Khulna University Studies Special Issue ICSTEM4IR: 803-816, 2022

ORIGINAL ARTICLE

SCIENCE & ENGINEERING

NEXT MUTATION PREDICTION OF SARS-COV-2 SPIKE PROTEIN SEQUENCE USING ENCODER-DECODER BASED LONG SHORT TERM MEMORY (LSTM) METHOD

Sumaiya Tasnim*, Kamrul Hasan Talukder, Anika Asfi

Computer Science and Engineering Discipline, Khulna University, Khulna-9208, Bangladesh

KUS: ICSTEM4IR-22/0142

Manuscript submitted: June 27, 2022 Accepted: September 28, 2022

Abstract

The recent world is facing a new pandemic which is caused by a virus named Coronavirus. Its fast mutation capability makes the situation worse affecting all the countries. Handling the virus is a challenging task now as there is still no permanent remedy for this. The doctors, engineers, scientists all are working together to fight against the virus. Revealing the genome sequencing and total structure of the virus paves the way for more research on this topic. Many researchers and scientists are working relentlessly on mutation analysis. Since spike proteins are one of the most important parts of SARS-CoV-2 for affecting humans, scientists are working for vaccine and drug discovery targeting S protein. Many Machine learning, Artificial Intelligence, Deep Learning methods are used on the genome datasets to detect the mutation position and predict further insights. The goal of this work is to predict the most probable next-generation Spike Protein sequence of SARS-CoV-2. We have proposed a model that uses the Encoder-Decoder based LSTM model on date-wise ordered protein sequence data of S-protein. This has worked effectively on predicting next generation sequence of S protein. We compared our model with other deep learning models i.e. CNN-LSTM and Attention-based LSTM. We also experimented our model with large datasets as well as with small datasets, and the results of the tests are effective and efficient in both ways.

Keywords: SARS-CoV-2, Machine Learning, LSTM, S Protein, Neural Network, Covid-19.

Introduction

Coronavirus has brought a pandemic situation throughout the whole world. All countries are affected by this pandemic. Coronavirus comes from the family Coronaviridae. It is named Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). The first case of COVID-19 was reported on 31st December 2019 in the Wuhan city of China (Wang et. al., 2020). World Health Organization (WHO) declared the outbreak a pandemic on 11th March 2020 when the confirmed cases of affected people were 118,000 in 110 countries (Ducharme, 2020). As the virus was unknown, it became very difficult for doctors to create vaccine or medicine for this disease. By applying modern techniques and methods the advancement in the way of

*Corresponding author: <sumaiya1704@cseku.ac.bd>
DOI: https://doi.org/10.53808/KUS.2022.ICSTEM4IR.0142-se

knowing the virus and inventing the vaccine has started to increase. There is still more to know by analyzing big data and developing innovative solutions to this growing virus.

Mutations are the key mechanism for this challenging evolution of COVID-19. Frequent mutations enable the virus to form highly diverse strains. Mutation generally occurs when there is an error while copying RNA to a new cell. The spike protein(S) of SARSCoV-2 plays the most important role in the receptor recognition and cell membrane fusion process. It is thought to become more infectious because of the mutation on S protein that enables the virus to bind to its host cells 10 times more effectively. Mutations generate great challenges in improvements of vaccines and drugs, since anti-viral vaccines and drugs may lose their targets due to mutations.

If there exists any mutation tracking data and if the next mutation can be predicted, it can play a vital role in the field of vaccine and drug discovery. So the prediction of the next mutation is essential. Many researchers are working together on this.(S. ALBERT, 2017)(Walsh et. al., 2016) Next-strain is one of the major contributions which is a real-time pathogen tracking system that consists of a viral genome database, a bioinformatics pipeline for phylodynamics analysis and a dynamic visualization platform together which represent a real-time view into the evolution.(Hadfield et. al., 2018) Abdelhafid Zeroual et. al. Have discussed different machine learning techniques like Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Bidirectional LSTM, Gated Recurrent Units (GRU), and Variational Auto-Encoder (VAE) for forecasting COVID-19 time series data. They gave an assumption about the future condition of the pandemic by predicting future affected cases and recovered cases.(Zeroual et. al., 2020) Refat Khan Pathan et. al. proposed a method for computing the mutation rate and predicting it for the future using time series data using Recurrent Neural Network-based LSTM method for their research which can predict the substitution mutation rate of the 400th patient in the future.(Pathan et. al., 2020).

Gurjit S. Randhawa et. al. used supervised machine learning with digital signal processing following the augmented decision tree approach on the learning component and used spearman's correlation coefficient to monitor the validity of the result for genomic analysis where they found that the genes for the coronavirus came from bats and the taxonomic classification of SARS-CoV-2 is Sarbecovirus which is under Betacoronavirus.(Randhawa, 2020) Mostafa A. Salama et. al. worked on a methodology for predicting possible point mutation on primary RNA of Newcastle virus dataset. A machine learning technique Neural Network is applied for forecasting new strains of the virus and to extract the mutation patterns they applied rough set algorithm. (Salama et. al., 2016) T. Koyama et. al. analyzed the genome variation of SARS-CoV-2 data using BEAST analysis which provide an understanding of how viruses travel from reservoirs and come into human contact which is an important hint about the future risks for novel infections.(Koyama et. al., 2020) Fatemah Kargarfard et. al. analyzed the influenza data and identified mutation positions in all protein segments which can enable to distinguish between pandemic and seasonal strains using Classification Based on association rule mining (CBA), Ripper, and Decision tree algorithm to discover rules among mutation. CBA has shown the best results for their work.(Kargarfard et. al., 2019)

Materials and Methods

Our proposed system worked on spike protein data. The workflow diagram is given in figure 1. For predicting mutation sequence at a certain time we used Long Short Term Memory Method (LSTM) on the protein sequence data of SARS-CoV-2 S protein. Our total work is divided into 3 phases, i.e. i) Data Preparation Phase, ii) Training Phase, iii) Testing Phase.

We preprocessed the data and extracted features then fed the input into Encoder-Decoder based Long Short Term Memory Method of sequence to sequence mechanism for getting the final output of prediction. We also used other predictive models to check the comparative results.

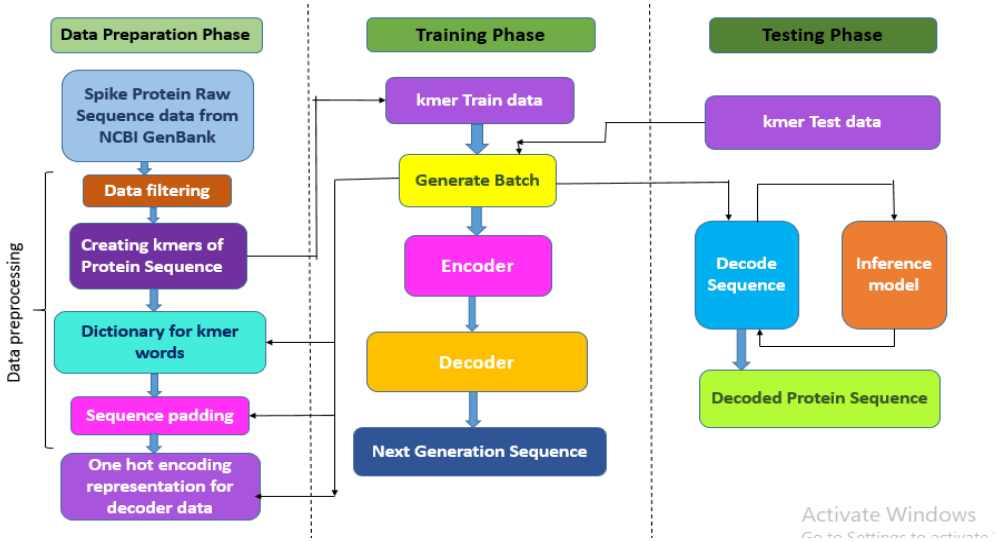

Figure 1. Workflow Diagram.

Data preparation phase

In this phase, we prepared the data for feeding it in our model. We took the Spike protein sequence data of COVID-19 from NCBI data bank ("SARS-CoV-2 protein datasets - NCBI Datasets", 2022) and will order the data according to the month. After that, we preprocessed the data by reducing duplicate data and un-useful incomplete data sequences. We divided the total dataset into an 8:2 ratio for training and testing respectively. We specially worked with the spike protein sequence as they are the most important part of COVID-19 attaching to the host through their spike protein. Hence the Scientists work on discovering vaccines targeting this protein and mutation on this protein sequence directly affects the phenotypic behavior.

Data preprocessing

Our preprocessing steps include Filtering out irrelevant data from the dataset, splitting the data, representing 3mers of protein sequence, making the dictionary for produced k-mers of input sequences and target sequences, padding the sequence with the highest length of the sequence and representing in a one hot encoding vector for decoder data that will be fed into the predictive model.

Collecting Data: We collected our protein spike sequence data from The National Center for Biotechnology Information (NCBI). It is part of the United States National Library of Medicine, a branch of the National Institutes of Health. We collected worldwide data so the mutation information can be generalized. We collected the data in fasta format and convert it into csv format only taking the primary sequence discarding other information.

Sorting Data: We sort the data by accession date so that the evolution of protein changes can be preserved.

Filtering Uut Un-useful Data: We filtered out those data which have less information and which are duplicates. We also filtered out data that is incomplete and have lengths below standard length.

Feature extraction

Feature extraction helps to shape data in a useful format that can be fed into the classifier. Features are machine understandable codes that describe the characteristics of different data. By feature extraction, data become ready as input for the next step. This helps the predictive model to work more efficiently in predicting the data. In our work, feature extraction methods includes data tokenization using k-mers counting, generating dictionary, protein sequence padding, and one hot encoding for decoder target data. The details are discussed in the following subsections.

Data tokenization using kmers counting

K-mers counting is a popular method in Natural Language Processing for making bigrams or trigrams of words. Here we used this k-mers counting method for making words from our spike protein sequences. In our work, we worked with 3-mers which is a meaningful word in protein sequences. To make the mutation more focusing we applied k-mers with overlapping. For example, if a sequence is MFVFLVLLP then the k-mers result will be MFV FVF VFL FLV LVL VLL LLP. The architecture is given in the following Figure 2.

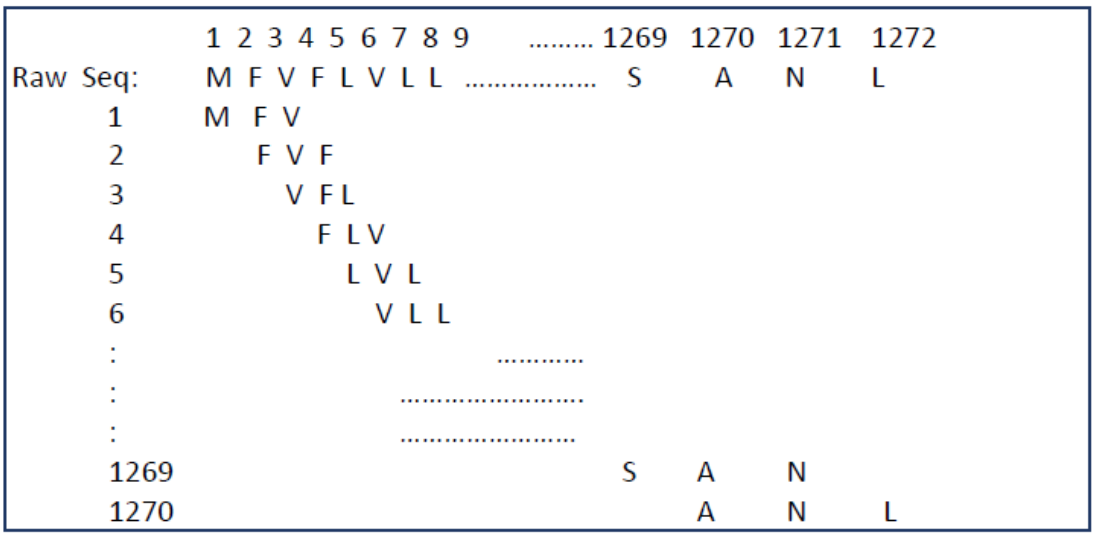

Figure 2. K-mers counting of protein sequence.

After k-mers counting our protein sequence, the data would look like Table 1.

Generating Amino Acid dictionary

Using a word dictionary is an efficient method for the operation of text sequence data. This method can be applied in a wide range of natural language processing problems i.e. text summarization, machine translation, picture captioning, video captioning, and so on. In this method, we can construct a distributed representation of biological sequences. So we used this method for creating the dictionary for our 3-mers words of protein sequence. The output will be for later use in the encoder-decoder model. After this the data dictionary would look like Table 2.

Table 1. 3-mers of protein sequence.

MFVFLVLLPLVSSQCVNLTTRTQLPPAYTNSFTRGVYYPDKVFRSSVLHSTQDLFLPFFSNVTWF HAIHVSGTNG…… …… …… ……..LQELGKYEQYIKW PWYIWLGFIAGLIAIVMVTIMLCCMTSCCSCLKGCCSCGSCCKFDEDDSEPVLKGVKLHYT
MFV FVF VFL FLV LVL VLL LLP LPL PLV LVS VSS SSQ SQC QCV CVN VNL NLT LTT TTR TRT RTQ TQL QLP LPP PPA PAY AYT YTN TNS NSF SFT FTR TRG RGV GVY VYY YYP YPD …… …… …… …….. KFD FDE DED EDD DDS DSE SEP EPV PVL VLK LKG KGV GVK VKL KLH LHY HYT

Table 2. Dictionary of 3-mers.

Created 3-mers: MFV FVF VFL FLV LVL VLL LLP LPL PLV LVS VSS SSQ SQC QCV CVN VNL NLT LTT TTR TRT RTQ TQL QLP LPP PPA PAY AYT YTN TNS NSF SFT FTR TRG RGV GVY VYY YYP YPD …… …… …… …… KFD FDE DED EDD DDS DSE SEP EPV PVL VLK LKG KGV GVK VKL KLH LHY HYT
'AAA': 1, 'AAE': 2, 'AAL': 3, 'AAR': 4, 'AAT': 5, 'AAX': 6, 'AAY': 7, 'ADA': 8, 'ADQ': 9, 'ADS': 10,…… …… …… …… …… ……..'YVP': 1831, 'YVS': 1832, 'YVT': 1833, 'YVX': 1834, 'YXG': 1835, 'YXK': 1836, 'YXX': 1837, 'YYH': 1838, 'YYK': 1839, 'YYP': 1840, 'YYV': 1841, 'YYX': 1842, 'YYY': 1843

We also did the reverse mapping from number to sequence word.

Sequence padding

After generating the dictionary of our words, the input will be then padded. The reason behind padding the input and the output is that the protein sequences can be of varying length while the LSTM expects instances with the same length. So, we changed the sequences into fixed-length vectors by using the padding. In

sequence padding, a specific length is defined for a sentence. In our case, we set the length of the longest sentence in the inputs and outputs to be used for padding the input and output sentences, respectively.

One hot encoding representation

The one hot representation is used here for the decoder target data. Each word is converted to a one-hot vector. A sequence of encoded words is then produced from the retrieved words of the protein sequence.

Training Phase

Spike protein primary sequence data from NCBI are used to assess the effectiveness of our proposed model. We worked with 25576 protein sequence data collected from December 2019 to December 2020. We trained the model with 80% of the total dataset and the remaining 20% are used for evaluating the model's effectiveness. In the following phase, we will describe the detail implementation of our experiment.

Recurrent Neural Network

RNN is an efficient method for modeling data sequencing that can handle time dependent learning problems. The basic task of RNN is to take the past information to generate output. It is also useful for learning temporal information. This method functions as follows:

$$h_t = \begin{cases} 0 & t = 0 \\ \varphi(W_{x_i}, x_t) & otherwise \end{cases} \quad (1)$$

And recurrent hidden step is updated like:

$$h_t = g(W_{x_i} + uh_{t-1}) \quad (2)$$

This method works efficiently on time series sequence data. But it has some pitfalls in considering sequences with bigger length. (Yan et. al., 2020) The Long Short Term Memory method works in the same way but it is more efficient for longer sequences.

Long short term memory (LSTM) method

Long short term memory method is one of powerful RNN models. This enables the data to learn dependencies and perform time series forecasting. It can handle vanishing problem and exploding gradients. Each LSTM includes memory cell and three gates to control the information flow named input, forge and output gates which are normally formed with logistic regression of weighted sums. The memory cell has the extended capacity capturing long term dependencies. (Hochreiter et. al., 1997) So this method served our purpose better. This encoder automatically detects the features from hidden state and updates on the 3-mer matrix are done in this phase.

Convolutional Neural Network (CNN) based long short term memory (LSTM) method

Inputs like images with spatial structure, cannot be trained easily with the standard LSTM. The CNN Long Short-Term Memory Network is an LSTM architecture which is specifically designed for sequence prediction problems with spatial inputs, like images or videos. The architecture of CNN LSTM comprises of using Convolutional Neural Network (CNN) layers for feature extraction on input data and the output of it goes with LSTMs to support sequence prediction. This model has also been used in the field of speech recognition and natural language processing problems where CNNs are used as feature extractors for the LSTMs on audio and textual input data. This model is appropriate for problems that:

- Have spatial structure in their input such as the 2D structure or pixels in an image or the 1D structure of words in a sentence, paragraph, or document.

- Requires the generation of output also with temporal structure like words in a textual description, or that have input with temporal structure like the order of or words in text or images in a video.

The basic Architecture of the model is illustrated in Figure 3.

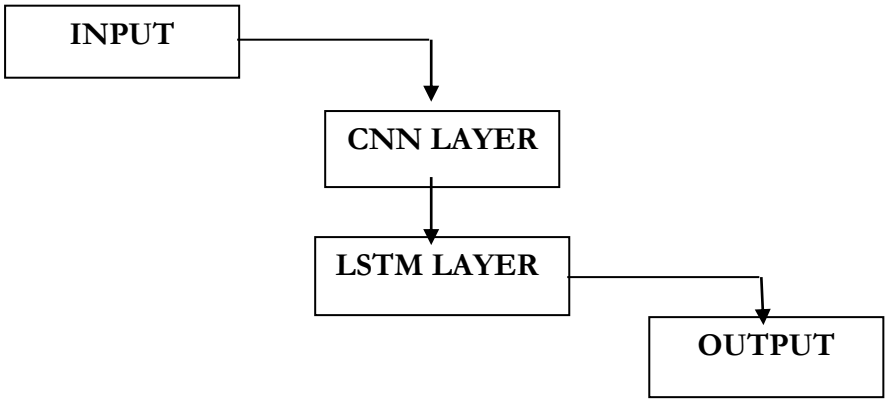

Figure 3. CNN-LSTM model basic architecture.

Encoder Decoder based long short term memory (LSTM) method

In our research we choose encoder decoder based sequence to sequence LSTM model.(Sutskever et. al., 2014) We choose it specially because it is more generalized. The problem in simple RNN is that it cannot handle large sequence of data. The pitfalls of Attention mechanism is space complexity for large sequences as it stores weights for the input in every words which requires much of space. And the problem with CNN based LSTM is that it works better for image which have spatial structures. The Architecture of our model is given in the below figure.

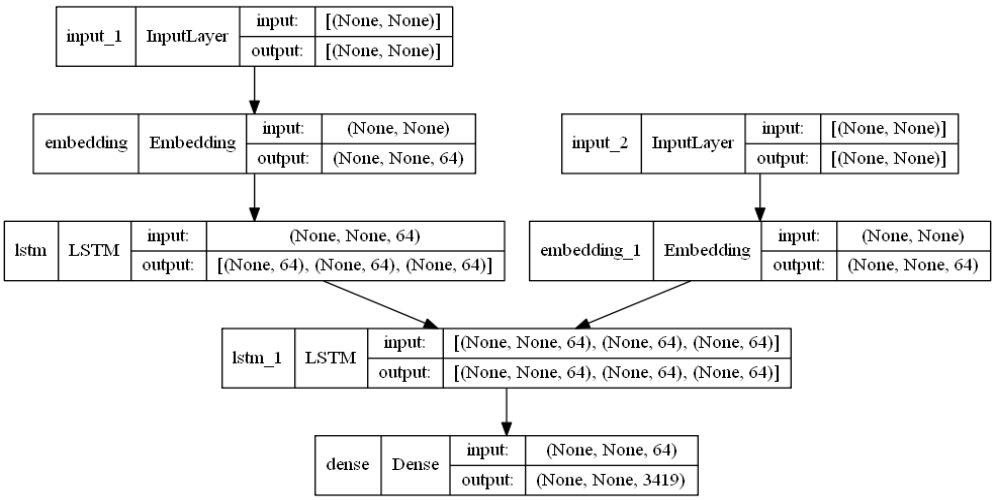

Figure 4. Architecture of proposed Encoder Decoder based LSTM model.

This architecture includes two models: one model is for reading the input sequence and encoding it into a fixed-length vector, and the second for decoding the fixed-length vector and outputting the predicted protein sequence. The Transformed words are fed into the model orderly, then the encoder gives output in an internal representation of input data then fed intro the decoder model where the internal representation gives output to y_t. Then the predicted output and the actual output are compared and minimize the loss or cross entropy between them. The optimizer used here is rmsprop.

Testing phase

We tested our model with the data that we had split before. The data are preprocessed. Then the data was given as input to the model to see the results. After that we evaluate our performance by accuracy and loss function using categorical cross entropy.

Preprocessed test data

The preprocessed 20% of data splitted in previous section 2 are now used for testing the result. As it is a predictive model, we took the last 20% data for testing our model. In the train test split the shuffle of data were off as we needed the data sequentially for our experimental analysis.

Generate batch

After taking the preprocessed data we pass it in the generate batch function where data are further processed and return sequences to fed into the encoder decoder model for predicting next possible sequence. After this the data are passed in the decode sequence function where they go through further processing.

Inference model

Inference model are specially used in machine learning model to predict our output sequences by considering weights from a pre-trained model. In other terms, we can conclude that it is a model that reduces computation for the properties that are learned in training phase and are now used for predicting new sequences.

Decode sequence

Decode sequence function is used to decode the sequences passing from inference model and concatenate word by word and make a complete protein sequence. Here the generated sequence is the targeted k-mers of the output sequence. We make this output so that any changes in protein sequence would affect the three consecutive k-mers. Hence it would be more focused in the mutation part.

Results

The model was implemented using Keras library, LSTM layers in Tensorflow 2.7. on jupyter notebook 6.4.5 on the corei 7 GPU with 24GB RAM 64-bit operating system, x64-based processor, and Panda library in Python was used for reading the CSV file and preprocessing protein sequences. We collected the data from the NCBI data source in FASTA format then run a python program to convert it into CSV form. To evaluate the prediction performance for different test cases, two measures, Accuracy and Loss are calculated. We compare our result with the CNN based LSTM model and Attention based LSTM. But in the case of Attention based LSTM larger amount of space is needed to feed with weights for each word in long sequences. So for this, we took less amount of data and experimented with it and analyze the results according to it.

We collected the data in FASTA format ordered by accession date then convert it into CSV format discarding all other information and just keeping the raw sequence which is already in order. All experiments are performed using python language. We used the following dataset. For our experimental analysis, we use an input sequence of length 1273, batch size of 50 with latent dimension = 64.

Table 3. Category of dataset

Virus Name	Protein Sequence name	Source	Length of complete Sequence	Number of samples	Ordered by	Time Duration
SARS-Cov-2	S protein	NCBI	1273	25576	Accession date	2019-2020

The experimental result of predicting future mutation using basic encoder decoder based LSTM are obtained by using the above dataset. As the mutation is not frequent enough compared with the huge amount of data we fed into our model, 99% accuracy is obtained in just 3 epochs. The experimental results are given in the following table.

Table 4. Experimental result of Encoder- Decoder based LSTM

Epoch	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss
1	13.34	56.86	1.40	0.81
2	87.02	98.77	0.39	0.12
3	99.61	99.64	0.04	0.01
4	99.85	99.87	0.006	0.004
5	99.89	99.89	0.0035	0.0029

Figure 5 shows the curve for training and testing accuracy from 1 to 5 epoch in x axis and accuracy in y axis. The curve of loss function is also shown in figure 6 where x axis denotes epochs and y axis is the scale for Loss function.

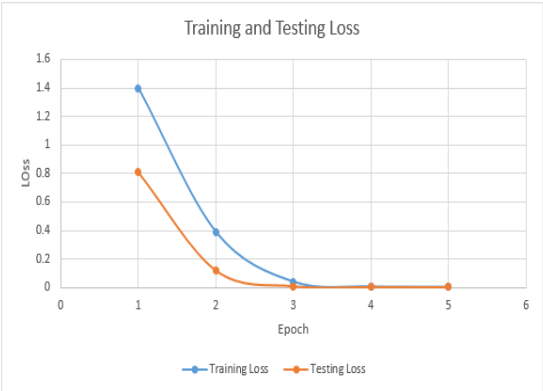

Figure 5. Train and Test Loss curve.

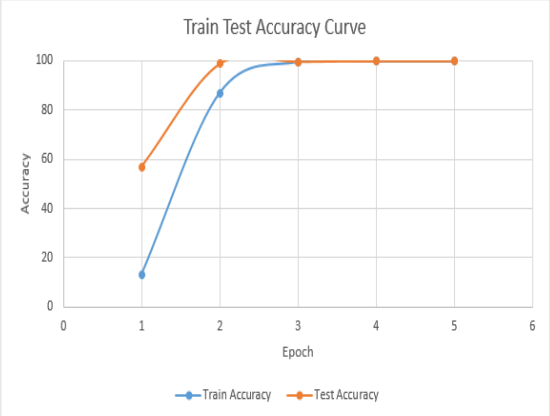

Figure 6. Training and Testing Accuracy Curve.

For predicting sequence we also experimented with CNN based LSTM model. In this case, all the parameters remained the same. For 25576 data 99.37% accuracy resulted in 10 epochs. The experimented results are illustrated in the following table.

Table 5. Experimental result for CNN based LSTM

Epoch	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss
1	20.11	38.91	2.6190	2.0767
2	46.65	78.09	1.8586	1.2904
3	68.81	92.62	1.4265	0.9750
4	78.51	97.03	1.2250	0.8302
5	83.39	98.18	1.1134	0.7578
6	86.19	99.06	1.0420	0.7055
7	87.99	99.13	0.9907	0.6726
8	89.27	99.20	0.9516	0.6585
9	90.20	99.37	0.9209	0.6260
10	90.96	99.37	0.8944	0.6094

We compared the result of CNN based LSTM with our proposed model and found that encoder decoder based LSTM gives better accuracy but in the case of time complexity CNN based LSTM performs efficiently. The comparative result analysis is given in the following Table 4.

Table 6. Comparative result analysis of encoder-decoder based LSTM and CNN based LSTM

Parameters	Encoder-Decoder based LSTM	CNN based LSTM
Training Accuracy	99.89	83.39
Testing Accuracy	99.89	98.18
Training Loss	0.0035	1.1134
Testing Loss	0.0029	0.7578
Epoch	5	5
Train-Test Split	8:2	8:2

As the attention-based model needed larger space for our data, so we took 1517 protein sequences of the first four months of 2020 and experimented with it to observe the comparative results. The encoder-decoder based LSTM again outperformed in the evaluation of accuracy and loss with the reduced amount of data. Table 5 shows the obtained results in experimental Analysis with Encoder-Decoder LSTM, CNN-LSTM and Attention- LSTM model keeping all the parameters same. Figure 7 shows the performance in bar chart form where y axis denotes accuracy.

Discussion

We focused on a model that can predict future protein sequence with possible mutations. So that it can help the researchers in effective vaccine designs that can also focus on future possible mutations. We individually took the protein sequence data for our analysis because it is the main cause of changing in phenotypic behavior of the viruses. When the protein changes due to mutation, there happens the phenotypic changes and the virus becomes more or less strong according to the changes in protein sequences. And we took the spike S protein

Table 7. Performance analysis of Encoder-Decoder LSTM, CNN-LSTM and Attention-LSTM

Parameter	Encoder Decoder LSTM	CNN-LSTM	Attention- LSTM
Data	1517	1517	1517
Train-Test Split	9:1	9:1	9:1
Training Accuracy	91.86	37.73	84.50
Testing Accuracy	93.66	44.83	90.18
Training Loss	0.454	2.14	1.64
Testing Loss	0.3876	1.97	1.48
Epoch	15	15	15

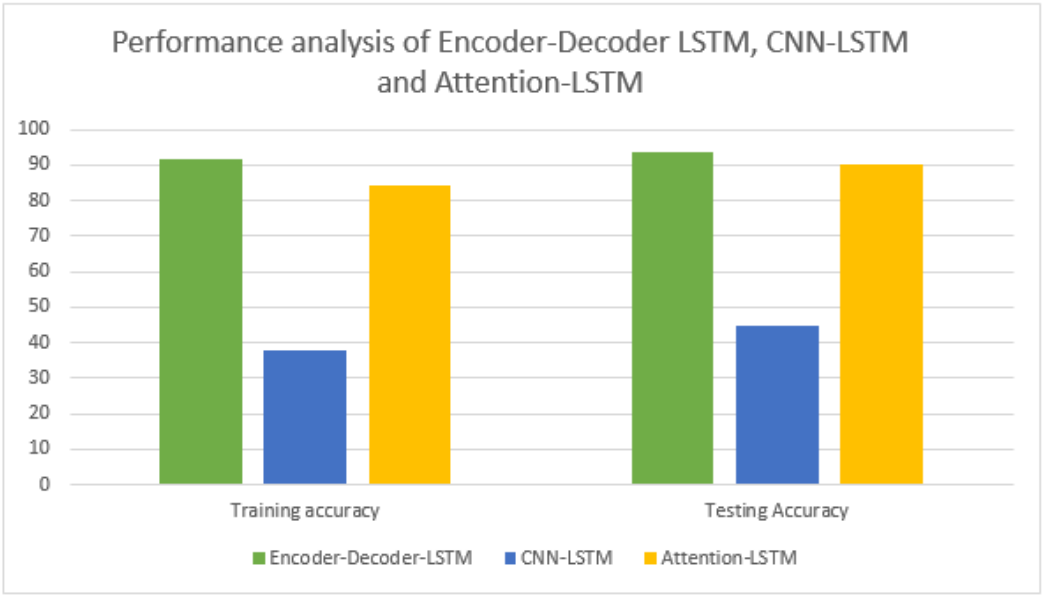

Figure 7. Performance analysis of Encoder-Decoder LSTM, CNN-LSTM and Attention-LSTM.

of SARS-CoV-2 because it is the most important part that is responsible for the virus to enter in our body and affect us.

We also did a comparative analysis about what we have done in our research with other most related researches. We took the research work of Takwa Mohammad et.al. and another research work of Refat Khan et. al. and analyzed them with our work. Table 6 denotes the comparative analysis.

Table 8. Analysis model with other experimental works.

Research work	Dataset	Source	Year range	Number of data	Length of Sequence	Accuracy (%)	Epoch
Takwa Mohamed et.al., 2021	Influenza genome Sequence (DNA) (H1N1)	Medline data bank	1935-2017	4609	535	96.98	50
Refat Khan et.al., 2020	Influenza Protein Sequence (H1N1, H3N2, H5N1)	Influenza Virus Resource	1991-2016	H1N1-8470, H3N2- 7703, H5N1- 2213	H1N1-566, H3N2- 566, H5N1- 568	95.2, 95.0, 98.9	15
Our model	SARS-CoV-2 S Protein Sequence	NCBI GenBank	2019-2020	Spike S Protein 25576	1273	99.89	05

The work is done using conventional LSTM methods that can usefully predict next generation protein sequence with high accuracy. The power of LSTM in extracting evolutionary data from sequence can be of great use in future with more robust and hybrid use of machine learning models that can assist the researchers in different sectors of biomedical engineering.

Conclusion

COVID-19 mutation prediction is very challenging as well as important task to trace the mutation pattern of it. In this article, an encoder decoder based LSTM model is proposed to predict next generation sequence. We applied encoder-decoder based long short term memory method on date wise ordered data of spike protein sequence and predicted the future mutation sequence which can give insights in the evolution of mutation. We compared our model with CNN-LSTM and Attention-LSTM for both in large dataset and small datasets. We found that Encoder-Decoder based LSTM model gives the best result in less epoch. But in term of time complexity CNN-LSTM worked efficiently. We believe this can broaden the way to a novel insight of the evolution of SARS-CoV-2 virus. It will also promote the possibilities of designing an effective vaccine against the virus. The traditional encoder-decoder based LSTM can be converted to a dynamic large scales of LSTM to have more intuitive about the evolution of virus mutations. Another avenue for research could be on the Attention Mechanism in putting attention on a specific site in the protein sequence data so that it reduces the space complexity which requires in depth knowledge of protein sequences as well as computational complexity.

Acknowledgement

Technical supports from the Khulna University Computer Lab has been acknowledged.

References

Wang, C., Horby, P. W., Hayden, F. G., & Gao, G. F. (2020). A novel coronavirus outbreak of global health concern. *The lancet*, *395*(10223), 470-473.

Ducharme, J. (2020). *The WHO Just Declared Coronavirus COVID-19 a Pandemic*. Time. Retrieved 28 May 2020, from https://time.com/5791661/who-coronavirus-pandemic-declaration/.

Hadfield, J., Megill, C., Bell, S. M., Huddleston, J., Potter, B., Callender, C., & Neher, R. A. (2018). Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, *34*(23), 4121-4123.

Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons & Fractals*, *140*, 110121.

Pathan, R. K., Biswas, M., & Khandaker, M. U. (2020). Time series prediction of COVID-19 by mutation rate analysis using recurrent neural network-based LSTM model. *Chaos, Solitons & Fractals*, *138*, 110018.

Dabbura, I. (2018, September 17). K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. Towards Data Science. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

Asgari, E., & Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, *10*(11), e0141287.

Randhawa, G. S., Soltysiak, M. P., El Roz, H., de Souza, C. P., Hill, K. A., & Kari, L. (2020). Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. *Plos one*, *15*(4), e0232391.

Salama, M. A., Hassanien, A. E., & Mostafa, A. (2016). The prediction of virus mutation using neural networks and rough set techniques. *EURASIP Journal on Bioinformatics and Systems Biology*, *2016*(1), 1-11.

Yin, R., Luusua, E., Dabrowski, J., Zhang, Y., & Kwoh, C. K. (2020). Tempel: time-series mutation prediction of influenza A viruses via attention-based recurrent neural networks. *Bioinformatics*, *36*(9), 2697-2704.

Koyama, T., Platt, D., & Parida, L. (2020). Variant analysis of SARS-CoV-2 genomes. *Bulletin of the World Health Organization*, *98*(7), 495.

Kargarfard, F., Sami, A., Hemmatzadeh, F., & Ebrahimie, E. (2019). Identifying mutation positions in all segments of influenza genome enables better differentiation between pandemic and seasonal strains. *Gene*, *697*, 78-85.

Bioinformatics. (n.d.). National Center for Biotechnology Information (NCBI). Retrieved Dec 1, 2020 from http://www.ncbi.nlm.nih.gov/Class/MLACourse/ Modules/MolBioReview/bioinformatics.html.

Luscombe, N. M., Greenbaum, D., & Gerstein, M. (2001). What is bioinformatics? An introduction and overview. *Yearbook of medical informatics*, *10*(01), 83-100.

Liu, H., Wang, Z., Wu, Y., Zheng, D., Sun, C., Bi, D., ... & Xu, T. (2007). Molecular epidemiological analysis of Newcastle disease virus isolated in China in 2005. *Journal of Virological Methods*, *140*(1-2), 206-211.

SARS-CoV-2 protein datasets - NCBI Datasets. NCBI. (2022). Retrieved 1 January 2021, from https://www.ncbi.nlm.nih.gov/datasets/coronavirus/proteins/.

Mohamed, T., Sayed, S., Salah, A., & Houssein, E. H. (2021). Long Short-Term Memory Neural Networks for RNA Viruses Mutations Prediction. *Mathematical Problems in Engineering*, *2021*.

Priya, P., Basit, A., & Bandyopadhyay, P. (2022). A strategy to optimize the peptide-based inhibitors against different mutants of the spike protein of SARS-CoV-2. *bioRxiv*.

Yan, S., & Wu, G. (2020, November). Application of neural network to predict mutations in proteins from influenza A viruses-A review of our approaches with implication for predicting mutations in coronaviruses. In *Journal of Physics: Conference Series* (Vol. 1682, No. 1, p. 012019). IOP Publishing.

Albert, S. (2017). A big data approach in mutation analysis and prediction. *Studia Universitatis Babes-Bolyai, Informatica*, *62*(1).

Walsh, I., Pollastri, G., & Tosatto, S. C. (2016). Correct machine learning on protein sequences: a peer-reviewing perspective. *Briefings in bioinformatics*, *17*(5), 831-840.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Wang, C., Liu, Z., Chen, Z., Huang, X., Xu, M., He, T., & Zhang, Z. (2020). The establishment of reference sequence for SARS-CoV-2 and variation analysis. *Journal of medical virology*, *92*(6), 667-674.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Mottaqi, M. S., Mohammadipanah, F., & Sajedi, H. (2021). Contribution of machine learning approaches in response to SARS-CoV-2 infection. *Informatics in Medicine Unlocked*, *23*, 100526.

Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International journal of computer applications*, *175*(4), 7-9.