Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 18(6): 8933–8950.
DOI: 10.3934/mbe.2021440
Received: 13 August 2021
Accepted: 22 September 2021
Published: 15 October 2021

Research article

A deep bidirectional recurrent neural network for identification of SARS-CoV-2 from viral genome sequences

Mohanad A. Deif[1], Ahmed A. A. Solyman[2], Mehrdad Ahmadi Kamarposhti[3,*], Shahab S. Band[4,*] and Rania E. Hammam[5]

[1] Department of Bioelectronics, Modern University of Technology and Information (MTI) University, Cairo 11571, Egypt
[2] Department of Electrical and Electronics Engineering, Istanbul Gelisim University, Avcılar 34310, Turkey
[3] Department of Electrical Engineering, Jouybar Branch, Islamic Azad University, Jouybar, Iran
[4] Future Technology Research Center, College of Future, National Yunlin University of Science and Technology, 123 University Road, Yunlin 64002, Taiwan
[5] Department of Bioelectronics, Modern University of Technology and Information (MTI) University, Cairo 11571, Egypt

* **Correspondence:** Email: mehrdad.ahmadi.k@gmail.com, m.ahmadi@jouybariau.ac.ir, shamshirbands@yuntech.edu.tw.

Abstract: In this work, Deep Bidirectional Recurrent Neural Networks (BRNNs) models were implemented based on both Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) cells in order to distinguish between genome sequence of SARS-CoV-2 and other Corona Virus strains such as SARS-CoV and MERS-CoV, Common Cold and other Acute Respiratory Infection (ARI) viruses. An investigation of the hyper-parameters including the optimizer type and the number of unit cells, was also performed to attain the best performance of the BRNN models. Results showed that the GRU BRNNs model was able to discriminate between SARS-CoV-2 and other classes of viruses with a higher overall classification accuracy of 96.8% as compared to that of the LSTM BRNNs model having a 95.8% overall classification accuracy. The best hyper-parameters producing the highest performance for both models was obtained when applying the SGD optimizer and an optimum number of unit cells of 80 in both models. This study proved that the proposed GRU BRNN model has a better classification ability for SARS-CoV-2 thus providing an efficient tool to help in containing the disease and achieving better clinical decisions with high precision.

Keywords: recurrent neural networks; deep learning; COVID-19; coronavirus; SARS-CoV-2; GRU; LSTM Multi-class classification

1. Introduction

In December 2019, a new, human-infecting SARS-Coronavirus-2, Coronavirus was recognized in Wuhan, China [1,2]. It was conveyed that the virus transferred between humans by droplets or close contact [3–5]. As of March 2020, the new SARS-Coronavirus-2 has above than 98,000 cases through 88 countries beyond China [6].

The virus is a pathogenic human coronavirus under the Beta coronavirus genus. The other two pathogenic species, including Severe Acute Respiratory Syndrome Coronavirus (SARS-Coronavirus) and the Middle East Respiratory Syndrome Coronavirus (MERS-Coronavirus) were outbreaks in China and the Middle East in 2002 and 2012 respectively [1]. On January 10, the full genome sequence of this massive RNA virus (SARS-Coronavirus-2) was first uncovered in China's laboratory [5,7] and placed in the NCBI GENBAND. Coronaviruses have enveloped viruses that contain a positive single-stranded ribonucleic acid (RNA) virus that affects humans and animals without segmentation [5,8]. The genomes of the Coronaviruses are formed of base pairs ranging from 26 kilobase-pairs (kbps) to 31 kbps, with GC contents fluctuating from 43 to 32%, and human-infecting coronaviruses including MERS-Coronavirus, HCoV-OC43, SARS-Coronavirus, HCoronavirus-NL63, HCoronavirus-229E, and HCoronavirus-HKU1 [9].

Coronaviruses are characterized by the ability to quickly evolve and adapt to various epidemiological conditions [10]. Each replication cycle of Coronavirus presents new genetic mutations, and its normal developmental rate is nearly from 4 to 10 nucleotide substitutions per site/ year [1]. During genomic information reproduction, SARS-Coronavirus-2 evolves [11]. The mutation happens because of certain errors when copying RNA to a new cell. SARS-Coronavirus-2 analyses can produce false-positive results if they are not targeted especially to SARS-Coronavirus-2, as the virus is difficult to distinguish from other Coronavirus organisms because of their genetic similarity. Therefore, it is essential to use improved diagnostic tools in order to correctly classify SARS Coronavirus-2 from other Coronaviruses.

Currently, Molecular approaches are commonly used to classify pathogens, such as quantitative real-time RT-PCR and nucleic acid sequencing methods [12]. However, owing to their comparatively new evolution and still not entirely understood characteristics, they have an overall unsatisfactory identification rate for this particular virus. Since coronaviruses are genetic diseases, viral sequencing techniques have been employed to detect the virus depending on its genome sequence and therefore avoiding the downsides of the classical diagnostic methods.

Classification using viral sequencing methods depends on strategies of alignment such as FASTA [13] and Basic Local Alignment Search Tool (BLAST) [14] algorithms. These methods count on the assumption that DNA sequences have common features, and their order predominates among various sequences [15,16]. However, there are limitations to these methods, which include the need for base sequences for the detection [17]. In addition, because viruses have a high level of mutation and most genomes do not have proper reference sequences, next-generation sequencing (NGS) genomic samples might not be identified by BLAST [18]. This method has also a negative impact of neglecting part of the vital information contained in the input sequence if it cannot completely fill a DNA sequence of fixed size.

Traditional machine learning approach using genomic signal processing technique was introduced to discern between Covid-19 and other Coronaviruses [19,20] by using their genomic sequences reported in the NCBI GENBAND in order to increase the accuracy of disease detection in lesser time. Some characteristics were then extracted, such as Discrete Fourier transformation, Discrete Cosine transformation, and incorporated into a classifier. However, this method needs to extract pre-selected features in order to recognize or classify the viral DNA sequences.

The deep learning approach has been recently introduced as various alternatives for DNA sequence recognition, as these strategies do not require pre-selected features to recognize or classify DNA sequences. Deep learning approaches [21,22] have evolved rapidly since they can be employed in the processing of large-scale genetic data, especially in the field of bioinformatics. Today, the size of these datasets exceeds the size of 10 million [23]. Studies in this area are focused on the analysis and classification of DNA and RNA sequences.

Deep learning to use Convolution Neural Networks (CNN) [24–26] have been efficiently used for the classification of DNA sequences. However, these methods encode the RNA sequences data into an input layer of the CNN therefore, numeric values have to be assigned to the different bases. In addition, CNN learns to recognize patterns across space while RNN helps in resolving temporal data problems.

Other alternative approaches focusing on deep learning have also been explored by slicing sequences into bits of fixed lengths, from 300 [16] to 3000 bps [23]. However, these methods ignore part of the data contained in the input sequence if it does not fill a piece of fixed size.

Because of the difficulties of differentiating between the SARS-Coronavirus-2 virus and other Coronavirus organisms (SARS-Coronavirus and MERS-Coronavirus) and respiratory infections because of their genetic similarity; it was aimed to develop an advanced system to correctly classify the virus based on its genomic sequence. Apart from the classical approaches, the proposed Deep Learning Bidirectional Recurrent Neural Networks (BRNNs) approach eliminates the need for pre-selected features to identify or classify the viral DNA sequences and also works on the full DNA input sequence as a whole thereby overcoming the problem of ignoring any data in the input sequence. It has also the advantages of rapid and high accuracy of disease detection and classification.

The rest of this article is organized as follows. The contribution of this research is presented in the next section. A background overview of BRNNs with different types of cell units is introduced in Section 4. The methodology, preparation of the data, and proposed algorithms are demonstrated in Section 5. Experimental results and a discussion of the results are presented in Section 6. Finally, concluding remarks are given in Section 7.

2. Contribution

In this study, Recurrent Neural Networks (RNN) based models with different cell units were implemented for the purpose of classification of the SARS-Coronavirus-2 virus from other Coronavirus and respiratory infections based on its DNA sequence. Apart from the classical approaches, the proposed Deep Learning Bidirectional Recurrent Neural Networks (BRNNs) approach eliminates the need for pre-selected features and also works on the full DNA input sequence as a whole thereby overcoming the problem of ignoring any data in the input sequence. A comparison between the presented deep network models based on LSTM and GRU unit cells was performed to attain the most efficient model for virus classification. Also, an investigation of the hyper-parameters that worked best with the BRNN models was presented to select the highest accuracy classification model.

3. Background

3.1 Recurrent neural network (RNN) overview

In recent times, the RNN model has been an extremely favored architecture [27], especially for sequential data. The RNN structure is demonstrated in Figure 1. Each node at a time step comprises an input from the preceding node and uses a feedback loop to continue. By employing the current input and previous hidden state, each node produces a current hidden state and output as follows:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{1}$$

$$o_t = f\left(W_{hy}h_t + b_o\right) \tag{2}$$

where x is the input, y is the output sequence, h is the hidden vector sequence at each time Step (t). W denotes the weight matrices, and b represents the bias for hidden and output states. f is the activation function used at the hidden layers.

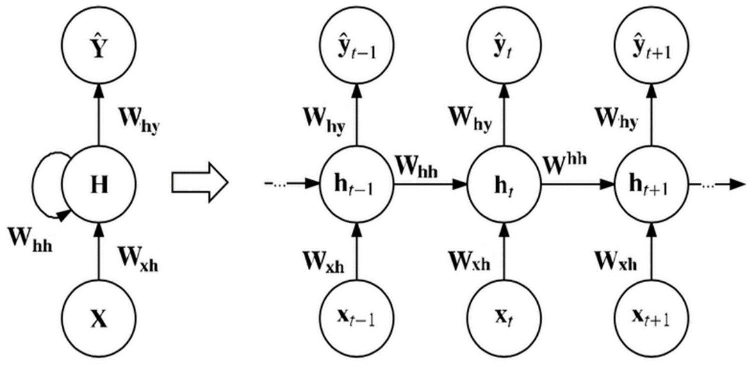


Figure 1. Basic RNN architecture.

3.2 Long short-term memory (LSTM) networks

The shared downside of the classical RNN models is that as the time step increases, the network fails to draw context from time steps of previous states far behind, such a phenomenon is known as long-term dependence. Moreover, exploding and vanishing gradient problems are often found quite commonly due to the deep layers of a network and repetitive behavior of the standard RNN,

To resolve this problem, LSTM models were implemented by positioning memory cells with multiple gates in a hidden layer [28]. Figure 2 displays a hidden layer block with an LSTM cell unit, and the three functions of gate controllers are demonstrated as follows:

- Forget gate (f_t) (should be omitted): It chooses which fragment of the long-term state c_t should be ignored.
- Input gate (i_t): It controls which part of c should be added to the long-term state c_t.
- Output gate (g_t): It defines which fragment of c should be fed to $h_t$and o_t.

The following equations demonstrate the cell's long-term and short-term states and each layer's output in time steps.

$$f_t = \sigma\left(W_{xf}^T \cdot x_t + W_{hf}^T \cdot h_{t-1} + b_f\right) \tag{3}$$

$$i_t = \sigma(W_x^T i \cdot x_t + W_h^T i \cdot h_{t-1} + b_i) \tag{4}$$

$$o_t = \sigma(W_x^T o \cdot x_t + W_h^T o \cdot h_{t-1} + b_i) \tag{5}$$

$$g_t = tanh(W_x^T g \cdot x_t + W_h^T g \cdot h_{t-1} + b_i) \tag{6}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \tag{7}$$

$$(o_t, h_t) = g_t \otimes \tanh(c_t) \tag{8}$$

where $W_{xf}, W_x i, W_x o, W_x g$ represent the weight matrices for the corresponding associated input vector and $W_{hf}, W_h i, W_h o, W_h g$ characterize the weight matrices of the short-term state of the previous time step, and b_f, b_i, b_o, b_g are bias.

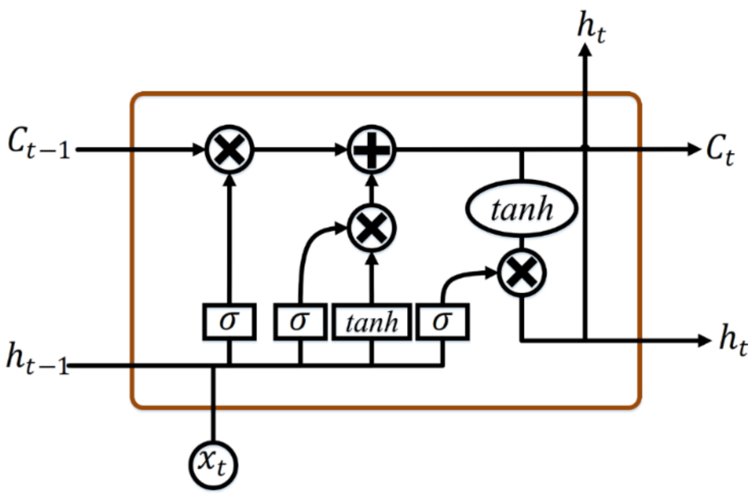


Figure 2. LSTM neural network unit architecture.

3.3 Gated recurrent unit (GRU) network

GRUs presented in 2014 resemble LSTMs but they have fewer parameters [11]. Figure 3 shows the block of a hidden layer with a GRU cell unit. They also have gated units as LSTMs that regulate the flow of context within the device without having separate memory cells. GRU doesn't have an output barrier, unlike LSTM, thereby revealing its full content. Mathematically, GRU formulations are expressed by the following equations:

$$r_t = \sigma(W_{xr}^T \cdot x_t + W_{or}^T \cdot o_{t-1} + b_r) \tag{9}$$

$$z_t = \sigma(W_{xz}^T \cdot x_t + W_{oz}^T \cdot o_{t-1} + b_z) \tag{10}$$

$$\tilde{o}_t = tanh\,(W_{x\tilde{o}}^T \cdot x_t + W_{o\tilde{o}}^T \cdot (r_t \otimes o_{t-1}) + b_{\tilde{o}}) \tag{11}$$

$$o_t = z_t \otimes o_{t-1} + (1 - z_t) \otimes \tilde{o}_t \tag{12}$$

where $W_{xr}, W_x z, W_{x\tilde{o}}$ indicate the weight matrices for the consistently associated input vector and W_{or}, W_{oz}, W_o characterize the weight matrices of the preceding time step, b_r, b_z, b are biasing.

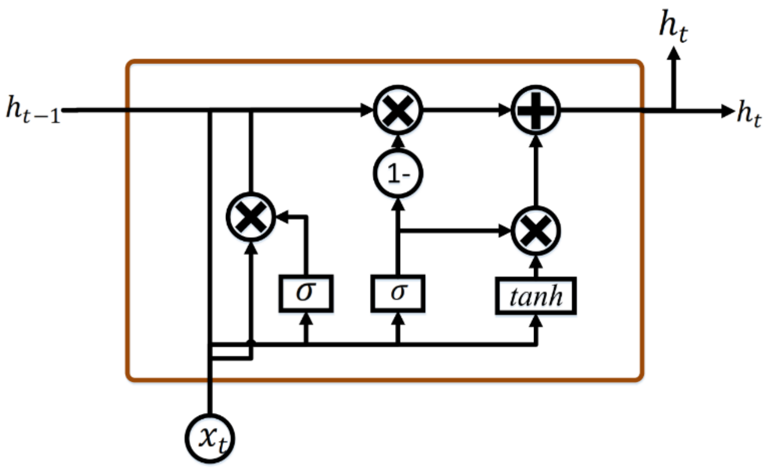


Figure 3. GRU neural network unit architecture.

3.4 Bidirectional recurrent neural network

Bidirectional Recurrent Neural Networks (BNN) is a version of the RNN that Schuster and Paliwal suggested in 1997 [17]. Standard RNN's use the previous context information, but Bidirectional RNNs explore also the future context. BRNN is a better alternative than plain RNN since the input sequence of bidirectional RNN processes has two RNN paths, one responsible for the forward states and the other for the backward states. Hence, Bidirectional RNN effectively increases the amount of information available to the network, improving the content available to the algorithm.

4. Methodology

4.1. Dataset

4.1.1. Dataset selection

The DNA sequence information was derived from the Novel Coronavirus Resource 2019 repository (2019nCoVR) [29]. All the accessible sequences with the Query Nucleotide Completeness = "Complete", Sequence Quality = "High", Host = "Homo Sapiens", DNA Sequence lengths = (29,800–29,900 bps) nucleotides for each epidemic, and Data Source = "GenBank" for a total of 2500 complete genome samples. 500 random complete genomes for each epidemic (COVID-19, MERSA–CORONAVIRUS-2, Common cold, ARI, and SARS) were chosen.

4.1.2. Data cleaning and preprocessing

To preprocess the data, all repeated sequences and sequences having a length that is out of the identified range (29,800–29,900 bps) were removed. Then the data was labelled and prearranged, as shown in Table 1. The data were grouped into 5 classes based on the disease type as follows:

- First Class "COVID-19": Includes the genome virus [SARS-CoV-2]
- Second Class "MERSA–CORONAVIRUS-2": Includes the genome virus [MERS-CoV]
- Third Class "Common Cold": Includes the genome viruses [HCoV-OC43/HCoV-229E] which are placed in one category, as they are coronaviruses accountable for normal cold [30], and [HCoV-4408] was also added because it is a Beta Coronavirus as HCoV-OC43.
- Fourth Class "ARI": Includes the genome viruses [HCoV-NL63/ HCoV-HKU1] in the same group as they are both accompanied by acute breathing infections [31].
- Fifth Class "SARS": Includes the genome viruses [SARS-Coronavirus-P2/SARS-Coronavirus/SARS-Coronavirus GDH-BJH01/SARS-Coronavirus HKU-39849 [32]] which are categorized in the same class because they are all strains of SARS.

Table 1 The virus name, type of disease, labelling, and number of samples of the input data sequence.

Virus name	Disease	Label
SARS Coronavirus -2	COVID 19	0
MERS- Coronavirus	MERSA – COV-2	1
Human coronavirus -OC43		
Human coronavirus -229E	Common cold	2
Human coronavirus -4408		
Human coronavirus -NL63	ARI	3
Human coronavirus -HKU1		
SARS Coronavirus		
SARS Coronavirus - P2	SARS	4
SARS Coronavirus HKU-39849		
SARS Coronavirus GDH-BJH01		

The samples were divided into two parts, 80% for training, 20% for testing, in tenfold cross-validation. The steps of implementation were summarized in Figure 4.

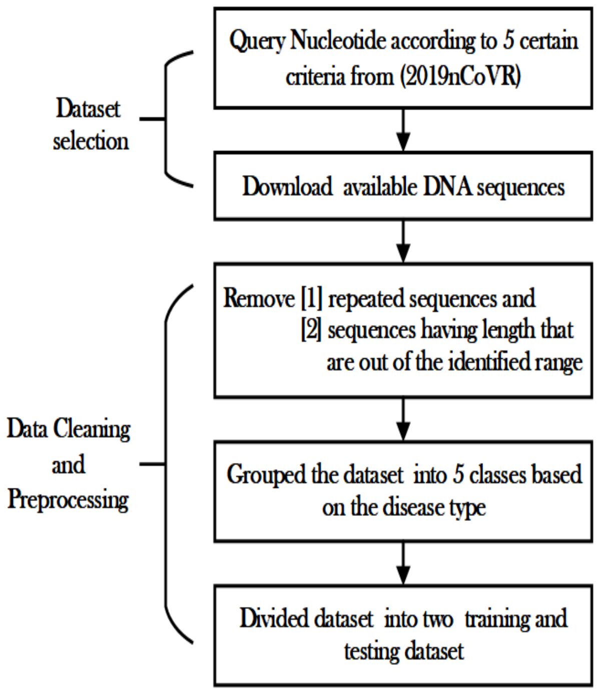

Dataset selection
Query Nucleotide according to 5 certain criteria from (2019nCoVR)
Download available DNA sequences
Data Cleaning and Preprocessing
Remove [1] repeated sequences and [2] sequences having length that are out of the identified range
Grouped the dataset into 5 classes based on the disease type
Divided dataset into two training and testing dataset

Figure 4. Overall dataset processing.

4.2. Proposed BRNN architecture

In this section, the proposed BRNN classifier model architectures were performed and shown in Figure 5. The proposed model depends on RNN with enhanced cell units, LSTM and GRU, in their hidden states, while the training procedure is applied in a bidirectional manner. The model comprises five layers that are described as follows:

Layer 1 (Input sequence): After selecting and preprocessing the DNA sequences as described in Section 5.1, it was fed into the presented RNN models.

Layer 2 (k-mers model)*:* The DNA sequence S was split into overlapping k-meters of length k (k =4 is employed in this work) by the use of a sliding window with stride s, the size of the window is the current k-mer. As illustrated in Figure 6. For DNA sequence S, the windows are located covering the first k bases, and it moves one right character at a time to generate the next k-mer. Given a sequence of length $n, S = (S_1, S_2, \dots, S_n)$ where $S_i \in \{A, C, G, T\}$, S is converted into $\tilde{n} = (n - k_{\text{high}} + 1\ )$ k-mer's numbers.

$$f(S) = \left(S_{1:k_1}, S_{2:2+k_2}, \dots S_{\tilde{n}:\tilde{n}+k_{\tilde{n}}}\right) \tag{13}$$

It was observed that k-mer embedding for representing input sequences has several advantages over one-hot encoding, which was reported in previous works. Firstly, it improves the model performance, secondly; it reduces computational time and space required to execute models as compared to one-hot encoding because the word vector in the one-hot method may be a high dimensional vector since it encodes each word in a large number of text data.

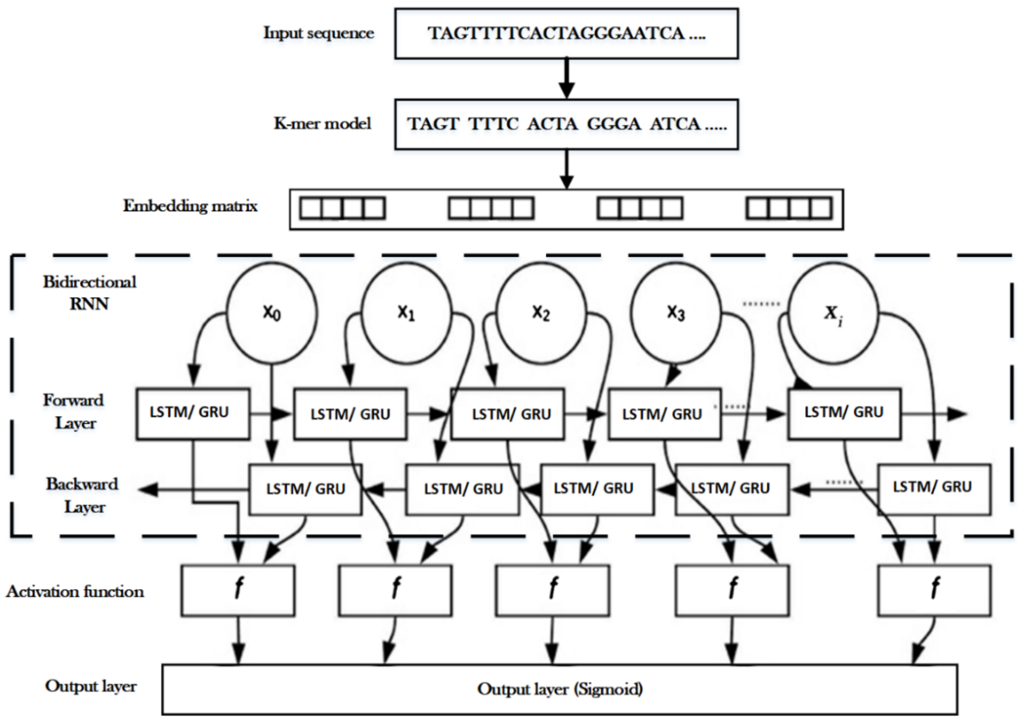


Figure 5. The proposed BRNN model architectural structure.

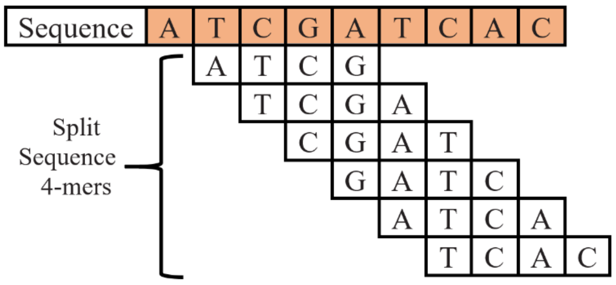


Figure 6. Converting of the sequence S into overlapping fixed length 4-mer based on sliding a window of length 4.

Layer 3 (Word2vec algorithm): By using the word2vec algorithm [33], each k-mer is mapped into a d-dimensional vector space. Word2vec can utilize either Continuous Bag-Of-Words (CBOW) or Continuous Skip-Gram (CSG) to produce a distributed representation of words. A continuous bag-of-words (CBOW) was employed for all our experiments because the CBOW is faster to train than skip-gram and CBOW works well with a large amount of training data.

Layer 4 (Bidirectional RNN): It represents the suggested models based on Bidirectional RNN, where LSTM or GRU cells are used as the hidden blocks. The forward track traces the data section from left to right, while in both BLSTM and BGRU, the backward track traces the input from right to left. The forward recurrent sequence and backward hidden sequence can be formulated as follows:

$$\vec{\mathrm{h}}_t = f\left(\mathrm{W}_{\mathrm{x},\vec{\mathrm{h}}}\mathrm{x}_t + \mathrm{W}_{\vec{\mathrm{h}},\vec{\mathrm{h}}}\vec{\mathrm{h}}_{t-1} + \mathrm{b}_{\vec{\mathrm{h}}}\right) \tag{14}$$

$$\overleftarrow{\mathrm{h}}_t = f\left(\mathrm{W}_{\mathrm{x},\overleftarrow{\mathrm{h}}}\mathrm{x}_t + \mathrm{W}_{\overleftarrow{\mathrm{h}},\overleftarrow{\mathrm{h}}}\overleftarrow{\mathrm{h}}_{t+1} + \mathrm{b}_{\overleftarrow{\mathrm{h}}}\right) \tag{15}$$

$$\mathrm{y}_t = \left(\mathrm{W}_{\overrightarrow{\mathrm{h}},\mathrm{y}}\overrightarrow{\mathrm{h}}_t + \mathrm{W}_{\overleftarrow{\mathrm{h}},\mathrm{y}}\overleftarrow{\mathrm{h}}_t + \mathrm{b}_\mathrm{y}\right) \tag{16}$$

where x_t is the input feature vector, $\overrightarrow{\mathrm{h}}_t$ $(\overleftarrow{\mathrm{h}}_t)$ is the activation vector on the forward (backward) hidden layer, $\mathrm{W}_{\mathrm{p,q}}$ is the weight matrix, b_r is the bias term, f $(\cdot)$ is the activation function on each node in the hidden layers, and y is the posterior probability vector of the output label.

Layer 5 (Output layer): A regular sigmoid function that is applied on the output layer in order to measure the expected probability of characters for each step of t and k in the alphabet. This performance is shown as follows:

$$\mathrm{y}_t = sigmoid\left(\mathrm{W}_{\overrightarrow{\mathrm{h}},\mathrm{y}}\overrightarrow{\mathrm{h}}_t + \mathrm{W}_{\overleftarrow{\mathrm{h}},\mathrm{y}}\overleftarrow{\mathrm{h}}_t + \mathrm{b}_\mathrm{y}\right) \tag{17}$$

4.3. Hyper-parameters

While training a model, it is important to know which hyper-parameters to select. Hyper-parameters refer to the parameters that are set before the start of Bidirectional RNN training. The time required to train and test a model is dependent on the types of hyper-parameters selected. The tune-ability of an algorithm, hyper-parameters, or interacting hyper-parameters is a measure of how much performance can be gained by tuning it. In this research, an investigation of the hyper-parameters that work best on the RNN models was performed in order to attain the highest accuracy for DNA sequence classification. The hyper-parameters that were focused on in this work are the optimizer type and the number of unit cells.

Four different types of gradient descent algorithm were employed on both the BRNN models based on LSTM and GRU cells including SGD, Adam, Adagrad, and RMSprop optimizers and three different unit numbers of cells ((50, 80 and 100). The performance of different parameter settings was recorded and compared to get the optimum parameter settings.

Gradient descent (GD) is a class of algorithms that aims to find the minimum point on a function by following the gradient. Vanilla gradient descent just follows the gradient (scaled by the learning rate). Two common tools to improve gradient descent are the sum of gradient (first moment) and the sum of the gradient squared (second moment). The Momentum method uses the first moment with a decay rate to gain speed. AdaGrad uses the second moment with no decay to deal with sparse features. RMSProp uses the second moment with a decay rate to speed up from AdaGrad. Adam uses both first and second moments and is generally the best choice.

In our research, didn't use a Particle Swarm Optimization (PSO) because a global optimization method PSO does not use the gradient of the problem being optimized, which means PSO was less effective compared to GD in our application. In addition to that, we analyzed the effect of different embedding dimensions (numbers of cells), including 150 and 200. The model complexity will be increased by more weight parameters, which are caused by a larger d and need to be learned in the embedding layer.

4.4. Performance evaluation

The performance of the proposed deep learning BRNN approach for SARS Coronavirus-2 virus classification was evaluated with the following classification performance measures which include accuracy, precision, recall, specificity, and F1-score and the corresponding confusion matrix was further developed. Confusion matrices represent a very important metric in machine learning evaluation since it provides further information on relations between classes and types of classification errors. The x-axis in the confusion matrix shows the predicted output (disease class) of the model and the y-axis shows the true class of the disease. The diagonal elements of the matrix represent the true positive values of the model, meaning all the correct classifications performed by the model and all other entries in the matrix show misclassifications. Off-diagonal elements for the row represent the probability that the disease of a certain class will be mis-classified as another class. The definition of the performance metrics are clarified as follows:

Accuracy: Represents the performance accuracy of the model

$$\text{Accuracy}= \frac{TP\ (Number\ of\ True\ Positives)+TN(Number\ of\ the\ true\ negatives)}{All\ predictions\ performed\ by\ the\ model} \tag{18}$$

Precision: It identifies the proportion of correct positive classifications.

$$\text{Precision}= \frac{TP}{TP+FP} \tag{19}$$

Recall (Sensitivity): It measures the proportion of actual positives correctly classified by the model.

$$\text{Recall}= \frac{TP}{TP+FN} \tag{20}$$

Specificity: It identifies the proportion of correct negative classifications.

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{21}$$

F1-score: It indicates the perfect balance between precision and recall values. It can be calculated as follows

$$\text{F1 score}= 2\,\frac{Precision*Recall}{Precisio+Recall} \tag{22}$$

Another significant metric is also assessed which is the area under the Receiver Operating Characteristic (ROC) curve. The Receiver Operating Characteristic (ROC) curve shows a relation between the True Positive Rate (TPR) and False Positive Rate (FPR) of the classification. The FPR is defined by:

$$\text{FPR}= \frac{FP}{FP+TN} \tag{23}$$

Therefore, the ROC Curve Region (AUC) tests the efficiency of all potential classification thresholds in a given model and therefore shows the accuracy of the findings independently of that model.

5. Results

In the first part of this section, experiments were conducted on two proposed BRNN models depending on two different types of cell units (GRU) and (LSTM) in order to attain a high-efficiency classification model based on viral DNA sequence. The RNN models were implemented using four different optimizers (SGD, Adam, Adagrad, and RMSprop) and the number of unit cells was altered between random values of 50, 80 and 100 to obtain the most suitable optimizer and the optimum number of cells that will lead to the highest classification efficiency.

A comparative study was performed and shown in Table 2 showing the classification accuracies of both models employing the four different types of optimizers and changing the unit number of cells in each optimizer. It was clear from the table that the highest accuracies were obtained when applying the SGD optimizer and the number of unit cells of 80 in both models. The results also revealed that the GRU BRNN model has achieved higher efficiencies than its counterpart model as it achieved an overall accuracy of 96.8 % while the LSTM RNN model achieved 95.8 %. This result can be attributed to the fact that the GRU cell in the BRNNs deploys a reset gate and an update gate in a hidden layer which is computationally more efficient than a usual LSTM network.

Furthermore, a plot was drawn and shown in Figure 7. In order to visualize the overall performance of the two models on all optimizers and applying the different number of unit cells. It was obvious from the plot that the performance of both models was nearly the same when applying a unit number of cells of 50. The highest classification accuracy achieved by the LSTM RNN model was 95.8% when applying the ADAM optimizer with a LSTM number of 50 and the SGD optimizer with a cell number of 80. However, the higher classification ability of the GRU was seen when applying a cell number of 80 and 100 for all optimizers reaching the highest value of 96.8% with the SGD optimizer and a GRU number of 80. The accuracy of both models didn't increase by increasing the number of unit cells concluding that the optimum number of cells was 80 and employing the SGD optimizer to achieve an efficient classification.

Table 2. Comparison between the classification accuracies of the LSTM BRNN and GRU BRNN Models based on different hyper-parameters.

Optimizer Type	Units number	LSTM BRNN Model (%)	GRU BRNN Model (%)
SGD_50	50	94.52	95.89
Adam_50	50	95.89	95.53
Adagrad_50	50	95.29	95.32
RMSprop_50	50	95.21	95.43
SGD_80	80	95.3	96.88
Adam_80	80	94.37	96.4
Adagrad_80	80	94.42	96.17
RMSprop_80	80	94.22	96.58
SGD_100	100	94.14	95.51
Adam_100	100	94.52	95.32
Adagrad_100	100	94.32	95.22
RMSprop_100	100	94.34	95.66

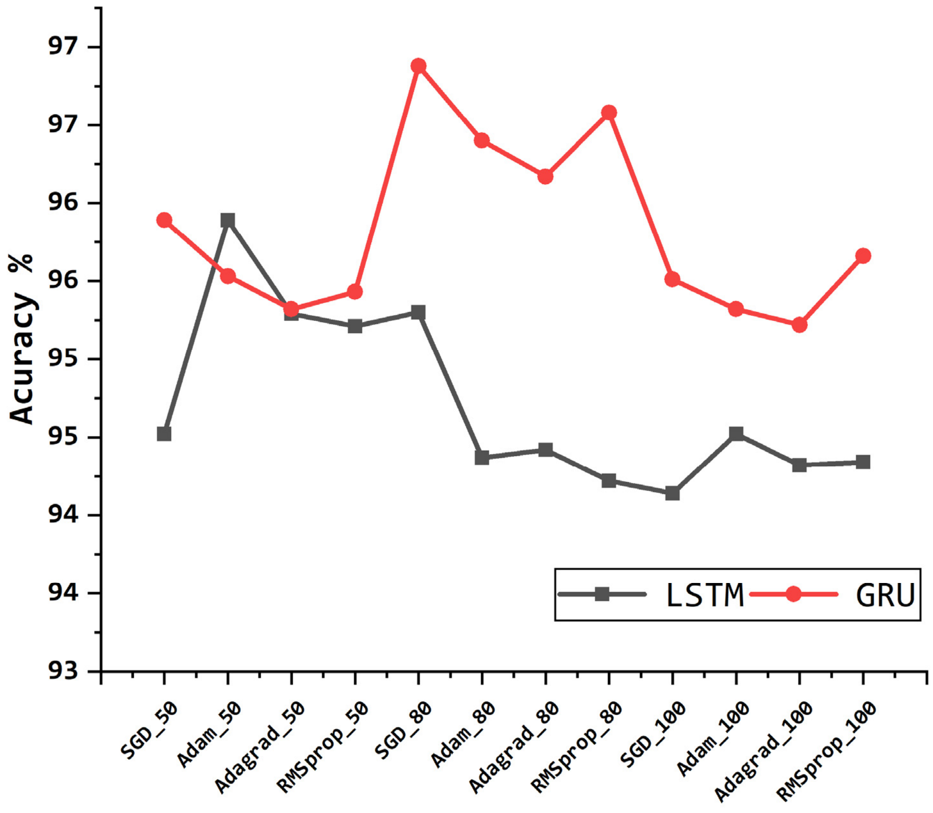
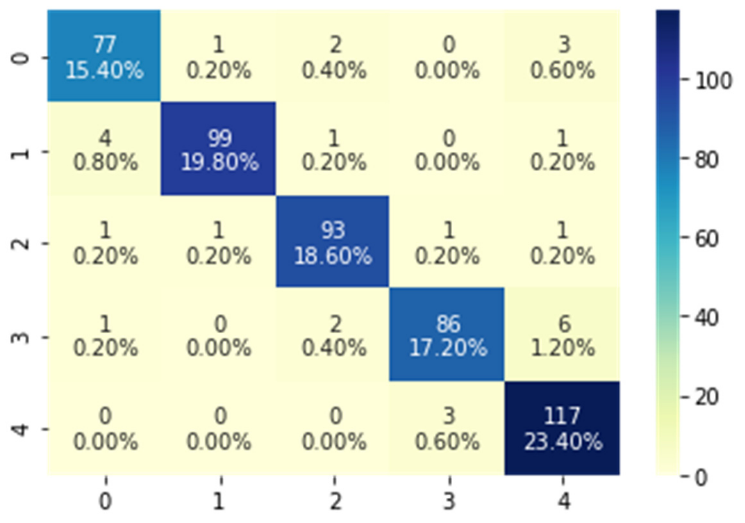
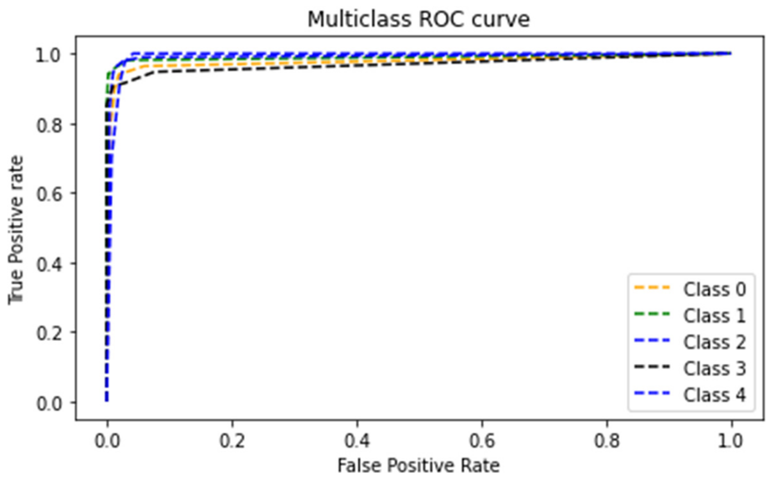


Figure 7. Classification accuracies of the LSTM and GRU BRNN Models.

To evaluate the effectiveness of the implemented BRNN models employing the best hyper-parameters (SGD optimizer with a cell number of 80) and its ability to discriminate between SARS-Coronavirus-2 and other viruses, the confusion matrices for the two models and the ROC curves were developed and shown in Figures 8 and 9. The resulting confusion matrix of the LSTM BRNN model reveals that 77 out of the 83 SARS-Coronavirus-2 sequences were correctly categorized, while the other 6 sequences were incorrectly classified in other groups. In the other hand, the GRU BRNN model correctly identified 37 of the 39 SARS-Coronavirus-2 sequences, while the remaining two were misclassified. By comparing the two matrices, it can be seen that the GRU has fewer false-positive and fewer false negative values than the LSTM, which reveals the higher accuracy of prediction of the GRU model. Another measure that shows successful classification is the area under ROC curve (AUC value). Generally, ROC curve is defined as a probability curve in classification studies and the AUC values express the degree of class separability within these possibilities. The higher the AUC value, the higher the classification predictions of the employed model. As showed from Figures 8 and 9 that the average AUC value (0.98) of the GRU model was higher than that of the LSTM (0.96) which also confirms the results of the present study that proved the higher ability of classification of the GRU as compared to the LSTM model. These promising results can help in the efficient detection and identification of the SARS Coronavirus-2 and discriminate it from other Coronaviruses with high accuracy results which can further help contain its spread.

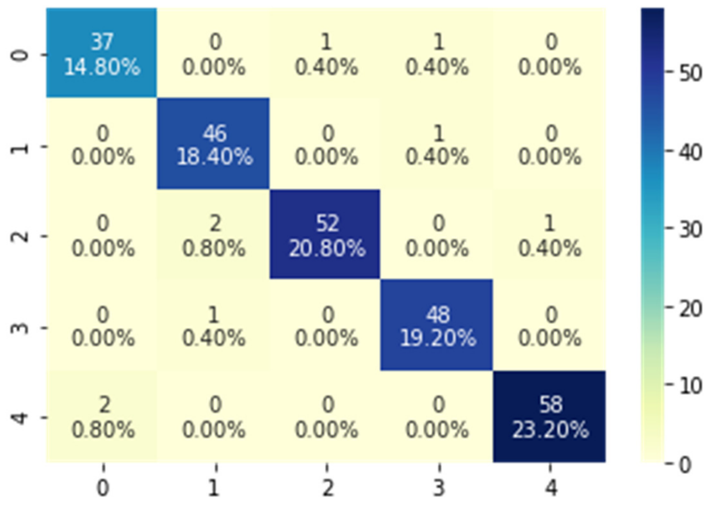


(a)

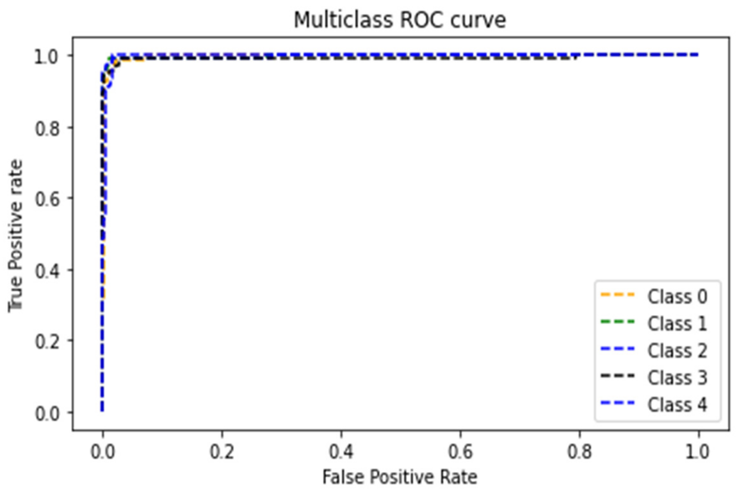


(b)

Figure 8. LSTM BRNN model. (a) Confusion matrix (b) ROC curves.



(a)



(b)

Figure 9. GRU BRNN model. (a) Confusion matrix (b) ROC curves.

Furthermore, Tables 3 and 4 show the classification performance metrics for all five classes of disease. It was clear from the tables that the accuracy of prediction of the presented models was higher than 97% for all classes of viruses, showing the efficient performance of both models. However, when comparing the ability of both models to classify class 0 (SARS-Coronavirus-2 virus), it was seen that the GRU achieved a higher classification performance than the LSTM since it attained an accuracy of 98.4% and precision, Recall, and F1 Score values of 0.95 which were higher than the LSTM performance values. Hence, it can be concluded that the presented GRU BRNN model has a better prediction ability for COVID 19, thus providing an efficient tool to help doctors achieve better clinical decisions with high precision.

Table 3. LSTM BRNN performance metrics for each virus class.

Classes	Accuracy (%)	Precision	Recall	F1 Score
0	97.6	0.93	0.93	0.93
1	98.4	0.94	0.98	0.96
2	98.2	0.96	0.95	0.95
3	97.4	0.91	0.96	0.93
4	97.2	0.97	0.91	0.94

Table 4. GRU BRNN Model Performance Metrics for each virus class.

Classes	Accuracy (%)	Precision	Recall	F1 score
0	98.4	0.95	0.95	0.95
1	98.4	0.98	0.94	0.96
2	98.4	0.95	0.98	0.96
3	98.8	0.98	0.96	0.97
4	98.8	0.97	0.98	0.97

6. Conclusions

Developing efficient diagnostic machine learning tools that target the genome in order to identify different classes of Corona Virus can help reduce the death rates and the time to identify the infection for proper clinical decisions. A Deep Learning Approach employing Bidirectional Recurrent Neural Networks (BRNN) based models was implemented to classify the SARS-Coronavirus-2 virus from other Coronaviruses and other respiratory infections based on its DNA sequence. It was aimed to propose a system with high-accuracy results that can classify the type of disease based on genome sequence leading to a rapid diagnosis of COVID-19, avoiding the disadvantages of the traditional diagnostic methods. An evaluation of the BRNN models depending on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) cells was investigated to determine the most effective model for SARS Coronavirus-2 sequence classification. A study of the hyper-parameters that work best on the BRNN models was also performed, including the optimizer type and the number of unit cells. The results revealed that the best hyper-parameters producing the highest performance for both models were got when applying the SGD optimizer and an optimum number of unit cells of 80 in both models. Moreover, the GRU BRNN model was able to discriminate between SARS Coronavirus-2 and other viruses with a higher overall classification accuracy of 96.8% as compared to that of the LSTM BRNN model having a 95.8% overall classification accuracy. When comparing the ability of both models to classify all classes of viruses, the accuracy of prediction were all higher than 97% for all classes, indicating the efficient performance of both models. However, when comparing the ability of both models to classify class 0 (SARS-Coronavirus-2 virus), it was seen that the GRU achieved a higher classification performance than the LSTM since it attained higher accuracies, precision, recall, and F1 score values. Hence, it was concluded that the proposed GRU BRNN model has a better prediction ability for COVID-19, thus providing an efficient tool to help achieve better clinical decisions with high precision.

Conflict of interest

All authors declare that there is no conflicts of interest in this paper.

References

1. R. Lu, X. Zhao, J. Li, P. Niu, B. Yang, H. Wu, et al., Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding, *Lancet*, **395** (2020), 565–574.
2. M. A. Deif, A. A. A. Solyman, R. E. Hammam, ARIMA Model Estimation Based on Genetic Algorithm for COVID-19 Mortality Rates, *Int. J. Inf. Technol. Decis. Mak.*, (2021), 1–24.
3. C. Wang, P. W. Horby, F. G. Hayden, G. F. Gao, A novel coronavirus outbreak of global health concern, *Lancet*, **395** (2020), 470–473.
4. D. Cucinotta, M. Vanelli, WHO declares COVID-19 a pandemic, *Acta Bio. Med. Atenei Parm.*, **91** (2020), 157.
5. M. Deif, R. Hammam, A. Solyman, Adaptive Neuro-Fuzzy Inference System (ANFIS) for Rapid Diagnosis of COVID-19 Cases Based on Routine Blood Tests, *Int. J. Intell. Eng. Syst.*, 2020.
6. *Rational use of personal protective equipment for coronavirus disease (COVID-19) and considerations during severe shortages: interim guidance*, World Health Organization, 2020.
7. J. Yang, Inhibition of SARS-CoV-2 Replication by Acidizing and RNA Lyase-Modified Carbon Nanotubes Combined with Photodynamic Thermal Effect, *J. Explor. Res. Pharmacol.*, (2020), 1–6.
8. M. Pal, G. Berhanu, C. Desalegn, V. Kandi, Severe acute respiratory syndrome Coronavirus-2 (SARS-CoV-2): An update, *Cureus*, **12** (2020), 3.
9. P. C. Y. Woo, Y. Huang, S. K. P. Lau, K. Y. Yuen, Coronavirus genomics and bioinformatics analysis, *Viruses*, **2** (2010), 1804–1820.
10. N. Decaro, V. Mari, G. Elia, D. D. Addie, M. Camero, M. S. Lucente, et al., Recombinant canine coronaviruses in dogs, Europe, *Emerg. Infect. Dis.*, **16** (2010), 41.
11. M. Pachetti, B. Marini, F. Benedetti, F. Giudici, E. Mauro, P. Storici, et al., Emerging SARS-CoV-2 mutation hot spots include a novel RNA-dependent-RNA polymerase variant, *J. Transl. Med.*, **18** (2020), 1–9.
12. L. Peñarrubia, M. Ruiz, R. Porco, S. N. Rao, M. Juanola-Falgarona, D. Manissero, et al., Multiple assays in a real-time RT-PCR SARS-CoV-2 panel can mitigate the risk of loss of sensitivity by new genomic variants during the COVID-19 outbreak, *Int. J. Infect. Dis.*, 2020.
13. W. R. Pearson, Rapid and sensitive sequence comparison with FASTP and FASTA, 1990.
14. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic local alignment search tool, *J. Mol. Biol.*, **215** (1990), 403–410.
15. L. Pinello, G. L. Bosco, G. C. Yuan, Applications of alignment-free methods in epigenomics, *Brief Bioinf.*, **5** (2014), 419–430.
16. S. Vinga, J. Almeida, Alignment-free sequence comparison–a review, *Bioinformatics*, **19** (2003), 513–523.
17. D. Bzhalava, J. Ekström, F. Lysholm, E. Hultin, H. Faust, B. Persson, et al., Phylogenetically diverse TT virus viremia among pregnant women, *Virology*, **432** (2012), 427–434.
18. A. Tampuu, Z. Bzhalava, J. Dillner, R. Vicente, ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples, *PLoS One*, **14** (2019), e0222271.

19. S. M. Naeem, M. S. Mabrouk, S. Y. Marzouk, M. A. Eldosoky, A diagnostic genomic signal processing (GSP)-based system for automatic feature analysis and detection of COVID-19, *Brief Bioinf.*, 2020.
20. M. A. Deif, R. E. Hammam, A. Solyman, Gradient Boosting Machine Based on PSO for prediction of Leukemia after a Breast Cancer Diagnosis, *Int. J. Adv. Sci. Eng. Inf. Technol.*, **11** (2021), 508–515.
21. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444.
22. J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, **61** (2015), 85–117.
23. M. Wainberg, D. Merico, A. Delong, B. J. Frey, Deep learning in biomedicine, *Nat. Biotechnol.*, **36** (2018), 829–838.
24. Y. Kim, Convolutional neural networks for sentence classification, preprint, arXiv: 1408.5882.
25. A. Lopez-Rincon, A. Tonda, L. Mendoza-Maldonado, E. Claassen, J. Garssen, A. D. Kraneveld, Accurate identification of sars-cov-2 from viral genome sequences using deep learning, *bioRxiv*, 2020.
26. M. A. Deif, R. E. Hammam, Skin lesions classification based on deep learning approach, *J. Clin. Eng.*, **45** (2020), 155–161.
27. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *IEEE Signal Process. Mag.*, **29** (2012), 82–97.
28. N. G. Nguyen, V. A. Tran, D. L. Ngo, D. Phan, F. R. Lumbanraja, M. R. Faisal, et al., DNA sequence classification by convolutional neural network, *J. Biomed. Sci. Eng.*, **9** (2016), 280.
29. *China National Center for Bioinformation,2019 Novel Coronavirus Resource (2019nCoVR)*, 2020, https://bigd.big.ac.cn/ncov/?lang=en.
30. A. Vabret, T. Mourez, S. Gouarin, J. Petitjean, F. Freymuth, An outbreak of coronavirus OC43 respiratory infection in Normandy, France, *Clin. Infect. Dis.*, **36** (2013), 985–989.
31. L. J. Cui, C. Zhang, T. Zhang, R. J. Lu, Z. D. Xie, L. L. Zhang, et al., Human coronaviruses HCoV-NL63 and HCoV-HKU1 in hospitalized children with acute respiratory infections in Beijing, China, *Adv. Virol.*, **2011** (2011).
32. F. Y. Zeng, C. W. M. Chan, M. N. Chan, J. D. Chen, K. Y. C. Chow, C. C. Hon, et al., The complete genome sequence of severe acute respiratory syndrome coronavirus strain HKU-39849 (HK-39), *Exp. Biol. Med.*, **28** (2003), 866–873.
33. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *Adv. Neural Inf. Process. Syst.*, **26** (2013), 3111–3119.