

Article

Identifying COVID-19 Severity-Related SARS-CoV-2 Mutation Using a Machine Learning Method

Feiming Huang^{1,†}, Lei Chen^{2,†}, Wei Guo^{3,†}, Xianchao Zhou⁴, Kaiyan Feng⁵, Tao Huang^{6,7,*} 
and Yudong Cai^{1,*} 

¹ School of Life Sciences, Shanghai University, Shanghai 200444, China; hfm123@shu.edu.cn

² College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China; lchen@shmtu.edu.cn

³ Key Laboratory of Stem Cell Biology, Shanghai Jiao Tong University School of Medicine (SJTUSM) and Shanghai Institutes for Biological Sciences (SIBS), Chinese Academy of Sciences (CAS), Shanghai 200025, China; gw_1992@sjtu.edu.cn

⁴ Center for Single-Cell Omics, School of Public Health, Shanghai Jiao Tong University School of Medicine (SJTUSM), Shanghai 200025, China; zhouch1@shanghaitech.edu.cn

⁵ Department of Computer Science, Guangdong AIB Polytechnic College, Guangzhou 510060, China; kyfeng@gdaib.edu.cn

⁶ Bio-Med Big Data Center, CAS Key Laboratory of Computational Biology, Shanghai Institute of Nutrition and Health, University of Chinese Academy of Sciences, Chinese Academy of Sciences, Shanghai 200031, China

⁷ CAS Key Laboratory of Tissue Microenvironment and Tumor, Shanghai Institute of Nutrition and Health, University of Chinese Academy of Sciences, Chinese Academy of Sciences, Shanghai 200031, China

* Correspondence: huangtao@sibs.ac.cn (T.H.); caiyudong@staff.shu.edu.cn (Y.C.); Tel.: +86-21-54923269 (T.H.); +86-21-66136132 (Y.C.)

† These authors contributed equally to this work.



Citation: Huang, F.; Chen, L.; Guo, W.; Zhou, X.; Feng, K.; Huang, T.; Cai, Y. Identifying COVID-19 Severity-Related SARS-CoV-2 Mutation Using a Machine Learning Method. *Life* **2022**, *12*, 806. <https://doi.org/10.3390/life12060806>

Academic Editor: Lixin Wan

Received: 12 April 2022

Accepted: 25 May 2022

Published: 28 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: SARS-CoV-2 shows great evolutionary capacity through a high frequency of genomic variation during transmission. Evolved SARS-CoV-2 often demonstrates resistance to previous vaccines and can cause poor clinical status in patients. Mutations in the SARS-CoV-2 genome involve mutations in structural and nonstructural proteins, and some of these proteins such as spike proteins have been shown to be directly associated with the clinical status of patients with severe COVID-19 pneumonia. In this study, we collected genome-wide mutation information of virulent strains and the severity of COVID-19 pneumonia in patients varying depending on their clinical status. Important protein mutations and untranslated region mutations were extracted using machine learning methods. First, through Boruta and four ranking algorithms (least absolute shrinkage and selection operator, light gradient boosting machine, max-relevance and min-redundancy, and Monte Carlo feature selection), mutations that were highly correlated with the clinical status of the patients were screened out and sorted in four feature lists. Some mutations such as D614G and V1176F were shown to be associated with viral infectivity. Moreover, previously unreported mutations such as A320V of nsp14 and I164ILV of nsp14 were also identified, which suggests their potential roles. We then applied the incremental feature selection method to each feature list to construct efficient classifiers, which can be directly used to distinguish the clinical status of COVID-19 patients. Meanwhile, four sets of quantitative rules were set up, which can help us to more intuitively understand the role of each mutation in differentiating the clinical status of COVID-19 patients. Identified key mutations linked to virologic properties will help better understand the mechanisms of infection and will aid in the development of antiviral treatments.

Keywords: SARS-CoV-2; mutation; machine learning; feature selection; decision rules

1. Introduction

The outbreak of novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) initiated the pandemic of the global infectious disease called COVID-19 [1]. The rapid

spread of SARS-CoV-2 has affected more than 300 million people from over 200 countries as of January 2022, and it has severely impacted public health and social economy. Scientists have achieved some successes in vaccine development, and vaccination programs are helping to prevent the infection and spread of SARS-CoV-2. However, emerging variants of coronavirus with increased pathogenicity and infectivity are still a challenge to public health [2,3]. SARS-CoV-2, as a type of single-stranded positive RNA virus, displays a stronger ability of evolution through highly frequent genomic mutations during transmission [4]. The rapid evolution of SARS-CoV-2 has brought great challenges to epidemic prevention. Viral genomics has become a major focus of current research which aims to explore the association between genomic mutations and virological properties, such as virulence, pathogenicity, and immunogenicity. Identifying the key mutations related to virological properties will largely contribute to understanding the mechanism of infection, which is important for antiviral treatment development.

SARS-CoV-2 belongs to the *Betacoronavirus* genus of the *Coronaviridae* family, which is considered to originate from bats and can widely spread in bats, civets, and humans, and usually causes respiratory illness in humans [5]. The genome of SARS-Cov-2 consists of several nonstructural proteins (nsp), which play crucial roles in viral replication, and four main structural proteins including envelope, membrane, nucleocapsid, and spike, which are involved in the process by which SARS-CoV-2 enters human cells via interacting with the host receptor ACE2 [6,7]. During the current pandemic, five variants of concern have emerged including B.1.1.7 (Alpha), B.1.351 (Beta), P.1 (Gamma), B.1.617.2 (Delta), and B.1.1.529 (Omicron).

The B.1.1.7 variant was first discovered in the UK in September 2020, and then it spread rapidly worldwide. The S protein of the B.1.1.7 strain has eight mutations (i.e., Δ 69–70 deletion, Δ 144 deletion, N501Y, A570D, P681H, T716I, S982A, and D1118H), which may have changed the transmission and infection ability of SARS-CoV-2 [8]. Nine mutations (i.e., L18F, D80A, D215G, R246I, K417N, E484K, N501Y, D614G, and A701V) were found in the S protein of the B.1.351 variant [9]. The E484K site located in the receptor-binding domain (RBD) of the S protein directly contacts human ACE2 residue, which reduces the neutralization susceptibility to convalescent serum. The P.1 variant has 10 mutations in S protein (i.e., L18F, T20N, P26S, D138Y, R190S, H655Y, T1027I V1176, K417T, E484K, and N501Y), among which 3 mutations (K417T, E484K, and N501Y) show striking similarity to the RBD of the B.1.351 variant [10]. The B.1.617.2 variant was first discovered in India in October 2020 and caused a second fatal wave of COVID-19 infections in India in April 2021, which quickly spread and attracted global attention. Its transmissibility has increased by 97% compared with that of previous circulating strains [11]. Three key mutations in the S protein of the B.1.617.2 strain (i.e., L452R, T478K, and P681R) reduce the probability of reinfection and the effectiveness of the vaccine. The genome of the Omicron variant strain has about 50 mutations, including more than 30 mutations in the S protein, which overlaps with the B.1.617.2 and B.1.1.7 variants [12]. Previous studies have shown that K417N, E484K, and N501Y mutations in the S protein exhibit the enhanced immune evasion ability of Omicron [13]. The S protein is thought to be the most critical affecting factor of the virulence and pathogenicity of SRAS-CoV-2.

The implications of SARS-CoV-2 genomic mutations on viral pathogenicity have been summarized in some reviews [14]. However, the impact of viral mutations on COVID-19 severity demands further exploration. Previous articles have also mainly focused on mutations in the S protein and drawn conclusions through clinical research of limited cohort sizes. Identifying the potential mutational risk that indicates disease severity will be important for medical care and targeted treatment. To this end, we integrate the available data describing SARS-CoV-2 variants and patients' outcomes in this study and perform a novel computational analysis to investigate the influences of mutations in the SARS-CoV-2 genome on COVID-19 severity. Mutations in the whole genome, including structural proteins, nonstructural proteins, and untranslated regions (UTRs), were included. Several advanced computational methods were applied in the feature selection process. The

associations of genomic mutations of SARS-CoV-2 with viral pathogenicity and patients' clinical outcome were revealed. Many mutations in the S protein of remarkable relevance to clinical outcome were obtained, such as V1176F and D614G. Several genomic mutations in the nucleocapsid gene or nonstructural proteins were also closely related to the severity of COVID-19 upon infection, which may indicate altered viral pathogenicity. This study provides a novel computational approach to identify new potential mutations related to viral pathogenicity. The key mutations identified in this study provide the theoretical basis of pathogenesis, and these results will impact the development and application of antiviral vaccines.

2. Materials and Methods

2.1. Data and Preprocessing

We downloaded 1513 viral genomes from the GISAID database (<https://www.gisaid.org/> accessed on 6 December 2021), and the accession numbers were provided by Nagy et al. [15]. Each patient of viral origin was accompanied by clinical follow-up data. Nagy et al. divided 716 of these patients of viral genomic origin into the "mild" group and the 797 others into the "severe" group [15]. To investigate the differences between two groups, a binary classification problem was set up where patients in the "mild" group were termed as positive samples and those in the "severe" group were considered as negative samples. Most of the patients in the severe group were in the ICU, and some had even died. The majority of patients in the mild group were stable, and some were asymptomatic so could be treated at home. Mutation features contained two main types: one was protein mutations, and the other was nonprotein mutations. The protein mutations were extracted from the analysis program of the Coronavirus Antiviral Research Database (<https://covdb.stanford.edu/> accessed on 6 December 2021) [16]. This analysis program was fed viral sequences in FASTA format as the input, using "Wuhan-Hu-1(NC_045512.2)" as the reference sequence. The nonprotein mutations were extracted using the MUMmer3.0 [17] software, and substitutions found in at least 10 genomes were chosen for further investigation. After filtering, 8 nonprotein mutations with frequencies greater than 10 were found out of 197 nonprotein mutations. A total of 3641 protein mutation features and 8 nonprotein mutation features were obtained for subsequent analysis. The first half of the name format for the protein mutation feature was the protein name, the second half was the amino acid substitution, and the number was the amino acid substitution position. Similarly, for nonprotein mutations (SNP features), the numbers represented the positions of nucleotides on the whole genome. Several studies have shown that age is an important predictor of clinical status in patients with COVID-19. Therefore, we used age after discretization as one of the essential features in our study [18,19]. In the feature matrix, a virus genome is denoted as "1" if it contains a certain feature and "0" otherwise, and this feature matrix can be found in Table S1.

2.2. Boruta Feature Filtering

A large number of redundant features exist among all mutation features, and they are not very helpful in distinguishing patients' clinical status and can become noisy features for subsequent modeling. Boruta can filter the set of all features that have a correlation with the dependent variable, and the experimental results are very stable and scalable [20,21]. Here, we used Boruta for the initial filtering of mutation features, which is implemented as described below.

Boruta uses a random forest (RF) approach to extract features and disrupt the order of features to calculate the importance of features. It iteratively runs an RF on an extended version of the data for a given a dataset. The extended version of the data is a copy of the original data with horizontally connected shuffled features in each iteration. This method keeps features that are (1) more important than the best random sorting feature in each iteration and (2) superior to random factors in terms of performance (using binomial distribution). Boruta adopts the Z-Score as a measure of relevance because it considers changes in the average loss of accuracy among trees in the forest. The importance of

Boruta's approach is that it can help us gain a better understanding of the features that influence the dependent variable and, therefore, provide better and more efficient feature selection.

The Boruta program was retrieved from <https://github.com/scikit-learn-contrib/boruta.py> accessed on 14 September 2020 and was run with default parameters in our study.

2.3. Feature Ranking Algorithms

Among the features filtered by Boruta, each feature has a different importance for classification. In this study, we ranked the Boruta-filtered features using four feature ranking algorithms. Their brief descriptions are as follows.

2.3.1. Max-Relevance and Min-Redundancy

Identifying features that are significant for prediction is important, but obtaining a collection of features that are not redundant is more crucial to boost robustness. When two features are strongly dependent on each other, removing one of them has little effect on the representative class's discriminative power. As a result, the minimal redundancy requirement can be introduced to choose mutually exclusive features. The max-relevance and min-redundancy (mRMR) criteria combine the two restrictions mentioned above [22–24]. Initially, mRMR uses mutual information to calculate the correlation between independent variables and features, as well as between features. Ensuring maximum correlation between features and independent variables, as well as minimum redundancy between features, during the mRMR calculation process is important.

The mRMR program from <http://home.penglab.com/proj/mRMR/> accessed on 2 May 2018 was performed with default parameters in the present research.

2.3.2. Monte Carlo Feature Selection

The Monte Carlo feature selection (MCFS) approach repeatedly selects many features at random before building a series of decision tree (DT) classifiers [25,26]. Intuitively, if a feature is chosen numerous times to form a classification tree, then it is significant because the classification model will choose the most distinguishing feature to become a node. A feature is given a score called relative importance depending on how well it behaves in these classifiers, which is calculated using the formula:

$$RI_g = \sum_{\tau=1}^{st} (wAcc)^u \sum_{n_g(\tau)} IG(n_g(\tau)) \left(\frac{no.in n_g(\tau)}{no.in \tau} \right)^v, \quad (1)$$

where $wAcc$ represents weighted accuracy; $IG(n_g(\tau))$ represents information gain (IG) of $n_g(\tau)$; $(no.in n_g(\tau))$ represents the sample numbers in $n_g(\tau)$; and $(no.in \tau)$ represents the sample numbers in the tree root. u and v represent two settled positive integers.

In this study, we adopted the MCFS program retrieved from <http://www.ipipan.eu/staff/m.draminski/mcfs.html> accessed on 4 June 2019. Such program was performed using its default parameters.

2.3.3. Light Gradient Boosting Machine

The light gradient boosting machine (LightGBM) is a framework for implementing the gradient boosting decision tree algorithm which supports efficient parallel training and has the benefits of faster training speed, lower memory consumption, better accuracy, and distributed support for fast data processing [27]. The number of times a feature is used in modeling is used to determine its importance in LightGBM, and features that are used more often are considered more important. In this case, we used LightGBM in Python, which was downloaded at <https://lightgbm.readthedocs.io/en/latest/> accessed on 7 May 2019 with the default parameters.

2.3.4. Least Absolute Shrinkage and Selection Operator

Robert Tibshirani first presented least absolute shrinkage and selection operator (LASSO) in 1996 based on Leo Breiman's nonnegative garrote [28,29]. It is a regularization-based regression analysis approach with a feature selection function with the goal of enhancing statistical model detection accuracy and interpretability.

In a given training sample set where $X = [x_1, x_2, \dots, x_N]^T \in R^{N \times d}$, x_i denotes the feature vector of the i -th sample, N denotes the number of training samples, and d denotes the feature dimension. $Y = [y_1, y_2, \dots, y_N]^T \in R^N$ denotes the classification labels corresponding to these samples. The objective function for the optimization of the LASSO feature selection method is:

$$\min_w \frac{1}{2} \|Y - X^T w\|_2^2 + \lambda \|w\|_1 \quad (2)$$

where w denotes the regression coefficient of the eigenvector. The regularization term $\|w\|_1$ using the L_1 paradigm will produce a sparse solution in the feature space. The coefficients corresponding to irrelevant and redundant features will be set to 0, while the features corresponding to nonzero coefficients will be retained for subsequent classification. The absolute value of the regression coefficient represents the importance of the features, and the impact on the classification results is greater when the absolute value of the regression coefficient is larger. In this study, we adopted the LASSO package integrated in Scikit-learn and default parameters were used.

Through each of above algorithms, one feature list was obtained. For convenience, the lists obtained by the LASSO, LightGBM, MCFS and mRMR methods were called the LASSO, LightGBM, MCFS and mRMR feature lists, respectively.

2.4. Incremental Feature Selection

Four feature lists were obtained by using different feature ranking algorithms. They measured the importance of each feature in different aspects. However, the number of features in each feature list, which can be used for constructing efficient classifiers, was unknown. Here, incremental feature selection (IFS) was employed to extract the optimal subset of features for building the optimal classifiers [30–32]. The specific implementation steps are as follows: (1) Setting a variable k ($k = 1, 2, \dots, n$), where n is the number of features filtered by Boruta. (2) The top k features are taken from one feature list. (3) Samples containing k features are fed into one given classification algorithm for training a classifier, and its classification performance is assessed by ten-fold cross-validation [33]. (4) After testing all possible k , the classifier with the best classification performance and its corresponding features are selected as the optimal classifier and the optimal features.

2.5. Synthetic Minority Oversampling Technique

Synthetic minority oversampling technique (SMOTE) is an over-sampling method which can produce new samples for minor classes so as to solve the imbalanced class problem [34,35]. The main calculation procedure of the SMOTE algorithm is as follows: (1) Randomly select one sample, e.g., x , from the minor class. (2) Its Euclidean distances to other samples in the minor class are calculated and k nearest neighbors are selected. (3) One neighbor, denoted by y , is randomly selected from these neighbors and a new sample is calculated according to the following formula:

$$x_{new} = x + rand(0, 1) \times (y - x) \quad (3)$$

where $rand(0, 1)$ stands for a random number between 0 and 1. The new sample is put into the minor class. (4) The above procedures are executed several times until samples in the minor class are same as those in the major class. In this study, the SMOTE method was employed when classifiers were constructed in the IFS method. New samples produced by SMOTE were not used in the feature analysis procedure.

The present study adopted the SMOTE obtained from <https://github.com/scikit-learn-contrib/imbalanced-learn> accessed on 24 March 2020. Default parameters were used.

2.6. Classification Algorithm

To execute the IFS method, one classification algorithm is necessary. In this study, we tried four algorithms: DT [36], k-nearest neighbor (KNN) [37], RF [38], and support vector machine (SVM) [39]. They have wide applications in dealing with biological and medical problems [40–49].

2.6.1. Decision Tree

A tree structure is referred to as a DT [36]. Each branch represents the output of this feature attribute on a value domain, and each leaf node carries a category. Starting at the root node, the associated feature attribute in the object to be classified is tested, the output branch is selected based on its value until it reaches the leaf node, and the category stored in the leaf node is taken as the decision result. This supervised learning method is based on the if-then-else rule, and the DT's rules are learned through training rather than by hand. In this study, we use optimal DTs to extract the classification rules. We conduct DT using the Scikit-learn package and the CART method with Gini coefficients as the information gain [50]. For convenience, default parameters were used to execute the above package.

2.6.2. k-Nearest Neighbor

KNN can be used for classification and regression predictive problems [37]. The closest neighbor technique uses a vector space model for categorization. The key assumption is that examples belonging to the same category have a high degree of similarity and that the categorization of cases belonging to unknown categories may be determined by comparing them to cases belonging to known categories. We achieve this precondition by specifying the default settings for the KNN model in the Scikit-learn package [50]. Likewise, default parameters were used.

2.6.3. Random Forest

RF is an ensemble learning model with strong predictive capacity that has great noise tolerance and unpredictability [38]. RF is a combinatorial classifier that relies on DTs as its foundation. This way alleviates some of the restrictions of a single classifier and allows for improved accuracy. It also alleviates some of the limitations of a single classifier and allows for improved prediction accuracy. At the same time, the unpredictability of the RF allows it to absorb the effects of outliers and noise to a higher extent, which reduces the overfitting problem of the DT method and improves generalization ability. In this case, we used RF from the Python Scikit-learn package [50]. This package was used with its default parameters.

2.6.4. Support Vector Machine

SVM is a quick and trustworthy classification system that works well with little amounts of data [39]. The principle of the SVM classification model is to use a kernel function for mapping sample points to a multidimensional feature space. Then, it constructs an optimal classification hyperplane that maximizes the distance between the hyperplane and a set of samples from different classes to maximize generalization ability. The SVM model from the Scikit-learn package is used in this research [50]. It was executed with its default parameters.

2.7. Performance Evaluation

For a binary classification, the predicted results can be counted as a confusion matrix containing four entries: true positive (TP), false negative (FN), false positive (FP), and true negative (TN). Based on these entries, several measurements can be computed. In this study, we used the following measurements: sensitivity (SN, also called recall), specificity

(*SP*), accuracy (*ACC*), *precision*, *F1-measure*, Matthews correlation coefficient (*MCC*), and G-Mean. They can be computed by:

$$SN = \frac{TP}{TP + FN} \quad (4)$$

$$SP = \frac{TN}{TN + FP} \quad (5)$$

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \quad (6)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$F1 - measure = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \quad (9)$$

$$G - mean = \sqrt{SN \cdot SP} \quad (10)$$

Among above measurements, the *F1-measure* was selected as the key measurement to help us select a classifier with the best performance.

3. Results

In this study, we extracted key mutational features to predict patients' clinical status and identified the rules used to distinguish different clinical states by a powerful computational analysis. Figure 1 depicts the whole computing process. The outcomes of each stage would be detailed in the following sections.

3.1. Results of Boruta and Feature Ranking Algorithms

The Boruta method was first used to analyze the importance of all mutant features. Irrelevant features were excluded. As a result, 57 mutant features were selected by the Boruta method, which can be found in Table S2.

The obtained 57 features were further investigated by four feature ranking algorithms, respectively. Each algorithm yielded a feature list, which can also be found in Table S2.

3.2. Results of IFS Method on Four Feature Lists

Four feature lists generated by different feature ranking algorithms were fed into the IFS method one by one, which incorporated four classification algorithms (DT, KNN, RF and SVM). For each list, the IFS method constructed all possible feature subsets, each of which contained some top features. A classifier based on one given classification algorithm was built on each feature subset and evaluated by ten-fold cross-validation. Detailed evaluation results are listed in Table S3.

3.2.1. Results of IFS Method on LASSO Feature List

For the LASSO feature list, the performances (*F1-measure*) of four classification algorithms for all possible feature subsets constructed from the list are illustrated in Figure 2. It can be observed that DT yielded the highest *F1-measure* of 0.798, which was obtained by using the top 57 features in the list. Accordingly, the optimal DT classifier was set up with these features. As for the other three classification algorithms, their highest *F1-measure* values were 0.732, 0.776 and 0.775, respectively. They were accessed by using the top 57, 6 and 6 features. Likewise, optimal KNN, RF and SVM classifiers could be built with corresponding features. Evidently, the optimal DT classifier provided a higher *F1-measure* than the other three optimal classifiers, indicating that it outperformed other optimal classifiers. Table 1 details the performance of the four optimal classifiers, from which we can see that

the optimal DT classifier provided the highest performance on almost all measurements except SN. This further confirmed the superiority of the optimal DT classifier.

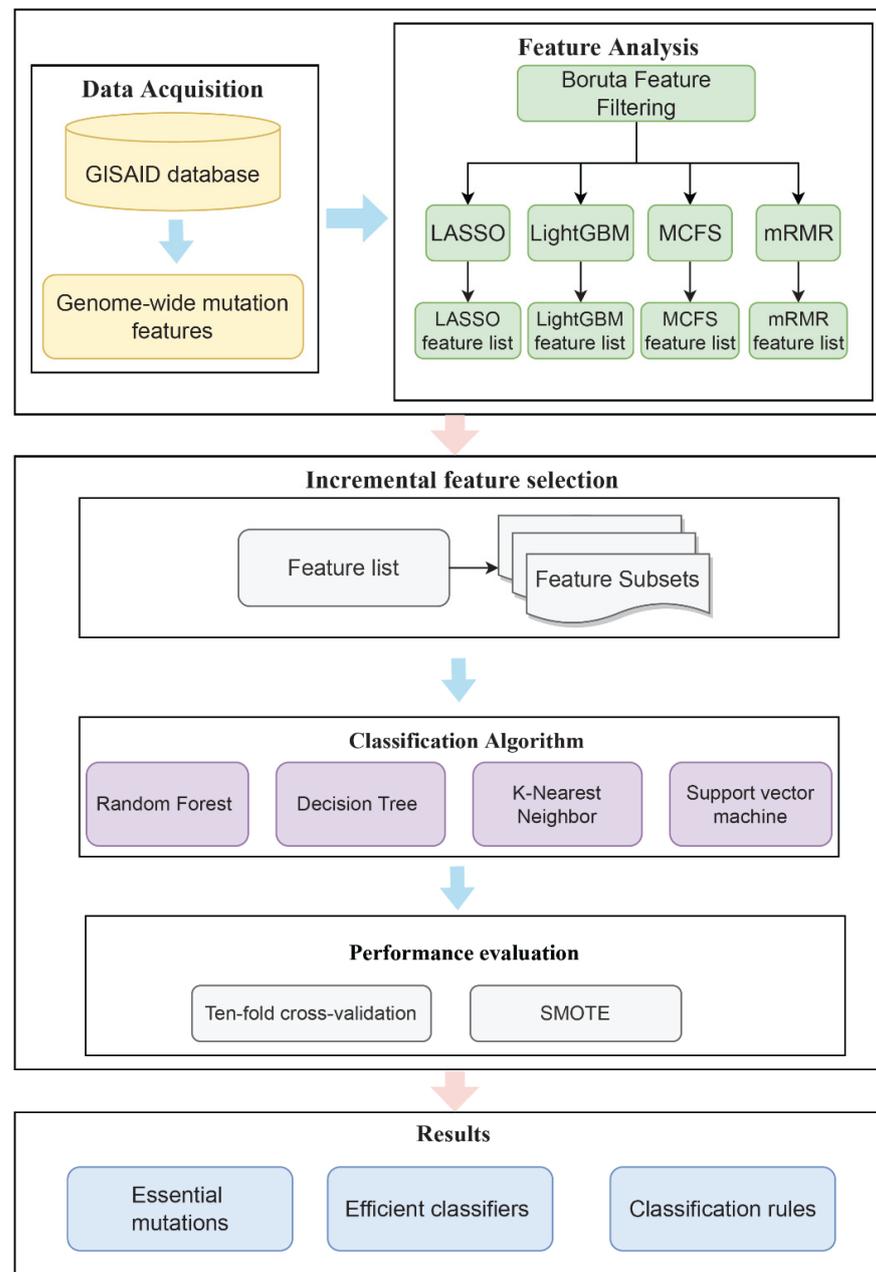


Figure 1. Flowchart of the whole analytical procedure of this research. Genome-wide mutation features of patients were obtained from the GISAID database and the Coronavirus Antiviral Research Database. Each patient was classified as “mild” or “severe” according to clinical status. Four lists of features were obtained after Boruta as well as four feature ranking algorithms. Subsequently, the optimal classifiers and the corresponding optimal features were obtained using the IFS method. The classification rules were mined by the optimal DT classifiers to obtain the classification basis for distinguishing the clinical status of different patients.

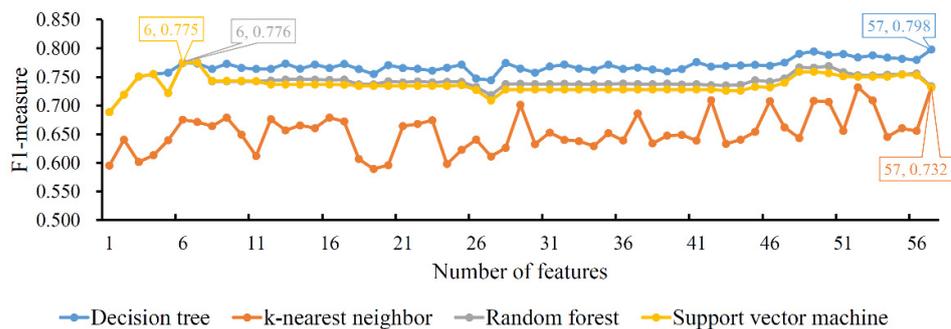


Figure 2. IFS curves of four classification algorithms based on the LASSO feature list. The four classification algorithms yield the highest *F1-measure* values of 0.798, 0.732, 0.776 and 0.775 when the top 57, 57, 6 and 6 features in the list are used.

Table 1. Performance of the optimal classifiers based on different classification algorithms and LASSO feature list ¹.

Term	Decision Tree	k-Nearest Neighbor	Random Forest	Support Vector Machine
Number of features	57	57	6	6
<i>SN</i>	0.788	0.824	0.821	0.820
<i>SP</i>	0.832	0.616	0.734	0.734
<i>ACC</i>	0.811	0.714	0.775	0.775
<i>MCC</i>	0.621	0.447	0.555	0.554
<i>Precision</i>	0.808	0.658	0.735	0.735
<i>F1-measure</i>	0.798	0.732	0.776	0.775
<i>G-mean</i>	0.809	0.712	0.776	0.776

¹ Numbers in bold are highest in the corresponding rows.

3.2.2. Results of IFS Method on LightGBM Feature List

The same procedures were performed on the LightGBM feature list. To clearly display the performance of four classification algorithms on all feature subsets, one IFS curve was plotted for each classification algorithm, as shown in Figure 3. DT/KNN/RF/SVM yielded the highest *F1-measure* with 0.803/0.758/0.785/0.783. This performance was achieved by using the top 24/52/24/24 features in the list. Accordingly, an optimal DT/KNN/RF/SVM classifier could be built. Similar to the results of the LASSO feature list, the optimal DT classifier also provided the best *F1-measure*. By observing the measurements of the four optimal classifiers, as listed in Table 2, we could further confirm that the optimal DT classifier was better than the other three optimal classifiers.

Table 2. Performance of the optimal classifiers based on different classification algorithms and LightGBM feature list ¹.

Term	Decision Tree	k-Nearest Neighbor	Random Forest	Support Vector Machine
Number of features	24	52	24	24
<i>SN</i>	0.844	0.878	0.813	0.816
<i>SP</i>	0.769	0.605	0.768	0.760
<i>ACC</i>	0.804	0.734	0.789	0.787
<i>MCC</i>	0.612	0.498	0.580	0.575
<i>Precision</i>	0.766	0.666	0.759	0.754
<i>F1-measure</i>	0.803	0.758	0.785	0.783
<i>G-mean</i>	0.805	0.729	0.790	0.788

¹ Numbers in bold are highest in corresponding rows.

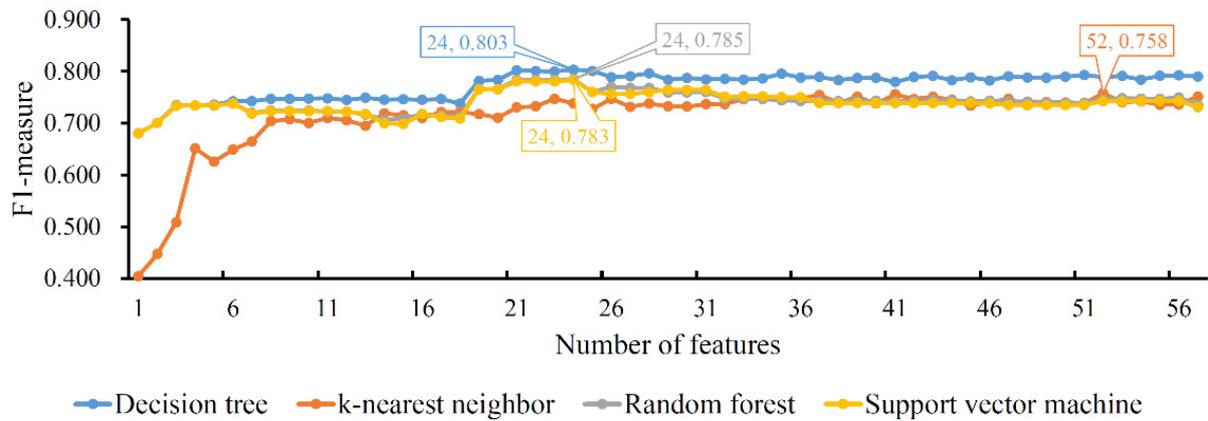


Figure 3. IFS curves of four classification algorithms based on the LightGBM feature list. The four classification algorithms yield the highest *F1-measure* values of 0.803, 0.758, 0.785 and 0.783 when the top 24, 52, 24 and 24 features in the list are used.

3.2.3. Results of IFS Method on MCFS Feature List

As for the MCFS feature list, the IFS method was also performed. An IFS curve was also plotted to show the performance of each classification algorithm on different feature subsets, as illustrated in Figure 4. The highest *F1-measure* values for four classification algorithms were 0.800, 0.745, 0.760 and 0.758, respectively, which were obtained by using the top 43, 55, 10 and 10 features in the list, respectively. Thus, we could set up the optimal DT, KNN, RF and SVM classifiers with the corresponding top features. Interestingly, the optimal DT classifier also generated the best *F1-measure*, conforming to the results of the LASSO and LighGBM feature lists. The performance details, including the seven measurements listed in Section 2.7, of the four optimal classifiers are provided in Table 3. It can be observed that the optimal DT classifier provided the best performance of all seven measurements, suggesting the high performance of the optimal DT classifier.

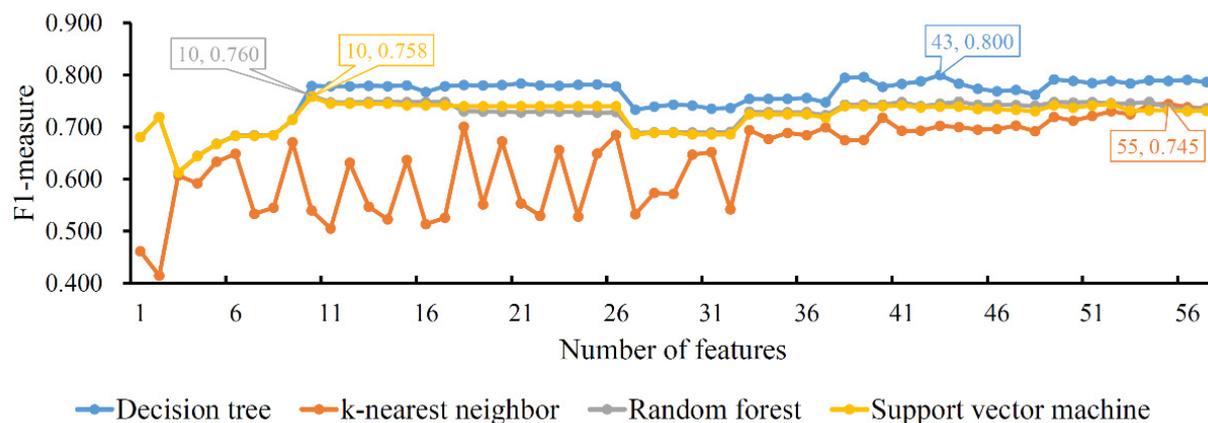


Figure 4. IFS curves of four classification algorithms based on the MCFS feature list. The four classification algorithms yield highest *F1-measure* values of 0.800, 0.745, 0.760 and 0.758 when the top 43, 55, 10 and 10 features in the list are used.

3.2.4. Results of IFS Method on mRMR Feature List

Finally, the IFS method was applied on the mRMR feature list. The *F1-measure* values yielded by each classification algorithm of the different feature subsets are illustrated in Figure 5, from which we can see that the highest *F1-measure* values for the four classification algorithms were 0.797, 0.759, 0.757 and 0.756, respectively. The highest performance was obtained by using the top 53, 52, 24 and 23 features in the list, respectively, with

which optimal DT, KNN, RF and SVM classifiers could be set up. Likewise, the optimal DT classifier also provided the best *F1-measure*, suggesting it was better than the other three optimal classifiers. Details of the performance of the four optimal classifiers are provided in Table 4. The optimal DT classifier generated the highest performance of the four measurements, indicating its superiority.

Table 3. Performance of the optimal classifiers based on different classification algorithms and MCFS feature list ¹.

Term	Decision Tree	k-Nearest Neighbor	Random Forest	Support Vector Machine
Number of features	43	55	10	10
SN	0.848	0.832	0.816	0.811
SP	0.755	0.637	0.704	0.704
ACC	0.799	0.730	0.757	0.755
MCC	0.603	0.476	0.521	0.516
Precision	0.757	0.673	0.712	0.711
F1-measure	0.800	0.745	0.760	0.758
G-mean	0.800	0.728	0.758	0.756

¹ Numbers in bold are highest in corresponding rows.

Table 4. Performance of the optimal classifiers based on different classification algorithms and the mRMR feature list ¹.

Term	Decision Tree	k-Nearest Neighbor	Random Forest	Support Vector Machine
Number of features	53	52	24	23
SN	0.777	0.885	0.666	0.665
SP	0.846	0.598	0.915	0.916
ACC	0.813	0.734	0.797	0.797
MCC	0.625	0.501	0.604	0.604
Precision	0.819	0.665	0.875	0.877
F1-measure	0.797	0.759	0.757	0.756
G-mean	0.810	0.728	0.781	0.780

¹ Numbers in bold are highest in the corresponding rows.

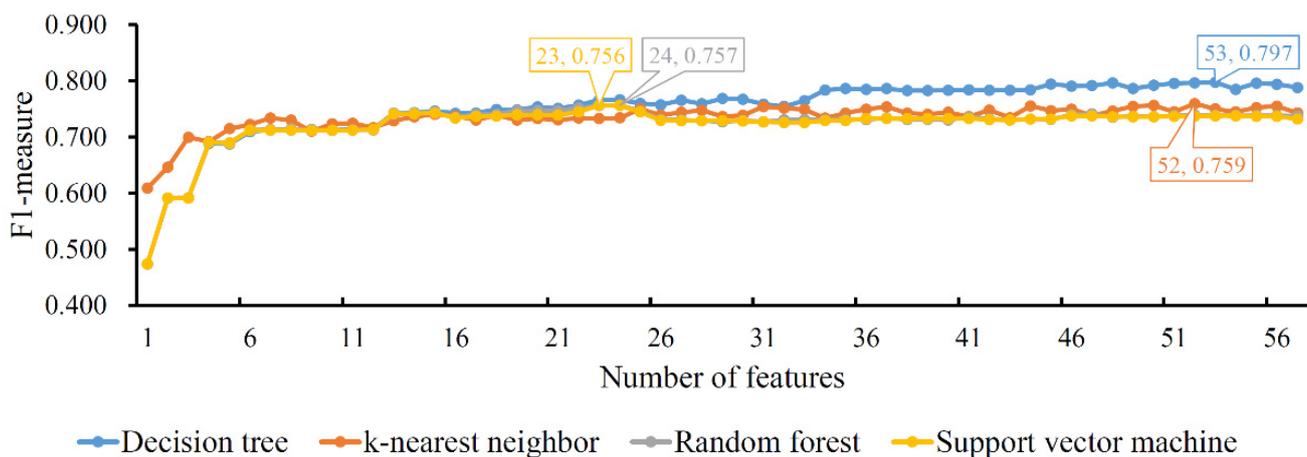


Figure 5. IFS curves of four classification algorithms based on the mRMR feature list. The four classification algorithms yield the highest *F1-measure* values of 0.797, 0.759, 0.757 and 0.756 when the top 53, 52, 24 and 23 features in the list are used.

According to the above descriptions, several optimal classifiers were set up. For each classification algorithm, four optimal classifiers were built based on four different feature lists. Their *F1-measure* values are illustrated in a box plot, as shown in Figure 6. It can be observed that the performance of each optimal classifier of different feature lists was

quite similar and the optimal DT classifier provided the best performance. Accordingly, for each feature list, we selected the features used in the optimal DT classifiers as the optimal features.

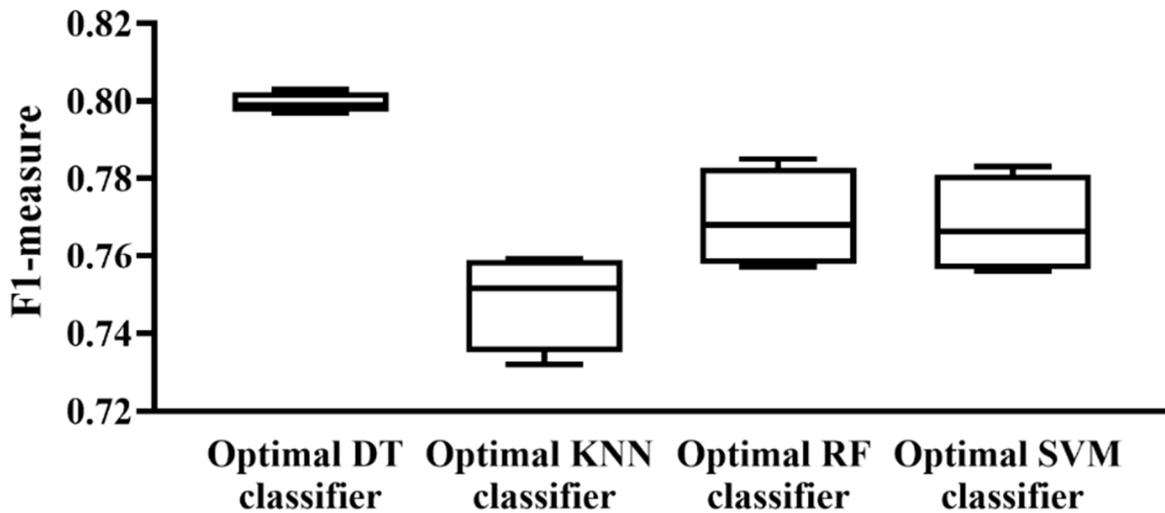


Figure 6. Box plot to show the performance of the optimal classifier based on different classification algorithms and feature lists. Each optimal classifier provided a similar performance in the different feature lists and the optimal DT classifier provided the highest performance.

3.3. Results of Intersection of Optimal Features on Different Feature Lists

By applying the IFS method to four feature lists, four sets of optimal features were accessed; these were the features used to construct the optimal DT classifier. The intersection of these four optimal feature subsets was investigated. Venn diagrams for these feature subsets can be found in Figure 7. It can be observed that 15 features occurred in all four feature subsets; 34 features existed in three feature subsets; 7 features belonged to exactly two feature subsets; and only 1 feature was in one feature subset. Features that appeared in four, three, two and one feature subsets are listed in Table S4. The presence of mutations in all four feature subsets indicated their importance to differentiate the clinical status of patients. A detailed explanation of these important mutations can be found in Section 4.1.

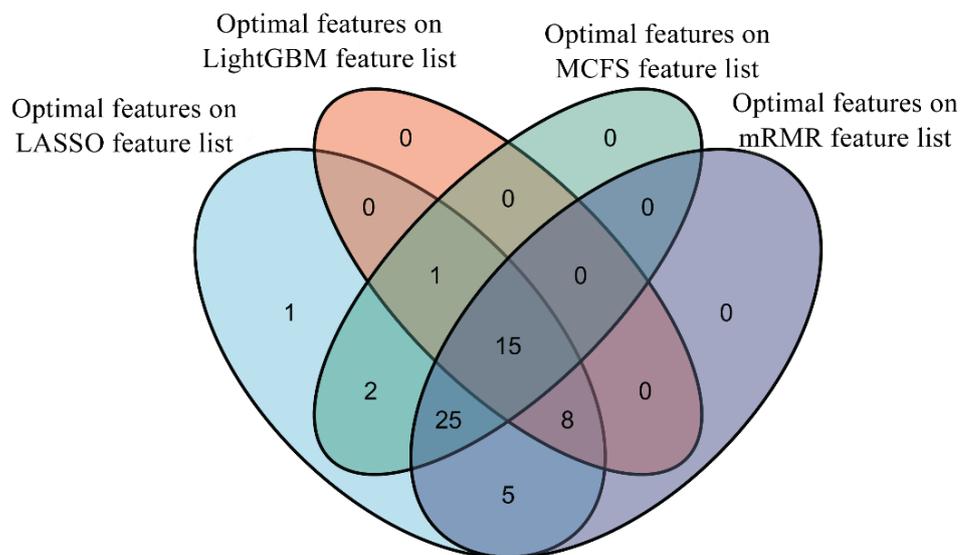


Figure 7. Venn diagram of the optimal feature subsets on four feature lists. Fifteen features occur in all four feature subsets, indicating their importance to differentiate the clinical status of patients.

3.4. Classification Rules

It is known that DT is a white-box algorithm, which are always generally weaker than most black-box classification algorithms, including KNN, RF and SVM used in this study. However, as mentioned above, the optimal DT classifiers were better than the optimal KNN/RF/SVM classifier of the investigated dataset. Further investigation into such DT classifiers is helpful to reveal the relationship between mutation and the clinical status of the patients with COVID-19.

Based on the features used in each optimal DT classifier, all samples were represented by them. DT was applied on this dataset, producing a large tree from which several classification rules could be constructed. Accordingly, four sets of classification rules were obtained which included 125, 97, 89 and 126 rules. Table S5 contains a detailed description of the rules. Each rule was made up of several conditions and one result. If a sample satisfied the condition, it would be classified into the corresponding group (“mild” or “severe”). Each condition involved one mutation feature and one threshold, most of which were 0.5. This threshold indicated that the presence or absence of a set of mutations could distinguish the clinical status of patients, which suggests the association of these features and indicates that these mutations may serve as potential targets for future therapy. We further investigated these similarity rules. A detailed discussion of these rules can be found in Section 4.2.

4. Discussion

In this study, data related to COVID-19 were deeply investigated by several computational algorithms. Four sets of optimal mutation features and decision rules that were strongly correlated with the severity of COVID-19 were identified. By combining features in four subsets, 57 mutation features were obtained which were deemed to be COVID-19 severity-related. Except the feature of “age”, we classified the remaining 56 mutations into distinct categories according to the genomic locations and protein functions. As shown in Figure 8, mutations in both untranslated regions and gene regions were associated with COVID-19 severity. The locations of some important spike protein mutations in the genome are shown in Figure 9. The analysis of these mutational signatures and decision rules may inform clinical evaluation and vaccine development.

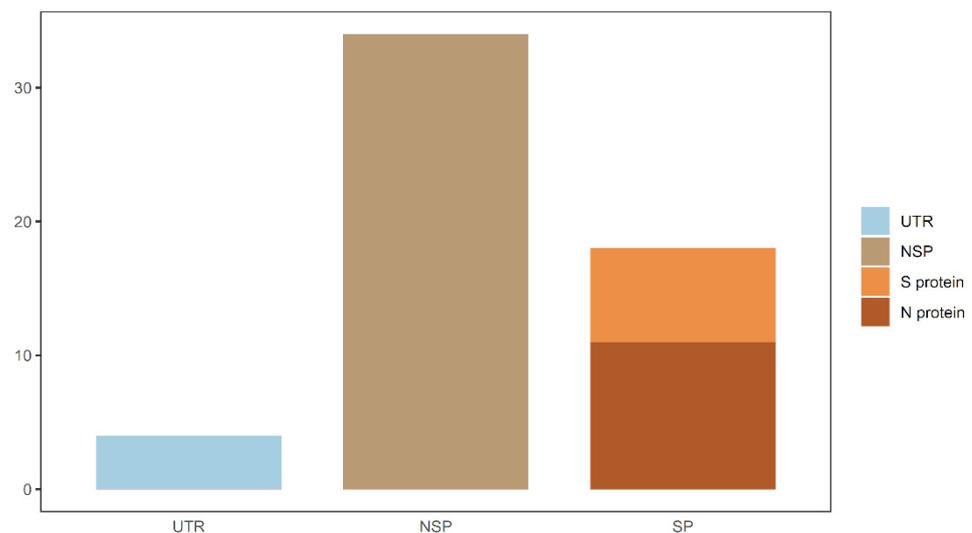


Figure 8. Summary of COVID-19 severity-related mutations. UTR—untranslated region; NSP—nonstructural protein; SP—structural protein; S protein—spike protein; N protein—envelope protein.

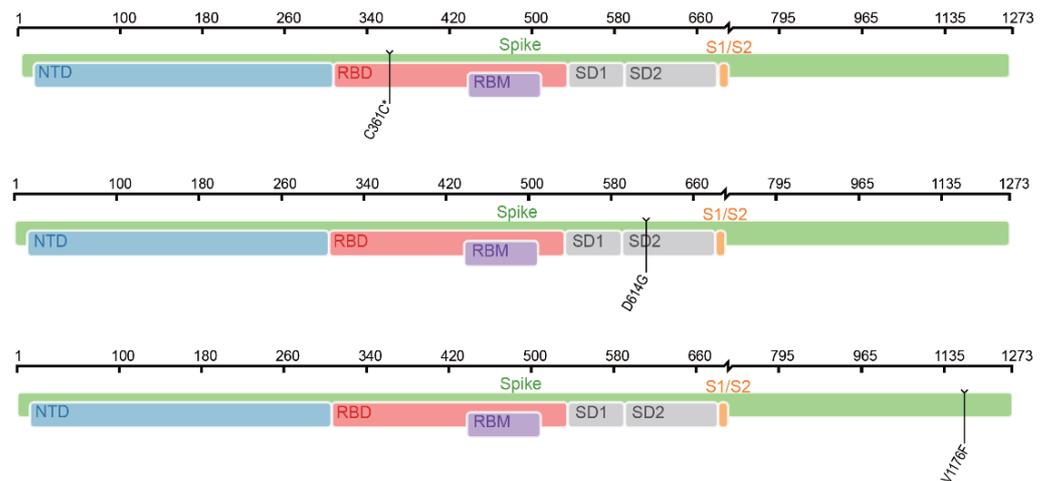


Figure 9. Location of some spike protein mutations in the genome. NTD—N-terminal domain; RBD—receptor-binding domain; RBM—receptor-binding motif; SD1—subdomain 1; SD2—subdomain 2; S1/S2—S1/S2 cleavage region.

4.1. Features Associated with COVID-19 Severity

We observed that 15 features (i.e., V1176F, age, ORF3a:Q57H, RdRP:P323L, N:I292T, nsp14:A320V, ORF6:I33T, SNP:C241T, nsp4:F308Y, nsp7:L71F, C361C*, D614G, nsp6:L37F, N:S202N, and nsp14:I164ILV) were simultaneously present in the results of the four groups involving age, S protein mutations, SNP mutations in the UTR, and mutations in other proteins. Most of these mutations have been widely reported to be associated with the pathogenicity of SARS-CoV-2, while some mutations have been less studied (e.g., nsp14:A320V, ORF6:I33T, C361C*, and N:S202N).

Among the 15 most significant mutational features associated with COVID-19 severity, V1176F is a classical mutational signature of the S protein and was originally identified in the Gamma variant. Studies suggest that V1176F may enhance S protein flexibility and is associated with higher patient mortality [51,52]. Another important S protein mutation in our results was D614G, and previous studies have shown that the mutation of D614G leads to the binding affinity of the virus to ACE2, thus increasing the efficiency of cell entry and further enhancing the infectivity of the virus [53]. Although previous studies of hospitalization outcomes have shown no significant association between D614G and disease severity, our results show that it is important for post-COVID-19 severity prediction [54]. Another S protein mutation, C361C*, is also associated with COVID-19 severity, which has not been investigated in depth in previous studies.

Numerous mutations in other encoded proteins have also been shown to correlate with disease severity. For example, nsp4:F308Y and nsp6:L37F have been reported to be associated with mild disease, while ORF3a:Q57H, RdRP:P323L, N:I292T, ORF6:I33T, and nsp7:L71F are associated with severe disease [55]. N:S202N is a hot spot mutation in a linker region which may be associated with enhanced RNA binding and altered serine phosphorylation, and no evidence has shown that N:S202N is associated with disease severity in COVID-19 patients. Our results also show that nsp14:A320V and nsp14:I164ILV are associated with the disease severity, which has not been reported in previous studies. Nsp14 is known to function in viral replication and transcription. Mutations in nsp14 may cause the alterations of viral functions, thus affecting virological properties and disease severity.

Age is also an important affecting factor of disease progression, and the infection rate of COVID-19 and the risk of serious disease increase with rising age [56,57]. Another important mutation with a small proportion is SNP:C241T in the UTR region. It is often accompanied with RdRP:P323L or D614G variants, and no studies have directly shown that SNP:C241T is associated with the severity of COVID-19 [58].

By combining existing studies, we found that 15 features are important for identifying the severity of COVID-19. This deduction not only demonstrates the reliability of our study, but also suggests that some under-appreciated mutations (e.g., C361C*, N:S202N, nsp14:A320V, nsp14:I164ILV, and SNP:C241T) may be associated with COVID-19 severity.

4.2. Decision Rules Related to the Severity of COVID-19 Infections

The four sets of decision rules (Table S5) were also similar, and we further analyzed the rules with higher passed counts to classify the severity of COVID-19 for most samples according to different mutations.

Our decision rules often involved a combination of multiple criteria, but we found that a combination of two genes (i.e., RdRP:P323L and V1176F) could partly determine cases of severe COVID-19 disease, and that the decision rules related to the results of the four groups of algorithms. This finding is consistent with that of existing studies; specifically, RdRP:P323L and V1176F are associated with poorer clinical presentation [52,55].

We also found that RdRP:P323L is often accompanied by ORF3a:Q57H, D614G, SNP:C241T, nsp2:T85I, and other mutations in severe COVID-19 patients. D614G is a hotspot mutation that appeared as early as the Alpha variant, and although the D614G mutation does not directly correlate with disease severity, other COVID-19 variants containing this mutation may accelerate infection due to the ability to facilitate viral entry into cells and further aggravate the disease [54,59]. ORF3a:Q57H and nsp2:T85I first appeared in the Beta variants and our results suggest that they may be associated with severe outcomes of SARS-CoV-2 infection [60]. Studies have shown that SNP:C241T mutation is associated with lower viral replication efficiency [61] and it can be observed in our results that mild patients with SNP:C241T mutation tend to be combined with less other severe mutation features (e.g., nsp2:T85I, ORF3a:Q57H, and V1176F), which suggests that this gene mutation may be indirectly associated with mild COVID-19.

In our results, some COVID-19 mutations associated with mild disease were also found, such as nsp14:I164ILV, ORF8:L84S, nsp4:F308Y, nsp6:L37F, ORF3a:G196V, N:P13L, nsp4:M324I, and N:G204R. These mutations tend to be present in low-risk patients with different combinations, and some have been reported to be associated with mild disease or disease transmissibility [55,60].

4.3. Comparison of the Previous Study

In Nagy et al.'s study [15], a computation analysis was also conducted on similar datasets. However, there were several differences between our study and theirs. The main differences were as follows.

- (1) The mutation features analyzed in our study were generated by a different platform. As different platforms have different advantages and disadvantages, analyzing mutation features obtained from different platforms can uncover novel mutations that are highly correlated with the clinical status of patients with COVID-19;
- (2) More machine learning algorithms were used in this study than in Nagy et al.'s study. We used five feature analysis methods (Boruta, LASSO, LightGBM, MCF5 and mRMR) and four classification algorithms (DT, KNN, RF and SVM). With these algorithms, each mutation feature was fully evaluated. The final mutation features were selected by multiple feature analysis methods, increasing the reliability of the results;
- (3) In this study, we not only discovered mutation features related to the clinical status of the patients with COVID-19 but also established rules to indicate more complicated mutation patterns of the clinical status of the patients with COVID-19. These patterns always included multiple mutation features, suggesting the relationships between a combination of mutation features and the clinical status of patients with COVID-19. Such form can be deemed an extension of single mutation biomarkers, and was not involved in Nagy et al.'s study;

- (4) Biological analysis was performed in our study, increasing the reliability of the results. In Nagy et al.'s study, the important mutation features were only listed and were not analyzed.

These differences indicate that the new findings reported in this study can provide novel insights to investigate the relationships between mutation and the clinical status of patients with COVID-19.

5. Conclusions

In this research, we utilized sophisticated and widely used computational approaches in relation to genome-wide mutation data to explore important mutations used to distinguish clinical status in COVID-19 patients. In short, our results demonstrated a set of mutations associated with SARS-CoV-2 severity, which could help rapidly identify SARS-CoV-2 infections carrying mutations associated with severe outcomes and guide the development of SARS-CoV-2 vaccines. We also suggested some effective classifiers for predicting COVID-19 patients' clinical status; these were trained with a vast quantity of data and performed well in classification. The quantitative decision rules in the optimal DT classifier provided direct clues to distinguish between different patient clinical states.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/life12060806/s1>, Table S1: SARS-CoV-2 genomic feature matrix. Rows represent samples, and columns represent mutational features. In the feature matrix, a virus genome is denoted as "1" if it contains a certain feature and "0" otherwise, Table S2: Feature list filtered by Boruta and ordered by four feature ranking algorithms, Table S3: Performance of IFS with different classifiers, Table S4: Intersection results of the optimal feature subsets in the four ranking algorithms. The column names represent features appearing in n ($n = 4,3,2,1$) feature sorting algorithms, Table S5: Classification rules generated by the optimal DT model based on four ranking algorithms.

Author Contributions: Conceptualization, T.H. and Y.C.; methodology, L.C. and K.F.; validation, T.H.; formal analysis, F.H., W.G. and X.Z.; data curation, T.H.; writing—original draft preparation, F.H. and L.C.; writing—review and editing, T.H.; supervision, Y.C. funding acquisition, T.H. and Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Strategic Priority Research Program of Chinese Academy of Sciences [XDA26040304, XDB38050200], National Key R&D Program of China [2018YFC0910403], the Fund of the Key Laboratory of Tissue Microenvironment and Tumor of Chinese Academy of Sciences [202002].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available at <https://www.gisaid.org/> (accessed on 6 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. CSG International. The species severe acute respiratory syndrome-related coronavirus: Classifying 2019-ncov and naming it SARS-CoV-2. *Nat. Microbiol.* **2020**, *5*, 536. [[CrossRef](#)] [[PubMed](#)]
2. Zhou, B.; Thao, T.T.N.; Hoffmann, D.; Taddeo, A.; Ebert, N.; Labroussaa, F.; Pohlmann, A.; King, J.; Steiner, S.; Kelly, J.N. SARS-CoV-2 spike d614g change enhances replication and transmission. *Nature* **2021**, *592*, 122–127. [[CrossRef](#)] [[PubMed](#)]
3. Hou, Y.J.; Chiba, S.; Halfmann, P.; Ehre, C.; Kuroda, M.; Dinnon, K.H.; Leist, S.R.; Schäfer, A.; Nakajima, N.; Takahashi, K. SARS-CoV-2 d614g variant exhibits efficient replication ex vivo and transmission in vivo. *Science* **2020**, *370*, 1464–1468. [[CrossRef](#)] [[PubMed](#)]
4. Pachetti, M.; Marini, B.; Benedetti, F.; Giudici, F.; Mauro, E.; Storici, P.; Masciovecchio, C.; Angeletti, S.; Ciccozzi, M.; Gallo, R.C. Emerging SARS-CoV-2 mutation hot spots include a novel rna-dependent-rna polymerase variant. *J. Transl. Med.* **2020**, *18*, 179. [[CrossRef](#)]
5. Cui, J.; Li, F.; Shi, Z.-L. Origin and evolution of pathogenic coronaviruses. *Nat. Rev. Microbiol.* **2019**, *17*, 181–192. [[CrossRef](#)]

6. Marra, M.A.; Jones, S.J.; Astell, C.R.; Holt, R.A.; Brooks-Wilson, A.; Butterfield, Y.S.; Khattra, J.; Asano, J.K.; Barber, S.A.; Chan, S.Y. The genome sequence of the sars-associated coronavirus. *Science* **2003**, *300*, 1399–1404. [[CrossRef](#)]
7. Wan, Y.; Shang, J.; Graham, R.; Baric, R.S.; Li, F. Receptor recognition by the novel coronavirus from wuhan: An analysis based on decade-long structural studies of sars coronavirus. *J. Virol.* **2020**, *94*, e00127–e00200. [[CrossRef](#)]
8. Leung, K.; Shum, M.H.; Leung, G.M.; Lam, T.T.; Wu, J.T. Early transmissibility assessment of the n501y mutant strains of SARS-CoV-2 in the united kingdom, october to november 2020. *Eurosurveillance* **2021**, *26*, 2002106. [[CrossRef](#)]
9. Mwenda, M.; Saasa, N.; Sinyange, N.; Busby, G.; Chipimo, P.J.; Hendry, J.; Kapona, O.; Yingst, S.; Hines, J.Z.; Minchella, P.; et al. Detection of b.1.351 SARS-CoV-2 variant strain—Zambia, december 2020. *MMWR Morb. Mortal. Wkly. Rep.* **2021**, *70*, 280–282. [[CrossRef](#)]
10. Faria, N.R.; Mellan, T.A.; Whittaker, C.; Claro, I.M.; Candido, D.d.S.; Mishra, S.; Crispim, M.A.; Sales, F.C.; Hawryluk, I.; McCrone, J.T. Genomics and epidemiology of the p. 1 SARS-CoV-2 lineage in manaus, brazil. *Science* **2021**, *372*, 815–821. [[CrossRef](#)]
11. Mlcochova, P.; Kemp, S.A.; Dhar, M.S.; Papa, G.; Meng, B.; Ferreira, I.A.; Datir, R.; Collier, D.A.; Albecka, A.; Singh, S. SARS-CoV-2 b. 1.617. 2 delta variant replication and immune evasion. *Nature* **2021**, *599*, 114–119. [[CrossRef](#)] [[PubMed](#)]
12. Callaway, E. Heavily mutated coronavirus variant puts scientists on alert. *Nature* **2021**, *600*, 21. [[CrossRef](#)]
13. Wang, Z.; Schmidt, F.; Weisblum, Y.; Muecksch, F.; Barnes, C.O.; Finkin, S.; Schaefer-Babajew, D.; Cipolla, M.; Gaebler, C.; Lieberman, J.A. Mrna vaccine-elicited antibodies to SARS-CoV-2 and circulating variants. *Nature* **2021**, *592*, 616–622. [[CrossRef](#)]
14. Abdullahi, I.N.; Emeribe, A.U.; Ajayi, O.A.; Oderinde, B.S.; Amadu, D.O.; Osuji, A.I. Implications of SARS-CoV-2 genetic diversity and mutations on pathogenicity of COVID-19 and biomedical interventions. *J. Taibah Univ. Med. Sci.* **2020**, *15*, 258–264. [[CrossRef](#)] [[PubMed](#)]
15. Nagy, Á.; Ligeti, B.; Szebeni, J.; Pongor, S.; Gyrfy, B. Covidoutcome-estimating covid severity based on mutation signatures in the SARS-CoV-2 genome. *Database J. Biol. Databases Curation* **2021**, *2021*, baab020. [[CrossRef](#)] [[PubMed](#)]
16. Tzou, P.L.; Tao, K.; Nouhin, J.; Rhee, S.Y.; Hu, B.D.; Pai, S.; Parkin, N.; Shafer, R.W. Coronavirus antiviral research database (cov-rdb): An online database designed to facilitate comparisons between candidate anti-coronavirus compounds. *Viruses* **2020**, *12*, 1006. [[CrossRef](#)] [[PubMed](#)]
17. Kurtz, S.; Phillippy, A.; Delcher, A.L.; Smoot, M.; Shumway, M.; Antonescu, C.; Salzberg, S.L. Versatile and open software for comparing large genomes. *Genome Biol.* **2004**, *5*, R12. [[CrossRef](#)]
18. Brodin, P. Immune determinants of COVID-19 disease presentation and severity. *Nat. Med.* **2021**, *27*, 28–33. [[CrossRef](#)]
19. Brodin, P. Why is COVID-19 so mild in children? *Acta Paediatr.* **2020**, *109*, 1082–1083. [[CrossRef](#)]
20. Kursu, M.B.; Rudnicki, W.R. Feature selection with the boruta package. *J. Stat. Softw.* **2010**, *36*, 1–13. [[CrossRef](#)]
21. Ding, S.; Li, H.; Zhang, Y.H.; Zhou, X.; Feng, K.; Li, Z.; Chen, L.; Huang, T.; Cai, Y.D. Identification of pan-cancer biomarkers based on the gene expression profiles of cancer cell lines. *Front. Cell Dev. Biol.* **2021**, *9*, 781285. [[CrossRef](#)] [[PubMed](#)]
22. Chen, L.; Zhang, Y.H.; Wang, S.; Zhang, Y.; Huang, T.; Cai, Y.D. Prediction and analysis of essential genes using the enrichments of gene ontology and kegg pathways. *PLoS ONE* **2017**, *12*, e0184129. [[CrossRef](#)] [[PubMed](#)]
23. Yu, X.; Pan, X.; Zhang, S.; Zhang, Y.H.; Chen, L.; Wan, S.; Huang, T.; Cai, Y.D. Identification of gene signatures and expression patterns during epithelial-to-mesenchymal transition from single-cell expression atlas. *Front. Genet.* **2020**, *11*, 605012. [[CrossRef](#)] [[PubMed](#)]
24. Peng, H.; Fulmi, L.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Anal. Mach. Intell. IEEE Trans.* **2005**, *27*, 1226–1238. [[CrossRef](#)] [[PubMed](#)]
25. Micha, D.; Rada-Iglesias, A.; Enroth, S.; Wadelius, C.; Koronacki, J.; Komorowski, J. Monte carlo feature selection for supervised classification. *Bioinformatics* **2008**, *24*, 110–117.
26. Li, J.; Lu, L.; Zhang, Y.H.; Xu, Y.; Liu, M.; Feng, K.; Chen, L.; Kong, X.; Huang, T.; Cai, Y.D. Identification of leukemia stem cell expression signatures through monte carlo feature selection strategy and support vector machine. *Cancer Gene Ther.* **2020**, *27*, 56–69. [[CrossRef](#)]
27. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NIP 2017)*. 2017. Available online: <https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html> (accessed on 11 April 2022).
28. Breiman, Leo. Better subset regression using the nonnegative garrote. *Technometrics* **1995**, *37*, 373–384. [[CrossRef](#)]
29. Tibshirani, R.J. Regression shrinkage and selection via the lasso. *J. R. Stat. Society. Ser. B Methodol.* **1996**, *73*, 273–282. [[CrossRef](#)]
30. Liu, H.; Setiono, R. Incremental feature selection. *Appl. Intell.* **1998**, *9*, 217–230. [[CrossRef](#)]
31. Chen, L.; Zeng, T.; Pan, X.; Zhang, Y.H.; Huang, T.; Cai, Y.D. Identifying methylation pattern and genes associated with breast cancer subtypes. *Int. J. Mol. Sci.* **2019**, *20*, 4269. [[CrossRef](#)]
32. Zhang, Y.H.; Guo, W.; Zeng, T.; Zhang, S.; Chen, L.; Gamarra, M.; Mansour, R.F.; Escorcía-Gutierrez, J.; Huang, T.; Cai, Y.D. Identification of microbiota biomarkers with orthologous gene annotation for type 2 diabetes. *Front. Microbiol.* **2021**, *12*, 711244. [[CrossRef](#)] [[PubMed](#)]
33. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence— Volume 2*; Morgan Kaufmann Publishers Inc.: Montreal, QC, Canada, 1995; pp. 1137–1143.

34. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [[CrossRef](#)]
35. Pan, X.; Chen, L.; Liu, I.; Niu, Z.; Huang, T.; Cai, Y.D. Identifying protein subcellular locations with embeddings-based node2loc. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 666–675. [[CrossRef](#)] [[PubMed](#)]
36. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [[CrossRef](#)]
37. Gorodkin, J. Comparing two k-category assignments by a k-category correlation coefficient. *Comput. Biol. Chem.* **2004**, *28*, 367–374. [[CrossRef](#)]
38. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
39. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
40. Chen, L.; Li, Z.; Zhang, S.; Zhang, Y.-H.; Huang, T.; Cai, Y.-D. Predicting rna 5-methylcytosine sites by using essential sequence features and distributions. *BioMed Res. Int.* **2022**, *2022*, 4035462. [[CrossRef](#)]
41. Ding, S.; Wang, D.; Zhou, X.; Chen, L.; Feng, K.; Xu, X.; Huang, T.; Li, Z.; Cai, Y. Predicting heart cell types by using transcriptome profiles and a machine learning method. *Life* **2022**, *12*, 228. [[CrossRef](#)]
42. Zhou, X.; Ding, S.; Wang, D.; Chen, L.; Feng, K.; Huang, T.; Li, Z.; Cai, Y.-D. Identification of cell markers and their expression patterns in skin based on single-cell rna-sequencing profiles. *Life* **2022**, *12*, 550. [[CrossRef](#)]
43. Li, X.; Lu, L.; Chen, L. Identification of protein functions in mouse with a label space partition method. *Math. Biosci. Eng.* **2022**, *19*, 3820–3842. [[CrossRef](#)] [[PubMed](#)]
44. Yang, Y.; Chen, L. Identification of drug–disease associations by using multiple drug and disease networks. *Curr. Bioinform.* **2022**, *17*, 48–59. [[CrossRef](#)]
45. Wu, Z.; Chen, L. Similarity-based method with multiple-feature sampling for predicting drug side effects. *Comput. Math. Methods Med.* **2022**, *2022*, 9547317. [[CrossRef](#)] [[PubMed](#)]
46. Chen, W.; Chen, L.; Dai, Q. Impt-fdnpl: Identification of membrane protein types with functional domains and a natural language processing approach. *Comput. Math. Methods Med.* **2021**, *2021*, 7681497. [[CrossRef](#)] [[PubMed](#)]
47. Baranwal, M.; Magner, A.; Elvati, P.; Saldinger, J.; Violi, A.; Hero, A.O. A deep learning architecture for metabolic pathway prediction. *Bioinformatics* **2019**, *36*, 2547–2553. [[CrossRef](#)]
48. Casanova, R.; Saldana, S.; Chew, E.Y.; Danis, R.P.; Greven, C.M.; Ambrosius, W.T. Application of random forests methods to diabetic retinopathy classification analyses. *PLoS ONE* **2014**, *9*, e98587. [[CrossRef](#)]
49. Sang, X.; Xiao, W.; Zheng, H.; Yang, Y.; Liu, T. Hmmpred: Accurate prediction of DNA-binding proteins based on hmm profiles and xgboost feature selection. *Comput. Math. Methods Med.* **2020**, *2020*, 1384749. [[CrossRef](#)]
50. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
51. Farkas, C.; Mella, A.; Haigh, J.J. Large-scale population analysis of SARS-CoV-2 whole genome sequences reveals host-mediated viral evolution with emergence of mutations in the viral spike protein associated with elevated mortality rates. *medRxiv* **2020**. [[CrossRef](#)]
52. Hahn, G.; Wu, C.M.; Lee, S.; Hecker, J.; Lutz, S.M.; Haneuse, S.; Qiao, D.; DeMeo, D.; Choudhary, M.C.; Etemad, B. Two mutations in the SARS-CoV-2 spike protein and rna polymerase complex are associated with COVID-19 mortality risk. *bioRxiv* **2020**. [[CrossRef](#)]
53. Ozono, S.; Zhang, Y.; Ode, H.; Sano, K.; Tan, T.S.; Imai, K.; Miyoshi, K.; Kishigami, S.; Ueno, T.; Iwatani, Y. SARS-CoV-2 d614g spike mutation increases entry efficiency with enhanced ace2-binding affinity. *Nat. Commun.* **2021**, *12*, 848. [[CrossRef](#)] [[PubMed](#)]
54. Korber, B.; Fischer, W.M.; Gnanakaran, S.; Yoon, H.; Theiler, J.; Abfalterer, W.; Hengartner, N.; Giorgi, E.E.; Bhattacharya, T.; Foley, B. Tracking changes in SARS-CoV-2 spike: Evidence that d614g increases infectivity of the COVID-19 virus. *Cell* **2020**, *182*, 812–827.e819. [[CrossRef](#)] [[PubMed](#)]
55. Nagy, Á.; Pongor, S.; Györfy, B. Different mutations in SARS-CoV-2 associate with severe and mild outcome. *Int. J. Antimicrob. Agents* **2021**, *57*, 106272. [[CrossRef](#)]
56. Guan, W.-J.; Ni, Z.-Y.; Hu, Y.; Liang, W.-H.; Ou, C.-Q.; He, J.-X.; Liu, L.; Shan, H.; Lei, C.-L.; Hui, D.S. Clinical characteristics of 2019 novel coronavirus infection in china. *medRxiv* **2020**. [[CrossRef](#)]
57. Davies, N.G.; Klepac, P.; Liu, Y.; Prem, K.; Jit, M.; Eggo, R.M. Age-dependent effects in the transmission and control of COVID-19 epidemics. *Nat. Med.* **2020**, *26*, 1205–1211. [[CrossRef](#)] [[PubMed](#)]
58. Nguyen, T.T.; Pham, T.N.; Van, T.D.; Nguyen, T.T.; Nguyen, D.T.N.; Le, H.N.M.; Eden, J.-S.; Rockett, R.J.; Nguyen, T.T.H.; Vu, B.T.N. Genetic diversity of SARS-CoV-2 and clinical, epidemiological characteristics of COVID-19 patients in hanoi, vietnam. *PLoS ONE* **2020**, *15*, e0242537. [[CrossRef](#)] [[PubMed](#)]
59. Eaaswarkhanth, M.; Al Madhoun, A.; Al-Mulla, F. Could the d614g substitution in the SARS-CoV-2 spike (s) protein be associated with higher COVID-19 mortality? *Int. J. Infect. Dis.* **2020**, *96*, 459–460. [[CrossRef](#)]
60. Patro, L.P.P.; Sathyaseelan, C.; Uttamrao, P.P.; Rathinavelan, T. Global variation in SARS-CoV-2 proteome and its implication in pre-lockdown emergence and dissemination of 5 dominant SARS-CoV-2 clades. *Infect. Genet. Evol.* **2021**, *93*, 104973. [[CrossRef](#)]
61. Chaudhari, A.; Chaudhari, M.; Mahera, S.; Saiyed, Z.; Nathani, N.M.; Shukla, S.; Patel, D.; Patel, C.; Joshi, M.; Joshi, C.G. In-silico analysis reveals lower transcription efficiency of c241t variant of SARS-CoV-2 with host replication factors madp1 and hnnp-1. *Inform. Med. Unlocked* **2021**, *25*, 100670. [[CrossRef](#)]