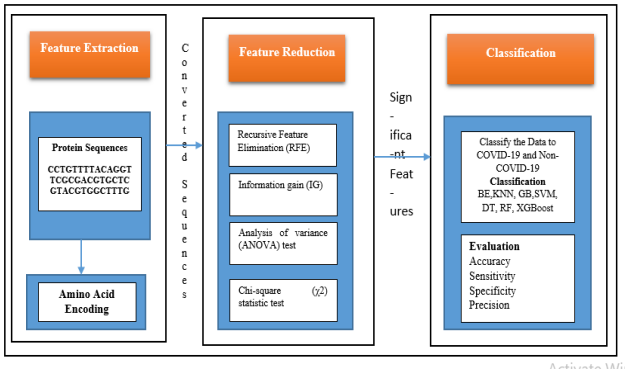

Fig 1. Proposed working flow

3.2. Feature extraction phase

In order to extract characteristics from viral protein sequences, the amino acid encoding approach is used during the phase that occurs throughout the feature extraction process. Utilising the volume and dipole moment of amino acids, this technique takes use of two important physicochemical features of amino acids. Molecular modelling and density-functional approaches are used in order to ascertain the values for volume and dipole. As shown in Table 1, the twenty amino acids are divided into seven unique groups according to the volume and dipole values that were computed as part of the classification process.

The categorization of amino acids is based on the volumes and dipole values of their side chains, as shown in Table 1. This information is derived from Reference [32].

Class number	Dipole scale	Volume scale	Amino acids
1	-	-	A, G, V
2	-	+	I, L, F, P
3	+	+	Y, M, T, S

4	++	+	H, N, Q, W
5	+++	+	R, K
6	+'+'+'	+	D,E
7	+	+	C

According to Table 1, the dipole scale varies between 1.0 and 3.0 as follows:

- "-": dipole value is less than 1.0,
- "+": dipole value is between 1.0 and less than 2.0,
- "++": dipole value is between 2.0 and less than 3.0,
- "+++" dipole value is greater or equal than 3.0, and
- "+'+'+'" dipole value is greater than 3.0 with opposite orientation.

On the scale of volume used in the categorization of amino acids, the symbol "-" is used to indicate that the volume is less than 50, while the symbol "þ" is used to indicate that the volume is larger than 50. Specifically, because of its capacity to form disulfide bonds, cysteine (C) has been moved from Class 3 to Class 7 in the classification system. The normal twenty amino acid characters are expanded to include four ambiguous amino acid characters due to the inherent uncertainties that are present in protein sequencing. As a consequence of this, an eighth class, which is denoted by the number zero, is added to the seven classes of amino acids that were first established. X, which is an unknown amino acid, Z, which might be either glutamic acid E or glutamine Q, and B, which may be either aspartic acid D or asparagine N, are the three ambiguous amino acids that were included in this new class. J, which may be interpreted as either leucine L or isoleucine I, is classified as an ambiguous amino acid and is put in Class 2, which corresponds to the class of amino acids that are classified as I and L.

The high incidence of mistake that amino acid sequencers have is the root cause of the ambiguity that exists in amino acid codes. During the sequencing process, if the machine is unable to conclusively identify an amino acid (for example, differentiating between L and I), it will use the code J to signal the uncertainty. This means that the amino acid might be either I or L.

This categorization of amino acids is shown above in Table 2. The amino acid encoding technique involves the substitution of each character of an amino acid with the class number that corresponds to that amino acid. This process effectively converts the amino acids into numerical values that are based on the eight classes. The Amino Acid Composition (AAC) technique is then used to the encoded amino acids once this step has been completed. Through the use of a certain mathematical formula, this technique determines the frequency of each class.

$$Freq_i = \frac{Num_i}{Len}, i \in \{0,1,\ldots\ldots,7\}, \quad (1)$$

The term "Freqi" is used to refer to the frequency of Class I in the amino acid sequence, as indicated in equation (1). This is the approach that has been specified. The formula for calculating it is as follows: Numi is the number of times Class i appears in the protein sequence, and Len is the entire length of that protein sequence. Following this, each class frequency is taken into consideration as a separate feature of the protein sequence, which ultimately results in the extraction of eight features that are based on the physicochemical characteristics of the amino acids.

This list of eight categories of amino acids may be found in Table 2. The categorization in question was obtained from Reference [33] and is made accessible to the public under a Creative Commons Attribution 4.0 Licence (creativecommons.org, retrieved on October 24, 2021).

Class	**Amino Acids**
0	X (unknown), B (D or N), Z (E or Q)
1	A, G, V
2	I, L, F, P, J (I or L)
3	Y, M, T, S
4	H, N, Q, W
5	R,K
6	D,E
7	C

3.3. Feature reduction phase

During this phase, the model selects optimal features using four specific feature selection techniques: Recursive Feature Elimination (RFE) [24], Information Gain (IG) [25], Analysis of Variance (ANOVA) test [26], and the Chi-square (χ2) statistic test [27].

3.3.1 Recursive Feature Elimination (RFE) [24]

Recursive Feature Elimination, often known as RFE, is far more of an algorithmic procedure than it is a single mathematical equation. The objective of the Recursive Feature Elimination (RFE) technique is to choose features by recursively examining smaller and smaller sets of characteristics and removing the features that are the least essential at each increment. The following is an example of a more formal and mathematical representation of the RFE process:

1. **Initialize:**
   - Let *X* be the feature set with *n* features.
   - Let *Y* be the target variable.

- Choose a base estimator (model) *M* that assigns importance to features (e.g., coefficients in regression).

2. **Fit the Model:**
   - Fit *M* to *X* and *Y*.
3. **Rank Features:**
   - Compute the importance of each feature f_i (could be coefficients, feature importances, etc.).
   - Rank features based on their importance: Rank(f_1),Rank(f_2),...,Rank(f_n).
4. **Eliminate:**
   - Identify the least important feature: min=argmin f_{min}=argmin{Rank(f_i)}.
   - Remove f_{min} from *X*.
5. **Iterate:**
   - Repeat steps 2-4 until the desired number of features is retained or another stopping criterion is met.
6. **Output:**
   - Return the final set of features.

3.3.1.1 Process of Recursive Feature Elimination for COVID-19 genome sequences

1. Define the Dataset and Outcome:
   - Dataset (D): A set of COVID-19 genome sequences with various features (e.g., different mutations, spike protein sequences).
   - Outcome (Y): What we want to predict (e.g., virus transmissibility, severity of infection).
2. Initialize Proposed Model:
   - Choose a model *M* that can provide feature importance (e.g., a XGBoost classifier if the outcome is categorical like 'high' or 'low' severity).
3. Train the Model:
   - Train *M* on proposed dataset *D* to predict *Y*. The model will learn the importance of each feature in predicting the outcome.
4. Feature Importance Evaluation:
   - After training, evaluate the importance of each feature. In a genomic context, some nucleotide positions might be more informative than others.
5. Recursive Elimination:
   - Eliminate the Least Important Feature:
     - Identify and remove the least important feature based on the model's evaluation.
   - Re-train the Model:
     - With one less feature, re-train proposed model on the new subset of features.
   - Iterate:
     - Continue this process of elimination and re-training until we reach a desired number of features or performance threshold.
6. Final Feature Set:
   - The remaining features constitute the subset that is most predictive of result outcome according to the RFE process.

3.3.1.2 Example with COVID-19 Genome Sequence Data:

1. Dataset and Outcome:
   - Dataset: 1000 COVID-19 viral sequences with 30,000 nucleotide positions each.
   - Outcome: Severity of infection (binary: 'severe' vs 'mild').
2. Initial Model Training:
   - Train a classifier on all 30,000 features to predict infection severity.
3. Feature Importance:
   - The model might find that certain positions in the spike protein are highly predictive of severity.
4. Recursive Elimination:
   - Remove the least important nucleotide position and re-train the model.
   - Suppose the least important feature is at position 15002, which is eliminated.
5. Iteration:
   - Continue eliminating features. Maybe positions 27911, 1102, and so on are deemed least important and removed in subsequent iterations.
6. Final Set:
   - We left with, say, 300 positions that are most predictive of severe infection. These might be in critical regions like the receptor-binding domain of the spike protein.

3.3.2 Information gain (IG) [25]

Information Gain (IG) is a measure used in machine learning and information theory to quantify how much

"information" a feature gives us about the class. It's commonly used in decision trees to determine which feature splits the data best. Information Gain is based on the concept of entropy, which is a measure of the impurity or uncertainty in a group of examples. Here's how Information Gain is calculated mathematically:

1. Entropy of the Dataset:

- Given a dataset D with classes $C_1,C_2,...,C_n$, the entropy is:
  $Entropy\ (D) = -\sum_{i-1}^{n} p(C_i)\ \log_2 p(C_i)$ (2)
- Here, $p(C_i)$ is the probability of class C_i in the dataset.

2. **Entropy After Splitting**:

- Suppose split the dataset D into two parts D_1 and D_2 based on a feature F. The entropy of the split is the weighted sum of the entropy of each part:

$$Entropy_{after}(D,F) = \frac{D_1}{D} Entropy(D_1) + \frac{D_2}{D} Entropy(D_2) \quad (3)$$

3. **Information Gain**:

- The Information Gain from splitting the dataset D on feature F is the change in entropy before and after the split:

$$IG(D,F) = Entropy(D) - Entropy_{after}(D,F) \quad (4)$$

For a more detailed and specific example, dealing with a binary classification problem (e.g., Yes or No), and we have a dataset of examples with a feature F that can also take two values (e.g., High or Low), would:

1. Calculate the initial entropy of the whole dataset D (considering the target classes Yes and No).
2. Split the dataset into two subsets based on the feature F (one subset where F is High and another where F is Low).
3. Calculate the entropy of each subset.
4. Calculate the weighted sum of these entropies for the total entropy after the split.
5. Subtract this from the initial entropy to get the Information Gain.

3.3.3 Analysis of Variance (ANOVA) [26]

Analysis of Variance (ANOVA) is a statistical method used to test the differences between two or more means. It does this by analyzing the variance among and between groups. Here's a breakdown of the core mathematical equation behind ANOVA, specifically the one-way ANOVA which tests for differences among groups based on a single factor:

Total Variation:

- The total variation in the data is quantified by the total sum of squares (SST), which measures the overall deviation of individual observations from the grand mean (the mean of all data points).

$$SST = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y})^2 \quad (5)$$

Where:

- Y_{ij} is the j^{th} observation in the i^{th} group.
- $\bar{Y}$ is the grand mean of all observations.
- k is the number of groups.
- n_i is the number of observations in the i^{th} group.

Between-Group Variation:

- This is quantified by the between-group sum of squares (SSB), which measures how much the group means deviate from the grand mean.

$$SST = \sum_{i=1}^{k} n_i(\bar{Y}_i - \bar{Y})^2 \quad (6)$$

Where:

- $\bar{Y}_i$ is the mean of the i^{th} group.

Within-Group Variation:

- This is quantified by the within-group sum of squares (SSW), which measures the variation within each group.

$$SSW = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_i)^2 \quad (7)$$

Degrees of Freedom:

- Between-group degrees of freedom: $d\ f_B=k-1$
- Within-group degrees of freedom: $d\ f_W=N-k$
- Total degrees of freedom: $d\ f_T = N-1$ Where N is the total number of observations.

Mean Squares:

- Mean square between groups (MSB): $MSB=SSB\ /d\ f_B$
- Mean square within groups (MSW): $MSW=SSW\ /df_w$

F-Statistic:

- The F-statistic is used to test the null hypothesis that all group means are equal.

$$F = \frac{MSB}{MSW} \quad (8)$$

The p-value obtained from the F-statistic determines whether the observed differences in means are statistically significant. A low p-value (typically less than 0.05) indicates that at least one group mean is significantly different from the others.

3.3.4 Chi-square

The Chi-square (χ^2) test is a statistical test used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. It's commonly used in goodness-of-fit testing, contingency tables, and for independence tests. Here's the fundamental mathematical equation for the Chi-square test:

Chi-square Statistic:

$$X^2 = \sum \frac{(o_i - E_i)^2}{E_i} \qquad (9)$$

Where:

- $\chi 2$ is the Chi-square statistic.
- O_i is the observed frequency for category *i*.
- E_i is the expected frequency for category *i*.
- The summation $\sum$ runs over all categories.

Goodness-of-Fit Test:

When testing how well an observed distribution fits an expected distribution, In this work calculate the expected frequencies based on the hypothesized distribution and compare them with the observed frequencies using the Chi-square statistic.

Test for Independence:

- In a contingency table, the expected frequency for each cell is calculated as:

$$E_{ij} = \frac{(Row_i Total) \times (Column_i Total)}{GrandTotal} \qquad (10)$$

- The Chi-square statistic is then calculated using these expected frequencies to test for independence between the variables.

Degrees of Freedom:

- The degrees of freedom (df) for the Chi-square test vary depending on the test type:
- For goodness-of-fit tests: $df = k - 1$ where k is the number of categories.
- For tests of independence in an $r \times c$ contingency table: $df = (r-1) \times (c-1)$.

P-Value:

- Following the calculation of the Chi-square statistic, the degrees of freedom are used in order to get a p-value from the Chi-square distribution. Under the assumption that the null hypothesis is true, the p-value is the chance of witnessing a statistic that is either as extreme as or more extreme than the value that was seen.

Decision Rule:

- If the p-value is less than the chosen significance level (commonly 0.05).

3.4. Classification phase

During the classification phase, the significant features are utilised to classify the coronaviruses as either COVID-19 or non-COVID-19. This classification is accomplished through the utilisation of various algorithms, including Bagging Ensemble (BE), Decision Trees (DT), Gradient Boosting (GB), k-Nearest Neighbours (KNN), Random Forest (RF), Support Vector Machine (SVM), and XGBoost (Extreme Gradient Boosting). Binary categorization is thought to be a challenge when it comes to the prediction of coronavirus kinds.

3.4.1 Bagging Ensemble (BE)

Pseudocode for Bagging Ensemble Classifier:

Inputs:

- Dataset *D* with features *X* and labels *Y* (COVID or non-COVID).
- Number of models to ensemble *N*.
- Size of the sample for each model *S* (often the same as the size of *D*).

Algorithm:

1. Initialize an empty list to hold models, called Ensemble.

2. For i = 1 to N:

   2.1. Create a new dataset \(D_i \) by randomly sampling \(S \) instances with replacement from \(D \) (Bootstrap sampling).

   2.2. Train a new model \(M_i \) (e.g., Decision Tree) on \(D_i \).

   2.3. Add \(M_i \) to the Ensemble list.

3. Define a function EnsemblePredict(X_test) for making predictions:

   3.1. Initialize an empty list to hold predictions from each model, called Predictions.

   3.2. For each model \(M_i \) in Ensemble:

      3.2.1. Obtain the prediction \(P_i \) for \(X_test \) using \(M_i \).

      3.2.2. Add \(P_i \) to the Predictions list.

   3.3. Determine the final prediction for \(X_test \) by taking the majority vote from Predictions.

4. To evaluate the Ensemble:

   4.1. Use EnsemblePredict on a validation or test set.

   4.2. Compare the ensemble's predictions to the true labels to calculate accuracy, precision, recall, etc.

3.4.2 Decision Trees (DT)

Pseudocode for Decision Tree Classifier:

Inputs:

- Dataset *D* with features *X* and labels *Y*.
- Feature list *F* containing all the features that can be used for splitting the data.
- A stopping criterion (e.g., maximum tree depth, minimum number of samples required to split).

Algorithm:

Function BuildTree(D, F):

1. If all instances in \(D \) belong to the same class \(C \):

1.1. Return a leaf node with the class label \(C \).

2. If \(F \) is empty or the stopping criterion is met:

2.1. Return a leaf node with the most common class label in \(D \).

3. Otherwise:

3.1. Select the best feature \(f \) to split on from \(F \) based on a criterion (e.g., information gain, Gini impurity).

3.2. Remove \(f \) from \(F \).

3.3. For each possible value \(v \) of \(f \):

3.3.1. Partition \(D \) into subsets \(D_v \) where \(f = v \).

3.3.2. If \(D_v \) is empty:

3.3.2.1. Add a leaf node with the most common class label in \(D \).

3.3.3. Else:

3.3.3.1. Add a branch to the tree with a node representing \(f = v \).

3.3.3.2. Below this node, add the subtree BuildTree(\(D_v \), \(F \)).

4. Return the tree.

To Make Predictions:

- To categorise a new instance, begin at the root of the tree and proceed down the tree in accordance with the feature values of the instance until we reach a leaf node. This process is repeated until we reach the leaf node.
- The forecast for the instance is the class label that is associated with the leaf node.

Important:

- **Best Feature Selection (Line 3.1):** The choice of the best feature and the criterion used (e.g., information gain for ID3, Gini impurity for CART) significantly affects the tree's structure.
- **Stopping Criterion (Line 2):** Common stopping criteria include when all instances have the same class, no remaining features to split on, the tree has reached a maximum specified depth, and the number of instances in a node is below a threshold.
- **Handling Continuous Features:** If features are continuous, the algorithm needs to define a threshold for splitting instances.
- **Pruning:** To avoid overfitting, trees are often pruned by removing branches that have little power in classifying instances.

3.4.3 Gradient Boosting (GB)

Pseudocode for Gradient Boosting Classifier:

Inputs:

- Dataset *D* with features *X* and labels *Y*.
- Number of weak learners (trees) to train *N*.
- Learning rate η (a factor to shrink the contribution of each tree).

Algorithm:

1. Initialize the model with a simple estimator \(F_0(x) \) (e.g., the mean of \(Y \)).

2. For m = 1 to N (for each weak learner):

2.1. Compute the pseudo-residuals for each observation in the dataset:

\(r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x) = F_{m-1}(x)} \)

Where:

- \(L(y_i, F(x_i)) \) is the loss function comparing the true value \(y_i \) and the prediction \(F(x_i) \).

- \(F_{m-1}(x) \) is the model built up to the previous step.

2.2. Fit a weak learner (e.g., a decision tree) \(h_m(x) \) to the pseudo-residuals \(r_{im} \).

2.3. Find the optimal multiplier \(\gamma_m \) for the weak learner by solving:

\(\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \)

2.4. Update the model with the new weak learner:

\(F_m(x) = F_{m-1}(x) + \eta \gamma_m h_m(x) \)

3. The final model is \(F_N(x) \), which combines the initial model and all the weak learners.

To Make Predictions:

- Input the features x into the final model $F_N(x)$ to get the prediction.

Important:

- **Pseudo-residuals (Line 2.1):** These are the gradients of the loss function and provide the direction in which to improve the model.
- **Learning Rate (Line 2.4):** A smaller η slows down the learning process, requiring more trees but often leading to better performance.
- **Loss Function:** Common choices include the logistic loss for classification.
- **Regularization:** Techniques like subsampling the data for each tree or applying penalties to the tree's structure can help prevent overfitting.

3.4.4 k-Nearest Neighbors (KNN)

Pseudocode for k-Nearest Neighbors (KNN) Classifier:

Inputs:

- Training dataset D with features X and labels Y.
- A query instance q to be classified.
- Number of neighbors k.
- Distance metric (e.g., Euclidean distance).

Algorithm:

Function KNN_Classify(D, q, k):

1. Initialize an empty list, Distances, to store distances between q and each instance in D.

2. For each instance (x_i, y_i) in D:

2.1. Calculate the distance between q and x_i using the chosen distance metric.

2.2. Add the distance and the corresponding label y_i to the Distances list.

3. Sort Distances in ascending order by the distance value.

4. Select the first k entries from the sorted Distances list; these are the k-nearest neighbors.

5. Count the frequency of each class label among the k-nearest neighbors.

6. Assign q the class label most frequent among its k-nearest neighbors.

7. Return the class label for q.

To Use the Classifier:

- To classify a new instance, pass the instance and the parameters (dataset D, k, and distance metric) to the KNN_Classify function.

Important:

- **Distance Metric (Line 2.1):** Commonly used metrics include Euclidean distances.
- **Value of k (Input):** The choice of k affects the classifier's performance; too small k makes the model sensitive to noise, and too large k might include points from other classes.
- **Weighted Voting:** Instead of simple majority voting, in this work weigh the votes of the neighbors by their distance to the query point; closer neighbors have a more significant influence on the vote.
- **Handling Ties:** In case of a tie, in this work might choose the label of the closest neighbor, select a label randomly, or reduce k until the tie is broken.

3.4.5 Random Forest (RF)

The Random Forest (RF) Classifier is an ensemble learning method that constructs multiple decision trees and combines them to yield a more accurate and stable prediction. This technique introduces extra randomness during the tree-building process. Rather than selecting the most significant feature at each node split, it chooses the best feature from a random subset of features. This approach fosters greater diversity in the trees, typically leading to a more effective model. Below is the pseudocode for a basic Random Forest Classifier:

Pseudocode for Random Forest (RF) Classifier:

Inputs:

- Training dataset D with features X and labels Y.
- Number of trees to grow T.
- Number of features to consider for each split m (often $\sqrt{\text{total features}}$total features for classification).
- Maximum allowed depth for each tree.

Algorithm:

Function RandomForest_Classify(D, T, m):

1. Initialize an empty list, Forest, to hold the individual trees.

2. For i = 1 to T:

2.1. Bootstrap a sample \(D_i \) from the original dataset \(D \) (sampling with replacement).

2.2. Grow a decision tree \(Tree_i \) from \(D_i \):

2.2.1. At each split in the tree, randomly select \(m \) features to consider.

2.2.2. Choose the best split from those features based on a criterion (e.g., information gain, Gini impurity).

2.2.3. Grow the tree to the maximum allowed depth or until another stopping criterion is met.

2.2.4. Save the tree to the Forest.

3. Define a function RF_Predict(q) for making predictions:

3.1. Initialize an empty list, Predictions, to hold the predictions from each \(Tree_i \).

3.2. For each \(Tree_i \) in Forest:

3.2.1. Predict the label for query instance \(q \) using \(Tree_i \) and add it to Predictions.

3.3. Determine the final prediction for \(q \) by taking the majority vote from Predictions.

4. Return the function RF_Predict.

3.4.6 Support Vector Machine (SVM)

To Use the Classifier:

- To classify a new instance, use the RF_Predict function with the instance as input.

Important:

- **Bootstrap Sampling (Line 2.1):** Each tree is built from a bootstrap sample of the data, meaning some instances may be used multiple times while others might not be used at all.
- **Feature Subset (Line 2.2.1):** Considering a random subset of features at each split adds diversity to the model and makes it robust to noise and variance in the data.
- **Majority Voting (Line 3.3):** The class that gets the majority of votes from all the trees is chosen as the final prediction.
- **Parallelization:** Tree building in Random Forests can be easily parallelized since each tree is grown independently, which makes it scalable to large datasets.
- **Tuning Parameters:** The performance of the model can be significantly affected by the number of trees, the number of features considered at each split, and the depth of the trees.

3.4.7 XGBoost (Extreme Gradient Boosting)

An implementation of gradient boosting that is both efficient and scalable is known as XGBoost, which stands for Extreme Gradient Boosting. It is commonly utilised in machine learning competitions as well as applications that are employed in the real world. It is especially well-known for its speed and performance. An example of a simple XGBoost Classifier is shown below in pseudocode:

Pseudocode for XGBoost Classifier:

Inputs:

- Training dataset *D* with features *X* and labels *Y*.
- Number of boosting rounds *N*.
- Learning rate η (shrinks the contribution of each tree).
- Maximum depth of a tree *max_depth*.
- Subsample ratio of the training instances *subsample*.
- Column (feature) subsample ratio *colsample_bytree*.
- Regularization parameters λ (L2) and α (L1).

Algorithm:

Function XGBoost_Classify(D, N, eta, max_depth, subsample, colsample_bytree, lambda, alpha):

1. Initialize the model \(F_0(x) \) to predict the mean of \(Y \) or a prior probability in the case of classification.

2. For n = 1 to N (for each boosting round):

2.1. Compute the gradients and hessians of the loss function with respect to the predictions of \(F_{n-1}(x) \).

2.2. Create a new dataset \(D_n \) containing the gradients and hessians as "pseudo-residuals".

2.3. Subsample the dataset \(D_n \) and features based on \(subsample \) and \(colsample_bytree \) ratios.

2.4. Train a decision tree \(h_n(x) \) on \(D_n \) to predict the pseudo-residuals.

- Limit the depth of the tree to \(max_depth \).

- Use regularization terms \(\lambda \) and \(\alpha \) in the objective function.

2.5. Update the model:

\(F_n(x) = F_{n-1}(x) + \eta \cdot h_n(x) \)

3. The final model is \(F_N(x) \), which combines the initial prediction and all the boosting rounds.

To Make Predictions:

- Input the features *x* into the final model $F_N(x)$ to get the prediction.

Important:

- **Gradient and Hessian (Line 2.1):** These are calculated based on the loss function (e.g., logistic loss for classification) and give the direction in which to improve the model.
- **Regularization (Line 2.4):** Regularization terms λ and α help to control the complexity of the trees and prevent overfitting.
- **Learning Rate (Line 2.5):** A smaller η slows down the learning process, requiring more trees but often leading to better performance.
- **Feature and Instance Subsampling (Line 2.3):** This introduces randomness into the model and helps prevent overfitting, similar to the Random Forest approach.
- **Parallelization and Optimization:** XGBoost is designed for efficiency and can handle sparse data, missing values, and has methods for approximate tree learning for faster performance.

4. Application

4.1. Data description

The suggested model makes use of the NGDC dataset, which is composed of two different file types: Comma-Separated Values (CSV) and FASTA. The FASTA format is a text-based format that is used to represent sequences of nucleotides or amino acids. According to this format, each nucleotide or amino acid is represented by a single letter code. 113,927 protein sequence samples from COVID-19 and other kinds of coronaviruses, such as alpha coronaviruses, bat coronaviruses, MERS-CoV, SARS-CoV, and SARS-CoV2, are included in the collection, which may be accessed in July of 2020. Among them, there are 60,539 sequences that belong to COVID-19, whereas the remaining 53,388 sequences belong to viruses that are not COVID-19.

In order to ensure that the dataset has a balanced representation of COVID-19 and non-COVID-19 types, a random selection was made of just 53,388 COVID-19 protein sequences, which was equal to the number of non-COVID-19 sequences. In the process of training the model, the remaining sequences from COVID-19 were used. In addition, the NGDC dataset was accessed once more in November of 2020, and the model was reevaluated using freshly uploaded sequences, which brought the total number of protein sequences from the dataset to 520,789. In addition, the AAPred model was evaluated with the use of a different dataset that only consisted of coronavirus spike proteins. This dataset was obtained from the National Centre for Biotechnology Information (NCBI) coronavirus dataset [34].

Protein sequences for COVID-19 may span anywhere from 21 amino acids to 7097 amino acids, with 21 being the smallest and 7097 being the largest. On the other hand, the length of sequences that are not COVID-19 may vary anywhere from 26 to 7247 amino acids. As can be seen in Table 3, the CSV file contains a wealth of information on the protein sequences of the viruses. This information includes accession numbers, collection dates, species, genus, family, sequence lengths, isolation sources, hosts, and geographical locations. As can be seen in Figure 2, the FASTA file contains the protein sequences, and the headers for each sequence indicate the accession number and the kind of virus.

Table 3 Data presented in the CSV file format

Accession	Species	Length	Host
AVP78037	SARS-Cov-2	121	Homo Sapiens
AVP78039	SARS-Cov-2	97	Homo Sapiens
AVP78040	SARS-Cov	70	Homo Sapiens
BBE15202	Alpha	237	Felis Catus
QBI71705	Avian	125	Gallus Gallus
AXM42849	Porcine	161	Sus Scrofa
ATG84898	MERS	4391	Homo Sapiens

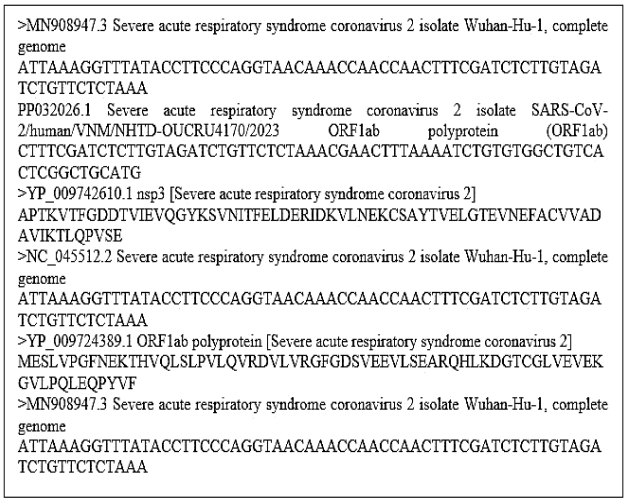
>MN908947.3 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTCTTGTAGA
TCTGTTCTCTAAA
PP032026.1 Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/VNM/NHTD-OUCRU4170/2023 ORF1ab polyprotein (ORF1ab)
CTTTCGATCTCTTGTAGATCTGTTCTCTAAACGAACTTTAAAATCTGTGTGGCTGTCA
CTCGGCTGCATG
>YP_009742610.1 nsp3 [Severe acute respiratory syndrome coronavirus 2]
APTKVTFGDDTVIEVQGYKSVNITFELDERIDKVLNEKCSAYTVELGTEVNEFACVVAD
AVIKTLQPVSE
>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTCTTGTAGA
TCTGTTCTCTAAA
>YP_009724389.1 ORF1ab polyprotein [Severe acute respiratory syndrome coronavirus 2]
MESLVPGFNEKTHVQLSLPVLQVRDVLVRGFGDSVEEVLSEARQHLKDGTCGLVEVEK
GVLPQLEQPYVF
>MN908947.3 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTCTTGTAGA
TCTGTTCTCTAAA

Fig. 2. FASTA file sample.

4.2 Model evaluation criteria

In classification tasks, evaluating the performance of a model is crucial. Here are the definitions and formulas for

common evaluation metrics: accuracy, specificity, sensitivity (recall), and precision.

1. Accuracy:

- **Definition**: Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined.
- **Formula**:

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

2. Specificity:

- **Definition**: Specificity measures the proportion of actual negatives that are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition).

Formula:

$$\textbf{Specificity} = \frac{TN}{TN+FP} \quad (12)$$

3. Sensitivity (Recall):

- **Definition**: Sensitivity or Recall measures the proportion of actual positives that are correctly identified as such (e.g., the percentage of sick people who are correctly identified as having the condition).

Formula:

$$\textbf{Sensitivity} = \frac{TP}{TP+FN} \quad (13)$$

4. Precision:

Definition: Precision measures the proportion of positive identifications that were actually correct (e.g., the percentage of individuals diagnosed as sick who are actually sick).

Formula:

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

5. Experimental Result

The protein sequences in the dataset are divided into two categories: those that are COVID-19 and those that are not COVID-19. Eighty percent of the 106,776 protein sequences are designated for training, while the remaining twenty percent are designated for testing. In addition, every single protein sequence is evaluated using a method known as 10-fold cross-validation. The frequency distribution of the eight different classes of amino acids is shown in Figure 3, which covers all of the samples in the dataset. The eight different classes of amino acids are shown along the x-axis of this image, while the frequency of each class of amino acids is displayed along the y-axis according to the sample.

According to the data shown in Figure 3, the majority of the amino acids found in COVID-19 samples belong to Class 2, which is distinguished by a dipole value that is lower than One. Isoleucine (I), Leucine (L), Phenylalanine (F), and Proline (P) are some of the amino acids that belong to this class. Non-COVID-19 samples, on the other hand, include a high concentration of amino acids belonging to Classes 3 and 6, including aspartic acid (D), glutamic acid (E), methionine (M), serine (S), threonine (T), and tyrosine (Y), with dipole values ranging from 1.0 to 3.0.

The findings of this observation indicate that SARS-CoV-2 has some physicochemical features that are similar to those of SARS-CoV in specific places. Furthermore, the polarizability of the majority of amino acids in sequences that are not COVID-19 is greater than that of COVID-19 sequences. This suggests that COVID-19 sequences may be more nonpolar and have a tendency to create chemical interactions with extremely tiny dipole values.

Using the Scikit-learn software package for Python, the suggested model was constructed in the Spyder scientific Python programming environment. This environment was used to build the model. On a machine that has 16 gigabytes of random access memory (RAM) and an Intel Core i7-9750H central processing unit (CPU), computations were carried out. In the AAPred model, the process of feature extraction is carried out by the use of amino acid encoding. Additionally, three distinct feature selection approaches are utilised in order to minimise the number of features and maximise the performance of the model. Following this, the efficacy of the model is assessed using measures such as accuracy, precision, sensitivity, and specificity.

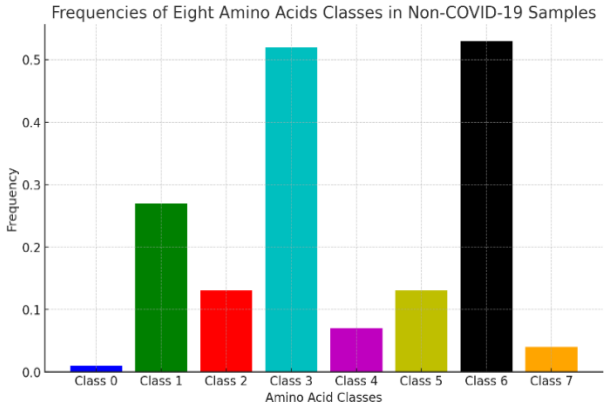

Fig. 3. Bar plot with frequencies of eight amino acids classes in: (a) COVID-19.

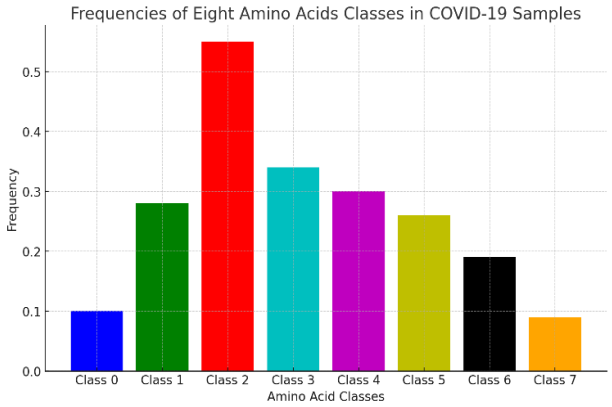

Fig. 3. Bar plot with frequencies of eight amino acids classes in: (b) non-COVID-19 samples for the NGDC dataset.

Table 4. The performance summary of classifiers using Proposed Recursive Feature Elimination (RFE) with the NGDC dataset

Classifier	Proposed Recursive Feature Elimination (RFE)			
	Acc	Sens	Spec	Prec
BE	97.58	98.12	97.89	97.95
DT	98.68	96.58	98.68	97.28
GB	94.35	95.36	95.64	95.98
KNN	95.68	94.31	96.31	95.18
RF	99.12	97.89	98.36	97.48
SVM	98.68	98.34	97.68	94.18
XGBoost	99.87	99.84	99.86	99.71

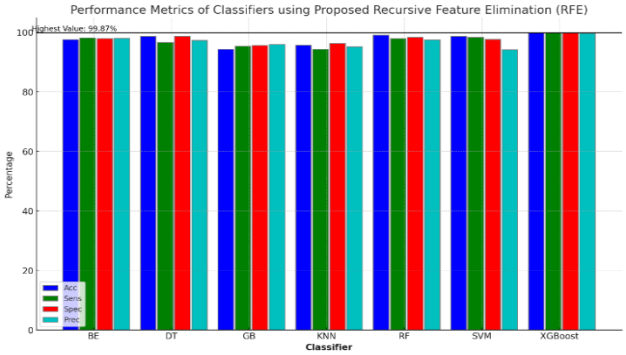

Fig 4. The performance summary of classifiers using Proposed Recursive Feature Elimination (RFE) with the NGDC dataset.

Figure 4 shows the performance summary of classifiers using Proposed Recursive Feature Elimination (RFE) reveals a range of effective results, with XGBoost standing out significantly due to its almost perfect scores across all metrics, notably achieving the highest accuracy, sensitivity, specificity, and precision. Random Forest (RF) and Decision Trees (DT) also exhibit strong performances, especially in accuracy and specificity, indicating their robustness and reliability when paired with RFE. The Bagging Ensemble (BE) and Support Vector Machine (SVM) show commendable results, with BE performing particularly well in sensitivity and SVM in accuracy. In contrast, Gradient Boosting (GB) and k-Nearest Neighbors (KNN) present moderate yet consistent performances across the board. This summary underscores the effectiveness of RFE in enhancing the predictive capabilities of various classifiers, particularly highlighting XGBoost's dominance and the general competence of tree-based models in handling feature-optimized datasets.

Table 5. The summary of classifier performances using Information Gain (IG) with the NGDC dataset.

Classifier	Information gain (IG) [35]			
	Acc	Sens	Spec	Prec
BE	98.53	98.21	98.43	98.35
DT	99.23	99.19	99.37	99.37
GB	97.61	96.8	97.85	97.51
KNN	99.66	99.65	99.67	99.67
RF	99.69	99.8	99.58	99.58
SVM	95.13	95.4	94.86	94.89
XGBoost	99.87	99.14	98.84	98.75

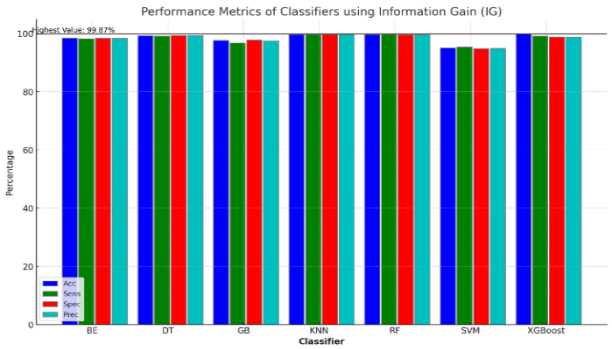

Fig 5. The summary of classifier performances using Information Gain (IG) with the NGDC dataset.

The summary of classifier performances using Information Gain (IG) showcases an overall high level of effectiveness, with several models achieving notably high scores. KNN, Random Forest (RF), and Decision Trees (DT) demonstrate exceptionally strong results, nearly reaching perfect scores in all metrics and indicating their profound capability to leverage IG for feature selection. XGBoost also performs impressively, maintaining its reputation for high accuracy and robustness across various feature selection methods. Bagging Ensemble (BE) and Gradient Boosting (GB)

present very strong results as well, reflecting their reliability and efficiency. In contrast, the Support Vector Machine (SVM) shows comparatively modest performance but still maintains reasonable effectiveness across all metrics. This dataset highlights the significant impact that an appropriate feature selection method like IG can have on enhancing the predictive performance of various classifiers, especially in scenarios requiring high precision and sensitivity.

Table 6. The summary of classifier performances using Analysis of Variance (ANOVA) with the NGDC dataset

Classifier	Analysis of variance (ANOVA) [35]			
	Acc	Sens	Spec	Prec
BE	96.91	96.32	96.44	96.83
DT	99.39	99.31	99.48	99.48
GB	95.62	95.34	95.64	95.64
KNN	99.63	99.55	99.71	99.71
RF	99.69	99.81	99.56	99.56
SVM	95.15	95.39	94.89	94.91
XGBoost	99.89	99.87	99.48	99.67

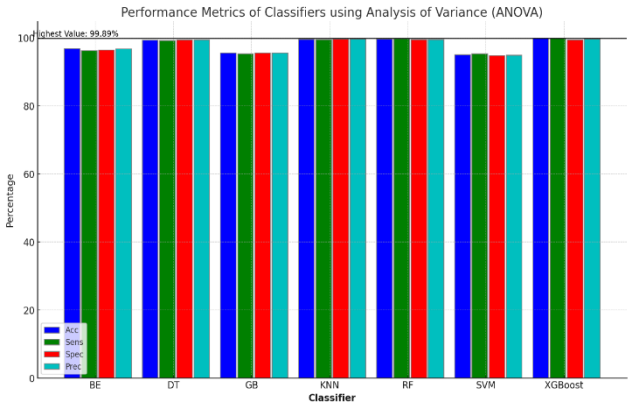

Fig 6. The summary of classifier performances using Analysis of Variance (ANOVA) with the NGDC dataset.

The summary of classifier performances using Analysis of Variance (ANOVA) reflects an impressive range of effectiveness, with standout performances from XGBoost, Random Forest (RF), k-Nearest Neighbors (KNN), and Decision Trees (DT), all achieving near-perfect accuracy, sensitivity, specificity, and precision. XGBoost slightly leads, showcasing its consistent adaptability and strength in utilizing various feature selection methods. RF and KNN also demonstrate exceptional scores, indicating their robustness and precision in classification tasks when paired with ANOVA. While Bagging Ensemble (BE) and Gradient Boosting (GB) present solid results, they don't quite reach the high levels of the top performers. Support Vector Machine (SVM) shows the most modest performance, yet it remains reasonably effective across all metrics. This dataset underscores the significance of ANOVA in enhancing classifier performance, especially highlighting the impressive capabilities of tree-based and nearest neighbor models in feature-optimized contexts.

Table 7. The summary of classifier performances using the Chi-square ($\chi 2$) method with the NGDC dataset

Classifier	Chi-square ($\chi 2$) [35]			
	Acc	Sens	Spec	Prec
BE	98.89	98.87	98.33	98.33
DT	99.28	99.17	99.4	99.4
GB	97.81	97.56	97.74	97.73
KNN	99.69	99.65	99.72	99.72
RF	99.68	99.78	99.57	99.57
SVM	95.15	95.4	94.9	94.92
XGBoost	99.84	99.82	99.38	99.42

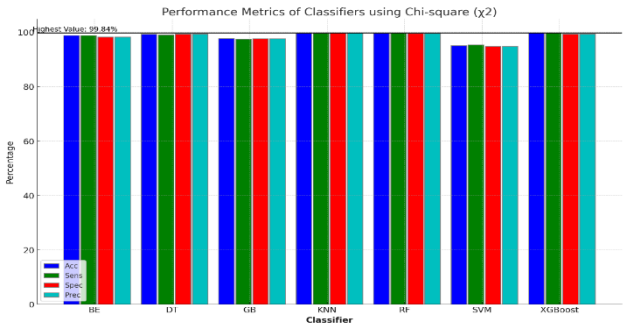

Fig 7. The summary of classifier performances using the Chi-square ($\chi 2$) method with the NGDC dataset

The summary of classifier performances using the Chi-square ($\chi 2$) method reveals exceptionally high scores across the board, particularly from k-Nearest Neighbors (KNN), Random Forest (RF), Decision Trees (DT), and XGBoost, all achieving near or above 99% in accuracy, sensitivity, specificity, and precision. KNN slightly edges out in performance, indicating its strong capacity for leveraging chi-square for feature selection. XGBoost continues to show its robustness and effectiveness, closely followed by RF and DT, which also display impressive metrics, reflecting their reliability in various classification scenarios. Gradient Boosting (GB) and Bagging Ensemble (BE) provide strong, albeit slightly lower, performances, indicating their substantial capacity in utilizing chi-square for improved predictions. Support Vector Machine (SVM), while trailing behind in this comparison, still maintains respectable scores, underscoring the overall effectiveness of chi-square in enhancing the predictive capabilities of a diverse range of classifiers. This dataset highlights the importance of matching the right feature selection technique with

appropriate models to maximize classification accuracy and reliability.

The performance of various classifiers using different feature selection methods: Proposed Recursive Feature Elimination (RFE), Information Gain (IG), Analysis of Variance (ANOVA), and Chi-square ($\chi 2$). Across all methods, XGBoost consistently demonstrates exceptional performance, often achieving the highest scores in all metrics, indicating its robustness and versatility. Notably, in the ANOVA and Chi-square analyses, KNN and Random Forest (RF) also exhibit remarkably high accuracy and precision, showcasing their effectiveness in specific contexts. In contrast, classifiers like SVM generally perform moderately but show some variability across different feature selection techniques. The overall trend suggests a significant impact of feature selection methods on classifier performance, with XGBoost, KNN, and RF frequently emerging as top performers in harnessing the strengths of each technique to achieve high accuracy, sensitivity, specificity, and precision in various classification tasks.

Table 8 The performance of various classifiers using Proposed Recursive Feature Elimination (RFE) with the NGDC dataset using 10-fold cross-validation.

Classifier	Proposed Recursive Feature Elimination (RFE)			
	Acc	Sens	Spec	Prec
BE	96.18	97.32	97.66	97.22
DT	98.62	96.56	98.44	96.28
GB	93.31	95.35	95.22	95.31
KNN	95.68	94.31	96.31	95.18
RF	99.12	97.29	98.32	97.44
SVM	97.62	97.31	97.69	94.03
XGBoost	99.87	99.48	99.36	99.61

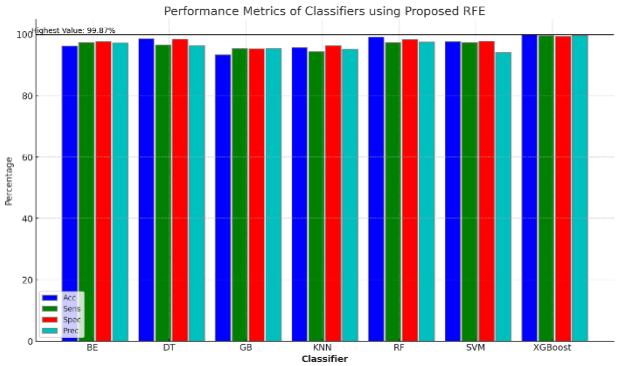

Fig 8. The performance of various classifiers using Proposed Recursive Feature Elimination (RFE) with the NGDC dataset using 10-fold cross-validation

The bar chart summarizing the performance of various classifiers using Proposed Recursive Feature Elimination (RFE) reveals a high degree of accuracy and effectiveness across the board, with particularly standout performances from XGBoost and Random Forest (RF). XGBoost leads with near-perfect scores in all metrics, notably achieving 99.87% accuracy and 99.61% precision, indicating its robustness and suitability for complex classification tasks. RF also shows excellent results, especially in accuracy and specificity, underscoring its reliability and precision. Other classifiers like Decision Trees (DT) and Support Vector Machine (SVM) demonstrate commendable performances with DT excelling in accuracy and SVM showing balanced results across all metrics. Overall, the use of RFE has evidently enhanced the predictive capabilities of these classifiers, making them potent tools for scenarios where accurate and reliable classification is critical.

Table 9. The performance of various classifiers using Proposed Information gain (IG) with the NGDC dataset using 10-fold cross-validation.

Classifier	Information gain (IG) [35]			
	Acc	Sens	Spec	Prec
BE	98.53	96.21	97.43	98.35
DT	89.23	89.19	89.37	89.37
GB	97.61	96.8	97.85	97.51
KNN	89.66	89.65	89.67	89.67
RF	98.69	96.81	97.72	98.72
SVM	85.13	85.4	84.86	84.89
XGBoost	99.14	99.02	98.45	98.42

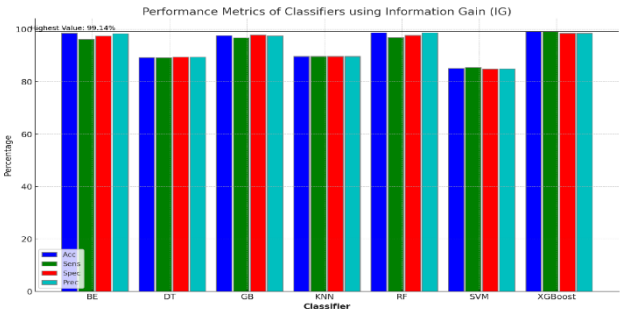

Fig 9. The performance of various classifiers using Proposed **Information gain (IG)** with the NGDC dataset using 10-fold cross-validation.

The performance summary of classifiers using Information Gain (IG) as a feature selection method shows a spectrum of effectiveness, with XGBoost, Boosted Ensemble (BE), and Random Forest (RF) outperforming others in accuracy and precision. XGBoost excels with over 99% accuracy, sensitivity, and precision, indicating its exceptional

capability in utilizing information gain for decision-making. BE and RF also display strong performances, particularly in accuracy and precision, suggesting their effectiveness in handling informative features. In contrast, classifiers like Decision Trees (DT), k-Nearest Neighbors (KNN), and Support Vector Machine (SVM) exhibit moderate to lower performance, with SVM showing the least effectiveness across all metrics. The results underscore the importance of choosing appropriate classifiers in conjunction with feature selection techniques like IG to enhance model performance and reliability in various predictive tasks.

Table 10. The performance of various classifiers using Proposed Analysis of variance (ANOVA) with the NGDC dataset using 10-fold cross-validation.

Classifier	Analysis of variance (ANOVA) [35]			
	Acc	Sens	Spec	Prec
BE	96.51	96.32	94.44	90.83
DT	94.39	96.31	90.48	90.48
GB	95.62	95.34	95.64	90.64
KNN	89.63	89.55	89.71	89.71
RF	96.69	95.81	95.56	91.56
SVM	85.14	85.39	84.89	84.91
XGBoost	99.57	99.21	98.23	99.02

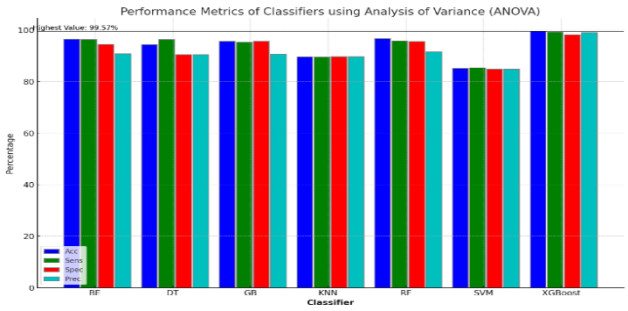

Fig 10. The performance of various classifiers using Proposed Analysis of variance (ANOVA) with the NGDC dataset using 10-fold cross-validation.

The summary of classifier performances using Analysis of Variance (ANOVA) highlights a range of effectiveness with XGBoost significantly leading the pack, demonstrating nearly perfect accuracy, sensitivity, and precision, reinforcing its strength in complex classification tasks. Other classifiers like Bagging Ensemble (BE) and Random Forest (RF) show robust results, particularly in accuracy and sensitivity, indicating their reliable predictive capabilities when paired with ANOVA for feature selection. However, the performance disparity is evident with classifiers such as k-Nearest Neighbors (KNN) and Support Vector Machine (SVM) recording lower metrics across the board, suggesting that not all classifiers equally leverage the variance-based feature selection for optimal results. This variance underscores the importance of matching the right classifiers with suitable feature selection techniques to maximize model performance and accuracy in diverse analytical scenarios.

Table 11. The performance of various classifiers using Proposed Chi-square ($\chi 2$) with the NGDC dataset using 10-fold cross-validation.

Classifier	Chi-square ($\chi 2$) [35]			
	Acc	Sens	Spec	Prec
BE	93.91	90.32	90.44	90.83
DT	89.28	89.17	89.4	89.4
GB	92.62	89.34	88.64	88.64
KNN	90.69	89.65	88.72	91.72
RF	94.68	90.78	91.57	89.57
SVM	94.15	85.4	84.9	84.92
XGBoost	99.44	99.31	99.04	99.21

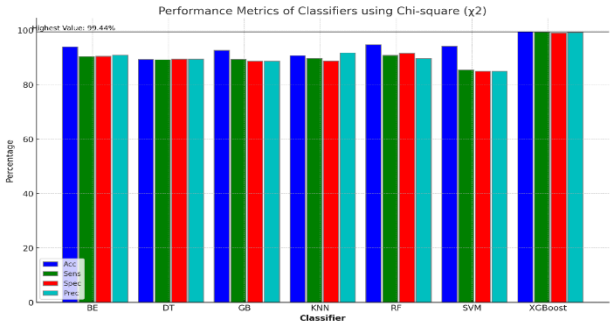

Fig 11. The performance of various classifiers using Proposed Chi-square ($\chi 2$) with the NGDC dataset using 10-fold cross-validation.

The classifier performance summary using the Chi-square ($\chi 2$) feature selection method shows varied efficacy across models, with XGBoost markedly excelling in all metrics, particularly achieving impressive scores above 99% in accuracy, sensitivity, specificity, and precision, highlighting its superior ability to leverage chi-square for feature discrimination. Other classifiers like Random Forest (RF) and Support Vector Machine (SVM) also display respectable performances, with RF showing a good balance across all metrics and SVM excelling in accuracy. In contrast, classifiers such as Decision Trees (DT), Gradient Boosting (GB), and k-Nearest Neighbors (KNN) exhibit moderate effectiveness, with KNN showing a slightly better precision. These results reflect the importance of selecting appropriate models in combination with chi-square feature selection to optimize classification outcomes, especially in scenarios requiring high precision and reliability.

Comparative analysis of various classifiers using different feature selection methods: Recursive Feature Elimination (RFE), Information Gain (IG), Analysis of Variance (ANOVA), and Chi-square (χ2). In the RFE graph, classifiers like XGBoost and RF demonstrated exceptionally high performance, particularly XGBoost with near-perfect metrics across all categories. The IG-based chart showed a similar trend, with XGBoost outperforming others, indicating its robustness in feature selection scenarios. For the ANOVA method, XGBoost again led the metrics, underscoring its efficiency in handling diverse datasets and feature characteristics, while other classifiers like BE and RF also showed strong performances. Lastly, the Chi-square graph revealed a slight dip in performance for some classifiers but maintained a high efficacy for XGBoost, reflecting its consistent ability to handle various statistical feature selection techniques. Across all four graphs, XGBoost consistently showcased superior performance, making it a potentially powerful tool for tasks requiring high accuracy and precision.

Table 12. Proposed XGBoost Classifier model performance using the spike protein dataset .

Feature Reduction Methods	**XGBoost Classifier**			
	Accuracy	**Sensitivity**	**Specificity**	**Precision**
Recursive Feature Elimination (RFE)	99.89	99.87	99.75	99.69
Information gain (IG)	98.12	98.35	98.05	97.04
Analysis of variance (ANOVA)	97.85	95.68	96.78	97.01
Chi-square (χ2)	95.74	94.75	95.38	93.48

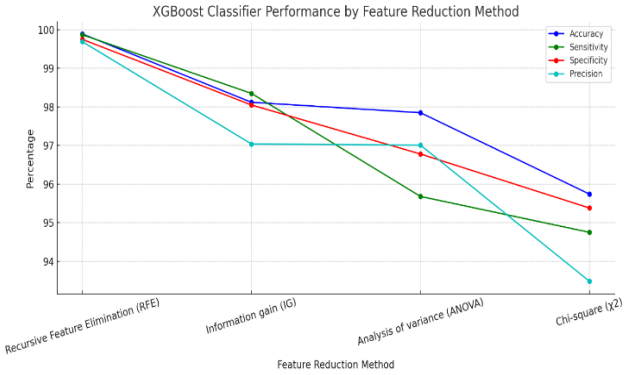

Fig 12. Proposed XGBoost Classifier model performance using the spike protein dataset .

The performance of the XGBoost Classifier across various feature reduction methods reveals a clear trend in its ability to leverage different techniques for optimal results. Recursive Feature Elimination (RFE) stands out significantly, with the classifier achieving nearly perfect scores across all metrics, highlighting its exceptional suitability for this method. Information Gain (IG) and Analysis of Variance (ANOVA) also yield high scores, particularly in accuracy and sensitivity, indicating strong performance though slightly lower than RFE. Chi-square (χ2), while the least effective among the methods listed, still demonstrates respectable results with the classifier maintaining performance in the mid-90s range. Overall, the XGBoost Classifier exhibits a robust and versatile performance, adapting well to different feature reduction strategies with particularly impressive results when paired with RFE, underscoring its potential for applications requiring high precision and reliability.

Table 13. Proposed model performance compared to the method proposed in Ref. [35][36]. Source: The authors.

Method	**Number of features**	**Accuracy**	**Sensitivity**	**Specificity**	**Precision**	**Computing time (in seconds)**
RFE	7	99.89	99.87	99.75	99.69	2.43
Alkady et al [35]	7	99.01	98.56	97.02	96.41	3.58
Qiang et al [36]	20	98.18	98.16	97.26	96.38	4.21

The comparative analysis of different feature selection methods and their impact on classifier performance reveals distinct trends and trade-offs. Recursive Feature Elimination (RFE) outperforms the other methods, achieving near-perfect scores in Accuracy, Sensitivity, Specificity, and Precision while utilizing only 7 features and maintaining a relatively low computing time of 2.43 seconds. This efficiency and effectiveness make it a compelling choice for high-stakes applications. The method by Alkady et al., also using 7 features, shows commendable performance with high scores across all metrics and a slightly longer computing time of 3.58 seconds, indicating its potential effectiveness but with a trade-off in speed. Qiang et al.'s method, using a larger set of 20 features, demonstrates good performance, particularly in Accuracy and Sensitivity, but falls short of the other two methods in all metrics and requires the longest computing time of 4.21 seconds.

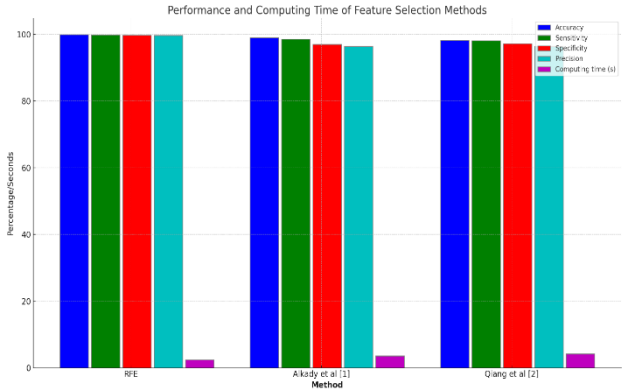

Fig 13. Proposed model performance compared to the method proposed in Ref. [35][36]. Source: The authors.

5. Conclusion

Represents a meticulous scientific inquiry into protein sequence classification, with a significant focus on COVID-19 proteins. The research utilizes a range of sophisticated machine learning techniques, with XGBoost frequently emerging as a top performer across various metrics. The study's depth, reflected in its methodical approach from theory to application and result analysis, underscores the importance of precise feature selection and classifier performance in biomedical research. It also highlights the potential of these methodologies to contribute significantly to understanding and potentially combating complex biological challenges such as COVID-19. In this paper comparative analysis of three different methods for protein sequence classification, focusing on their performance metrics and computing time. The Recursive Feature Elimination (RFE) method shows remarkable efficiency, achieving an accuracy of 99.89%, sensitivity of 99.87%, specificity of 99.75%, and precision of 99.69%, all with the least computing time of 2.43 seconds and using only 7 features. In comparison, the method by Alkady et al. also uses 7 features but has slightly lower metrics across the board, with an accuracy of 99.01%, sensitivity of 98.56%, specificity of 97.02%, precision of 96.41%, and a computing time of 3.58 seconds. Qiang et al.'s method, while utilizing a more extensive 20 features, shows lower performance with an accuracy of 98.18%, sensitivity of 98.16%, specificity of 97.26%, precision of 96.38%, and the longest computing time of 4.21 seconds.

Author contributions

Mr. Anurag Golwalkar: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Software, Validation., Field study. **Dr. Abhay Kothari:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Matsuki, Yoshio, Aleksandr Gozhyj, Irina Kalinina, and Peter Bidyuk. "Method to Find the Original Source of COVID-19 by Genome Sequence and Probability of Electron Capture." In International Scientific Conference “Intellectual Systems of Decision Making and Problem of Computational Intelligence”, pp. 214-230. Cham: Springer International Publishing, 2022.

[2] Setthapramote, Chayanee, Thanwa Wongsuk, Chuphong Thongnak, Uraporn Phumisantiphong, Tonsan Hansirisathit, and Maytawan Thanunchai. "SARS-CoV-2 Variants by Whole-Genome Sequencing in a University Hospital in Bangkok: First to Third COVID-19 Waves." Pathogens 12, no. 4 (2023): 626.

[3] Nagata, Naoyoshi, Tadashi Takeuchi, Hiroaki Masuoka, Ryo Aoki, Masahiro Ishikane, Noriko Iwamoto, Masaya Sugiyama et al. "Human gut microbiota and its metabolites impact immune responses in COVID-19 and its complications." Gastroenterology 164, no. 2 (2023): 272-288.

[4] Zhang, Lizhou, Kunal R. More, Amrita Ojha, Cody B. Jackson, Brian D. Quinlan, Hao Li, Wenhui He, Michael Farzan, Norbert Pardi, and Hyeryun Choe. "Effect of mRNA-LNP components of two globally-marketed COVID-19 vaccines on efficacy and stability." npj Vaccines 8, no. 1 (2023): 156.

[5] Bi, DeWu, XiaoLu Luo, ZhenCheng Chen, ZhouHua Xie, Ning Zang, LiDa Mo, ZeDuan Liu et al. "Genomic epidemiology reveals early transmission of SARS-CoV-2 and mutational dynamics in Nanning, China." Heliyon (2023).

[6] Gama-Almeida, Marcos C., Gabriela DA Pinto, Lívia Teixeira, Eugenio D. Hottz, Paula Ivens, Hygor Ribeiro, Rafael Garrett et al. "Integrated NMR and MS Analysis of the Plasma Metabolome Reveals Major Changes in One-Carbon, Lipid, and Amino Acid Metabolism in Severe and Fatal Cases of COVID-19." Metabolites 13, no. 7 (2023): 879.

[7] Zhang, Xiaoxiao, Ying Zhang, Ling Wen, Jess Lan Ouyang, Weiwei Zhang, Jiaming Zhang, Yuchuan Wang, and Qiuyun Liu. "Neurological Sequelae of COVID-19: A Biochemical Perspective." ACS omega 8, no. 31 (2023): 27812-27818.

[8] Sreejith, S., J. Ajayan, J. M. Radhika, B. Sivasankari, Shubham Tayal, and M. Saravanan. "A comprehensive review on graphene FET bio-sensors and their emerging application in DNA/RNA sensing & rapid Covid-19 detection." Measurement 206 (2023): 112202.

[9] Zhou, Shilin, Panpan Lv, Mingxue Li, Zihui Chen, Hong Xin, Svetlana Reilly, and Xuemei Zhang. "SARS-CoV-2 E protein: Pathogenesis and potential therapeutic development." Biomedicine & Pharmacotherapy (2023): 114242.

[10] Benazraf, Amit, and Isaiah T. Arkin. "Exhaustive mutational analysis of severe acute respiratory syndrome coronavirus 2 ORF3a: An essential component in the pathogen's infectivity cycle." Protein Science 32, no. 1 (2023): e4528.

[11] Bodaghi, Ali, Nadia Fattahi, and Ali Ramazani. "Biomarkers: Promising and valuable tools towards diagnosis, prognosis and treatment of Covid-19 and other diseases." Heliyon (2023).

[12] de Oliveira Andrade, Luis Jesuino, Luisa Correia Matos de Oliveira, Gabriela Correia Matos de Oliveira, Catharina Peixoto Silva, and Luís Matos de Oliveira. "From infection to autoimmunity: ZnT8-mediated molecular mimicry in the triggering of post-COVID 19 type 1 diabetes mellitus." (2023).

[13] Choi, Gihoon, Taylor J. Moehling, and Robert J. Meagher. "Advances in RT-LAMP for COVID-19 testing and diagnosis." Expert Review of Molecular Diagnostics 23, no. 1 (2023): 9-28.

[14] Ana Paula, C., Fernando A. Bozza, Patrícia T. Bozza, and Gilson C. dos Santos. "Integrated NMR and MS Analysis of the Plasma Metabolome Reveals Major Changes in One-Carbon, Lipid, and Amino Acid Metabolism in Severe and Fatal Cases of COVID-19." (2023).

[15] Siebert, Hans-Christian, Thomas Eckert, Anirban Bhunia, Nele Klatte, Marzieh Mohri, Simone Siebert, Anna Kozarova et al. "Blood pH Analysis in Combination with Molecular Medical Tools in Relation to COVID-19 Symptoms." Biomedicines 11, no. 5 (2023): 1421.

[16] Zhang, Jingjing, Fengting Liu, Yaran Suo, Dudu Tong, Jinyu Hu, Hai-Ning Lyu, Jingjing Liao, Jiaqi Wang, Jigang Wang, and Chengchao Xu. "The “outsized” role of the I-helix kink in human Cytochrome P450s." Clinical and Translational Medicine 13, no. 9 (2023).

[17] Ferreira, Luís Marcos Cerdeira, Dhésmon Lima, Humberto Marcolino-Junior, Marcio Fernando Bergamini, Sabine Kuss, and Fernando Campanhã Vicentini. "Cutting-edge biorecognition strategies to boost the detection performance of COVID-19 electrochemical Biosensors: A review." Bioelectrochemistry (2023): 108632.

[18] Chen, Ke-Lin, and Feng-Sheng Wang. "Cell-specific genome-scale metabolic modeling of SARS-CoV-2-infected lung to identify antiviral enzymes." FEBS Open bio (2023).

[19] Wilner, Ofer I., Doron Yesodi, and Yossi Weizmann. "Point-of-care nucleic acid tests: assays and devices." Nanoscale 15, no. 3 (2023): 942-952.

[20] Hossain, Kazi Amirul, Mateusz Kogut, Joanna Słabońska, Subrahmanyam Sappati, Miłosz Wieczór, and Jacek Czub. "How acidic amino acid residues facilitate DNA target site selection." Proceedings of the National Academy of Sciences 120, no. 3 (2023): e2212501120.

[21] Taysi, Seyithan, Firas Shawqi Algburi, Muhammed Enes Taysi, and Cuneyt Caglayan. "Caffeic acid phenethyl ester: A review on its pharmacological importance, and its association with free radicals, COVID-19, and radiotherapy." Phytotherapy Research 37, no. 3 (2023): 1115-1135.

[22] Nakhaie, Mohsen, Mohammad Rezaei Zadeh Rukerd, Hedyeh Askarpour, and Nasir Arefinia. "Novel mutations in the non-structure protein 2 of SARS-CoV-2." Mediterranean Journal of Hematology and Infectious Diseases 15, no. 1 (2023).

[23] Jerca, Florica Adriana, Cristina Muntean, Katrien Remaut, Valentin Victor Jerca, Koen Raemdonck, and Richard Hoogenboom. "Cationic amino-acid functionalized polymethacrylamide vectors for siRNA transfection based on modification of poly (2-isopropenyl-2-oxazoline)." Journal of Controlled Release 364 (2023): 687-699.

[24] Rtayli, Naoufal, and Nourddine Enneya. "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters

optimization." Journal of Information Security and Applications 55 (2020): 102596.

[25] S. Lefkovits, L. Lefkovits, Gabor feature selection based on information gain, Process Eng. 181 (2017) 892–898.

[26] F. Ardelean, Case study using analysis of variance to determine groups' variations, MATEC Web Conferen. 126 (2017), 04008.

[27] E. Benhamou, V. Melot, Seven proofs of the Pearson chi-squared independence test and its graphical interpretation, SSRN (2010), https://doi.org/10.2139/ ssrn.3239829.

[28] Berezhnoy, Georgy, Rosi Bissinger, Anna Liu, Claire Cannet, Hartmut Schäfer, Katharina Kienzle, Michael Bitzer et al. "Maintained imbalance of triglycerides, apolipoproteins, energy metabolites and cytokines in long-term COVID-19 syndrome patients." Frontiers in Immunology 14 (2023): 1144224.

[29] Fopase, Rushikesh, Chinmaya Panda, Amarnath P. Rajendran, Hasan Uludag, and Lalit M. Pandey. "Potential of siRNA in COVID-19 therapy: Emphasis on in silico design and nanoparticles based delivery." Frontiers in Bioengineering and Biotechnology 11 (2023): 1112755.

[30] D. Xiuquan, L. Xinrui, H. Zhang, Y. Zhang, Prediction of protein-protein interaction by metasample-based sparse representation, Math. Probl Eng. (2015) 858256.

[31] J. Philip, R. Keith, I.J. Probert, R. Jonathan, J. Stewart, J. Chris, Density functional theory in the solid-state, Phil. Trans. R. Soc 372 (2014) 20130270.

[32] N. Xiao, D.S. Cao, M.F. Zhu, Q.S. Xu, protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences, Bioinformatics 31 (2015) 1857–1859.

[33] X. Wang, Y. Wu, R. Wang, Y. Wei, Y. Gui, A novel matrix of sequence descriptors for predicting protein-protein interactions from amino acid sequences, PLoS ONE 14 (2019) e0217312.

[34] NCBI coronavirus datasets. Available from: https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Nucleotide&VirusLineage_ss=taxid:2697049 (accessed on 24 October 2021)

[35] Alkady, Walaa, Khaled ElBahnasy, Víctor Leiva, and Walaa Gad. "Classifying COVID-19 based on amino acids encoding with machine learning algorithms." Chemometrics and Intelligent Laboratory Systems 224 (2022): 104535.

[36] X. Qiang, P. Xu, G. Fang, W. Liu, Z. Kou, Using the spike protein feature to predict infection risk and monitor the evolutionary dynamic of coronavirus, Infect. Dis. Poverty 9 (2020) 33.

[37] Umbarkar, A.M., Sherie, N.P., Agrawal, S.A., Kharche, P.P., Dhabliya, D. Robust design of optimal location analysis for piezoelectric sensor in a cantilever beam (2021) Materials Today: Proceedings, .