AI

MDPI

Article

Attention-Based Hybrid Deep Learning Models for Classifying COVID-19 Genome Sequences

A. M. Mutawa [1,2]

[1] Computer Engineering Department, College of Engineering and Petroleum, Kuwait University, Safat 13060, Kuwait; dr.mutawa@ku.edu.kw
[2] Computer Sciences Department, University of Hamburg, 22527 Hamburg, Germany

Abstract: Background: COVID-19 genetic sequence research is crucial despite immunizations and pandemic control. COVID-19-causing SARS-CoV-2 must be understood genomically for several reasons. New viral strains may resist vaccines. Categorizing genetic sequences helps researchers track changes and assess immunization efficacy. Classifying COVID-19 genome sequences with other viruses helps to understand its evolution and interactions with other illnesses. **Methods**: The proposed study introduces a deep learning-based COVID-19 genomic sequence categorization approach. Attention-based hybrid deep learning (DL) models categorize 1423 COVID-19 and 11,388 other viral genome sequences. An unknown dataset is also used to assess the models. The five models' accuracy, f1-score, area under the curve (AUC), precision, Matthews correlation coefficient (MCC), and recall are evaluated. **Results**: The results indicate that the Convolutional neural network (CNN) with Bidirectional long short-term memory (BLSTM) with attention layer (CNN-BLSTM-Att) achieved an accuracy of 99.99%, which outperformed the other models. For external validation, the model shows an accuracy of 99.88%. It reveals that DL-based approaches with an attention layer can accurately classify COVID-19 genomic sequences with a high degree of accuracy. This method might assist in identifying and classifying COVID-19 virus strains in clinical situations. Immunizations have lowered COVID-19 danger, but categorizing its genetic sequences is crucial to global health activities to plan for recurrence or future viral threats.

Keywords: attention layer; convolutional neural network; COVID-19; deep learning; genome sequencing; sequence classification

Academic Editor: Ioannis Kakkos

Received: 25 October 2024
Revised: 15 December 2024
Accepted: 23 December 2024
Published: 2 January 2025

Citation: Mutawa, A.M. Attention-Based Hybrid Deep Learning Models for Classifying COVID-19 Genome Sequences. *AI* **2025**, *6*, 4. https://doi.org/10.3390/ai6010004

1. Introduction

The COVID-19 pandemic, induced by the SARS-CoV-2 virus, has posed a considerable worldwide health issue [1]. It has resulted in unprecedented death, disease, and economic effects. Given the size of the problem and the fact that the disease's progress is hard to predict, it is essential to find and focus on those most at risk [2]. The pandemic has been mitigated in several regions globally due to the advancement and dissemination of successful vaccinations, including the mRNA-based Pfizer-BioNTech and Moderna vaccines. Notwithstanding these developments, the analysis of COVID-19 genomic sequences remains essential [3].

The genetic code tells us a lot about how the pandemic has changed. It makes it easier to make medicines to fight the virus [4]. With genetic material information, health workers could work out which virus is infecting a patient. Nucleic acid, also called deoxyribonucleic acid (DNA), is the molecule that saves an organism's genetic information and which is

necessary to continue developing and processing. The four nucleotide bases that make up a genome are Adenine (A), G (Guanine), C (Cytosine), and T (Thymine). With this chain of nucleotides and DNA, a ribonucleic acid (RNA) molecule can be made [5]. Global genomic real-time monitoring should be a vital part of any reaction to an outbreak [4].

The Global Initiative for Sharing Avian Influenza Data (GISAID) fosters beneficial collaboration between researchers by collecting and storing genomes worldwide for further comparison [6]. The United States of America (USA) and the United Kingdom (UK) have the highest genome sequence published in GISAID [7]. The multinational HapMap project aims to provide the typical patterns of variability in human genes and how they relate to health, disease, response to therapy, and environmental variables [8,9]. In addition, hundreds of COVID-19 genomes have been given to GISAID, where researchers release data from patient samples from different nations in light of the significant worldwide diversity in the death rate caused by COVID-19 [10–12].

Insights into the virus's origins, modes of transmission, genetic variety, and evolutionary history have been made possible by sequencing the SARS-CoV-2 genome. Comprehending the genomic sequences of SARS-CoV-2 is crucial for several reasons. It enables scientists to observe the virus's development and identify the introduction of new variations, which might affect vaccination effectiveness and public health efforts. Secondly, genetic investigations elucidate the virus's transmission patterns and origins, and therefore inform current and future pandemic responses [13].

Despite the pandemic being managed, ongoing genomic surveillance remains essential to identify possible alterations that may result in vaccine resistance or enhanced transmissibility. As artificial intelligence (AI) solutions have developed, they have become indispensable for handling the ever-growing databases linked with viral genome research [14]. Machine learning (ML) is applied in bioinformatics, which seeks to understand biological data via computing. One of the trickiest parts of genomics is determining how to classify genes as healthy or diseased [15,16].

In this research, we use the COVID-19 genome sequence dataset to develop a classification model with a few different deep learning (DL) techniques. The classification of the genome sequence provides medical professionals with a helpful tool for the early detection of viruses. The primary goals of this investigation are as follows: to carry out genome sequence analysis, which assists in the detection of the COVID-19 genome, and to carry out deep learning models, such as hybrid DL models with attention layers. The different models employed for the study are Bidirectional gated recurrent units (BGRU) with attention (BGRU-Att), Bidirectional long short-term memory (BLSTM) with attention (BLSTM-Att), Convolutional neural network (CNN) with BLSTM-Att (CNN-BLSTM-Att), and CNN-BGRU-Att. The study's most significant contributions are:

- The study proposes a hybrid DL-based approach for efficiently classifying COVID-19 genome sequences using CNN with BLSTM, BGRU, and an attention layer. It is a novel contribution, as existing studies have used CNN with BLSTM models for sequence classification;
- While previous studies may have used k-mer counting with a single k value, the k values employed in this study (3 to 6) may differ from previous works. It could lead to different and potentially better results;
- The study employs the sliding window method to overcome the class imbalance problem. It is a novel contribution, as existing studies of genome classification use other oversampling approaches like the Synthetic Minority Oversampling Technique;
- The study evaluated the proposed approach on an unseen external dataset, adding confidence to the findings and potentially aiding reproducibility.

Thus, the research question for the study is as follows: How does a hybrid attention DL-based approach (BLSTM-Att, BGRU-Att, CNN-BLSTM-Att, and CNN-BGRU-Att) perform accurately in classifying COVID-19 genomic sequences, and how well does it generalize to an unseen external dataset?

The remaining report layout can be broken down into the following sections: Section Two is a background study of genome sequence classification. It is followed by methodology in section three and study results in section four. The discussions of the results with reference to related work are explained in section five, followed by the study's conclusion (section six).

2. Literature Review

To fully grasp the evolution, medical, and epidemiological aspects of COVID-19 and the necessity of early diagnosis and therapy, it is essential to understand the genetic sequence of SARS-CoV-2 [17]. According to Ahmad et al. [18], it is critical to comprehend the COVID-19 genomic alterations that have taken place. In the research presented by Hu et al. [19], the modeling of large input sequences (200 kb) was investigated, and it was shown that the model architecture needed to include self-matched modularization. The second innovation is the establishment of harmony between the predictability of models and their interpretability, which has led to the latter being increasingly relevant in meeting biological criteria.

The DL models help predict the genome sequence and can be used in various contexts, such as customized medicine and the detection of diseases [20]. Genomic sequencing could identify the genotype of the virus in a blood specimen, which can help track and trace potential transmission sources [21]. Methods for diagnosis involve real-time polymerase chain reactions to analyze nucleic acids and viral genome sequencing for the localization of infectious sources. Determining the viral load helps track how the disease progresses [22]. Zhu et al. [23] explain in their study how they created a statistical framework for COVID-19 by using SARS-CoV-2 complete genome sequencing in conjunction with electronic medical records.

Table 1 explains the background papers on ML and DL in genome sequence classification. DL has many applications in sequence analysis, like performing imputation based on correlation with genes, dimensionality reduction with specific algorithms, cell annotations, and computation [24]. COVID-19 data analysis can be accelerated and improved by concerted efforts to promote accessible research and information sharing [25]. Purohit [26] studied the correlation and alignment analysis of different virus genome sequences with COVID-19 and concluded a low annealing temperature with the COVID-19 virus. It is the reason why COVID-19 can be found in a wide variety of forms throughout different countries.

Table 1. Previous studies of AI models for COVID-19 genome sequence.

Reference	Model	Genome Sequence	Results
[27]	DQNN	COVID-19 genome sequences	Accuracy = 94.10%
[28]	MLP	Five different coronaviruses with 10,000 sequences	Accuracy = 97.32%
[29]	SVM	COVID-19 and influenza virus with 107,000 sequences	Accuracy = 99.40%
[30]	NB, SVM, KNN	COVID-19 and non-COVID-19 virus with 7331 sequences	Accuracy = 99.39%

Table 1. *Cont.*

Reference	Model	Genome Sequence	Results
[31]	Capsule network	COVID-19 and non-COVID-19 virus with 10,988 segments	Accuracy = 100.00%
[32]	RF, KNN, SVM, DT	COVID-19 sequence in 6 countries	Accuracy = 98.90%
[33]	Neurochaos Learning	COVID-19 and other viruses (multi-class) with 361 sequences	Accuracy = 99.80%
[34]	CNN, CNN-LSTM, CNN-BLSTM	86,637 sequences with COVID-19 and seven other viruses.	Accuracy = 94.88%
[35]	KNN, SVM	260 sequences of COVID-19 and healthy patients	Accuracy = 98.84%
[36]	XGB, RF, LR, KNN, DT, SVM, NB	300 sequences with COVID-19 and three other viruses.	Accuracy = 97.00%
[37]	SVM, RF, DT, KNN, GB	113,927 protein sequence of COVID-19 and non-COVID-19 viruses	Accuracy = 98.69%
[38]	CNN, CNN-LSTM, CNN-BLSTM	329 sequences of COVID-19 and non-COVID-19 viruses	Accuracy = 99.95%
[39]	KNN, RF, DT, SVM	1582 sequences of COVID-19 and non-COVID-19 viruses	Accuracy = 97.47%
[40]	CNN, CNN-LSTM, CNN-BLSTM	66,153 sequences of COVID-19 and five other viruses	Accuracy = 93.16%
[41]	SVM, RF, KNN, DT, AB, MLP	1615 sequences of COVID-19 and non-COVID-19 viruses	Accuracy = 99.80%
[42]	SVM, KNN, NB, RF, DT	1334 sequences of COVID-19 and non-COVID-19 viruses	Accuracy = 93.00%
[43]	LR, KNN, SVM, DT, RF	9238 COVID-19 with 27 countries	Accuracy = 100.00%

Abbreviations: DQNN: Deep quantum neural network, AUROC: area under the receiver operating characteristic curve, AUC: area under the curve, ROC: receiver operating characteristic, DT: Decision tree, SVM: Support vector machine, MLP: Multi-layer perceptron, KNN: K-nearest neighbors, CNN: convolutional neural network, LR: Logistic regression, AB: Ada boosting, XGB: extreme gradient boosting, RF: Random forest, NB: naïve Bayes, LSTM: long short term memory, BLSTM: bi-directional LSTM, LD: linear discriminant, DNA: deoxyribonucleic acid, RNA: ribonucleic acid, HIV: human immunodeficiency virus, HCV: hepatitis C virus.

Most previous studies have not used external data to test the prediction. According to Riley et al. [44], the model should be externally tested on new data to ensure that it is reliable and works well on real data.

3. Materials and Methods

The methodology of the study is illustrated in Figure 1. Python (version 3.9) with TensorFlow (version 2.10) and Scikit-Learn (version 1.0) library was used in this work [45,46]. For genome sequence analysis the Biopython (version 1.79) tool was used [47].

3.1. Dataset and Data Balancing

The genome sequence dataset was collected from open-source data [48]. It contained 1557 COVID-19 genome sequences (Label-0) and 11540 other viruses' genomes (Label-1). After cleaning the dataset, the total count consisted of 12,811 with 1423 Label-0 and 11,388 Label-1 sequences. The average sequence length of the COVID-19 genome was 29,837, and that of the other viruses was 15,789. The classes are highly imbalanced, with Label-0 having a lesser sequence count than Label-1.

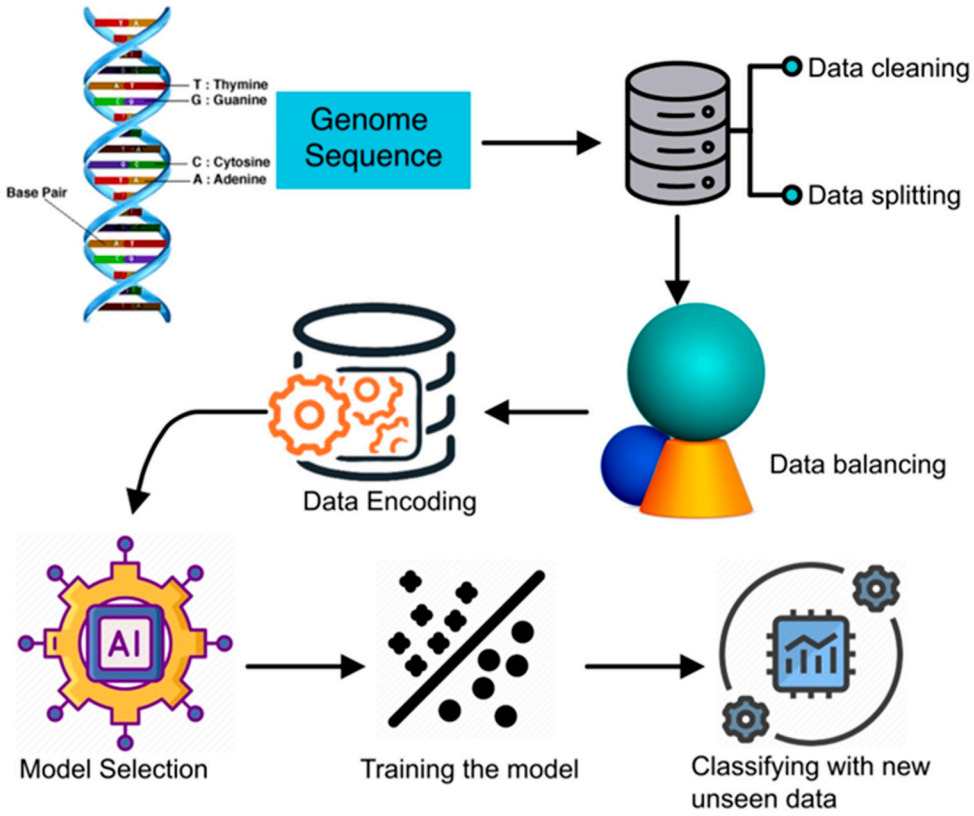

Figure 1. Methodology of the study.

To balance the class frequencies, a sliding window method was used to create a new set of minority classes (Label-0) [49]. The sliding window from the starting position of a COVID-19 sequence was used to extract a window of nucleotides. This window could then be replaced with a new sequence of nucleotides generated by a random DNA sequence generator. The process could be repeated, sliding the window along the DNA sequence until it reached the complete length and replaced the nucleotides within the window with new, randomly generated sequences. The new sequences were developed with a size of 10,000. After generating a new series based on COVID-19, the total count for Label-0 changed to 11,375, which resulted in a complete sequence count of 22,763. This dataset was used for training and validating the model with a ratio of 70:30.

The external validation was based on a new unseen dataset downloaded from the National Center for Biotechnology Information (NCBI), which has complete DNA sequences for viruses and is accessible to the public [50]. The collected DNA sequence databases include COVID-19, MERS, Dengue, Ebola, Influenza, and Rota. Since the trained model is binary classification, viruses other than COVID-19 were grouped into one and labeled as Label-1. Table 2 presents the count of every virus in the external dataset. The total count was 4622.

Table 2. Detail of genome sequence external dataset.

Virus	Total Sequences	Maximum Length	Minimum Length
COVID-19	722	29,903	29,454
Dengue	700	10,736	202
Ebola	600	19,043	587
MERS	1000	30,484	110
Influenza	1000	2396	159
Rota	600	3538	441

It is common for real-world datasets to have class imbalances, meaning that some classes may have significantly more or fewer samples than others. In such cases, it is often necessary to balance the class distribution in the training data to ensure that the model is not biased toward the majority class. However, when testing the model on unseen data, it is unnecessary to balance the class distribution as it reflects the real-world distribution.

3.2. Data Pre-Processing

The character sequence was encoded with *k*-mer counting. *K*-mer counting is used in sequence read error correction, metagenomic sequencing, and genome and transcriptome assembly. *K*-mers are simply length *k* subsequences. Equation (1) displays the sequence's overall length after *k*-mer counting.

$$Total\ sequence\ length = L - k + 1 \quad (1)$$

where *k* is the length given in the *k*-mer, and *L* is the overall length of the input sequence. For a DNA sequence, *n* is 4 with four nucleotides: A, C, G, and T; and *k* represents the sequence's potential monomers.

After applying the *k*-mer counting encoding method, the raw sequence was converted to English-like statements. For example, consider a random sequence as 'GGAAAATC-TATTGGT.' Then a window of length three is made, and one character is moved from left to right at a time. So, the sequence is split into 'GGA', 'GAA', 'AAA', 'AAA', 'AAT', 'ATC', 'TCT', 'CTA', 'TAT', 'ATT', 'TTG', 'TGG', and 'GGT'. So, the total sequence length of *k*-mer counting is 15 − 3 + 1 = 13. Here, in the example, 13 sequences are generated. In the proposed study, we used *K*-mer counting with *k* value ranges from 3 to 6. The best performance of the model based on each *k* value was evaluated. The countVectorizer function in the Scikit-Learn module was then used to vectorize each English-like sequence using the character-level analyzer. The data were split into 70% training and 30% testing. The model was externally evaluated on the new unseen data.

3.3. Deep Learning Models

BLSTM-Att, BGRU-Att, CNN-BLSTM-Att, and CNN-BGRU-Att are the DL models utilized in this study. BLSTM is architecture for recurrent neural networks (RNNs) that combines forward and reverse information flows to improve the accuracy of sequence classification tasks. It is an expansion of the standard long short-term memory (LSTM) model that handles sequence inputs in both models' routes. The processing of the input sequence by the forward LSTM in the order in which it was supplied produces a hidden state sequence. The reverse LSTM oppositely processes the input sequence, and the hidden state sequence is created as a result. The output of the BLSTM model is produced by appending the forward and backward sequences of the hidden states. BLSTM is better than standard LSTM because it can acquire contextual information from both the input sequence's past and future states. As a result, it is more effective when used for sequence classification tasks. In natural language processing (NLP), BLSTM has been used for tasks including part-of-speech tagging, text categorization, and sentiment analysis [51,52].

Because of their ability to accurately describe sequential data with long-term dependencies, BGRU are widely used in various industries. BGRU are an extension of Gated Recurrent Units (GRUs) that includes two layers of GRUs operating in opposing directions. This allows the model to capture forward and backward dependencies in the input sequence. The GRU is a specialized form of the RNN that has gating mechanisms. These mechanisms allow the network to selectively update and reset its hidden state based on the input at each time step. BGRU models can capture each input token's past and future

contexts because they are constructed using two GRU layers stacked in opposite directions. This enables them to learn more complicated relationships in the input sequence [53,54].

Convolutional neural networks, more often referred to as CNNs, have garnered significant interest in recent years because they are so good at various NLP tasks, including the classification of text [55]. CNNs were initially designed to perform tasks associated with image processing; nevertheless, their performance in NLP tasks has been attributed to their ability to acquire hierarchical and local properties from the input data. CNNs can learn these properties, which helps them perform well in natural language processing (NLP) tasks [56].

This research uses a hybrid model that incorporates a CNN layer to extract features from the input sequence, which is then followed by a BGRU or BLSTM layer. Upon inputting a sequence into the model, it is initially transformed into an integer index using a tokenizer. This index pertains to a certain character inside the vocabulary. The Embedding layer functions as a lookup table that associates integer indices with their respective dense vector representations. If the input index for a character is 5, the Embedding layer extracts the 5th row from the embedding matrix, which holds the dense vector corresponding to that character. A convolutional layer, a max pooling layer, a BGRU layer or a BLSTM layer, an attention layer, and a dense layer are included in the model, and their purpose is to categorize the target label. The complete summary of the models is depicted in Figure 2.

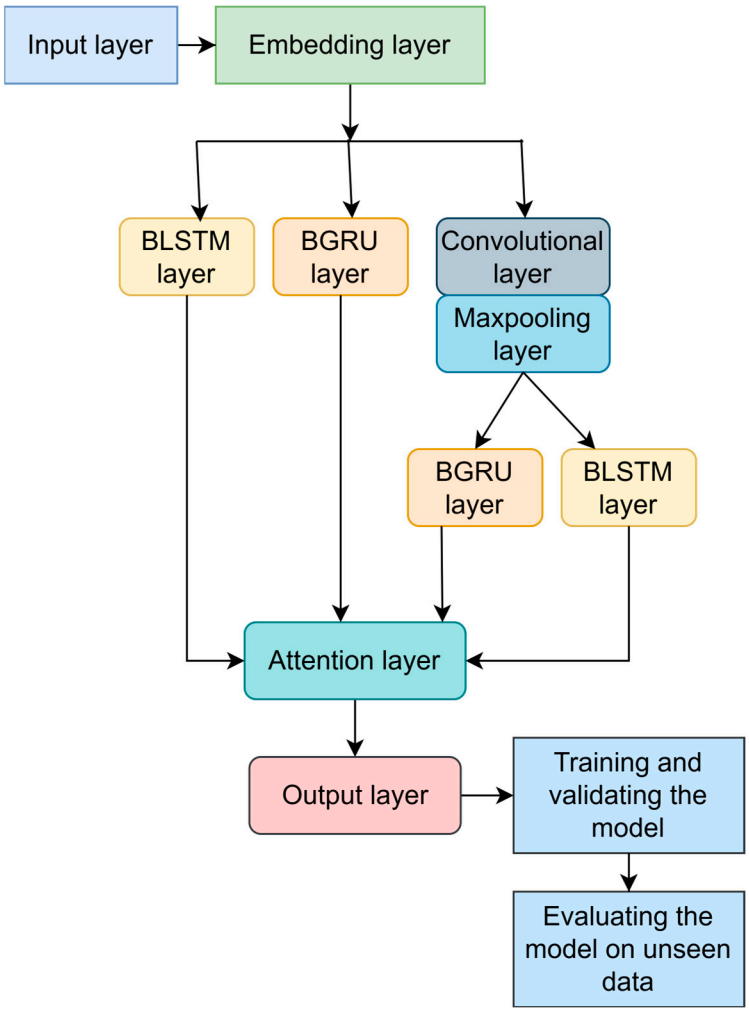

Figure 2. The proposed study architecture for sequence classification.

DL uses "attention" to boost the efficiency of models that operate on sequential data, such as text or time-series information. Introduced by Vaswani et al. (2017) in "Attention is All You Need," the attention mechanism transformed sequence modeling by allowing models to dynamically focus on the most pertinent sections of input sequences [57]. Attention may be utilized to increase performance [58–60]. During the process of making a forecast, the purpose of attention is to enable the model to focus on the aspects of the input sequence that are most significant to the task. When applied to neural networks, attention may be understood as a system that learns a set of weights over the input sequence. These weights indicate the relative significance of each component in the sequence concerning the prediction job. After that, these weights are utilized to compute a weighted sum of the input sequence. This weighted sum is then sent to the subsequent layer of the model in the form of input data, enabling the model to emphasize the most relevant areas of the sequence.

To guarantee interpretability in the proposed study, a single-layer attention mechanism is utilized. Attention is included before the model's final output layer and after the CNN's feature extraction stage. This placement improves classification performance by allowing the model to concentrate on the most relevant characteristics that were taken from the genomic data. The attention layer's scoring mode is the dot-product or Luong-style attention. The Attention layer receives the query (Q) and key(K)-value(V) pair. The query and key-value pair are obtained from the RNN layer output. Hence, Q = K = V, which sets up the self-attention mechanism. This approach emphasizes temporal or geographical aspects in genomic data by focusing on connections within the same sequence. At each timestep, the bidirectional RNN collects sequence dependencies and produces rich feature representations. These characteristics feed into the attention mechanism, so the model learns to prioritize the sequence depending on the task. The attention score (AS) and output are defined as:

$$AS = Q.K^T \tag{2}$$

$$Attention\ output = softmax\left(\frac{AS}{\sqrt{d_k}}\right).V \tag{3}$$

where Q is the query vector, K is the key vector, V is the value vector, T is the sequence length, and d_k is the dimensionality of the bidirectional output. The complete architecture is explained in Figure 3.

The parameters of the models used in this work are mentioned in Table 3. The number of units in the CNN layer was set as 128 with a 2 × 2 kernel size. For this study, only one layer was evaluated for CNN, BGRU, and BLSTM models. All other parameters employed for the CNN, BGRU, and BLSTM were the default values. The output space dimensionality for BLSTM and BGRU was set to 64. Thus, the attention layer operated on a 128-dimensional space.

Table 3. Parameters of the model.

Parameters	Value
Loss	binary_crossentropy
Epochs	50
Activation	Sigmoid
Optimizer	Adam
learning rate	0.001
Batch size	64
EarlyStopping	Validation loss, patience = 5
ReduceLRonPlateau	Validation loss, patience = 2

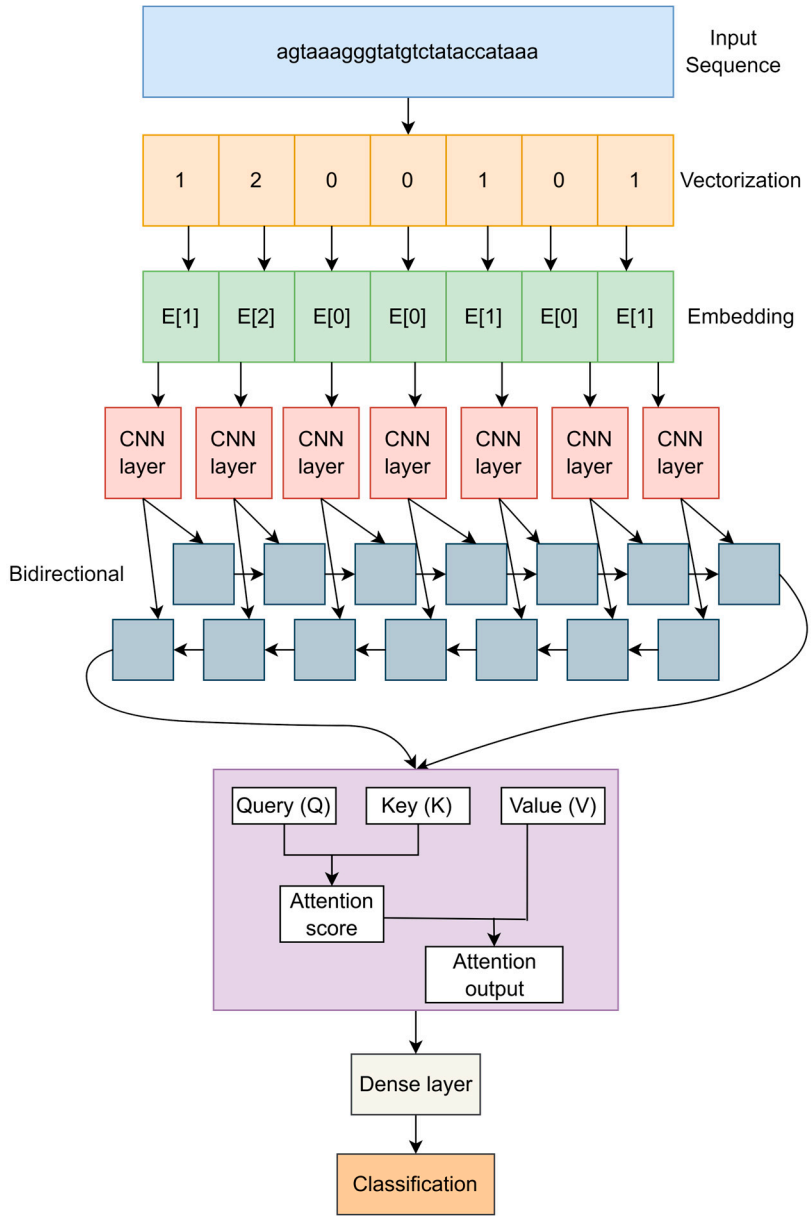

Figure 3. The workflow of each layer in the hybrid CNN model. The embedding layer will be given as the input for bidirectional models without the CNN layer.

The tuning parameter is the learning rate. The EarlyStopping and ReduceLROnPlateau were used to overcome the overfitting problem in DL models. The epoch was set to 50, and the patience for ReduceLROnPlateau was set to 2. If the validation loss was stable or increasing, the learning rate parameter was updated to a factor of 0.1. (The initial learning rate was set to 0.001.) The model exited the training if the validation loss in EarlyStopping was not decreasing for five epochs. The pseudo-code for the proposed study is mentioned in Figure 4.

3.4. Evaluation Metrics

Accuracy, precision, recall, and f1-score were the criteria used to analyze the classification result. In addition, the confusion matrix for each approach was considered. Data that were not evenly distributed may not have been correctly measured using accuracy [61]. As a direct consequence of this, f1-score, precision, and recall were also utilized [62]. A confusion matrix was employed to evaluate performance measures based on true positives (trPos), true negatives (trNeg), false positives (faPos), and false negatives (faNeg). A cal-

culation was made to determine how many classified samples were relevant, known as precision (as shown in Equation (4)). Recall determined how many relevant samples were classified (Equation (5)). F1-score is the mean of recall and precision (Equation (6)) and the accuracy of the model in Equation (7). The Matthews Correlation Coefficient (MCC), as shown in Equation (8), is a statistic utilized to assess the efficacy of binary classifications. It offers a balanced metric applicable even when the classes vary significantly in size.

$$Precision = \frac{trPos}{trPos + faPos} \tag{4}$$

$$Recall = \frac{trPos}{trPos + faNeg} \tag{5}$$

$$F1\text{-}Score = \frac{2trPos}{2trPos + faPos + faNeg} \tag{6}$$

$$Accuracy = \frac{trPos + trNeg}{trPos + trNeg + faPos + faNeg} \tag{7}$$

$$MCC = \frac{trPos.trNeg - faPos.faNeg}{\sqrt{(trPos + faPos)(trPos + faNeg)(trNeg + faPos)(trNeg + faNeg)}} \tag{8}$$

The ROC Curve, also known as the Receiver Operating Characteristic Curve, is an efficient method for determining the accuracy of binary classifiers. The behavior of the classifier may be understood by graphing the True Positive Rate (TPR) and the False Positive Rate (FPR) concerning each threshold. As the model better classifies the data, the ROC curve moves closer and closer to the top left corner. To determine how much of the graphic falls inside the range of the curve, we computed the AUC (area under the curve). The model is better when the AUC becomes closer to one [63].

Input: The genome sequence based on COVID-19 and other-viruses
Output: The classification labels and its performance metrics

begin
Step 1: Load the dataset
Step 2: Preprocess the data including cleaning, normalization, encoding
Step 3: Splitting the data to training and validation
Step 4: Balance the training data using sliding window method
Step 5: Define the model architecture, optimizer and loss function
Step 6: Define the EarlyStopping parameter
Step 7: Train and validate the model
Step 8: Monitor the evaluation metrics
Step 9: Test the model with unseen external data and calculate the performance metrics
end

Figure 4. The pseudo-code for the proposed genome sequence classification study.

4. Results

In the study, the data were split into 70–30 methods for training and validation. The models employed in this work consist of four DL models: BLSTM-Att, BGRU-Att, CNN-BLSTM-Att, and CNN-BGRU-Att. The k-mer counting with k values ranging from 3 to 6 was tuned for each model. The model was checked on in respect to unseen new data to evaluate whether the model's classification was reliable [44]. The platforms employed in this study were Nvidia GeForce GTX 1080 Ti (19 GB memory) and Nvidia Titan V (20 GB memory). Python 3.9, TensorFlow 2.10, Scikit-learn 1.0, and BioPython 1.79 were the

library packages employed to build the models. There are 1423 COVID-19 sequences and 11,388 genome sequences from different viruses that make up the complete dataset. The sample nucleotides in DNA from COVID-19 and different virus categories are specified in Figures 5 and 6. The letters A (Adenine), G (Guanine), C (Cytosine), and T (Thymine) are referred to as the DNA's four bases. Figure 5 demonstrates that a random sample sequence taken from the COVID-19 target label has 8954 instances of adenine, 9594 instances of thymine, 5863 instances of guanine, and 5492 instances of cytosine.

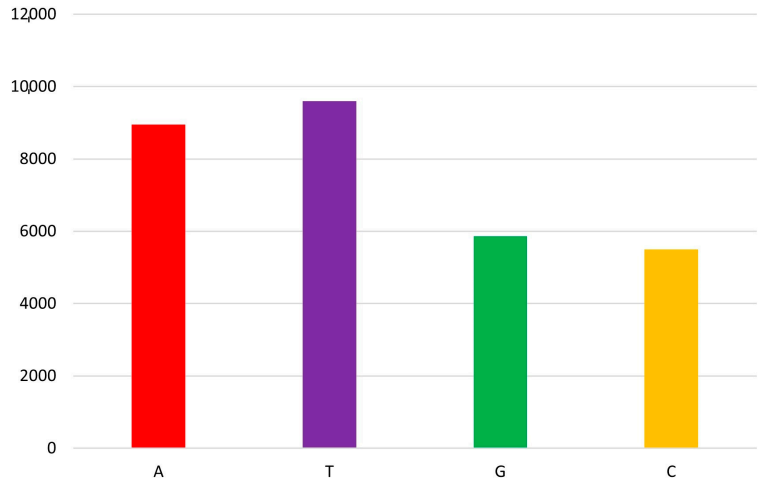

Figure 5. The DNA nucleotide on a random sample of COVID-19 genome sequence.

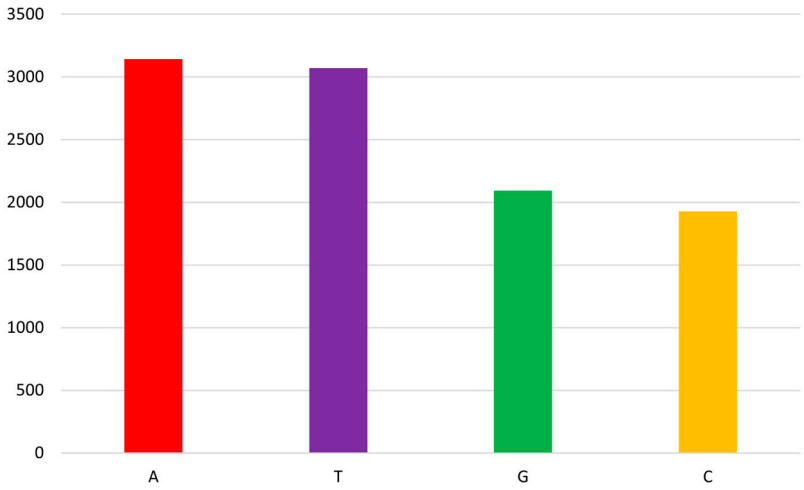

Figure 6. The DNA nucleotide on a random sample of other viruses' genome sequence.

Similarly, other viral labels' random sample genome sequences have also been given in Figure 6. It reveals that adenine has a count of 3143, guanine has a count of 2092, thymine has a count of 3070, and cytosine has a count of 1927. Figures 5 and 6 provide a comparison examination of the nucleotide composition (counts of A, T, G, and C) between random sequences of COVID-19 and other viral genomes. This is essential for comprehending the genetic traits and variations that may occur across various virus strains. Visualizing nucleotide counts assists in understanding possible changes in genomic structure that may affect the behavior of these viruses.

The comparison of two sequences can be performed using a dot matrix. It functions based on the DNA sequence, and it will map a dot whenever it detects a match on the base of the DNA sequence. In most cases, understanding the sequence alignment better is beneficial. Figure 7 shows a sample dot matrix of COVID-19 and other viruses' genome

sequences. This form of representation can assist in recognizing patterns in the sequences, such as those that are similar or that repeat themselves. Only the first 20 nucleotide bases of the random sequence are shown here since its length is too great to be shown in its entirety.

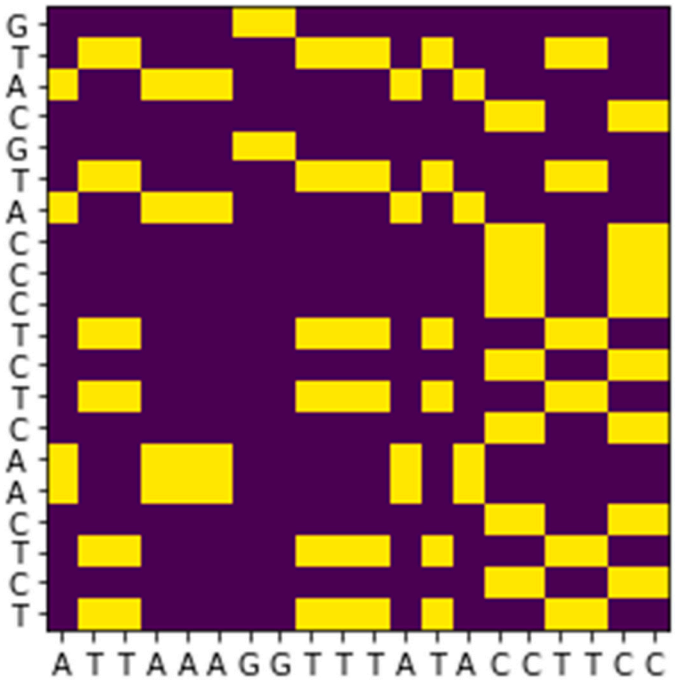

Figure 7. Dot plot of the genome sequence of COVID-19 and other viruses. A yellow color indicates matching nucleotides.

Classification Results

The genome sequence classification used K-Mer counting with a countVectorizer for encoding the data. The k size of the K-Mer counting was used as a tuning parameter. The DL models were BLSTM-Att, BGRU-Att, CNN-BLSTM-Att, and CNN-BGRU-Att. The parameters were tuned to obtain better performance for each model. The tuning parameter is the learning rate. Since the models reach the best accuracy with one hidden layer, it was not tuned further, as increasing the hidden layer increases the time to fit the model.

Table 4 shows the accuracy of the validation data for the k-mer counting method. Looking at the metrics, we can see that the models generally performed well across the board. For all the metrics, higher values indicate better performance. The models consistently achieved high accuracy, with values above 0.99 for most cases. AUC values are also high, meaning solid predictive ability. CNN-BLSTM-Att showed the highest accuracy of 99.93% with k values of 6 and 5. For the corresponding k value, both the models achieved high precision, recall, and f1-score values (0.9991, 0.9994, and 0.9993). The AUC score is higher for CNN-BLSTM-Att, with a value of 0.9993. It indicates that the model accurately distinguished between positive and negative samples or correctly ranked the predicted probabilities for the classes.

To evaluate the model performance based on all four models and k-mer values, analysis of variance (ANOVA) was employed in this study. The metrics, accuracy, and AUC are used here to analyze performance. The null hypothesis states no significant difference between the means of measuring parameters according to the four DL models. The *p*-value, according to the model, is depicted in Table 5. The null hypothesis is rejected if the *p*-value is less than 0.05. The measures other than recall specify a significant difference between the mean values. It depicts that one group is different from the other. The Tukey test was performed to identify which group is diverse [64].

Table 4. Classification results of all models with different k-mer methods for validation.

Model	k-mer	Accuracy	AUC	Precision	Recall	F1-Score
BLSTM-Att	k = 3	0.9970	0.9971	0.9965	0.9976	0.9970
	k = 4	0.9949	0.9949	0.9920	0.9976	0.9949
	k = 5	0.9980	0.9979	0.9982	0.9976	0.9980
	k = 6	0.9980	0.9981	0.9970	0.9991	0.9980
BGRU-Att	k = 3	0.9938	0.9939	0.9906	0.9970	0.9938
	k = 4	0.9943	0.9943	0.9903	0.9982	0.9943
	k = 5	0.9962	0.9962	0.9962	0.9962	0.9962
	k = 6	0.9953	0.9953	0.9912	0.9994	0.9953
CNN-BLSTM-Att	k = 3	0.9968	0.9968	0.9959	0.9976	0.9968
	k = 4	0.9980	0.9981	0.9973	0.9988	0.9980
	k = 5	0.9993	0.9993	0.9991	0.9994	0.9993
	k = 6	0.9993	0.9993	0.9991	0.9994	0.9993
CNN-BGRU-Att	k = 3	0.9965	0.9965	0.9953	0.9976	0.9965
	k = 4	0.9966	0.9966	0.9944	0.9988	0.9966
	k = 5	0.9980	0.9981	0.9973	0.9988	0.9980
	k = 6	0.9985	0.9985	0.9976	0.9994	0.9985

Table 5. The ANOVA results for analyzing the model's performance.

Factor	Response	*p*-Value
Model	Accuracy	0.01
	AUC	0.01
	Precision	0.02
	Recall	0.362
	F1-score	0.01

The results of the Tukey test are shown in Figure 8. It shows that the models, CNN-BLSTM-Att, and BGRU-Att, are significantly different. All other models are not different according to the mean value. So, the CNN-BLSTM-Att is considered the best model with 99.93% accuracy.

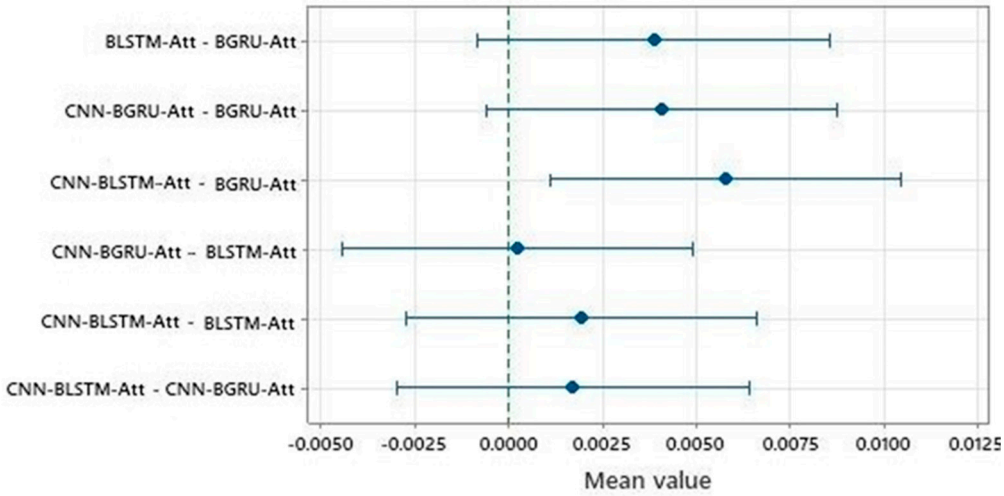

Figure 8. Difference of means using the Tukey test.

5. Discussion

The open-source genome sequence dataset with 1423 COVID-19 sequences and 11,388 other virus sequences was considered for classifying the genome, which can help

health workers understand the COVID-19 virus and infection. This study employed a sliding window method to solve the class imbalance problem during training. The k-mer method with countVectorizer was used to encode and vectorize the sequence. The k value, 6, showed the highest performance, as shown in Table 4. The result shows 99.93% accuracy with the CNN-BLSTM-Att model for classifying the test sequence. The model better classifies the sequence of COVID-19 and other viruses.

The suggested models were evaluated across different k values to find the best balance between classification accuracy and overfitting. The results found that k = 6 had the greatest accuracy and generality for the genomic sequences studied, which is a good balance. In genomic sequence analysis, the k-mer size decision is a crucial factor as it directly affects the data representation and, thus, the model's classification performance. Smaller k values, like k = 3, capture local patterns; greater k values, such as k = 6, allow deeper contextual representation.

The study analyzed how well the models performed on different datasets. The external data had 722 COVID-19 sequences and 3922 other virus sequences (Table 2). The data were not balanced as they reflected real-world distribution. The results are depicted in Table 6. The CNN-BLSTM-Att performed better with an AUC score of 0.9886, accuracy of 99.61%, 100% precision, a 0.9772 recall value, and an f1-score of 0.9885. The MCC score is a robust metric for binary classification with class imbalance. The CNN-BLSTM-Att showed a 0.9863 MCC score, which indicates a perfect prediction.

Table 6. Classification result of unseen external data.

Model	k-mer	Accuracy	AUC	Precision	Recall	F1-Score	MCC
BLSTM-Att	k = 3	0.9566	0.8727	1.0000	0.7454	0.8541	0.8416
	k = 4	0.9656	0.8996	0.9982	0.7994	0.8878	0.8753
	k = 5	0.9690	0.9090	1.0000	0.8180	0.8998	0.8879
	k = 6	0.9702	0.9125	1.0000	0.8250	0.9041	0.8924
BGRU-Att	k = 3	0.9615	0.8886	0.9945	0.7781	0.8731	0.8597
	k = 4	0.9855	0.9579	0.9984	0.9160	0.9555	0.9481
	k = 5	0.9624	0.8898	1.0000	0.7795	0.8761	0.8636
	k = 6	0.9920	0.9771	0.9985	0.9545	0.9760	0.9716
CNN-BLSTM-Att	k = 3	0.9580	0.8775	0.9981	0.7553	0.8600	0.8470
	k = 4	0.9884	0.9659	1.0000	0.9317	0.9647	0.9586
	k = 5	0.9940	0.9822	1.0000	0.9644	0.9819	0.9785
	k = 6	0.9961	0.9886	1.0000	0.9772	0.9885	0.9863
CNN-BGRU-Att	k = 3	0.9730	0.9227	0.9950	0.8464	0.9147	0.9030
	k = 4	0.9852	0.9566	1.0000	0.9132	0.9546	0.9472
	k = 5	0.9876	0.9784	0.9630	0.9644	0.9638	0.9563
	k = 6	0.9925	0.9780	1.0000	0.9559	0.9775	0.9733

The AUC score and the ROC curve of the external dataset evaluated on four models are illustrated in Figure 9. From the plot, the CNN-BLSTM-Att model shows better classification with an AUC score of 0.9886 and k = 6. The AUC score of CNN-BGRU-Att is 0.9780, of BLSTM-Att is 0.9125, and of BGRU-Att is 0.9771.

5.1. Cross-Validation Results

A five-fold cross-validation (CV) was executed, wherein the dataset was partitioned into five equal-sized folds. In each iteration, one fold served as the validation set, while the remaining four folds were utilized for training. This procedure was executed five times, guaranteeing that each fold functioned as the validation set precisely once.

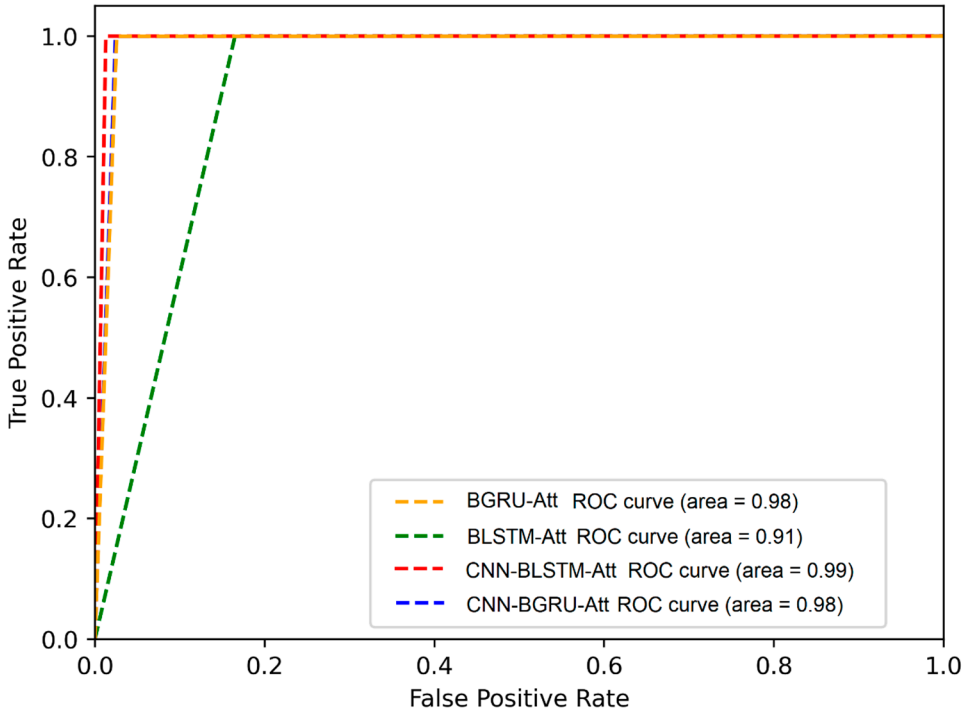

Figure 9. Receiver operating characteristic plot with k-mer value = 6.

It was performed on the best k value in k-mer (k = 6 from the results obtained in Table 6) to reduce the bias from the train/validation split. Examining various k values may yield further insights into the model's efficacy and its responsiveness to diverse k-mer sizes. This issue will be tackled in future studies, by performing a more thorough examination of k values ranging from 3 to 6, which might potentially strengthen the validity of our conclusions.

For each fold, the epoch was set to 25. The average accuracy from all five folds of all models reached a performance of 99%. The best result was achieved by the CNN-BLSTM-Att model and each fold's accuracy, and loss is plotted in Figure 10. The average accuracy from all the folds for the CNN-BLSTM-Att model was 99.99%.

The unseen data were evaluated on the best model and are depicted in Table 7. The CNN-BLSTM-Att model outperformed all other models with an accuracy of 99.88%, an AUC score of 0.9965, and an MCC score of 0.9957.

Table 7. Classification result of unseen data based on 5-fold CV with k = 6.

Models	Accuracy	AUC	Precision	Recall	F1-Score	MCC
BLSTM-Att	0.9915	0.9965	0.9873	0.9957	0.9915	0.9897
BGRU-Att	0.8904	0.9313	0.6096	0.8694	0.7554	0.7240
CNN-BLSTM-Att	0.9988	0.9965	1.0000	0.9929	0.9964	0.9957
CNN-BGRU-Att	0.9976	0.9957	0.9929	0.9929	0.9929	0.9957

5.2. Comparison of Performance with Previous Studies

Khodaei et al. [29] showed an accuracy of 99.4% with influenza and the COVID-19 virus. This was achieved using the SVM model. In another study by Hammad et al. [30], the sequence classification using KNN achieved 99.39% accuracy with first and second-order extracted features. Similarly, Bihter Das [35] has shown that extracting and selecting features from DNA sequences and model evaluation using SVM and KNN performs better in classifying COVID-19 and normal sequences. The CNN combined with BLSTM and LSTM was analyzed by Whata et al. [38] with 329 sequences, which achieved 99.95% accuracy.

Most previous studies have not used external data to test the prediction. Table 8 shows the comparison of the proposed research in relation to prior studies.

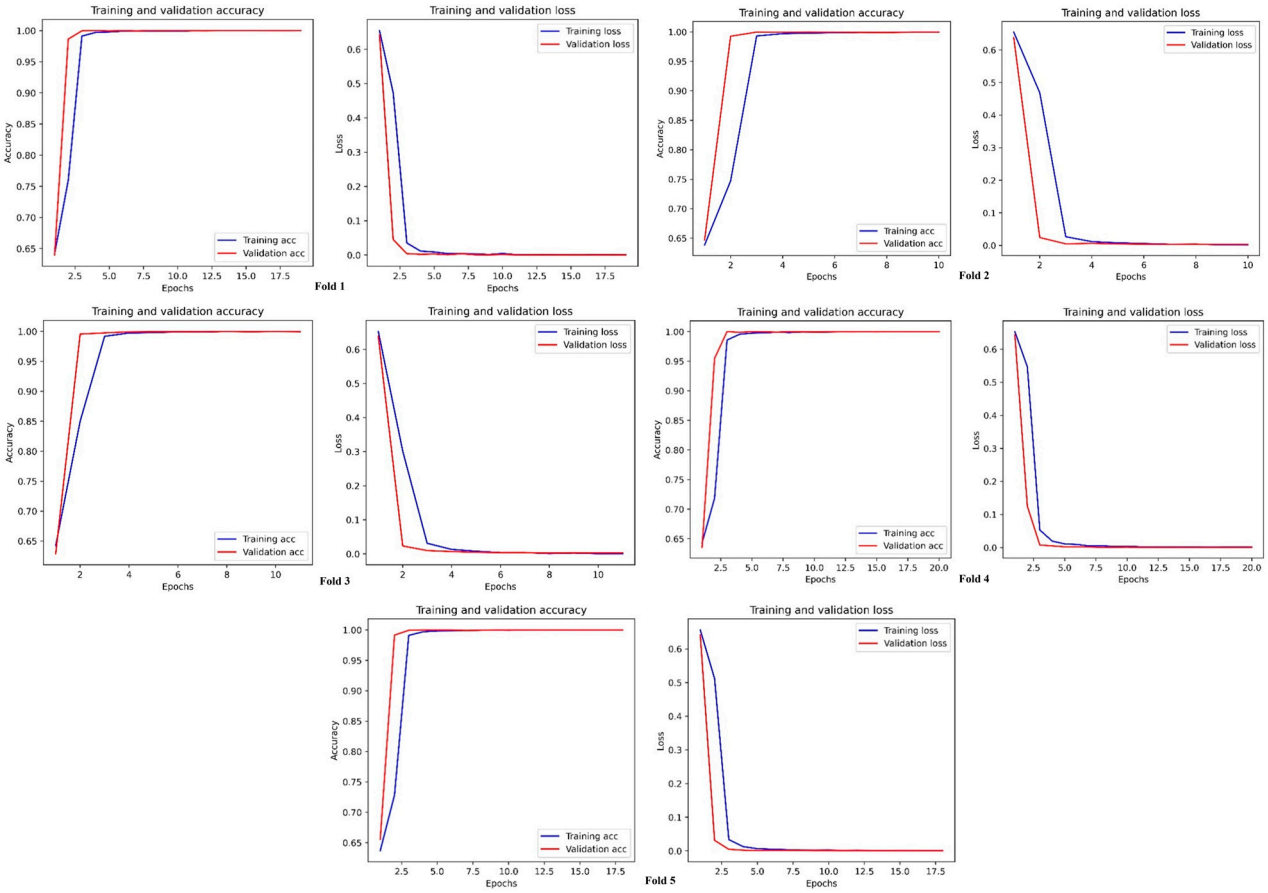

Figure 10. The 5-fold CV results for training and validation set with CNN-BLSTM-Att model.

Table 8. A comparison of the proposed research in relation to prior studies with binary classification.

Authors	Sequence Length	Method	Accuracy	External Test Sequence Length	External Test Accuracy
Khodaei et al. [29]	107,000	SVM	99.40%	-	-
Hammad et al. [30]	7331	KNN	99.39%	-	-
B. Das and S. Toraman [31]	10,988	Capsule network	100%	-	-
Bihter Das [35]	260	SVM	98.84%	-	-
Alkady et al. [37]	113,927	RF	98.69%	-	-
Whata et al. [38]	329	CNN-BLSTM	99.95%	-	-
Singh et al. [39]	1582	RF	97.47%	-	-
H. Arslan [41]	1615	KNN	99.8%		
Proposed study	12,811	CNN-BLSTM-Att	99.99%	4622	99.88%

5.3. Limitations and Future Work

The study employed a novel hybrid DL model with an attention mechanism and used external data for validation. Different k values were also tuned for the k-mer encoding method. Also, a sliding window approach was used for class imbalance problems. Even though the work had these advantages, there were limitations, such as dataset size. Future studies could benefit from larger and more diverse datasets to enhance the robustness of the classification models. It would enable researchers to determine the generalizability and

adaptability of the models to different viral pathogens, thereby enhancing their potential applications beyond COVID-19.

6. Conclusions

The study proposed a DL-based approach for efficiently classifying COVID-19 genome sequences, aiming to address the urgent need for accurate and efficient methods in classifying the virus strains. Four DL models, namely, BLSTM-Att, BGRU-Att, CNN-BLSTM-Att, and CNN-BGRU-Att, and two different datasets were analyzed for classifying the COVID-19 genome sequences. The training and testing data with 12811 sequences were first classified with an accuracy of 99.99% by CNN-BLSTM-Att and a k-mer value of six. Then new unseen data were tested on the models as an external validation, with 4662 sequences, and an accuracy of 99.88% was achieved. This showcases the efficacy of DL-based approaches with attention layers in accurately classifying COVID-19 genomic sequences. The high degree of accuracy achieved suggests the potential of implementing this strategy in clinical settings to aid in identifying and classifying future pandemic responses. Although the study attained encouraging outcomes with the existing dataset, augmenting it to encompass a broader range of viral genomes might substantially enhance the model's efficacy. This will not only improve the model's generalization capabilities but also augment the expanding corpus of knowledge on genetic diversity.

Funding: This work was funded by the Kuwait Foundation for the Advancement of Sciences (KFAS) under grant number PN20-15QE01/CORONA PROP 70.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: [48,50].

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Riou, J.; Althaus, C.L. Pattern of early human-to-human transmission of Wuhan 2019 novel coronavirus (2019-nCoV), December 2019 to January 2020. *Eurosurveillance* **2020**, *25*, 2000058. [CrossRef] [PubMed]
2. Supady, A.; Curtis, J.R.; Abrams, D.; Lorusso, R.; Bein, T.; Boldt, J.; Brown, C.E.; Duerschmied, D.; Metaxa, V.; Brodie, D. Allocating scarce intensive care resources during the COVID-19 pandemic: Practical challenges to theoretical frameworks. *Lancet Respir. Med.* **2021**, *9*, 430–434. [CrossRef] [PubMed]
3. Mayoclinic. COVID-19 and Related Vaccine Development and Research. Available online: https://www.mayoclinic.org/diseases-conditions/history-disease-outbreaks-vaccine-timeline/covid-19 (accessed on 21 October 2024).
4. Lancet, T. Genomic sequencing in pandemics. *Lancet* **2021**, *397*, 445. [CrossRef] [PubMed]
5. Divya, S.; Bhavani, Y.; Thota, M.K. A Survey on Genomic Dataset for Predicting the DNA Abnormalities Using Ml. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 311–319. [CrossRef]
6. Shu, Y.; McCauley, J. GISAID: Global initiative on sharing all influenza data–from vision to reality. *Eurosurveillance* **2017**, *22*, 30494. [CrossRef]
7. Furuse, Y. Genomic sequencing effort for SARS-CoV-2 by country during the pandemic. *Int. J. Infect. Dis.* **2021**, *103*, 305–307. [CrossRef]
8. Bogner, P.; Capua, I.; Lipman, D.J.; Cox, N.J. A global initiative on sharing avian flu data. *Nature* **2006**, *442*, 981. [CrossRef]
9. Gibbs, R.A.; Belmont, J.W.; Hardenbol, P.; Willis, T.D.; Yu, F.; Yang, H.; Ch'ang, L.-Y.; Huang, W.; Liu, B.; Shen, Y. The international HapMap project. *Nature* **2003**, *426*, 789–796. [CrossRef]
10. Cortés-Ciriano, I.; Lee, J.J.-K.; Xi, R.; Jain, D.; Jung, Y.L.; Yang, L.; Gordenin, D.; Klimczak, L.J.; Zhang, C.-Z.; Pellman, D.S. Comprehensive analysis of chromothripsis in 2,658 human cancers using whole-genome sequencing. *Nat. Genet.* **2020**, *52*, 331–341. [CrossRef]

11. Murthy, S.; Gomersall, C.D.; Fowler, R.A. Care for critically ill patients with COVID-19. *JAMA* **2020**, *323*, 1499–1500. [CrossRef]
12. Li, X.; Geng, M.; Peng, Y.; Meng, L.; Lu, S. Molecular immune pathogenesis and diagnosis of COVID-19. *J. Pharm. Anal.* **2020**, *10*, 102–108. [CrossRef] [PubMed]
13. Saravanan, K.A.; Panigrahi, M.; Kumar, H.; Rajawat, D.; Nayak, S.S.; Bhushan, B.; Dutt, T. Role of genomics in combating COVID-19 pandemic. *Gene* **2022**, *823*, 146387. [CrossRef] [PubMed]
14. Suster, C.J.E.; Pham, D.; Kok, J.; Sintchenko, V. Emerging applications of artificial intelligence in pathogen genomics. *Front. Bacteriol.* **2024**, *3*, 1326958. [CrossRef]
15. Dixit, P.; Prajapati, G.I. Machine learning in bioinformatics: A novel approach for dna sequencing. In Proceedings of the 2015 Fifth International Conference on Advanced Computing & Communication Technologies, Haryana, India, 21–22 February 2015; pp. 41–47.
16. Huo, D.; Wang, X. A new era in healthcare: The integration of artificial intelligence and microbial. *Med. Nov. Technol. Devices* **2024**, *23*, 100319. [CrossRef]
17. Raskin, S. Genetics of COVID-19. *J. Pediatr.* **2021**, *97*, 378–386. [CrossRef]
18. Ahmad, S.U.; Kiani, B.H.; Abrar, M.; Jan, Z.; Zafar, I.; Ali, Y.; Alanazi, A.M.; Malik, A.; Rather, M.A.; Ahmad, A. A comprehensive genomic study, mutation screening, phylogenetic and statistical analysis of SARS-CoV-2 and its variant omicron among different countries. *J. Infect. Public Health* **2022**, *15*, 878–891. [CrossRef]
19. Hu, X.; Fernie, A.R.; Yan, J. Deep learning in regulatory genomics: From identification to design. *Curr. Opin. Biotechnol.* **2023**, *79*, 102887. [CrossRef]
20. Adebiyi, M.; Enwere, M.N.; Shekari, A.; Adebiyi, A.; Osang, F.B. Digitization Techniques for the Representation of Genomic Sequences in LSTM-Based Models. In *Intelligent Sustainable Systems: Selected Papers of WorldS4 2022*; Springer: Singapore, 2023; Volume 1, pp. 653–661. [CrossRef]
21. Noronha, P.M. Staring into the human genome to diagnose COVID. *Nature* **2022**, *603*, 354. [CrossRef]
22. Zhou, Y.; Zhang, L.; Xie, Y.H.; Wu, J. Advancements in detection of SARS-CoV-2 infection for confronting COVID-19 pandemics. *Lab. Investig.* **2022**, *102*, 4–13. [CrossRef]
23. Zhu, L.; Marsh, J.W.; Griffith, M.P.; Collins, K.; Srinivasa, V.; Waggle, K.; Van Tyne, D.; Snyder, G.M.; Phan, T.; Wells, A. Predictive model for severe COVID-19 using SARS-CoV-2 whole-genome sequencing and electronic health record data, March 2020–May 2021. *PLoS ONE* **2022**, *17*, e0271381. [CrossRef]
24. Brendel, M.; Su, C.; Bai, Z.; Zhang, H.; Elemento, O.; Wang, F. Application of Deep Learning on Single-cell RNA Sequencing Data Analysis: A Review. *Genom. Proteom. Bioinform.* **2022**, *20*, 814–835. [CrossRef] [PubMed]
25. Chiara, M.; D'Erchia, A.M.; Gissi, C.; Manzari, C.; Parisi, A.; Resta, N.; Zambelli, F.; Picardi, E.; Pavesi, G.; Horner, D.S.; et al. Next generation sequencing of SARS-CoV-2 genomes: Challenges, applications and opportunities. *Brief Bioinform.* **2021**, *22*, 616–630. [CrossRef] [PubMed]
26. Purohit, S.; Satapathy, S.; Sibi Chakkaravarthy, S.; Zhang, Y.-D. Correlation based analysis of COVID-19 virus genome versus other fatal virus genomes. *Arab. J. Sci. Eng.* **2023**, *48*, 11015–11027. [CrossRef] [PubMed]
27. Gugulothu, P.; Bhukya, R. Coot–Lion optimized deep learning algorithm for COVID-19 point mutation rate prediction using genome sequences. *Comput. Methods Biomech. Biomed. Eng.* **2024**, *27*, 1410–1429. [CrossRef] [PubMed]
28. Mo, W.; Wen, J.; Huang, J.; Yang, Y.; Zhou, M.; Ni, S.; Le, W.; Wei, L.; Qi, D.; Wang, S. Classification of Coronavirus Spike Proteins by Deep-Learning-Based Raman Spectroscopy and its Interpretative Analysis. *J. Appl. Spectrosc.* **2023**, *89*, 1203–1211. [CrossRef]
29. Khodaei, A.; Shams, P.; Sharifi, H.; Mozaffari-Tazehkand, B. Identification and classification of coronavirus genomic signals based on linear predictive coding and machine learning methods. *Biomed. Signal Process. Control* **2023**, *80*, 104192. [CrossRef]
30. Hammad, M.S.; Mabrouk, M.S.; Al-atabany, W.I.; Ghoneim, V.F. Genomic image representation of human coronavirus sequences for COVID-19 detection. *Alex. Eng. J.* **2023**, *63*, 583–597. [CrossRef]
31. Das, B.; Toraman, S. New Coronavirus 2 (SARS-CoV-2) Detection Method from Human Nucleic Acid Sequences Using Capsule Networks. *Braz. Arch. Biol. Technol.* **2023**, *66*, e23220316. [CrossRef]
32. Muflikhah, L.; Rahman, M.A.; Widodo, A.W. Profiling DNA sequence of SARS-Cov-2 virus using machine learning algorithm. *Bull. Electr. Eng. Inform.* **2022**, *11*, 1037–1046. [CrossRef]
33. Harikrishnan, N.B.; Pranay, S.Y.; Nagaraj, N. Classification of SARS-CoV-2 viral genome sequences using Neurochaos Learning. *Med. Biol. Eng. Comput.* **2022**, *60*, 2245–2255. [CrossRef]
34. El-Tohamy, A.; Maghwary, H.A.; Badr, N. A Deep Learning Approach for Viral DNA Sequence Classification using Genetic Algorithm. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 530–538. [CrossRef]
35. Das, B. An implementation of a hybrid method based on machine learning to identify biomarkers in the Covid-19 diagnosis using DNA sequences. *Chemom. Intell. Lab. Syst.* **2022**, *230*, 104680. [CrossRef] [PubMed]
36. Ahmed, I.; Jeon, G. Enabling artificial intelligence for genome sequence analysis of COVID-19 and alike viruses. *Interdiscip. Sci. Comput. Life Sci.* **2022**, *14*, 504–519. [CrossRef]

37. Alkady, W.; ElBahnasy, K.; Leiva, V.; Gad, W. Classifying COVID-19 based on amino acids encoding with machine learning algorithms. *Chemom. Intell. Lab. Syst.* **2022**, *224*, 104535. [CrossRef]
38. Whata, A.; Chimedza, C. Deep Learning for SARS-CoV-2 Genome Sequences. *IEEE Access* **2021**, *9*, 59597–59611. [CrossRef]
39. Singh, O.P.; Vallejo, M.; El-Badawy, I.M.; Aysha, A.; Madhanagopal, J.; Faudzi, A.A.M. Classification of SARS-CoV-2 and non-SARS-CoV-2 using machine learning algorithms. *Comput. Biol. Med.* **2021**, *136*, 104650. [CrossRef]
40. Gunasekaran, H.; Ramalakshmi, K.; Rex Macedo Arokiaraj, A.; Deepa Kanmani, S.; Venkatesan, C.; Suresh Gnana Dhas, C. Analysis of DNA Sequence Classification Using CNN and Hybrid Models. *Comput. Math. Methods Med.* **2021**, *2021*, 1835056. [CrossRef]
41. Arslan, H. COVID-19 prediction based on genome similarity of human SARS-CoV-2 and bat SARS-CoV-like coronavirus. *Comput. Ind. Eng.* **2021**, *161*, 107666. [CrossRef]
42. Arslan, H. Machine Learning Methods for COVID-19 Prediction Using Human Genomic Data. *Proceedings* **2021**, *74*, 20. [CrossRef]
43. Afify, H.M.; Zanaty, M.S. Computational predictions for protein sequences of COVID-19 virus via machine learning algorithms. *Med. Biol. Eng. Comput.* **2021**, *59*, 1723–1734. [CrossRef]
44. Riley, R.D.; Ensor, J.; Snell, K.I.; Debray, T.P.; Altman, D.G.; Moons, K.G.; Collins, G.S. External validation of clinical prediction models using big datasets from e-health records or IPD meta-analysis: Opportunities and challenges. *BMJ* **2016**, *353*, i3140. [CrossRef] [PubMed]
45. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. Available online: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf (accessed on 18 July 2022).
46. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467. [CrossRef]
47. Cock, P.J.; Antao, T.; Chang, J.T.; Chapman, B.A.; Cox, C.J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422–1423. [CrossRef] [PubMed]
48. Barbosa, R.M.; Fernandes, M.A.C. Data stream dataset of SARS-CoV-2 genome. *Data Brief* **2020**, *31*, 105829. [CrossRef]
49. Dai, Q.; Liu, J.-w.; Yang, J.-P. SWSEL: Sliding Window-based Selective Ensemble Learning for class-imbalance problems. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105959. [CrossRef]
50. NCBI Virus. Available online: https://www.ncbi.nlm.nih.gov/labs/virus/vssi/#/virus?SeqType_s=Nucleotide (accessed on 15 March 2023).
51. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991. [CrossRef]
52. Xu, G.; Meng, Y.; Qiu, X.; Yu, Z.; Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [CrossRef]
53. Yu, Q.; Wang, Z.; Jiang, K. Research on text classification based on bert-bigru model. *J. Phys. Conf. Ser.* **2021**, *1746*, 012019. [CrossRef]
54. Kenarang, A.; Farahani, M.; Manthouri, M. BiGRU attention capsule neural network for persian text classification. *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 3923–3933. [CrossRef]
55. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *arXiv* **2015**, arXiv:1509.01626. [CrossRef]
56. Zhang, Y.; Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv* **2015**, arXiv:1510.03820. [CrossRef]
57. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 1–11.
58. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
59. Kavianpour, P.; Kavianpour, M.; Jahani, E.; Ramezani, A. A cnn-bilstm model with attention mechanism for earthquake prediction. *arXiv* **2021**, arXiv:2112.13444. [CrossRef]
60. Pustokhin, D.A.; Pustokhina, I.V.; Dinh, P.N.; Phan, S.V.; Nguyen, G.N.; Joshi, G.P. An effective deep residual network based class attention layer with bidirectional LSTM for diagnosis and classification of COVID-19. *J. Appl. Stat.* **2023**, *50*, 477–494. [CrossRef]
61. Sturm, B.L. Classification accuracy is not enough. *J. Intell. Inf. Syst.* **2013**, *41*, 371–406. [CrossRef]
62. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2020**, *17*, 168–192. [CrossRef]

63. Fan, J.; Upadhye, S.; Worster, A. Understanding receiver operating characteristic (ROC) curves. *Can. J. Emerg. Med.* **2006**, *8*, 19–20. [CrossRef]
64. Abdi, H.; Williams, L.J. Newman-Keuls test and Tukey test. *Encycl. Res. Des.* **2010**, *2*, 897–902. [CrossRef]