Research Article

Long Short-Term Memory Neural Networks for RNA Viruses Mutations Prediction

Takwa Mohamed ¹, Sabah Sayed ², Akram Salah ² and Essam H. Houssein ¹

¹Computer Science, Faculty of Computers and Information, Minia University, Minia, Egypt

²Computer Science, Faculty of Computers and Information, Cairo University, Giza, Egypt

Correspondence should be addressed to Essam H. Houssein; essam.halim@mu.edu.eg

Received 26 March 2021; Revised 22 May 2021; Accepted 18 June 2021; Published 27 June 2021

Academic Editor: Peng-fei Jiao

Copyright © 2021 Takwa Mohamed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Viral progress remains a major deterrent in the viability of antiviral drugs. The ability to anticipate this development will provide assistance in the early detection of drug-resistant strains and may encourage antiviral drugs to be the most effective plan. In recent years, a deep learning model called the seq2seq neural network has emerged and has been widely used in natural language processing. In this research, we borrow this approach for predicting next generation sequences using the seq2seq LSTM neural network while considering these sequences as text data. We used hot single vectors to represent the sequences as input to the model; subsequently, it maintains the basic information position of each nucleotide in the sequences. Two RNA viruses sequence datasets are used to evaluate the proposed model which achieved encouraging results. The achieved results illustrate the potential for utilizing the LSTM neural network for DNA and RNA sequences in solving other sequencing issues in bioinformatics.

1. Introduction

Families of viruses are grouped supported by their style of nucleic acid as genetic material, DNA, or RNA. DNA viruses usually contain double-stranded DNA (dsDNA) and infrequently single-stranded DNA (ssDNA). Viruses can replicate using DNA-dependent DNA polymerase. RNA viruses have typically ssRNA and also contain dsRNA. There are two groups of ssRNA viruses, and they are positive-sense (ssRNA (+)) and negative-sense (ssRNA (-)).

The genetic material of ssRNA (+) viruses is like mRNA and might be directly translated by the host cell. ssRNA (-) viruses carry RNA complementary to mRNA and must be changed to positive RNA using RNA polymerase at some point in time. A special case of this type is the retroviruses, which replicate through DNA intermediates utilizing invert transcriptase in spite of having RNA genomes [1].

The strands of DNA are composed of nucleotides. All nucleotides are made of a nitrogen-containing nucleobase, which are either guanine (G), adenine (A), thymine (T), or

cytosine (C). Most of the DNA molecules consist of two threads wrapped around each other to form a double helix. DNA strands used this arrangement to generate RNA in a process known as translation. In any case, unlike DNA, RNA is regularly found as a single strand. One sort of RNA is the flag-bearer RNA (mRNA), which carries data from the ribosome, which is where the protein is synthesized. The mRNA sequence is what indicates the chain of amino acids that made up the protein. Moreover, DNA and RNA are the most common components of viruses. A few of the viruses are DNA-based, whereas others are RNA-based such as Newcastle, HIV, and flu [2].

RNA viruses differ from DNA viruses in that they have many mutations, and subsequently, they have higher versatility. This mutation causes continuous development that leads to resistance, so the virus becomes more destructive [3].

Influenza viruses are stuck in negative RNA viruses. Influenza A and B contain eight parts of viral RNA, while influenza C contains seven parts of viral RNA [4]. Several viruses with widespread potential have developed over time.

The 2002 rise of the extremely intense respiratory disorder coronavirus (SARS-CoV), the widespread H1N1 flu in 2009, which resulted in the circulation of flu H5N1 and H5N7 strains, and the later rise of the Middle East respiratory disorder coronavirus (MERS-CoV) outline the current risk of these viruses [1, 4]. In spite of the main contrasts in their structure and the study of disease transmission, these widespread viruses share a number of vital properties. They are zoonotic encompassed RNA respiratory viruses that occasionally transmit between people in their original shape but seem to be transforming to facilitate more productive human-to-human transmission [5].

Machine learning is one of the tools that are used for analyzing the mutation data. Machine learning techniques offer assistance by predicting the impacts of non-synonymous single nucleotide polymorphisms on protein stability, work, and drug resistance [6].

Learning the rules of depicting mutations that influence protein behavior and utilizing them to gather unused important mutations that will be resistant to certain drugs is one of the purposes that machine learning techniques are used for [7]. Another purpose is to predict the potential secondary structure based on essential structure sequences [8–10].

Another trend is to predict the variants of RNA sequences' single nucleotide. RNA sequence is considered to be a series of four separate states and, thus, follows nucleotide substitutions during the evolution of the sequence. In this direction, it has been assumed that the different nucleotides in the sequence evolved symmetrically, and this is justified by using the neutral evolution state of nucleotides. Later, other research denied this assumption and identified the relevant neighbor-dependent substitution processes.

Finally, the prediction of the mutation process helps in developing drugs that can overcome these mutations [11, 12]. Long-term memory (LSTM) is a specific recurring neural network structure (RNN) designed for modeling time series, and long-term dependencies are more accurate than traditional networks RNN [13]. LSTM architecture is based on RNNs, which were introduced to address the vanishing gradient problem of classical RNNs [14]. The seq2seq LSTM architecture is an encoder-decoder architecture, which consists of two LSTM networks: the encoder LSTM and the decoder LSTM. The LSTM cell takes an input in the form of the one-hot encoded version at t position, as well as two recurrent states, the hidden state $h(t)$ and the cell state $c(t)$, that are vectors with predetermined dimensions. These states and the input are regulated by trainable input [15].

Each LSTM cell remembers values of the hidden state and cell state over time intervals and the three gates modify the information received from the previous time step. Finally, the outputs of each gate are combined to update the hidden state and the cell state of the current cell [14]. The LSTM networks deal with texts, and they create dictionaries of unique words for all text data without ordering them. During training, the data from the dictionaries were used to modify the LSTM on the basis of only one word. Every word is represented by one vector value. We create as many input vectors as the number of words in one sequence. In this case,

the sequence is one sentence, one set of words. In this study, a machine learning technique is proposed that is based on deep learning. It predicts the probabilities of nucleotide mutations that do not occur in the primary sequence. This paper presents a proof of concept by applying this technique on a set of successive generations of the influenza A Virus (H1N1) from USA and generations of SARS-COV-2. The actual sequence and the predicted sequence are then compared in order to validate the ability of the machine learning technique to predict mutations in RNA sequence evolution and its prediction accuracy. In this method, a training phase is used in which all the input iteration may be a sequence of one generation of the virus, while the output is the sequence of the next generation of the virus. Each feature within the input may be a nucleotide within the sequence corresponding to a feature within the output. The method presented in this paper predicts potential mutations in a sequence based on previous sequences. This comparison results in a ratio that is calculated based on the number of nucleotides matched between both the expected and actual sequences versus the total number of nucleotides in the sequence.

The main contribution of this study is to predict the next sequence of DNA sequence using the LSTM deep learning utilizing a different manner, which is dividing the long sequences of DNA into amino acids as words, tokenizing them to create an amino acid dictionary of unique words. The results outperform the results of studies in the literature.

The rest of the paper is organized as follows. Section 2 provides related work in the application of machine learning techniques in genetic problems, as well as the proposed technique to solve these problems. Methods, Results, and Discussion appear in Sections 3–5, and finally, a Conclusion is presented in Section 6.

2. Related Work

Predicting strategies has been practiced within the field of genetics for a long time, and one of the trends has been the description of genome sequence changes in the influenza virus after it has invaded humans from other animal hosts. Based on the composition of a few oligonucleotide-based hosts, a strategy has been proposed for predicting changes to the directivity sequence of the virus and control strains that are potentially dangerous when introduced to human populations from nonhuman sources adapted toward distinctive directions [16].

Authors in [17] used another direction depending on the relationship of mutations between 241 H5N1 hemagglutinins of influenza. A virus was decided according to its amino acid and RNA codon sequences with these independent six features and the likelihood or nonappearance of mutation as being dependent utilizing logistic regression. Generally, logistic regression can capture a relationship, but this relationship cannot be captured when only a few mutations are contained in the regression. This means that all the mutations need to be pooled in the first hemagglutinin sequence, followed by meaning of all independents into a single hemmagglutinin sequence for the regression analysis rather

than to regress each hemagglutinin sequence with its mutations.

In [18, 19], authors go to another direction depending on the randomness. An attempt is made to apply the traditional neural network for modeling the relationship between cause and mutation to predict potential sites of mutation, and then the possibility of mutation of amino acids is used to predict the amino acids that may be developed in the expected positions. The results confirmed the possibility of using the relationship and the cause of the mutations with the neural network model to predict the positions of mutations and the possibility of using the mutated amino acids to predict the amino acids that may be developed.

Authors in [11] depend on prediction host tropism using a machine learning algorithm (random forest) where computational models of 11 influenza proteins were generated to predict the host tropism. Prediction models were trained on influenza protein sequences isolated from both bird and human samples. The model could determine host distension for individual influenza proteins using the features of 11 proteins to build a model for predicting the virus host influenza.

In [12], they developed software for discrimination of pandemic from non-pandemic influenza sequences at both nucleotide and protein levels via the CBA method.

In [20], the possible point mutations of primary RNA sequence structure alignments are predicted. They predicted the genotype of each nucleotide in the RNA sequence and demonstrated that the nucleotides in the RNA sequence influence adjacent nucleotide mutations in the sequence using a neural network technique in order to predict new strains, and then predicted the mutation patterns using the rough set theory. The data used in this model are several aligned RNA isolates of time-series species of the Newcastle virus. Two datasets from two different sources were used for model validation. This method results in nucleotide prediction in the new generation exceeding 75%.

In this paper, a seq-2-seq LSTM-RNN model is introduced for predicting next-generation sequence inspired by the recent work in language modeling. This model, the entire genome for training, predicts the next sequence generation with accuracy that outperforms what had been previously reported.

3. Materials and Methods

3.1. Dataset. This study presents a proof of concept by applying this technique on a set of successive generations of the Newcastle Disease Virus (NDV) and influenza virus, as described in Table 1.

The first dataset of NDV consists of Eighty-three DNA (RNA reverse transcription) sequences that were obtained from different birds in China over the course of 2005; samples were taken from ill or dead poultry. The data were collected and presented in [21].

The second dataset of influenza virus consists of DNA (RNA reverse transcription) sequences of 4609 H1N1 of influenza A virus from 1935 to 2017. It was obtained from the Medline data bank [22].

TABLE 1: Data description table.

NO.	Dataset	#Samples	Sequence length
1	New Castle Disease Virus	83	1778
2	H1N1 Influenza Virus	4609	535

3.2. Proposed Model Architecture. In this study, each amino acid is predicted in the virus sequence, and it was demonstrated that the amino acids in the sequence influence adjacent amino acid mutations in the sequence using a LSTM deep neural network technique in order to predict new strains.

We proposed a model for the Virus Mutations Prediction. The aim of this approach was to optimize the prediction accuracy by preparing sequencing data in a new way for optimal accuracy. The proposed approach consists of four main phases. In the first phase, sequences of datasets are preprocessed. In the second phase, once we have preprocessed sequences of data, they are transformed into a format that is suitable for training an LSTM network. In this case, a one-hot encoding of the integer values is used where each value is represented by a binary vector that is all "0" values except the pointer to the word, which is set to 1. In the third phase, the input data are prepared to train on LSTM encoder. After that, it is the role of the decoder to take the output from encoder as integers and transform it to sequences. Finally, the obtained results are evaluated. The overall process of the proposed method is illustrated in Figure 1.

3.2.1. Preprocessing Phase. As shown in Figure 2, to achieve better results with sequencing data, the data preprocessing phase is based on three steps; the first step is the DNA translation, the second is the tokenization, and the third is the padding.

(1) DNA Sequencing. DNA sequencing is the sequence of sequential letters without space. There are no words in the DNA sequence. We propose a method to translate sequences of DNA to words, and then apply the representation technique of text data without losing any information of the position of each nucleotide in sequences. Figure 3 gives an example of translating a DNA sequence into words. Using a window with fixed size and sliding it through the given sequence with a fixed step's stride equals to zero. This segment is considered as a word and is added to the destination sequence. Finally, a series of words derived from a specific DNA sequence has been produced. The word size is 3 nucleotides. Three-letter code of RNA may be used to represent each amino acid in the sequence.

(2) DNA Sequence Tokenization for Amino Acid Dictionary. The next step is tokenizing the DNA sequences. It divides a sentence into the corresponding list of words. A dictionary with 64 different words is used as shown in Figure 2. Each sequence in the dataset has four nucleotides bases (A, C, T, G) and each word consists of three nucleotides, so we generate a dictionary from $64 (4^3)$ unique words where words are the keys and the corresponding integers are the

FIGURE 1: Proposed model architecture.

FIGURE 2: Preprocessing phase.

FIGURE 3: Translating sequence of DNA into words.

values. This is extremely important since deep learning and machine learning algorithms work with numbers.

(3) *DNA Sequence Padding.* Next, the input will be padded. The reason behind padding the input and the output is that, content sentences can be of varying length; however, the LSTM expects input instances with the same length. In this manner, the sentences are changed into fixed-length vectors. One way to do this is by using the padding. In padding, a certain length is defined for a sentence. In our case, the length of the longest sentence in the inputs and outputs will be used for padding the input and output sentences, respectively. Each word is then represented by a one-hot vector. A sequence of words is produced from the given DNA sequence.

3.2.2. Proposed Prediction Model. Borrowing the idea from seq-to-seq learning [23], the proposed LSTM Model is composed of connected LSTM-based encoder and decoder networks (Figure 4). The input to the encoder LSTM is the generation sequence (i). The final output hidden states and cell states are concatenated and passed to the decoder, which employs it, beside the following generation sequence ($i + 1$) (during training), to create a next-generation sequence. Hidden states (h) and cell states (c) capture the context information that will be used to inform the decoder. The

decoder consists of an LSTM like the encoder. Each cell receives the cell and hidden states from the previous cell, except for the first cell, which obtains those from the encoder (h and c). The cell (k) in the dense layer is used to predict a probability distribution of word at position t of the next-generation sequence.

Hyperparameters used in the model are shown in Table 2; they were set based on a compromise between training time and accuracy on the validation set.

(1) *The Model Training.* As shown in Figure 5, LSTM bi-directional sequence to the sequence is made of an encoder and decoder. An input sequence (ACT CTA TTG in Figure 5) is given as input to the encoder. Each LSTM cell takes information as input from the previous cell, in the form of a hidden state vector and cell state vector (represented in arrows) and combines it with the one-hot encoded vector. The output of the encoder is the concatenation of the hidden and cell-state vectors. In the decoder, each cell receives as input $[i]$ in the one-hot encoded version and the previous word of the next generation generated by the model, as well as the hidden state and cell-state vectors from the previous cell. It not only passes those two vectors after updating them to the next cell but also feeds the hidden state vector to the dense layer (output layer), which outputs a probability distribution over the next generation word at that position. A SoftMax function is applied to obtain probability distribution. To reduce overfitting, early stopping is used to end training when the validation loss did not decrease.

(2) *The Model Evaluation.* To use a trained LSTM model to simulate the evolution of a given generation sequence, the model is used as described previously, with a few small

FIGURE 4: Proposed prediction model.

TABLE 2: Hyperparameters used in the proposed prediction model.

Hyper-parameter	Value
Epochs	20 for first dataset; 50 for second dataset
Batch size	10
Activation function	Softmax
Training data size	80% of all data size
Validation data size	20% of all data size

FIGURE 5: Training model.

modifications. The functionality of the encoder remains the same. The sentence in the original language is passed through the encoder and the hidden state, and the cell state is the output from the encoder. To make predictions, the decoder output is passed through the dense layer. In the tokenization steps, we converted words to integers. The outputs from the decoder will also be integers. However, the output should be a sequence of words of the next generation. Conversion of the integers back to words is done. A new dictionary will be created for both inputs and outputs where the keys will be the integers and the corresponding values will be the words. Finally, the words in the output are concatenated using a space and the resulting string is returned.

As shown in Figure 6, in Step 1, the hidden state and cell state of the encoder are used as input to the decoder. The decoder predicts a word, y_1 , which may or may not be true.

At Step 2, the decoder hidden state and cell state from Step 1, along with y_1 , is used as input to the decoder, which predicts y_2 . Then, the process continues.

4. Results

The model was implemented using Keras library, LSTM layers in Tensorflow 2.1, and Biopython library in Python was used for reading the FASTA file and preprocessing genome sequences. To evaluate the prediction performance for different test cases, two measures, Accuracy and Loss function, are used as shown in Figures 7–10 and Tables 3–6.

The proposed model is applied on two datasets of different viruses as described in Table 1. The dataset is divided into training and testing portions. In the training phase, each

FIGURE 6: Evaluation model.

FIGURE 7: Experimental results of the influenza dataset.

FIGURE 8: Loss function of the influenza dataset.

FIGURE 9: Results of Newcastle dataset.

FIGURE 10: Loss function of Newcastle dataset.

TABLE 3: Results varying according to epoch's numbers of the influenza dataset.

Epochs	Training	Validation	Epochs	Training	Validation
1	35.51	93.95	11	98.79	98.81
2	91.47	98.44	12	98.82	98.88
3	97.19	98.47	13	98.85	98.9
4	97.94	98.55	14	98.88	98.62
5	98.24	98.71	15	98.9	98.91
6	98.43	98.66	16	98.92	98.78
7	98.54	98.4	17	98.95	98.84
8	98.63	98.49	18	98.96	98.94
9	98.69	98.86	19	98.97	98.75
10	98.75	98.9	20	98.99	98.91

TABLE 4: Decreasing loss function according to epoch numbers of the influenza dataset.

Epochs	Training	Validation	Epochs	Training	Validation
1	2.7743	0.643	11	0.0545	0.0789
2	0.4941	0.129	12	0.0545	0.0786
3	0.157	0.1008	13	0.0529	0.0781
4	0.107	0.1001	14	0.0512	0.085
5	0.0885	0.0871	15	0.05	0.0779
6	0.0776	0.0861	16	0.0489	0.0762
7	0.0708	0.0896	17	0.0477	0.0761
8	0.0659	0.0853	18	0.0467	0.0767
9	0.062	0.081	19	0.046	0.0781
10	0.0589	0.0795	20	0.0452	0.0762

TABLE 5: Results varying according to the epoch numbers of the Newcastle dataset.

Epochs	Training	Validation	Epochs	Training	Validation
1	7.31	8.58	26	88.41	93.43
2	9.94	10.02	27	90.31	92.38
3	12.16	12.78	28	89.63	92.67
4	14.16	18.9	29	90.53	94.61
5	18.74	19.98	30	91.24	93.46
6	20.64	21.52	31	91.88	94.35
7	23.64	22.31	32	92.65	93.92
8	26.11	34.08	33	92.94	94.15
9	33.21	32.83	34	92.26	94.81
10	38.26	43.58	35	93.2	95.66
11	42.02	48.11	36	93.52	95.79
12	53.48	62.54	37	94.02	95.53
13	56.42	68.95	38	94.04	96.16
14	67.35	66.91	39	94.4	95.47
15	68.8	69.14	40	94.67	94.68
16	73.96	84.88	41	94.9	94.94
17	77.08	86.13	42	94.9	96.88
18	81.87	84.09	43	94.9	96.91
19	82.82	88.01	44	95.07	95.79
20	84.25	91.03	45	95.63	97.17
21	83.54	89.58	46	95.63	96.19
22	85.85	91.88	47	95.44	96.19
23	87.49	91.55	48	95.56	95.89
24	87.6	91.46	49	95.88	96.19
25	88.19	93.3	50	96.12	96.98

TABLE 6: Decreasing loss function according to the epoch numbers.

Epochs	Training	Validation	Epochs	Training	Validation
1	4.0003	3.7102	26	0.5854	0.3878
2	3.6434	3.5906	27	0.5051	0.3707
3	3.5486	3.3866	28	0.4927	0.3611
4	3.4181	3.1821	29	0.4599	0.2925
5	3.1585	3.0986	30	0.4251	0.3074
6	2.9704	2.8028	31	0.388	0.2892
7	2.7521	2.6392	32	0.3589	0.2743
8	2.523	2.3341	33	0.9294	0.2594
9	2.3444	2.2772	34	0.3426	0.2555
10	2.1713	1.9458	35	0.3083	0.214
11	2.0551	1.921	36	0.2873	0.2164
12	1.8859	1.6624	37	0.2684	0.2057
13	1.7944	1.534	38	0.262	0.1755
14	1.6047	1.4616	39	0.2508	0.2028
15	1.5118	1.3282	40	0.243	0.2072
16	1.3722	1.1242	41	0.2251	0.196
17	1.2746	1.0356	42	0.218	0.1503
18	1.1337	1.0117	43	0.2043	0.1444
19	1.0464	0.8692	44	0.2087	0.1722
20	0.947	0.7218	45	0.1884	0.1364
21	0.9063	0.6648	46	0.1863	0.1747
22	0.8096	0.5972	47	0.1848	0.1503
23	0.7136	0.5329	48	0.1769	0.1861
24	0.6682	0.5075	49	0.1723	0.1498
25	0.6164	0.4318	50	0.1597	0.1267

segment of a sequence is considered as a single training entry. The required output corresponding to the input DNA segment is the next scaled DNA segment from the next generation of the training dataset. For each I/O sequence, the weights are continuously updated until reaching the highest possible accuracy calculated according to the number of correct predicted nucleotides to the total number of nucleotides in the sequence. After that, the next entry is the DNA sequence produced in the current step; the last RNA segment is left for testing.

In the first experiment, the effectiveness of the proposed model is verified by using the influenza virus dataset. The accuracy of this dataset is 98.9%. This accuracy is quite high compared with other approaches such as finding good representations for sequences or feature selection, which was applied before. It is shown that features extracted by LSTM layers of RNN neural network are very useful for the virus mutations prediction.

The second experiment aims to compare the proposed model with another model in [16]. The proposed model is applied on the New Castle Disease dataset, which is used in [16]. The resulting accuracy is 96.9%, which outperforms the 70% accuracy in [16].

As noticed, the results of the proposed model show a better performance on the influenza virus dataset than the New Castle Disease dataset. This is because the number of samples in the dataset increased and subsequently, the model learned better. On the other hand, the computational complexity increased.

5. Discussion

Accurate and fast prediction mutations of RNA viruses can significantly improve vaccine development. In this research,

we accomplished improvements by utilizing the LSTM recurrent neural network, a deep learning model with the high power of representing complicated issues. In addition, the ability to deal with a huge number of sequences, compared with the common approach of considering a small number of sequences, could be a critical quality of data mining-based approaches. One hot vector is used to represent the sequences in order to preserve the position of the information for each individual word or amino acid in the sequence. Using H1N1 and New Castle Disease datasets, the proposed model achieved a high-performance prediction. It can be used as a reliable tool to support a study of these types of data. To study these types of data, you should only use the predictions of our model as references. One limitation of this research is that we experimentally selected hyperparameters such as word size, area size, and network architecture configuration. Through the results of several experiments, we realized that these hyperparameters significantly influence the prediction performance of the proposed model. The computational and financial cost of this method is very low, while the speed, range, and accuracy are remarkably high. Also, these methods can be used as preprocessing expensive and lengthy steps of laboratory experiments. The only requirement of these methods is reliable and comprehensive sequencing data that are becoming increasingly available in the public domain.

6. Conclusions

The recurrent neural network has shown remarkable implementation in many research fields. This research succeeded in manipulating A, C, T, and G nucleotides for DNA data. By using a single-hot vector to represent DNA sequences and applying the LSTM repetitive neural network model, this work paves the way for a new horizon where the prediction of mutations, such as the evolution of the virus, is possible. It can help to plan new drugs for potential drug-resistant strains of the virus at some point in the recent times of a potential outbreak. Moreover, it can provide assistance in formulating a diagnosis for early detection of cancer and possibly for early initiation of treatment. This work examined the relationship between nucleotides in RNA including the effect of each nucleotide in the genotypic sequence of other nucleotides. The bases for these relationships are investigated and visualized to predict which mutations will emerge over the next generations and are trained by two datasets isolated from two distinct viruses. This work proved the existence of a correlation between the mutations of nucleotides, and successfully predicted the nucleotides in the next generation. The proposed model achieved significant performance improvements in all evaluation.

Data Availability

Previously reported that the viruses' datasets were used to support this study and are available at [Liu, Hualei, et al. "Molecular epidemiological analysis of Newcastle disease

virus isolated in China in 2005." *Journal of virological methods* 140.1-2 (2007): 206–211. and Medline data bank]. These prior studies (and datasets) are cited at relevant places within the text as references [21, 22].

Conflicts of Interest

The authors declare that they have no Conflicts of Interest.

References

- [1] S. Durmuş and Ö. Kutlu, "Comparative interactomics for virus–human protein–protein interactions: DNA viruses versus RNA viruses," *FEBS Open Bio*, vol. 7, pp. 96–107, 2017.
- [2] S. F. Elena and S. Rafael, "Adaptive value of high mutation rates of RNA viruses: separating causes from consequences," *Journal of Virology*, vol. 79, no. 18, pp. 11555–11558, 2005.
- [3] B. A. Wilson, "The population genetics of drug resistance evolution in natural populations of viral, bacterial and eukaryotic pathogens," *Molecular Ecology*, vol. 25, no. 1, pp. 42–66, 2016.
- [4] R. J. Cox, K. A. Brokstad, and P. L. Ogra, "Influenza virus: immunity and vaccination strategies. Comparison of the immune response to inactivated and live, attenuated influenza vaccines," *Scandinavian Journal of Immunology*, vol. 59, no. 1, pp. 1–15, 2004.
- [5] J. A. Otter, "Transmission of SARS and MERS coronaviruses and influenza virus in healthcare settings: the possible role of dry surface contamination," *Journal of Hospital Infection*, vol. 92, no. 3, pp. 235–250, 2016.
- [6] E. Capriotti, "A three-state prediction of single point mutations on protein stability changes," *BMC Bioinformatics*, vol. 9, no. 2, pp. 1–9, 2008.
- [7] S. WANG, D. U. Zhijuan, and M. E. N. G. Xiaofeng, "Research progress of large-scale knowledge graph completion technology," *Scientia Sinica Informationis*, vol. 50, no. 4, pp. 551–575, 2020.
- [8] B. Rost and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *Journal of Molecular Biology*, vol. 232, no. 2, pp. 584–599, 1993.
- [9] J. A. Cuff and G. J. Barton, "Application of multiple sequence alignment profiles to improve protein secondary structure prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 40, no. 3, pp. 502–511, 2000.
- [10] D. Frishman and A. Patrick, "Seventy-five percent accuracy in protein secondary structure prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 27, no. 3, pp. 329–335, 1997.
- [11] C. L. P. Eng, J. C. Tong, and T. W. Tan, "Predicting host tropism of influenza A virus proteins using random forest," *BMC Medical Genomics*, vol. 7, no. 3, pp. 1–11, 2014.
- [12] F. Kargarfard, A. Sami, and E. Ebrahimi, "Knowledge discovery and sequence-based prediction of pandemic influenza using an integrated classification and association rule mining (CBA) algorithm," *Journal of Biomedical Informatics*, vol. 57, pp. 181–188, 2015.
- [13] H. Sak, A. W. Senior, and F. Beaufays, *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*, 2014.
- [14] M. Sundermeyer, R. Schlüter, and N. Hermann, "LSTM neural networks for language modeling," in *Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association*, Portland, OR, USA, September 2012.

- [15] F. A. Gers, S. Jürgen, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," in *Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99*, pp. 850–855, Edinburgh, UK, September 1999.
- [16] Y. Iwasaki, "Novel bioinformatics strategies for prediction of directional sequence changes in influenza virus genomes and for surveillance of potentially hazardous strains," *BMC Infectious Diseases*, vol. 13, pp. 1–9, 2013.
- [17] G. Wu and S. Yan, "Prediction of possible mutations in H5N1 hemagglutinins of influenza A virus by means of logistic regression," *Comparative Clinical Pathology*, vol. 15, p. 255, 2006.
- [18] G. Wu and S. Yan, "Prediction of mutations engineered by randomness in H5N1 hemagglutinins of influenza A virus," *Amino Acids*, vol. 35, no. 2, pp. 365–373, 2008.
- [19] G. Wu and S. Yan, "Prediction of mutations in H1 neuraminidases from North America influenza A virus engineered by internal randomness," *Molecular Diversity*, vol. 11, no. 3, pp. 131–140, 2007.
- [20] M. A. Salama, "Aboul Ella Hassanien, and Ahmad Mostafa. "The prediction of virus mutation using neural networks and rough set techniques," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 1, pp. 1–11, 2016.
- [21] H. Liu, "Molecular epidemiological analysis of Newcastle disease virus isolated in China in 2005," *Journal of Virological Methods*, vol. 140, pp. 1–2, 2007.
- [22] <https://www.ncbi.nlm.nih.gov/genomes/FLU/Database/nph-select.cgi?go=database>.
- [23] I. Sutskever and V. Quoc, "Sequence to sequence learning with neural networks," 1409.3215, 2014.