Deep Learning Applied to the SARS-CoV-2 Classification

Karolayne Azevedo
Federal University of Rio Grande do Norte
Luísa Souza
Federal University of Rio Grande do Norte
Maria Coutinho
Federal University of Rio Grande do Norte
Raquel Barbosa
Federal University of Rio Grande do Norte
Marcelo Fernandes (mfernandes@dca.ufrn.br)
Federal University of Rio Grande do Norte

Research Article

Keywords: SARS-CoV-2, COVID-19, viral classification, deep learning

Posted Date: September 5th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3290221/v1

Additional Declarations: No competing interests reported.

Deep Learning Applied to the SARS-CoV-2 Classification

Karolayne Azevedo[1], Luísa Souza[1], Maria Coutinho[1], Raquel Barbosa[1], Marcelo Fernandes[1,2,3*]

[1]InovAI Lab, Federal University of Rio Grande do Norte, Natal, 59078-970, RN, Brazil.
[2]Bioinformatics Multidisciplinary Environment (BioME), Federal University of Rio Grande do Norte, Natal, 59078-970, RN, Brazil.
[3*]Department of Computer Engineering and Automation (DCA), Federal University of Rio Grande do Norte, Natal, 59078-970, RN, Brazil.

*Corresponding author(s). E-mail(s): mfernandes@dca.ufrn.br;

Abstract

Purpose: The primary objective of this study was to develop and evaluate a deep neural network model based on convolutional neural networks (CNNs) for accurately classifying SARS-CoV-2 viral sequences and other subtypes within the Coronaviridae family. With the rapid evolution of viral genomes and the increasing need for timely classification, we aimed to provide a robust and efficient tool that could enhance the accuracy of viral identification and classification processes. By harnessing the power of deep learning, we sought to contribute to advancing viral genomics research and aid in surveilling emerging viral strains.
Methods: We designed and implemented a CNN-based deep neural network architecture capable of processing complete cDNA genomic sequences to achieve our goal. We used a dataset comprising diverse viral subtypes, including SARS-CoV-2, for training and testing. The dataset was partitioned using a 5-fold cross-validation strategy to ensure rigorous evaluation. Our model's performance was assessed using various metrics, including accuracy, precision, sensitivity, specificity, F1-score, and AUROC. Additionally, artificial mutation tests were conducted to evaluate the model's generalization ability across sequence variations. We also used the BLAST algorithm and conducted comprehensive processing time analyses for comparison.
Results: The developed CNN-based model demonstrated exceptional performance across various evaluation metrics. In the training phase, the model consistently achieved maximum values for accuracy, sensitivity, specificity, and

other key metrics, indicating its robust learning ability. Notably, during testing on over 10,000 viral sequences, the model exhibited a sensitivity of over 99% for sequences with fewer than 2,000 mutations. The CNN-based model showcased superior accuracy and significantly reduced processing times compared to the BLAST algorithm. These findings underscore the model's effectiveness in accurately classifying viral sequences and its potential to revolutionize viral genomics research.

Conclusion: This study introduces a CNN-based deep neural network model as a powerful tool for precisely classifying viral sequences, specifically focusing on SARS-CoV-2 and other Coronaviridae family subtypes. Our model's superiority is evident through rigorous evaluation and comparison with existing methods, offering enhanced accuracy and efficiency. The application of artificial mutation testing demonstrated the model's robustness in handling sequence variations. By harnessing deep learning capabilities, our model significantly contributes to viral classification and genomics research. As viral surveillance becomes increasingly critical, our model holds promise in aiding rapid and accurate identification of emerging viral strains.

Keywords: SARS-CoV-2, COVID-19, viral classification, deep learning

1 Introduction

One particular virus has made of attention of the entire world, the SARS-CoV-2. The virus belongs to the family Coronaviridae, which contains one of the largest viral genomes, ranging from 26,000 base pairs (bp) to 31,700 bp [1]. The SARS-CoV-2 causes the COVID-19 disease, which has caused the death of thousands of people worldwide due to its high virulence rate in conjunction with your rapid spread. [2, 3]. The novel and timely classification systems are necessary for more insights into the evolution of underlying mechanisms of increased epidemicity and enhanced virulence compared to related lineages [4, 5].

Viral classification is a task largely applied for many scientists around the world. This activity assigns a certain sequence to a specific group based on known genomic sequences which share common characteristics and traits [6]. The conventional methods for characteristics extraction of the virus are based on sequence alignment [7, 8]. Alignment-based techniques search for regions of similarity between biological sequences from a previously characterized reference sequence. These techniques can also be used for viral identification [6]. Alignment-based techniques are used in algorithms like BLAST (Basic Local Alignment Search Tool) [9], MALT (Megan alignment tool) [10], FASTP (FASTQ preprocessor) [11], ClustalW [12] and USEARCH [13]. However, these methods have some limitations: low accuracy and limited genomic sequence length used. The use of long genomic sequences implies a high computational cost due to the nature of the problem [7, 14]. Works presented in [6] and [7] draw attention to the evidence that alignment-based methods are not quite satisfactory when applied to genomes susceptible to large genetic variations, which is the case of the vast majority of the viruses. In order to minimize these problems, free-alignment

(FA) techniques emerged, which are based on features from linear algebra, information theory and statistical mechanics to calculate the similarity or distance between sequences [6, 7].

According to [6, 15, 16], to provide the best results, the viral classification based on free-alignment algorithms uses the artificial intelligence approach based on machine learning (ML) techniques to perform the feature extraction of the genomic sequences. Recent studies indicate that ML algorithms and techniques have been widely used in research related to genomics, including viral classification, for offering a set of methods capable of identifying highly complex patterns in an automated, efficient way and with the minimal human intervention [17, 18]. Works in the literature show that machine learning based on Deep Learning (DL) techniques provides excellent results for genomic sequences applications, including classification problems [19, 20].

Mottaqi [18] and Lalmuanawma [21] show that among many ML algorithms, the Convolutional Neural Networks (CNN) have been frequently used for data analysis based on genomic sequence for their ability to extract intrinsic characteristics of the sequences and present promising results in their applications. However, most of these tools and techniques use genomic sequences of limited length or are aimed at other purposes such as protein prediction [22, 23].

Fabijańska proposes a deep viral genome classifier, named VGDC (Viral Genome Deep Classifier), able to identify viral subtypes from different families such as dengue, hepatitis B and C, HIV-1, and influenza A presented F1-score between 0.85 and 1 [24]. Tampuu *et al.* presented an architecture to recognize the presence of viruses by the raw metagenomic contigs of various human samples. The methodology proposed was named ViraMiner and made use of two CNNs. They reached a Receiver Operating Characteristic (ROC) curve of 0.923[25].

The work presented by Whata *et al.* used a CNN and a Bi-LSTM (bi-directional long short-term memory), which he called CNN-Bi-LSTM (convolutional neural network bidirectional long short-term memory). This model achieved a classification accuracy of 99.95%, AUC of 100.00%, specificity of 99.97%, and sensitivity of 99.97% as from 34 sequences from the SARS-CoV-2 virus and 295 samples from other viruses of the same family [26].

The study presented by Adetiba *et al.* used a CNN to perform a multiclass classification of genomic sequences of three viral subtypes, MERS-CoV (Middle East Respiratory Syndrome CoV), SARS-CoV (Severe Acute Respiratory Syndrome CoV), and SARS-CoV -2 (Severe Acute Respiratory Syndrome Coronavirus 2). The authors used the GSP (Genomic Signal Processing) technique to transform the genomic sequences into RGB images and later applied them to a CNN, using only 300 samples for training . The model obtained an accuracy of 95% for MERS-CoV, 95% for SARS-CoV, and 95% for SARS-CoV-2, titled by the authors DeepCOVID-19 [27].

Classification between SARS-CoV-2, MERS-CoV, SARS-CoV, hepatitis-A, dengue, and influenza was proposed by Gunasekaran *at al.*. Therefore, the authors use the CNN, CNN-LSTM, and CNN-Bidirectional LSTM architectures with k-mers to verify which architectures present better performance. According to the tests performed, it was observed that CNN and CNN-Bidirectional LSTM with k-mers offered the highest accuracy metrics, reaching 93.16% and 93.13%, respectively [28]. A neural

network called miRNA proposed by Lopez-Rincon *et al.* was applied at viral classification. The architecture has a few layers and was also used to classify viruses from the Coronaviridae family. This model showed an accuracy of 98%, specificity of 0.9939, and sensitivity of 1.00 [20].

Several viral genomic sequences of different sizes were analyzed by [29], which used the area under the receiver operating characteristic (AUROC) as their performance metric. The research obtained AUROC values of 0.95, 0.93, 0.97, and 0.98, for the genomic sizes 300, 500, 1000, and 3000 bp, respectively. The architecture used was called DeepVirFinder and consists of a CNN of multiple layers [29].

Given this context, the present work aims to present a technique capable of classifying the Coronaviridae family's viruses and recognizing the SARS-Cov-2 virus. That approach uses the CNN that receives complete genomic sequences of cDNA as input, codified by the one-hot-encoding technique. Thus, this work makes the following specific contributions:

- Develop an alignment-free method to classify SARS-CoV-2 sequences between viruses of the same family.
- Develop a deep learning algorithm that can efficiently classify from the complete cDNA sequences of the virus.
- Comparison of performance of the proposed model with the BLAST algorithm with the number of samples found or correctly classified and the processing time both tools took to present their results.

2 Results

2.1 Training and validation

As mentioned in Section 3.1, the dataset used for training the network comprises 501 samples referring to the Non-SARS group and receiving label 0 and 501 samples from SARS, in which they obtained label 1. In this way, we obtained a training set balanced and homogeneous consisting of 1,002 samples. Cross-validation was used to train and validate the classification model (see Section 3.2). The performance metrics for the k-fold ($k = 5$) cross-validation corresponded to the average between all the values obtained in each fold. The classification results of validation (after training) were presented through the confusion matrix (see Figure 1), the ROC (see Figure 2), and measured by the sensitivity, specificity, precision, accuracy, and F1-score metrics (see Table 1). As a result, the model results in maximum performance values for the training and validation sets, as shown in Table 1.

Figure 1 presents the results of the mean classification of the samples referring to the validation set (SARS-Cov-2 and Not SARS-Cov-2) and shows that for all subsets, all sequences were correctly grouped according to their respective class. The ROC curve for this problem is shown in Figure 2 and presents sensitivity and specificity values equal to 100%, according to Table 1.

Figures 3 and 4 illustrate the training and validation learning curve for accuracy and loss, respectively. Each iteration point represents the mean and standard deviations of the 5-fold cross-validation. The accuracy learning curve of training and

Table 1 Performance metrics results for the classification of SARS-Cov-2 from the architecture proposed in this work for the validation set.

Metrics	Performance
Sensitivity	100%
Specificity	100%
Precision	100%
Accuracy	100%
F1-score	100%

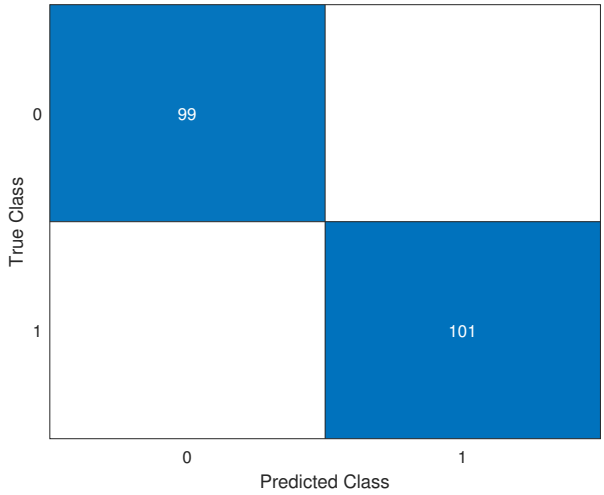

Fig. 1 Confusion matrix of the proposed approach to the classification problem of being SARS-CoV-2 and Non SARS-CoV-2. Non SARS-CoV-2 samples are represented by label 0, and SARS-CoV-2 samples are represented by label 1. The model is able to correctly classify all samples according to their classes.

validation (see 3) corroborates with the results presented in Table 1, and these curves show that the model does not suffer from overfitting (high variance) or underfitting (high bias). Furthermore, the reduced difference (almost zero) between the training and validation curves consolidates the absence of overfitting. The training was concluded after 10 epochs with 72 iterations, as shown in Figures 3 and 4. It is observed that the error was stabilized after the 30th iteration (see Figure 4).

2.2 SARS-Cov-2 prediction tests

Similar to the methodology used in [14], two tests were performed to evaluate the SARS-Cov-2 prediction of the proposed deep learning model after training. The tests were composed of samples not used in the training stage, that is, samples that remained from the initial dataset belonging to the SARS-CoV-2 virus (see Section 3.3). The tests, called Prediction test 1 and Prediction test 2, are described below.

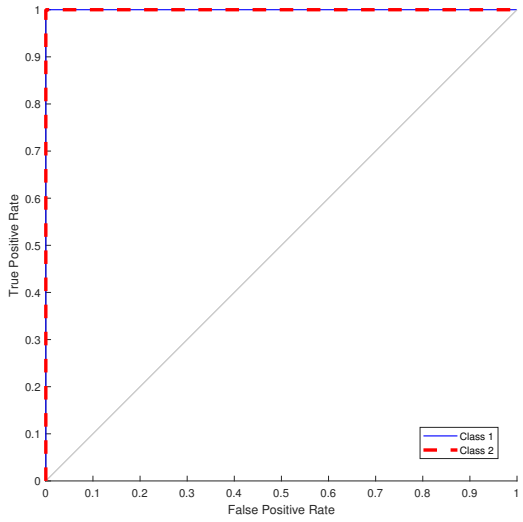

Fig. 2 ROC curve for classification of SARS-CoV-2 and Non SARS-CoV-2.

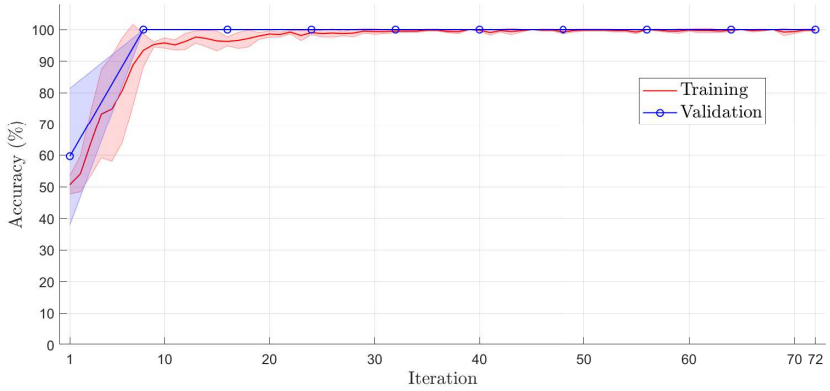

Fig. 3 The learning curve of training and validation accuracy of the training set using 5-fold cross-validation.

2.2.1 Prediction test 1

Of the remaining 16,891 SARS-CoV-2 samples from the initial dataset, 12,000 were randomly chosen to compose this experiment. These samples obtained label 1 indicating that they were SARS-CoV-2. The objective of this experiment was to test the model for identifying SARS-CoV-2.

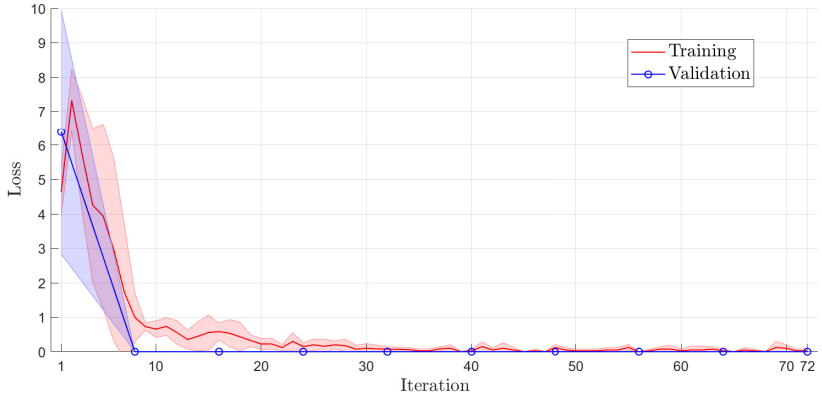

Fig. 4 The learning curve of training and validation loss of the training set using 5-fold cross-validation.

2.2.2 Prediction test 2

For this experiment, 10,000 samples of SARS-CoV-2 were used (of the remaining 16,891 SARS-CoV-2 samples from the initial dataset), in which they were divided into two groups, each with 5,000 samples. In one of these groups, we applied the artificial mutation method discussed in Subsection 3.4 to investigate the architecture's sensitivity and robustness to possible mutations in the SARS-CoV-2 virus. In this way, a group was created with 5,000 samples of the SARS-CoV-2 virus, which suffered artificial mutations, and another group, also with 5,000 samples, which did not undergo any mutation. The artificial mutation strategy used $V_{max} = 31{,}029$ and $\gamma = 5\%$, i.e., $N_{\text{mut}} = 1{,}551$ nucleotides have changed per sequence.

2.2.3 Prediction test results

The results of Prediction tests 1 and 2 are shown in Table 2. For prediction test 1, 11,996 were correctly classified to their respective group (SARS-CoV-2), and only 4 samples were not classified correctly, reaching 99.99%, 100%, 99, 94%, and 99, 96% for the sensitivity, precision, F1-score, and accuracy, respectively. As described above, prediction test 2 verified the ability of the trained model to classify SARS-CoV-2 samples even after changing their genomic structure through the artificial mutation technique in half of the dataset samples. Even applying modifications to the sequences, the model is quite sensitive to possible mutations that the sequences may suffer, reaching a sensitivity value of 99.77%. This result strongly attests to the model's ability to generalize, given that, even with the samples changing, the network can identify who is SARS-CoV-2 through low false negative results (accuracy about 99.96%).

The results obtained through the experiments carried out and detailed in Section 3.3, are promising, consistent with the performance obtained in the network training phase. Furthermore, the sensitivity and precision values derived from the set of

Table 2 Results associated with prediction tests 1 and 2.

Pt	Sensitivity	Precision	F1-Score	Accuracy
Pt-1	99.99%	100%	99.94%	99.96%
Pt-2	99.77%	100%	99.88%	99.96%

experiments remain high regardless of the class labels, which is very important, considering that high rates of false negatives directly corroborate the increase in infected people. Finally, the proposed model's characteristics and results will be compared and discussed with works found in the literature below.

3 Methods

3.1 Database and data balancing

The National Genomics Data Center (NGDC) provides open and free access to a set of database resources that have the resources of the New Coronavirus 2019 Data Resource - 2019nCoVR. The 2019nCoV maintains daily updates and brings together a comprehensive collection of genomic sequences and clinical information, not only about SARS-CoV-2 but also regarding other viruses that belong to the coronaviridae family worldwide and from other traditional repositories, such as the National Center for Biotechnology Information - NCBI [30]. The 2019nCoV was the chosen repository to download the dataset. Sequences belonging to the coronaviridae family were selected, whose size ranges from 25,000 bp to 35,000 bp, covering the size of all viruses in the family without losing any crucial genetic information. The selected host was the Homo Sapiens. The download of the dataset used in this research was carried out in August 2020, when the variants of concern were not yet available.

The database used is formed by 17,893 genomic sequences of nine types of viruses of the coronaviridae family, coming from 62 different countries. Figure 5 shows all countries with genomic samples on the database. It is observed that the United States has the highest number of sequences, followed by Australia, India, and China. From the 17,893 samples, 17,392 belong to the SARS-CoV-2 virus 97.2% of all), of which 11,140 are coming from the United States (62.25% of all).

The data used for viral classification are cDNA sequences, whose length varies from 26,342 bp to 31,029 bp. Table 3 summarizes some properties related to viral subtypes present in the database. The BetaCoronaVirus shows the most extensive sequence length among all virus subtypes, varying between 31,029 bp and 30,536 bp. In addition to having the same sequence length (30,499 bp), the *CoronaVirus cya-BetaCov/2019*, *CoronaVirus cyb-BetaCov/2019*, and *CoronaVirus cyc-BetaCov/2019* are the viruses that have the smallest amount of samples in the database. They are long genomic samples and very similar viruses, so a robust model is required to provide the appropriate classification [24].

As shown in Table 3, the largest amount of samples in the database belong to the SARS-CoV-2 virus, which causes the COVID-19 disease, followed by the MERS-CoV virus. In this context, it was necessary to balance the data to improve the network's

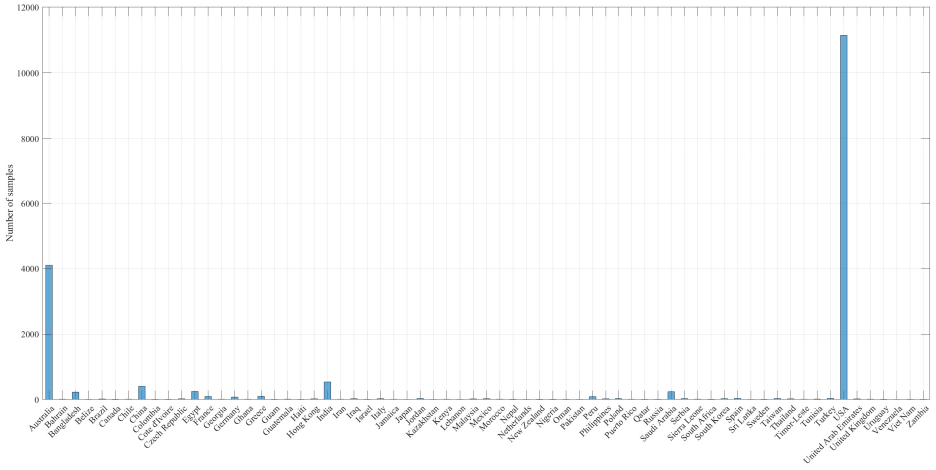

Fig. 5 Countries that contain genomic samples of the Coronaviridae family in the database.

Table 3 Viral subtypes on the database created for this work.

Virus	Number of samples	Minimum sequence length	Maximum sequence length
BetaCoronaVirus	140	30,536	31,029
CoronaVirus cya-BetaCov/2019	1	30,499	30,499
CoronaVirus cyb-BetaCov/2019	1	30,499	30,499
CoronaVirus cyc-BetaCov/2019	1	30,499	30,499
HCoV-229E	27	26,592	27,307
HCoV-HKU11	18	29,367	29,983
HCoV-NL63	55	27,302	27,832
MERS-CoV	258	29,267	30,150
SARS-CoV-2	17,392	26,342	28,784

performance and avoid problems such as Overfitting due to the disproportion of samples from the other viruses.

The dataset was divided into two groups: non SARS-CoV-2 and SARS-CoV-2, as illustrated in Figure 6. The non SARS-CoV-2 group comprises eight viral subtypes different from the SARS-CoV-2 virus, totaling 501 samples. Therefore, 501 samples were taken from all countries that presented genomic sequences of the SARS-CoV-2 virus randomly and uniformly, guaranteeing diversity and representativeness of each viral subtype in the training and validation sets, as illustrated in Figure 7. The dataset used for the training and validation phases contains 1,002 samples in total. The samples were labeled by 0 and 1, where 0 is associated with the non SARS-CoV-2 samples, and 1 is related to the SARS-CoV-2 samples. Part of the remaining genomic samples was used to test the performance of the network.

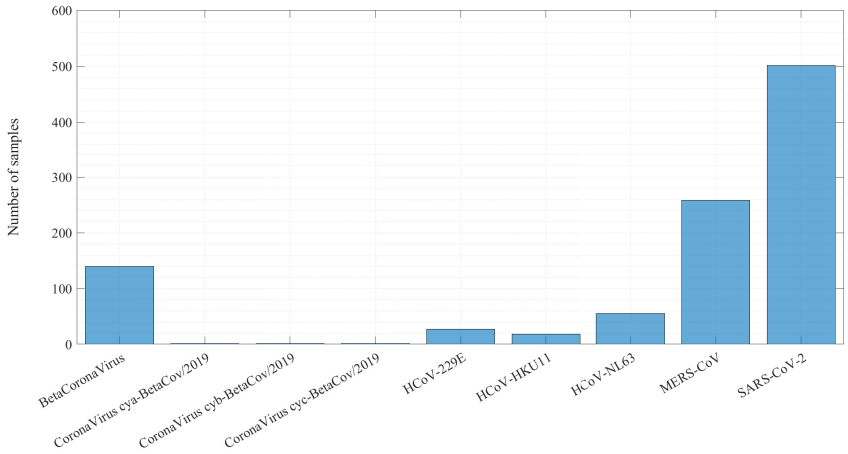

Fig. 6 Dataset of all viral subtypes after the data balancing process.

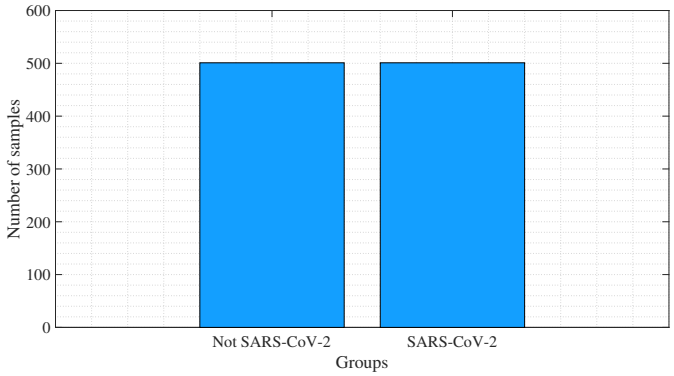

Fig. 7 Dataset after balancing the samples according to their groups.

3.2 CNN architecture and parameters

Based on the length of the sequences in the database presented in Table 3, it appears that the most prolonged sequences correspond to BetaCoronaVirus. Therefore, all genomic sequences will have the same length ($N_{\max}$ = 31,029) to be processed by CNN. Then, for each m-th sample, the CNN receives as entry 5 channels of dimension $31{,}029 \times 1$. As described in Section 3.3, this strategy allows all M viral sequences have the same length.

The CNN used in this work comprises twenty-six layers, divided into 1D convolutional layers and fully connected layers. The 1D convolutional layers are responsible for extracting characteristics of the cDNA genomic sequences, and the fully connected layers are responsible for classifying the data extracted from the previous layers, generating a total of 14,545,426 parameters across all layers, as shown in Table 4. Figure

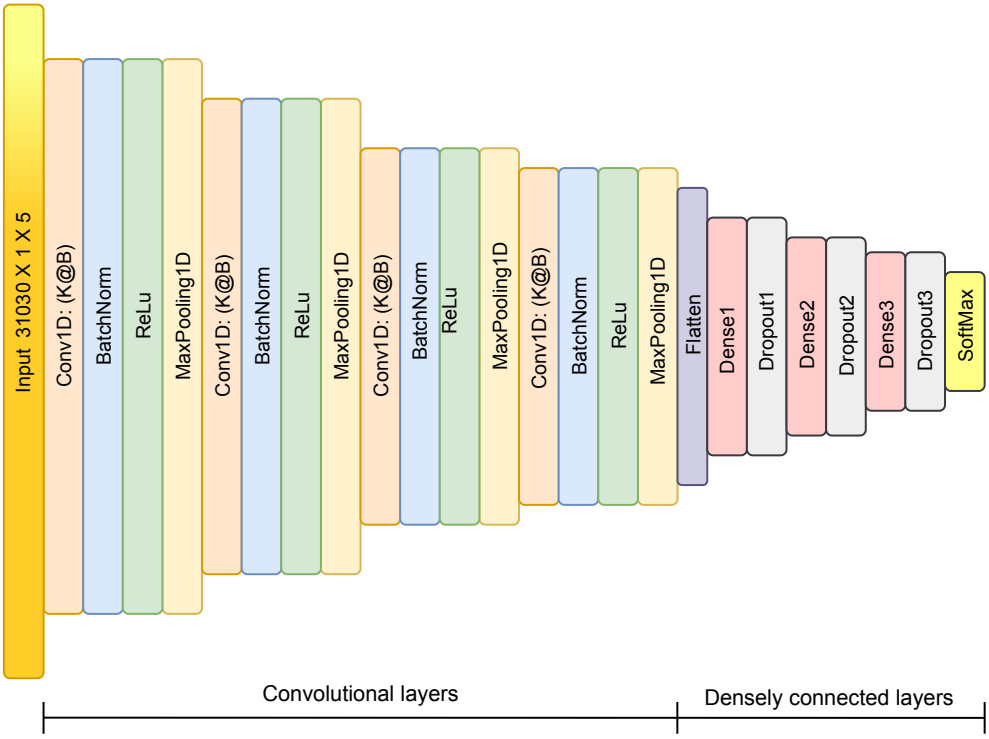

Fig. 8 CNN used for the viral classifier proposal presented in this work.

8 details the CNN architecture used in the appropriate viral classifier for the database described in Section 3.1.

The CNN comprises four convolutional layers, followed by a normalization layer and the activation function ReLu (Rectified Linear Unit). The MaxPool function is applied after each activation layer, with windows ranging in size from 8, 16, 32 and 64. In addition to the convolutional layers, the CNN structure contains four fully connected layers with 64, 32, 16, and 2 neurons, respectively. The number of neurons in the last layer corresponds to the number of classes to be classified, followed by the softmax function that will output the probability that each sequence belongs to a specific class.

The cross-validation k-fold was used to evaluate the proposed model, where k refers to the number of subsets, or folds, into which the dataset will be divided. We defined the value of $k = 5$ so that the dataset will be divided into five subsets, each fold containing 201 samples. In the cross-validation scheme, $k - 1$-folds are used for model training (801 samples), and 1-fold is used for model validation (201 samples), totaling 1,002 samples.The optimizer chosen for updating the network weights was the adam (Adaptive Moment Estimation), whose learning rate was 0.001 (see Table 5). An optimizer is a function that aims to reduce the error between the results obtained by a model concerning the desired results. Among the various optimizers, adam is one of

Table 4 CNN architecture used in this work with four convolutional layers and four fully connected layers.

Layers	Description	Values
1	Input ($L \times 1 \times 5$)	$N = 31{,}030$
2	Conv1d (K_1@B_1)	$K_1 = 256$ and $B_1 = 8$
3	BatchNorm	-
4	ReLU	-
5	MaxPool1D (P_s)	$P_s = 8$
6	Conv1D (K_2@B_2)	$K_2 = 64$ and $B_2 = 16$
7	BatchNorm	-
8	ReLU	-
9	MaxPool1D (P_s)	$P_s = 16$
10	Conv1D (K_3@B_3)	$K_3 = 32$ and $B_3 = 8$
11	BatchNorm	
12	ReLU	-
13	MaxPool1D (P_s)	$P_s = 32$
14	Conv1D (K_4@B_4)	$K_4 = 32$ and $B_4 = 64$
15	BatchNorm	-
16	ReLU	-
17	MaxPool1D (P_s)	$P_s = 64$
18	Flatten	-
19	Dense1 (P_1)	$P_1 = 64$
20	Dropout (a_1)	$a_1 = 0.4$
21	Dense2 (P_2)	$P_2 = 32$
22	Dropout (a_2)	$a_2 = 0.4$
23	Dense3 (P_3)	$P_3 = 16$
24	Dropout (a_3)	$a_3 = 0.4$
25	Dense4 (P_4)	$P_4 = 2$
26	Softmax	2 Classes

the most used in the literature, especially in deep learning. This optimizer is indicated in problems that involve a large amount of data or parameters because it is easy to implement, has a low computational cost, and requires a low amount of memory. [31]. The training converged in approximately 10 epochs. Given the nature of the problem and through tests and works found in the literature, a mini-batch of size 128 was applied due to the number of samples and training parameters as recommended in [24]. The parameters used in the architecture training phase are shown in Table 5. A mini-batch of 128 was used based on the long length of the viral genomes and the large number of samples used to train the model. Other parameters were adjusted to decrease the training time and the loss function as recommended in [16, 20, 24]. The training converged in approximately 10 epochs with 72 iterations (see Figures 3 and 4 in Subsection 2.1).

Table 5 Hyperparameters used in the training phase of the proposed architecture.

Hyperparameters	Values
Mini-Batches	128
MaxEpochs	12
InitialLearnRate	0.001
Optimizer	Adam

3.3 Pre-processing and data mapping

The methodology used in this work can be divided into two stages: 1) pre-processing and data mapping; 2) methods to verify and test the model's generalization. For CNN to perform feature extraction and classification, it is necessary to pre-process the data, which involves converting the nucleotides of the genomic sequences, represented by the characters (A, C, G, T, N), into numerical data, precisely ones and zeros. Once encoded, the data will be mapped into vectors of a dimension and depth of 5, using the one-hot-encode technique to be presented to CNN, indicating whether or not it is SARS-CoV-2.

The Figure 9 illustrates the overview of the technique proposed in this work. Considering a database with M samples of DNAc viral sequences, each m-th sample, $\mathbf{s}_m$ is mapped in a characteristic matrix, $\mathbf{S}_m$, that will be processed by the CNN. The CNN provides a binary classification in which the SARS-CoV-2 will be identified or not.

Each m-th sample of viral sequence de entrada is expressed by

$$\mathbf{s}_m = [s_{1,m}, \ldots, s_{N_m,m}] \tag{1}$$

where each i-th element of a m-th sample, $s_{i,m}$ represents a possible nucleotide of a set $S \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, and N_m is the length of the m-th viral sequence sample. Each element of S corresponds to one of the nitrogenous bases Adenine (A), Cytosine (C), Guanine (G) and Thymine (T).

The characteristic matrix associated with the m-th sample, $\mathbf{s}_m$, is constructed by the one-hot encode technique, which can be expressed as

$$\mathbf{S}_m = \begin{bmatrix} a_{1,1,m} & \cdots & a_{1,5,m} \\ \vdots & \ddots & \vdots \\ a_{N_{\max},1,m} & \cdots & a_{N_{\max},5,m} \end{bmatrix} \tag{2}$$

where

$$a_{i,j,m} = \begin{cases} 1 & \text{for } j = 1 \ \& \ s_{i,m} = \mathrm{A} \\ 1 & \text{for } j = 2 \ \& \ s_{i,m} = \mathrm{C} \\ 1 & \text{for } j = 3 \ \& \ s_{i,m} = \mathrm{G} \\ 1 & \text{for } j = 4 \ \& \ s_{i,m} = \mathrm{T} \\ 0 & \text{for } \forall j \ \& \ s_{i,m} \notin S \end{cases} \tag{3}$$

and $N_{\max}$ is the size of the largest sequence among all the M viral sequence samples, that is, $N_{\max} = \max\{N_1, \ldots, N_M\}$. So, the characteristic matrix has the same

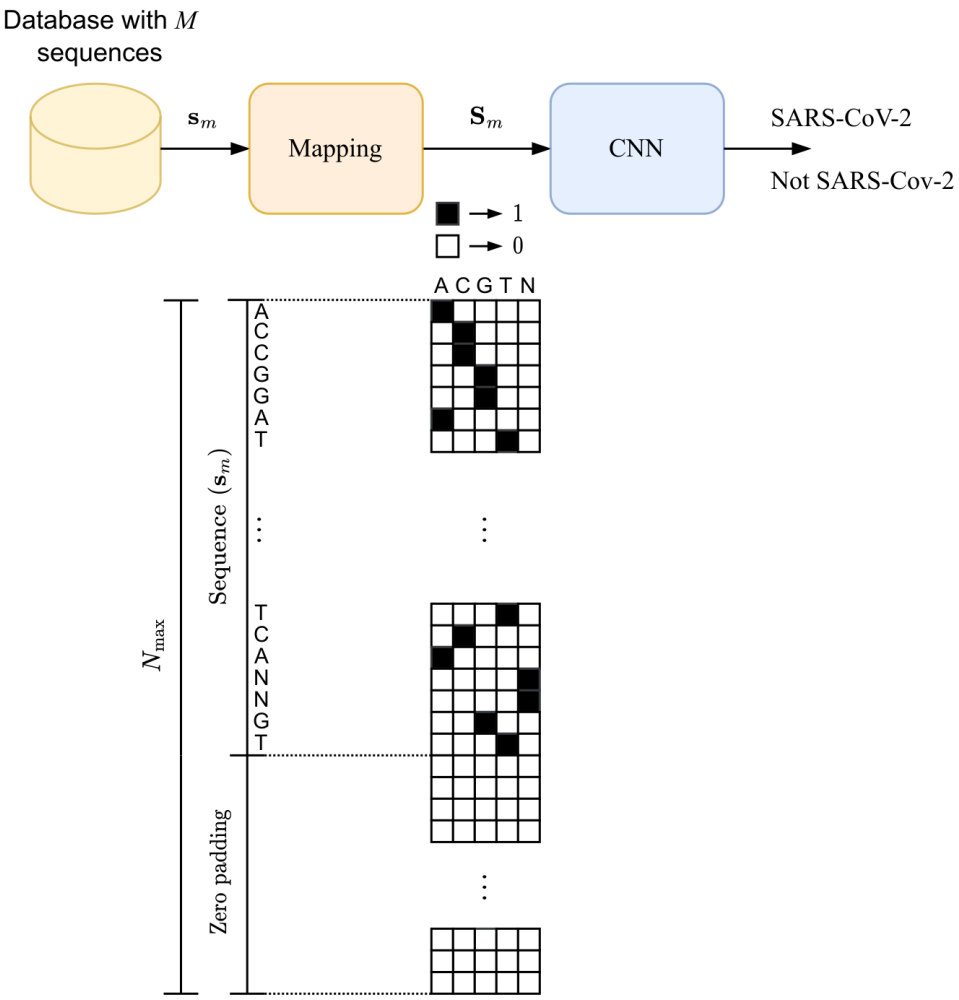

Fig. 9 Overview of the proposed technique.

dimension ($N_{\max} \times 5$) for all the M samples of viral sequences. If the size of the m-th sequence is less than the maximum sequence ($N_m < N_{\max}$), $N_{\max} - N_m$ zeros are inserted (zero padding).

Before entering into the CNN, the characteristic matrix of each m-th sample, $\mathbf{S}_m$, is transformed into a matrix of dimension $N_{\max} \times 1 \times 5$, expressed as

$$\mathbf{B}_m = \left[\mathbf{b}_{1,m} \ \ldots \ \mathbf{b}_{5,m} \right] \tag{4}$$

where

$$\mathbf{b}_{j,m} = \begin{bmatrix} b_{1,1,j,m} \\ \vdots \\ b_{N_{\max},1,j,m} \end{bmatrix} \tag{5}$$

which $b_{i,1,j,m} = a_{i,j,m}$. This transformation allows the CNN to process each mth sequence as an input formed by 5 channels of dimension vectors ($N_{\max} \times 1$), $\mathbf{b}_{j,m}$.

3.4 Artificial Mutation Technique

The artificial mutation process is initiated by searching for the maximum sequence length among the samples. So, for the set H of samples, $V_{max} = \max\{N_1, \ldots, N_H\}$, where N_i is the length of the sequences and V_{max} is the length of the most extensive sequence. After this step, the insertion of zeros is performed in each i-th sequence, s_i, where $N_i < V_{max}$. Each i-th sequence is completed with zeros until filling the value of V_{max}, i.e., the amount of zeros entered for the i-th sequence is $V_{max} - N_i$. After that, all the chosen H samples will have the same size, V_{max}. The artificial position mutation rate, γ, is defined at the end of this step. The value of γ establishes the percentage of the number of nucleotides positions that will change, N_{mut}, which can be expressed as

$$N_{\text{mut}} = \left\lfloor \frac{\gamma \times V_{max}}{100} \right\rfloor . \tag{6}$$

After the definition of the N_{mut}, the position of the N_{mut} nucleotides that will be changed is randomly defined, which is stored in the vector $\mathbf{k}_{\text{mut}} = [k_1, \ldots, k_{N_{\text{mut}}}]$. From the position vector, $\mathbf{k}_{\text{mut}}$, two methods are applied to change the selected nucleotides for artificial mutation. The first method was applied to the first half of the selected nucleotides, i.e., the positions $\left[k_1, \ldots, k_{N_{\text{mut}}/2}\right]$, and the second method was used for the second half of the position vector $\left[k_{N_{\text{mut}}/2+1}, \ldots, k_{N_{\text{mut}}}\right]$.

The first method changes the position of the nucleotides, considering the pairs, i.e.

$$\begin{gathered} \left[k_1, k_2, \ldots, k_{N_{\text{mut}}/2-1}, k_{N_{\text{mut}}/2}\right] \Rightarrow \\ \left[k_2, k_1, \ldots, k_{N_{\text{mut}}/2}, k_{N_{\text{mut}}/2-1}\right] \end{gathered} . \tag{7}$$

Moreover, the second method changes nucleotide values to

$$s_{k_i} = \begin{cases} \text{A} & \text{if } s_{k_i} = \text{T} \\ \text{T} & \text{if } s_{k_i} = \text{A} \\ \text{C} & \text{if } s_{k_i} = \text{G} \\ \text{G} & \text{if } s_{k_i} = \text{C} \\ \text{N} & \text{if } s_{k_i} = \text{T} \end{cases} . \tag{8}$$

4 Discussion

4.1 Blast comparison

The strategy proposed in this work was compared with the BLAST algorithm. The comparison obtained results associated with the correctness rate in the classification of sequences through various values of artificial position mutation rate (see section 3.4) and the average processing time to classify these sequences. In the comparison, 34 sequences belonging to the Coronaviridae family were used (17 SARS-Cov-2 and 17 Not SARS-Cov-2) that did not participate in the deep learning training.

The BLAST software version 2.13.0 made available by the NCBI [30] was downloaded and installed locally. The BLAST software used a database of 6,180,834 Betacoronavirus sequences (updated Sep 8, 2022) found in [30]. The database was also downloaded for local use. Using the BLAST software locally, accessing a local database allows a fairer comparison in terms of processing time with the deep learning strategy proposed in this work. The same computer used to run BLAST with its database was also used to train and run the CNN strategy. The computer has the following configurations: Intel(R) core(TM) i7-10700 CPU 2.9 GHz, 128 GBytes of RAM, 512 GBytes NVMe HD and an NVIDIA GeForce RTX 3060 GPU with 12 GBytes of RAM.

Figure 10 presents the relationship between the artificial position mutation rate (see section 3.4) applied in the 34 test sequences and the correctness rate (in percentage terms) of both the BLAST and the proposed CNN. It is possible to observe that up to $\gamma \approx 2\%$ ($N_{\text{mut}} \approx 620$ nucleotides), the correctness rate for BLAST and CNN-based strategy is the same, that is, 100%. However, for values of $\gamma > 2\%$, the correctness rate of BLAST drops rapidly to 50%, in which $\gamma \approx 19\%$ ($N_{\text{mut}} \approx 5{,}895$ nucleotides). On the other hand, the proposal based on CNN has a correctness rate of 100% up to $\gamma \approx 13\%$ ($N_{\text{mut}} \approx 4{,}033$ nucleotides) and decays more slowly than BLAST, with $\gamma > 13\%$. For $\gamma \approx 19\%$, a proposal based on CNN has a correctness rate of around 95.88% and BLAST around 50%. For values of γ between $\approx 32\%$ ($N_{\text{mut}} \approx 9{,}929$ nucleotides) and $\approx 45\%$ ($N_{\text{mut}} \approx 13{,}963$ nucleotides), the correctness rate of BLAST rapidly decays to zero while the proposal with CNN decays more slowly to 50%. Table 6 presents the values of correctness rate, artificial position mutation rate, γ, and the number of nucleotides that mutated, N_{mut}, for each point in the graphs shown in Figure 10.

Table 7 presents the average processing time obtained for BLAST and CNN at each point presented in the graphs in Figure 10. The data presented for CNN are the time required to perform the inference of the 34 test sequences, given that the training is performed only once. However, the time for training the CNN was approximately 341 seconds (around 6 minutes). It is possible to observe that CNN has a constant processing time while BLAST has a variable processing time that depends on the value of γ.

For sequences with many mutations, $\gamma > 25.78$ ($N_{\text{mut}} > 8{,}000$), BLAST has a faster response (shorter processing time) than for sequences with few mutations $\gamma < 3.22$ ($N_{\text{mut}} < 1{,}000$). Sequences with many mutations allow BLAST to reduce the search space due to the high dissimilarity between the query sequence and the sequences

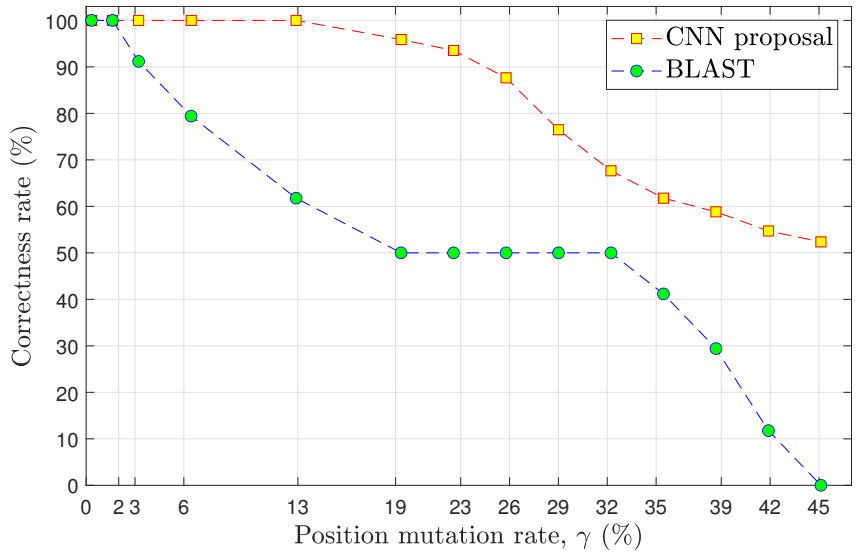

Fig. 10 Comparison of the correctness rate between BLAST and CNN (proposed in this work) for a test set of 34 sequences according to the increase of the artificial position mutation rate, γ.

Table 6 Values of correctness rate, artificial position mutation rate, γ, and the number of nucleotides that mutated, N_{mut}, for each point in the graphs shown in Figure 10.

γ (%)	N_{mut}	BLAST Correctness rate (%)	CNN Correctness rate (%)
0.32	100	100.00	100.00
1.61	500	100.00	100.00
3.22	1,000	91.18	100.00
6.45	2,000	79.41	100.00
12.89	4,000	61.76	100.00
19.34	6,000	50.00	95.88
22.56	7,000	50.00	93.53
25.78	8,000	50.00	87.65
29.01	9,000	50.00	76.47
32.23	10,000	50.00	67.65
35.45	11,000	41.18	61.76
38.67	12,000	29.41	58.82
41.90	13,000	11.76	54.71
45.12	14,000	0.00	52.35

stored in the base. On the other hand, when the value of g decreases, the BLAST processing time increases to obtain a better similarity value between the query sequence and the sequences stored in the base.

Table 7 Time processing, artificial position mutation rate, γ, and the number of nucleotides that mutated, N_{mut}, for each point in the graphs shown in Figure 10.

γ (%)	N_{mut}	BLAST	CNN
		Time processing (seconds)	Time processing (seconds)
0.32	100	94,261.48 ($\approx$ 26.2 hours)	0.33
1.61	500	94,261.48 ($\approx$ 26.2 hours)	0.35
3.22	1,000	93,202.74 ($\approx$ 25.9 hours)	0.39
6.45	2,000	92,172.83 ($\approx$ 25.6 hours)	0.45
12.89	4,000	91,176.66 ($\approx$ 25.3 hours)	0.68
19.34	6,000	64,122.58 ($\approx$ 17.8 hours)	0.68
22.56	7,000	24,587.36 ($\approx$ 6.8 hours)	0.68
25.78	8,000	68,17.63 ($\approx$ 1.9 hours)	0.68
29.01	9,000	44,35.43 ($\approx$ 1.2 hours)	0.68
32.23	10,000	2,155.14 ($\approx$ 0.6 hours)	0.68
35.45	11,000	1,940.66 ($\approx$ 0.5 hours)	0.68
38.67	12,000	1,831.33 ($\approx$ 0.5 hours)	0.69
41.90	13,000	1,832.6 ($\approx$ 0.5 hours)	0.69
45.12	14,000	1,801.26 ($\approx$ 0.5 hours)	0.68

The gain in CNN processing time over BLAST is significant, being around 2,600 times faster for $\gamma = 45.12\%$ ($N_{\text{mut}} = 14{,}000$) and 130,000 times faster for $\gamma = 0.32\%$ ($N_{\text{mut}} = 100$). It is essential to point out that BLAST needs a database of sequences already stored to find or classify the viral genome, and with this, it needs to carry out a search procedure which can take a long time. CNN stores the information needed to classify the viral genome in its models after the training process. After training, the CNN performs only a simple inference process, not needing to perform a search and a database.

The proposed CNN model can be an excellent alternative and ally in the rapid virus classification process, given its high sensitivity in detecting changes in the virus structure (represented by random mutations in its nucleotides), corroborating SARS-Cov-2 surveillance. In addition, this model enables the analysis of more significant amounts of complete genomic samples, at a lower computational cost, compared to techniques that use alignment and even BLAST.

4.2 State of the art comparison

The tables 9 and 8 summarize a set of approaches from the main works found in the literature that perform viral classification using CNNs presented in this work. Characteristics such as number of layers and size of genomic sequences will be presented in the Table 8.

When applying longer sequences, the works presented in [24], [25], and [29] had a considerable reduction in the performance of their models. This point implied the use of more extensive networks as in [24] and the reduction of sequence sizes as in works [25] and [29].

Regarding [20], despite making use of complete genomic sequences and presenting a smaller number of layers, the author makes use of a small dataset for the training and

Table 8 Comparison from the proposed architecture with related works.

References	Codification	Layers	Sequence length
Fabijańska e Grabowski [24]	ASCII	30	3,257-24,751 bp
Ren {et al.}[29]	One-Hot Encoded	6	150-3,000 bp
Tampuu{et al.}[25]	One-Hot Encoded	2 CNNs with 7 layers each	300 bp
Lopez-Rincon{et al.}[20]	Assigned values from 0 to 1 to the channels	10	31,029
Proposed Architecture	One-Hot Encoded	26	31,029

validation of his model, which may lead to generalization problems and consequently on the performance of your network by presenting new samples. Table 9 compares the performance results of the proposed architecture with the available results of the models in Table 8.

Although it presents an architecture with many layers, the variation in the performance values of the VGDC architecture was observed as the size of the genomic sequences used in the network increased. Although it uses two convolutional branches, the ViraMiner tool achieved 92.3% and 32% of the sensitivity and precision values, even using relatively short sequences.

Table 9 Performance metrics comparison from the proposed architecture with related works.

Ref.	Accuracy	Precision	Sensibility	Specificity	F1-score	AUROC
[24]	$0.99 - 1$	$0.83 - 1$	$0.84 - 1$	$0.99 - 1$	$0.83 - 1$	–
[29]	–	–	–	–	–	0.8635 0.9210 0.9496 0.9668
[25]	0.90	0.90	0.32	–	–	0.923
[20]	0.985	0.98	1	0.9939	0.9797	0.92
This work	1	1	1	1	1	1

The DeepVirFinder architecture provided only the AUROC values obtained in its model, reaching the maximum value of 96.68% for samples with 3,000 bp. Despite having obtained the sensitivity value of 100% and accuracy of 98%. The work presented by [20] obtained the AUROC value of 92%. The results obtained in the proposed model are superior for all architectures and performance metrics presented in Table 9, indicating the high performance and robustness of the model.

5 Conclusion

The classification and prediction of viral sequences using deep neural networks (DNN) have shown to be very promising in recent years. This work proposes using a CNN like DNN capable of classifying SARS CoV 2 through a binary classification from complete cDNA genomic sequences of eight viral subtypes belonging to the Coronaviridae family.

For this experiment, the technique of cross validation with folder k=5 was used, which reached maximum values in all evaluation metrics for the 960 samples used in training. More than 10,000 sequences were used to test the performance of the DNN after training. An artificial mutation technique was also used to test model generalization with sensitivity ¿ 99% for less than 2,000 mutations in the sequence. A test set formed by 34 samples from the two classes underwent different position mutation rates and was processed by the model proposed in this work together with the BLAST algorithm to verify their performance concerning the accuracy rate according to the two classes. Furthermore, the processing time that both techniques took to obtain their results. In addition, the main results were compared with other viral classification works found in the literature. The proposed model was superior, indicating that the tool proposed in this work can be applied to classify viruses of the Coronaviridae family and viruses of different species.

Declarations

Ethics approval. Not applicable

Consent to participate. Not applicable

Consent for publication. Not applicable

Availability of data and materials. The datasets generated and/or analysed during the current study are available in the Mendeley Data repository, https://data.mendeley.com/datasets/zmhsn2gz7w/1

Competing interests. No competing interest is declared.

Funding. Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001

Authors' contributions. All the authors have contributed in various degrees to ensure the quality of this work (e.g., K.S.A., L.C.d.S., M.G.F.C., R.d.M.B. and M.A.C.F. conceived the idea and experiments; K.S.A., L.C.d.S., M.G.F.C., R.d.M.B. and M.A.C.F. designed and performed the experiments; K.S.A., L.C.d.S., M.G.F.C., R.d.M.B. and M.A.C.F. analyzed the data; K.S.A., L.C.d.S., M.G.F.C., R.d.M.B. and M.A.C.F. wrote the paper. M.A.C.F. coordinated the project). All authors have read and agreed to the published version of the manuscript.

Acknowledgments. The authors wish to acknowledge the financial support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for their financial support.

References

[1] Wang, H. *et al.* The genetic sequence, origin, and diagnosis of sars-cov-2. *European Journal of Clinical Microbiology & Infectious Diseases* 1–7 (2020).

[2] Maghdid, H. S., Ghafoor, K. Z., Sadiq, A. S., Curran, K. & Rabie, K. A novel ai-enabled framework to diagnose coronavirus covid 19 using smartphone embedded sensors: Design study. *arXiv preprint arXiv:2003.07434* (2020).

[3] Chowdhury, M. E. H. *et al.* Can ai help in screening viral and covid-19 pneumonia? *IEEE Access* **8**, 132665–132676 (2020).

[4] Toyoshima, Y., Nemoto, K., Matsumoto, S., Nakamura, Y. & Kiyotani, K. Sars-cov-2 genomic variations associated with mortality rate of covid-19. *Journal of human genetics* **65**, 1075–1082 (2020).

[5] Remita, M. A. *et al.* A machine learning approach for viral genome classification. *BMC bioinformatics* **18**, 1–11 (2017).

[6] Lebatteux, D., Remita, A. M. & Diallo, A. B. Toward an alignment-free method for feature extraction and accurate classification of viral sequences. *Journal of Computational Biology* **26**, 519–535 (2019).

[7] Zielezinski, A., Vinga, S., Almeida, J. & Karlowski, W. M. Alignment-free sequence comparison: benefits, applications, and tools. *Genome biology* **18**, 1–17 (2017).

[8] Nooij, S., Schmitz, D., Vennema, H., Kroneman, A. & Koopmans, M. P. Overview of virus metagenomic classification methods and their biological applications. *Frontiers in microbiology* **9**, 749 (2018).

[9] Altschul, S. F. *et al.* Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research* **25**, 3389–3402 (1997).

[10] Vågene, Å. J. *et al.* Salmonella enterica genomes from victims of a major sixteenth-century epidemic in mexico. *Nature ecology & evolution* **2**, 520–528 (2018).

[11] Lipman, D. J. & Pearson, W. R. Rapid and sensitive protein similarity searches. *Science* **227**, 1435–1441 (1985).

[12] Thompson, J. D., Higgins, D. G. & Gibson, T. J. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research* **22**, 4673–4680 (1994).

[13] Edgar, R. C. Search and clustering orders of magnitude faster than blast. *Bioinformatics* **26**, 2460–2461 (2010).

[14] Randhawa, G. S. *et al.* Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: Covid-19 case study. *Plos one* **15**, e0232391 (2020).

[15] Randhawa, G. S., Hill, K. A. & Kari, L. Ml-dsp: Machine learning with digital signal processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels. *BMC genomics* **20**, 267 (2019).

[16] Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A. & Sun, F. Virfinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome* **5**, 1–20 (2017).

[17] Davenport, T. & Kalakota, R. The potential for artificial intelligence in healthcare. *Future healthcare journal* **6**, 94 (2019).

[18] Mottaqi, M. S., Mohammadipanah, F. & Sajedi, H. Contribution of machine learning approaches in response to sars-cov-2 infection. *Informatics in Medicine Unlocked* 100526 (2021).

[19] Park, Y. & Kellis, M. Deep learning for regulatory genomics. *Nature biotechnology* **33**, 825–826 (2015).

[20] Lopez-Rincon, A. *et al.* Accurate identification of sars-cov-2 from viral genome sequences using deep learning. *bioRxiv* (2020).

[21] Lalmuanawma, S., Hussain, J. & Chhakchhuak, L. Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review. *Chaos, Solitons & Fractals* 110059 (2020).

[22] Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics* **20**, 389–403 (2019).

[23] Zou, J. *et al.* A primer on deep learning in genomics. *Nature genetics* **51**, 12–18 (2019).

[24] Fabijańska, A. & Grabowski, S. Viral genome deep classifier. *IEEE Access* **7**, 81297–81307 (2019).

[25] Tampuu, A., Bzhalava, Z., Dillner, J. & Vicente, R. Viraminer: Deep learning on raw dna sequences for identifying viral genomes in human samples. *PLOS ONE* **14**, e0222271 (2019).

[26] Whata, A. & Chimedza, C. Deep learning for sars cov-2 genome sequences. *Ieee Access* **9**, 59597–59611 (2021).

[27] Adetiba, E. *et al.* Deepcovid-19: A model for identification of covid-19 virus sequences with genomic signal processing and deep learning. *Cogent Engineering* **9**, 2017580 (2022).

[28] Gunasekaran, H. *et al.* Analysis of dna sequence classification using cnn and hybrid models. *Computational and Mathematical Methods in Medicine* **2021**

(2021).

[29] Ren, J. *et al.* Identifying viruses from metagenomic data using deep learning. *Quantitative Biology* 1–14 (2020).

[30] NCBI. GenBank Overview. https://www.ncbi.nlm.nih.gov/genbank/ (2020).

[31] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).