

Predicting Novel coronavirus(nCoV) strains  
detecting the mutation process applying neural networking

by

Abdullah Hasan Sajjad Rafi

19301097

Arkadeep Das

19101431

Moumita Das

19301209

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
January 2024

© 2024. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**



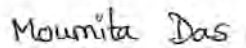
---

Abdullah Hasan Sajjad Rafi  
19301097



---

Arkadeep Das  
19101431



---

Moumita Das  
19301209

# Approval

The thesis/project titled “Predicting Novel coronavirus(nCoV) strains detecting the mutation process applying neural networking” submitted by

1. Abdullah Hasan Sajjad Rafi (19301097)
2. Arkadeep Das (19101431)
3. Moumita Das (19301209)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 9, 2024.

## Examining Committee:

Supervisor:  
(Member)



---

Arif Shakil  
Lecturer

Department of Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)

---

Md. Golam Rabiul Alam, PhD  
Professor

Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)

---

Sadia Hamid Kazi, PhD  
Chairperson and Associate Professor  
Department of Computer Science and Engineering  
Brac University

## Abstract

As viruses undergo rapid evolution, the SARS-CoV-2 which is known as Covid-19 has persisted in human populations for approximately three and a half years rapidly, continually exhibiting swift and unpredictable mutations. The relentless emergence of various new strains of SARS-CoV-2 has posed a significant challenge, leaving researchers grappling for effective strategies. This study employs a machine learning approach known as the Seq2Seq model to predict future new variants of the Human Coronavirus family by using the genome sequences of Human Coronaviruses in time series manner based on their first evolution. Through this methodology, the research successfully predicts and generates the future possible variants genome sequence of Human Coronavirus. This model would be a useful tool to predict genome sequences of future Human Coronaviruses and get important insights of the future variants to tackle the problem of fast evaluation of the Human Coronaviruses.

**Keywords:** Novel coronavirus(nCoV); Neural Network; mutation; machine learning; Prediction; variants; genome sequences; parent viruses; child viruses

## **Acknowledgement**

Firstly, all praise to the Almighty for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Mr. Arif Shakil sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

# Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Nomenclature	viii
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Research Problem . . . . .	2
1.3 Research Objectives . . . . .	4
1.4 Document Outline . . . . .	5
<b>2 Literature Review</b>	<b>6</b>
2.1 Background Study . . . . .	6
2.2 Related Work . . . . .	7
2.2.1 COVID Variant Prediction . . . . .	7
2.2.2 AI for Genome Analysis . . . . .	7
2.2.3 Predictive Genetics Analytics . . . . .	7
2.2.4 MutaGAN for Mutation Prediction . . . . .	8
2.2.5 Alignment-Free Genome Analysis . . . . .	8
2.2.6 ML in DNA Sequence Mining . . . . .	8
2.2.7 Nucleotide Patterns in COVID . . . . .	9
2.2.8 PCA in Genome Studies . . . . .	10
2.2.9 Neural Network Mutation Prediction . . . . .	10
2.2.10 Indian SARS-CoV-2 Analysis . . . . .	10
2.2.11 Mutation Prediction with LSTM . . . . .	11
<b>3 Methodology</b>	<b>12</b>
3.1 Data Acquisition . . . . .	12
3.2 Data Description . . . . .	13

3.2.1	SARS . . . . .	13
3.2.2	Human Coronavirus NL63 . . . . .	13
3.2.3	Human Coronavirus HKU1 . . . . .	14
3.2.4	MERS . . . . .	14
3.2.5	Human Coronavirus 229E . . . . .	14
3.2.6	COVID-19 . . . . .	15
3.2.7	Delta Variant . . . . .	15
3.2.8	Gamma Variant . . . . .	15
3.2.9	Omicron Variant . . . . .	16
3.3	Data Pre-Processing . . . . .	16
3.3.1	Removing Unnecessary characters . . . . .	16
3.3.2	Feature and Label . . . . .	16
3.3.3	Train Validation and Test set . . . . .	17
3.3.4	One Hot Encoding . . . . .	17
3.3.5	Slicing . . . . .	18
3.4	Working procedure of the model . . . . .	18
3.5	Neural network . . . . .	19
3.6	Recurrent Neural Network (RNN) . . . . .	21
3.7	Long Short Term Memory networks (LSTMs): . . . . .	23
3.8	SEQ2SEQ model . . . . .	25
3.9	Inference Model to generate mutated genome sequence . . . . .	27
<b>4</b>	<b>Result and Discussion</b>	<b>28</b>
4.1	Validation and train graph . . . . .	28
4.1.1	Train and Validation accuracy graph . . . . .	28
4.1.2	Train and Validation loss graph . . . . .	31
<b>5</b>	<b>Conclusion</b>	<b>35</b>
	<b>Bibliography</b>	<b>38</b>
	<b>Appendix: Google Drive link of all the Result Graphs</b>	<b>39</b>

# List of Figures

3.1	Workflow Overview . . . . .	12
3.2	Dataset . . . . .	13
3.3	Categories of sequence problems . . . . .	19
3.4	Single layer feed forward neural network . . . . .	20
3.5	Multiple layer feed forward neural network . . . . .	20
3.6	RNN Unfolded architecture. (Poudel, 2023) [30] . . . . .	22
3.7	Architecture of a LSTM Unit [31] . . . . .	24
3.8	Architecture of the seq2seq model . . . . .	26
3.9	Seq2seq model architecture in our problem . . . . .	26
3.10	Encoder model architecture for inference model . . . . .	27
3.11	Decoder model architecture for inference model. . . . .	27
4.1	Few graphs of train and validation accuracy. The given graphs are for the genome sequence 0 to 2699 . . . . .	31
4.2	Few graphs of train and validation loss. The given graphs are for the genome sequence 0 to 2699 . . . . .	34



# List of Tables

3.1	SARS . . . . .	13
3.2	Human Coronavirus NL63 . . . . .	13
3.3	Human Coronavirus HKU1 . . . . .	14
3.4	MERS . . . . .	14
3.5	Human Coronavirus 229E . . . . .	14
3.6	COVID-19 . . . . .	15
3.7	Delta Variant . . . . .	15
3.8	Gamma Variant . . . . .	15
3.9	Omicron Variant . . . . .	16
3.10	Feature and Label . . . . .	16
3.11	Training Set . . . . .	17
3.12	Validation set . . . . .	17
3.13	Test set . . . . .	17

# Chapter 1

## Introduction

### 1.1 Introduction

The COVID-19 outbreak, initially known as SARS-CoV-2 which emerged in China in December of 2019. In a brief period, this disease rapidly spread worldwide, causing a significant rise in both the amount of infected people and the amount of deaths. Despite a substantial decline in both daily new cases and deaths since mid-2022, new cases were yet persistent, and novel variants continue to emerge. According to WHO, on December 25, 2022, the total number of weekly new cases reached its peak, around 44.2 million. Even the WHO's latest weekly epidemiological update from 20 November to 17 December 2023, states that nearly 850,000 new COVID-19 cases and over 3000 deaths were reported globally. As of 17 December 2023, WHO tallied over 772 million confirmed cases and nearly 7 million deaths reported worldwide. The abrupt onset and rapid evolution of novel coronavirus (nCoV) strains created significant challenges in combating the pandemic. Within a mere couple of years, it brought numerous new variants and strains, increasing the difficulty of curbing its propagation and devising effective defenses. From the beginning, researchers and health organizations have encountered tremendous difficulty in early detection of the disease. Initially, COVID-19 cases were examined using a test called the RT-PCR (Reverse Transcription Polymerase Chain Reaction) test which detects the genetic material of the virus in a person's sample, usually collected from a nasal swab. It was the gold standard for diagnosing COVID-19 in the early stages. Throughout advancement, various other diagnostic methodologies emerged, including antigen tests. These tests work by detecting specific proteins on the surface of the virus. They are quicker and easier to implement but less sensitive than PCR tests. Later, further advancements were made in testing technology. Some countries introduced rapid tests that could provide results in minutes, allowing for quicker identification of cases although there was a record of some errors. Efforts were made to improve the accuracy further and accessibility of testing, with innovations like at-home test kits becoming available widely.

Despite substantial advancements in virus detection and global vaccination efforts, the emergence of new strains boasting enhanced transmissibility and immune evasion capabilities continued to precipitate upheaval from time to time in different regions. New strains of the virus continue to surface due to the nature of viruses to mutate over time. The virus responsible for COVID-19, like many viruses, un-

dergoes genetic changes as it spreads among populations. These mutations can lead to the emergence of new variants with altered characteristics, such as increased transmissibility or the ability to partially evade immunity from previous infections or vaccinations. While vaccination and efforts to curb transmission can limit the virus's spread and mutation rates, they cannot completely halt these processes. Additionally, variations in vaccination coverage, virus containment measures, and population behaviors across regions can create opportunities for the virus to evolve differently in various areas, leading to sporadic outbreaks or surges driven by newly emerged, potentially more infectious variants.

To solve this problem, this study suggests using a seq2seq model for predicting the genome sequence of future mutation viruses of the CoronaVirus family. This study's main goal is to employ neural networking to explore the virus mutation tree and parent and offspring virus genomic sequences. We seek to locate potential variations by using the machine learning algorithm to recognize patterns in these sequences. The key finding of this study is the application of machine learning techniques to the detection of mutation of future corona viruses. The issue at hand is the nCoV's rapid evolution, which has led to the discovery of more than 250,000 distinct missense variations in the viral protein-coding regions. High-risk variants with improved transmissibility and immune evasion abilities have been produced by several of these alterations. The emergence of variants such as Alpha, Beta, Delta, and Omicron has underscored the importance of preemptive detection and risk evaluation of novel variants before their widespread prevalence.

The seq2seq model that has been used in this paper used the genome sequences of the coronavirus family in the time series manner based on their evolving time and trained a machine learning system to predict new unique variants. This approach enables ongoing risk evaluation of various variants and offers the chance to research new high-risk variants earlier, shortening the response time to novel variants. Predicting new nCoV variations before they emerge, comprehending the alterations and ramifications of these variants, and creating efficient defenses against them are some of the research's goals. We further intend to acquire insights into the alterations that provide the virus advantages, such as greater transmissibility and immune evasion, by integrating epidemiological investigations, laboratory tests, and machine learning methods.

To summarize, this study aims to leverage seq2seq methodologies to proactively anticipate and discern emerging nCoV variants, seeking to address the challenges posed by these evolving strains. Doing this adds to ongoing attempts to comprehend the behavior of the virus, evaluate the hazards linked to various varieties, and create prompt responses to lessen their negative effects on public health.

## 1.2 Research Problem

A primary concern revolves around the emergence of Covid-19 is the fast evolving nature of the virus. From the beginning of its epidemic, more than 250,000 variants have been identified till November 2021 and the protein-coding sequences have been stored in the GISAID database and have numerous lineages linked with it.

More than 12,750 mutations have been found in Spike protein which are considered as the targets for neutralizing antibodies. Variants, being intricate entities, consist of a collection of diverse mutations, each harboring the potential to induce unexpected alterations in the SARS-CoV-2 virus. Viruses propagate by replicating their genomes, a process prone to generating variations due to imperfect copying. Mutations in viruses typically either decrease overall fitness or remain neutral in effect, yet certain combinations of mutations can lead to the emergence of high-risk variants (HRVs). These mutations alter immune characteristics, enhancing transmissibility and other advantageous traits that become more likely with unchecked virus reproduction, facilitated by rapid population spread or encounters with hosts with compromised immune systems, such as individuals with HIV or those undergoing immune-weakening medical treatments. For example, the Alpha variant exhibited a broader spread compared to the original Wuhan strain due to heightened transmissibility, while the Beta (B.1.351) variant showed reduced susceptibility to neutralization. Conversely, the highly transmissible Delta (B.1.617.2) variant has led to increased mortality and resurgence of infections across various nations, irrespective of vaccination rates. Moreover, the extensively mutated Omicron (B.1.1.529) variant, marked by multiple concerning mutations, has been classified as a Variant of Concern (VOC) primarily due to alterations in the Spike protein.

In situations where a specific sequence of changes confers increased effectiveness, a variant may gain prevalence, prompting epidemiologists to designate it as a "variant of concern." Scientific endeavors have been dedicated to understanding the alterations introduced by these variants and their implications for months. Notably, the spread of a variation may not always be attributed to beneficial mutations; external factors, such as inadvertent transportation by a small number of individuals, can contribute to its propagation in new areas, a phenomenon termed "The Founder Effect." Unraveling the reasons behind the appearance of a variation necessitates a combination of epidemiological investigations to identify and track new variants and laboratory research to elucidate the evolving characteristics of the virus resulting from mutations.

Studies are beginning to reveal mutations that confer advantages to the virus, including increased transmissibility and potential evasion of natural and vaccine-derived immunity. For instance, the D614G mutation, identified early in the pandemic, affects coronavirus particles and spike proteins used for cellular entry, rendering the new version more contagious. Another noteworthy spike protein mutation, N501Y, associated with enhanced transmissibility, is observed in the B.1.1.7, B.1.351, and P.1 strains, all designated as variants of concern. The E484K mutation, also known as Eek, found in the B.1.351 and P.1 variants, raises concerns about "immune escape." Studies indicate reduced vaccine efficacy and potential evasion of certain virus-blocking antibodies by this variant, particularly observed in South Africa. Despite the evolutionary dynamics of the virus, it evolves more slowly than other viruses like influenza, suggesting the continued efficacy of existing vaccinations to some extent. Nonetheless, scientists remain vigilant in addressing the threat posed by variants, considering potential actions to mitigate their impact.

Therefore, predicting new variants and their probable characteristics before they

evolve is a matter of significant importance. That is why experiments are becoming vital in forecasting a variant’s transmissibility. For instance, in vitro neutralization experiments using serum from participants who have been vaccinated or serum from patients who were infected previously and have additional SARS-CoV-2 strains. Every day, numerous novel variations undergo sequencing, raising concerns about their potential to evolve into more adaptable forms and develop resistance to immunity. Predicting mutations and their potential characteristics, like heightened transmissibility, before their emergence could offer vital proactive measures in managing and containing viral spread. If such predictions were possible, targeted surveillance and intensified monitoring could have been implemented earlier to track and contain the spread of high-risk variants. Moreover, preemptive vaccine development or adjustments could have been initiated to tailor vaccines against potential alterations in the viral spike protein or other key areas, potentially mitigating the impact of these variants on vaccine efficacy.

Thus, in this paper, we propose a methodology which is the Seq2Seq approach for predicting future mutations of coronavirus variants using neural network modeling where we demonstrate that it retains features of a variant’s fitness in addition to its capacity for immunological escape. We created a system that uses the entire genome sequence of the coronavirus family viruses to train and predict new coronaviruses. Using our system we can ensure that before the new variants evolve we can detect them using a machine learning approach. It enables continuous risk assessment of different variants of novel viruses and provides the opportunity to study the new high-risk variants earlier which reduces the dealing time with new variants. Therefore, this research inquiries if using Neural Networks can facilitate the prediction of future variants and probable characteristics of the COVID-19 virus.

### 1.3 Research Objectives

memory (LSTM) and seq2seq models can help predict future changes to COVID-19 and understand new forms of the virus. With that goal in mind, this paper aims to create a methodology that amalgamates the domains of genome sequencing and advanced neural network architectures to accurately forecast the genomic sequences of emerging SARS-CoV-2 variants. After that, it aims to further explore and get a comprehensive understanding of the predictive capabilities of these models in anticipating mutations within the viral genome and extrapolating potential viral characteristics, including transmissibility and immune evasion. Thus the objectives of this research include -

**Neural Network Model Implementation:** The core focus of this research is the meticulous implementation and optimization of LSTM, GRU, and Sequence-to-Sequence models to effectively capture the sequential dependencies within the genome sequences of the coronavirus family. These models will be trained to learn and predict future mutations. The objective is to assess the ability of these models to discern patterns and evolutionary trajectories leading to variant emergence.

**Performance Evaluation and Comparison:** A comprehensive evaluation of the model’s performance will be conducted. This assessment will involve measuring the accu-

racy, precision, recall, and F1 scores in predicting known mutations and identifying emerging variants. Furthermore, a comparative analysis among these neural network architectures will be executed to determine their individual strengths and weaknesses in predicting viral mutations.

**Genomic Sequence Prediction Accuracy:** A critical aspect of this research involves scrutinizing the accuracy of predicted genomic sequences against real-time and recent data available. The aim is to meticulously examine the fidelity of the predicted mutations and variants with the actual observed mutations in the database, validating the models' predictive capabilities.

**Characterization of Probable Variant Traits:** Beyond mutation prediction, this research endeavors to extrapolate and characterize the potential traits of emerging variants. This involves investigating the implications of predicted mutations on viral characteristics such as transmissibility, immune evasion potential, and other key phenotypic attributes. The goal is to correlate these predicted mutations with known phenotypic changes observed in previous variants.

**In-Depth Analysis and Documentation:** A comprehensive analysis and documentation of the research findings, methodologies, and model performances will be conducted. Through this research, we aspire to contribute to the ongoing discourse on viral evolution prediction and proactive containment strategies that can inform public health strategies, vaccine development, and overall pandemic management.

## **1.4 Document Outline**

There are five chapters in this thesis report. The first chapter contains the “Introduction” part which gives an overview of the context, research problem, its necessity, and finally the objectives of this research. The second chapter “Literature Review” discusses background study and existing related works so far, connected to this research. The third chapter, “Methodology” describes data acquisition, its pre-processing mechanism, and the working procedure of the proposed models. After that, the results of the experimentation through these models and their evaluation are discussed in the fourth chapter, “Result and Discussion”. Lastly, the fifth chapter, “Conclusion” concludes the research report with a summary of the proposed work and its outcomes.

# Chapter 2

## Literature Review

### 2.1 Background Study

The Disease Prevention & Control Center in Europe states that it is crucial to monitor the spreading of many SARS-COV-2 variants across international borders. Complete genome sequencing, or at least partial complete spike (S) genome sequencing, is the most efficient method for defining a specific variation. A virus's whole or partial genome's exact nucleotide sequence can be determined using genome sequencing. One way to identify the SARS-CoV-2 mutation is to use a technology that reveals the genetic coding of the virus and its variations. In order to enable the prevalence estimate of variants of interest (VOI), volatile organic compounds (VOCs) and variants under monitoring (VUM), alternative methodologies for pre-screening and early detecting technique have been developed. These approaches include diagnostic screening NAAT based tests. NATT means nucleic acid amplification technology. While many of these techniques can precisely distinguish between the different variations, others need verification by sequencing the entire or a portion of the Spike (S) gene's genomic position in a subcluster of samples. However, genome sequencing is expensive, time-consuming, and requires specific equipment and expertise [20]. Machine learning is a technique which aids the investigation of structure-activity connections, sequence error correction and the prediction of secondary and tertiary structure evolution. It is feasible to identify potential point of mutations that might result from alignments of the main RNA sequence structure by using a machine learning technique. Evidence is shown to demonstrate how a nucleotide in an RNA sequence responds to other nucleotides in the sequence and forecasts each nucleotide's genotype [13]. On genome sequence databases, a variety of Machine Learning (ML), Deep Learning (DL) and Artificial Intelligence (AI) techniques have already applied to identify the positions of mutations and forecast more insights [24], also neural network techniques are utilized in order to predict new strains [13]. Several techniques like alignment-free (AF) sequence, SPM (Sequential Pattern Mining) and mutation analysis [29][18], Convolution Neural Network (CNN), Neural Network (NN) applied in general. Another techniques like Gradient Neurons, complete genome sequence study combined with pattern recognition [28], Fast Vertical Mining of Sequential Patterns using co-occurrence info (CM-SPAM), closed SPM applying sparse and vertical id-lists CloFAST, and coherent mining of Top-k method of Sequential Patterns (TKS) [32], Principal Component Analysis (PCA) [19], CNN-LSTM and Attention-based LSTM [24], LSTM-RNN (Long Short-Term Memory-

Recurrent Neural Network) and GRU-RNN (Gated Recurrent Unit-Recurrent Neural Network) [22] sequence-to-sequence [26], PyR0 forecasts [21] has also been applied so far. The objective can be compiled to predict mutations of SARS-CoV-2 performing Neural Network approaches for predicting sequences from the existing genome sequences.

## 2.2 Related Work

### 2.2.1 COVID Variant Prediction

Ullah, Amin, et al. [25] unveiled a groundbreaking network for crafting COVID-19 variants through VAE, predicting new strains via uncertainty calculations. Their study established a unified framework utilizing CNN and a self-attention model to classify COVID-19 variants. They integrated the latest data on variations, employing diverse baseline methods for classification, with the proposed network delivering optimal results. The study underwent a thorough assessment, showcasing the high efficacy of new variant generation and prediction methods for the defined objectives.

Despite its achievements, the study bears limitations. The entropy score, rooted in a singular model's prediction, yielded a comparatively lower AUC for uncertainty. This drawback could be rectified through Bayesian techniques or robust ensembles to enhance uncertainty prediction efficiency. Furthermore, the study lacked insight into specific nucleotide sequence patterns influencing classification decisions. Future initiatives aim to introduce Explainable Artificial Intelligence (XAI) into COVID-19 analysis, unraveling how neural networks discern mutations and categorize sequences.

### 2.2.2 AI for Genome Analysis

Nawaz, M. Saqib, et al. [18] suggested two methods for examining and analyzing COVID-19 genomic sequences in this paper. Using pattern mining techniques, the first approach identifies nucleotide bases that occur frequently in the sequences, their patterns, and the sequential link among them. Additionally, a variety of sequence prediction models were assessed using genome sequences, and AKOM (All-K-Order-Markov) outperformed other cutting-edge algorithms. The second method was the proposal of an algorithm to examine mutations in COVID-19 genomic sequences. In order to evaluate the mutation rates, the system locates the region or locations within COVID-19 strains where nucleotide bases are altered. Their research presents methods that are not specific to the SARS-CoV-2 virus. They might also be applied to the study of other human viruses.

### 2.2.3 Predictive Genetics Analytics

Kakulapati, V., et al. [28] significantly improved sequence models by incorporating a corpus, aiming to ascertain the predictability of nucleotide bases from one to the next. To conduct a comprehensive mutation study on genome sequences, they employed an algorithm to identify mutated positions in the sequences and quantify the



extent of mutation. The primary objective of this analysis was to ascertain the presence of genetic mutations associated with primary immunodeficiencies in COVID-19 cases.

In the pursuit of swift and accurate COVID-19 detection, the study utilized the technique of XGBoost, the technique of Convolutional Neural Networks (CNN), the technique of Gradient Neurons, and also the technique of pattern recognition in conjunction with full genomic sequence analysis. This was done to assess the effectiveness of gradient boosting tree (GBT) and neural network (NN) applications. XGBoost designs were continually employed to predict the ongoing applicability of GBT. Over time, genetic factors were observed to contribute to the emergence of novel variations, enhancing the prediction of genome sequence codon pairings and families. The proposed method played a crucial role in identifying and mapping global outbreaks of harmful viruses and local genetic variants.

#### **2.2.4 MutaGAN for Mutation Prediction**

Berman, Daniel S., et al. [26] offered MutaGAN, a novel approach in Deep Learning(DL) framework that learns a generalized time-reversible evolving model using seq2seq models and GANs. In order to achieve this, they developed a model that can replicate important features of the phylogenetic tree by creating mutations for a given input parent sequence. They then showed that it can effectively predict the mutations seen in the phylogenetic data of the HA protein of the H3N2 influenza A virus. It was the first attempt at modeling and predicting the evolution of a protein using deep learning under no human oversight and with very little human input.

#### **2.2.5 Alignment-Free Genome Analysis**

Nawaz, M. Saqib, et al.'s [29] research employed Alignment-Free (AF) sequence analysis and sequential pattern mining (SPM) to scrutinize SARS-CoV-2 genome sequences and extract pertinent information. AF techniques were utilized to discern (dis)similarities in the genomic sequences of SARS-CoV-2, assess the effectiveness of diverse interventions, and construct phylogenetic trees using various distance metrics. SPM methods were applied to identify common amino acid patterns and their interactions, aiding in the prediction of amino acids through various sequence-based models. Additionally, the study introduced a mutation analysis approach for genomic sequences. This algorithm identifies locations in the genome sequences where amino acids undergo alterations and computes the mutation rate. The outcomes underscore that SARS-CoV-2 genome sequences harbor information and patterns useful for investigating differences and evolutionary aspects between strains, a characteristic evident in both AF and SPM approaches.

#### **2.2.6 ML in DNA Sequence Mining**

Yang, Aimin, et al. [17] stated that Recent decades have seen tremendous progress in hardware technology, opening up new opportunities for life science scientists to gather vast amounts of data in areas such as biological imaging, omics and medical imaging. This development does, however, provide substantial obstacles to the

use of data mining techniques to bioinformatics study. A growing field of study focuses on architecture, algorithm development, and novel functionalities for biological information analyzing at the nexus of machine learning and bioinformatics. Interdisciplinary techniques, coupled with discoveries in Artificial Neural Networks(ANN), Deep Learning(DL), and Reinforcement Learning, are changing machine intelligence. Machine learning’s incorporation into bioinformatics promises to yield more insightful mining outcomes, which will advance society.

Their research highlights two significant aspects of DNA sequence analysis machine learning research. First of all, it highlights the necessity of resolving the sensitivity and specificity problems that are common in current algorithms by including the biological importance of DNA sequences into the data mining process. Second, when data quantities increase, conventional analytical tools become less effective in terms of processing time. This calls for the creation of effective calculating techniques, especially when combining distributed and parallel computing to improve mining efficiency.

Additionally, their studies underscores the importance of selecting appropriate DNA sequence coding methods adapted to specific tasks, aiming to improve algorithm performance and reduce training time. Sequencing technology, DNA sequence data structure, sequence similarity, machine learning techniques, and difficulties encountered in biological sequence data mining are all covered in detail in this thorough examination. In order to shed light on their biological importance, the article explores four important uses of machine learning in DNA sequence data: alignment, classification, clustering, and pattern mining. An international body of research has synthesized current findings and proposed future research paths, pointing to a tighter integration of biology and machine learning for better mining outcomes in the field of DNA sequence data.

## 2.2.7 Nucleotide Patterns in COVID

UMAR, AQSA, et al. [32] research carried out in this paper is based on the genome sequences for COVID-19 strains of Pakistan, India, Spain, United Kingdom, China, and Brazil collected fromNCBI’s GenBank.The genome sequences have been examined using four SPM algorithms: CM-SPAM,VMSP, CloFAST, and TKS. The COVID19 genomic sequence is further analyzed using the TKS method after frequent nucleotides are first retrieved using patternmining tools. Six strains of COVID19 genomic sequences are used to derive top-k sequential patterns, from which patterns encoding amino acid codons are further examined in each strain. The statistics collected indicate that the majority of amino acid codons originated in Brazil, China, Spain, and the United Kingdom. The highest percentage of support for the pattern GCA, which encodes the amino acid alanine from India, is 1.69, while the lowest percentage of support for the pattern ACC, which encodes the amino acid threonine from China, is 1.10 from all other nations.

### 2.2.8 PCA in Genome Studies

Wang and Jiang [19] conducted research that adapted frequency method-based Principal Component Analysis (PCA) techniques, originally designed for protein sequences, to accommodate the vast COVID-19 genome sequences, comprising approximately 30,000 positions post-alignment. Initially demonstrated with a randomly selected shortlist of sequences, including seventy-five (75) animal samples from bats and pangolins, the PCA approach proved effective for distinguishing between human and animal samples, as indicated by the PCA score plot.

The study revealed that the presence of end sequence ambiguity and gap positions minimally impacted the grouping on the score plot, a favorable outcome considering the diverse and potentially error-prone nature of the global COVID-19 data. The approach was applied to over 20,000 sequences, offering insights into the potential trajectory of virus mutations over time. Despite potential limitations in classifications, the PCA technique is envisioned to provide a rapid analytical method with minimal data cleaning requirements. For classification outcomes, the authors recommend combining the tool with other techniques. As the COVID-19 genome sequence database expands, this approach presents an excellent opportunity for investigating coronavirus mutations and may serve as a preprocessing step for methods like tree-building approaches. The PCA score plot analysis also suggests potential viral evolution routes, warranting further investigation into possible COVID-19 mutations.

### 2.2.9 Neural Network Mutation Prediction

Saldivar-Espinoza, Bryan, et al. [23] devised an innovative method employing an artificial neural network for the prediction of RM in the SARS-CoV-2 genome. Utilizing the SARS-CoV-2 genome sequence, SHAPE-Seq reactivity values, and additional data, they successfully forecasted the occurrence, location, and associated amino acid changes of mutations. Validation of their predictions involved a test set featuring four genes, including M-pro and spike genes, as well as practical scenarios like predicting RM in VoCs. Notably, their approach demonstrated robustness in predicting mutations over an extended period, with certain initially identified false positives transitioning into real positives over time.

Furthermore, the prediction technique discerned both positively and negatively selected locations within the SARS-CoV-2 genome. Unanticipated harmful mutations among false positives were identified, underscoring the unpredictable nature of genomic changes. Conversely, positively chosen positions occasionally occurred among false negatives, exhibiting a positive impact on the virus's spread. These findings suggest the potential discovery of antiviral medications effective against future SARS-CoV-2 mutations through such studies.

### 2.2.10 Indian SARS-CoV-2 Analysis

Saha, Indrajit, et al. [16] examined 566 Indian SARS-CoV-2 genomes in order to identify substitution, deletion, and insertion mutations as well as SNPs. 64 SNPs, 1609 point mutations categorized as substitution, deletion, and insertion, and 100 clusters of mutations—mostly deletions—have all been found by our study. Of these

64 SNPs, six coding areas include 57 of them. Finding SNPs will help determine which genomic regions may be addressed in order to categorize the viral strain that is prevalent in India. Apart from this, the main benefit was that these SNPs might be utilized to establish the vaccination dosage for a tailored vaccine after determining the appropriate viral strain.

### **2.2.11 Mutation Prediction with LSTM**

Tasnim, Sumaiya, Kamrul Hasan Talukder, and Anika Asfi. [24] implemented 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 information in the evolution of mutation. They conducted comparisons between their model and CNN-LSTM and Attention-LSTM on both small and big datasets. The best result in less epochs is obtained by the LSTM model based on encoder-decoder, they discovered. But CNN-LSTM performed well in terms of temporal complexity. They thought that this would open doors to new understandings of the SARS-CoV-2 virus's evolutionary history. ..Additionally, it will raise the likelihood of creating a vaccine that effectively combats the infection. A dynamic large-scale LSTM may be created by converting the conventional encoder-decoder based LSTM, which will provide a more intuitive understanding of how viral mutations evolve. Another area of study might be the Attention Mechanism, which focuses attention on a particular region in the protein sequence data to lower the space complexity. This needs a thorough understanding of both computational complexity and protein sequences.

# Chapter 3

## Methodology

Our thesis proposes a distinctive approach to predicting the genome sequence of the Human Coronavirus using an LSTM Encoder-Decoder-based SEQ-2-SEQ model. This model enables us to forecast the entire genome sequence of a newly mutated Human Coronavirus genome. The workflow of the entire process is summarized here:

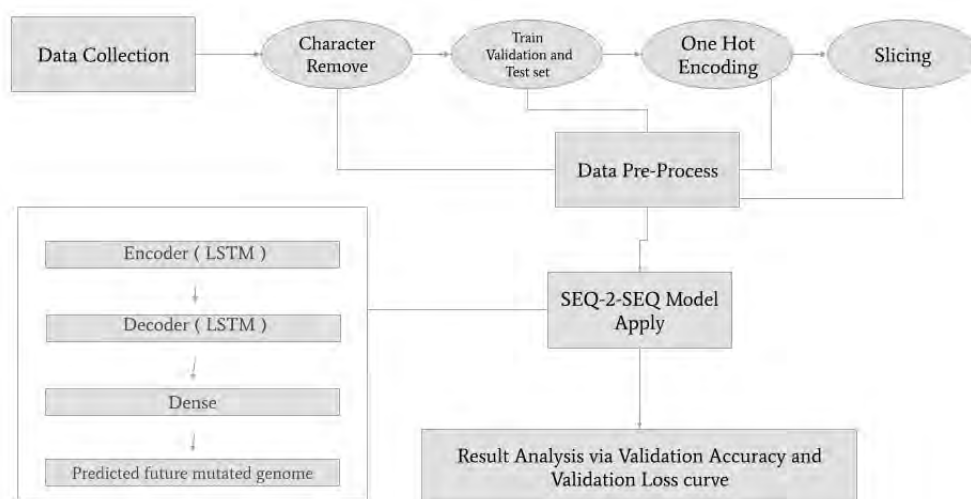


Figure 3.1: Workflow Overview

### 3.1 Data Acquisition

Our proposed system worked on genome sequence data of SARS-CoV-2's variants over the time. For predicting mutation sequences in a time series we used sequence-to-sequence approach on the SARS-CoV-2 genome sequence. We have collected information about all the variants that SARS-CoV-2 had till this date. The genome sequences were collected from the published work registered in NCBI (National Center for Biotechnology Information). The genome sequence was collected as the FASTA format. FASTA is a text-based, bioinformatic data format used to record nucleotide or amino acid sequences (e.g. Deoxyribonucleic Acid [DNA] or Ribonu-

cleic Acid [RNA]). Then the collected work was prepared as datasheet as a time series according to the following figure 1

Virus name	Medical terminology	First collected	Link of genome sequence	GenomeSequence
SARS	SARS-CoV	November 2002	<a href="https://www.ncbi.nlm.nih.gov/genbank/3573015">https://www.ncbi.nlm.nih.gov/genbank/3573015</a>	>AY310126.1 SARS coronavirus FRA, complete genome
Human Coronavirus NL63	HCoV-NL63	collected in January 2003	<a href="https://www.ncbi.nlm.nih.gov/genbank/394831.2">https://www.ncbi.nlm.nih.gov/genbank/394831.2</a>	>NC_055531.2 Human Coronavirus NL63, complete genome
Human Coronavirus HKU1	HCoV-HKU1	January 2004	<a href="https://www.ncbi.nlm.nih.gov/genbank/AY597011">https://www.ncbi.nlm.nih.gov/genbank/AY597011</a>	>AY597011.2 Human coronavirus HKU1 genotype A, complete genome
MERS	MERS-CoV	June 2012	<a href="https://www.ncbi.nlm.nih.gov/genbank/67489330">https://www.ncbi.nlm.nih.gov/genbank/67489330</a>	>NC_019843.3 Middle East respiratory syndrome-related coronavirus isolate HCoV-EMC2012, complete genome
Human Coronavirus 229E	HCoV-229E	1965, its collected in 2016	<a href="https://www.ncbi.nlm.nih.gov/genbank/MF542265">https://www.ncbi.nlm.nih.gov/genbank/MF542265</a>	>MF542265.1 Human coronavirus 229E strain 229E/Habi-1/2016, complete genome
COVID-19	SARS-CoV-2	December 2019	<a href="https://www.ncbi.nlm.nih.gov/genbank/328174254">https://www.ncbi.nlm.nih.gov/genbank/328174254</a>	>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome
Delta Variant	SARS-CoV-2 (B.1.617.2)	Oct 2020	<a href="https://www.ncbi.nlm.nih.gov/genbank/CM417166.1">https://www.ncbi.nlm.nih.gov/genbank/CM417166.1</a>	>CM417068.1 Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/USANW1749/2021, complete genome
Gamma Variant	SARS-CoV-2 P.1 (B.1.1.28.1)	Nov 2020	<a href="https://www.ncbi.nlm.nih.gov/genbank/MN985205.1">https://www.ncbi.nlm.nih.gov/genbank/MN985205.1</a>	>MN985205.1 Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/UKRY/385/2021, complete genome
Omicron Variant	SARS-CoV-2 (B.1.1.529)	Nov 2021	<a href="https://www.ncbi.nlm.nih.gov/genbank/OL672336">https://www.ncbi.nlm.nih.gov/genbank/OL672336</a>	>OL672336.1 Severe acute respiratory syndrome coronavirus 2 isolate SARS-CoV-2/human/BEL/ega-20174/2021, complete genome

Figure 3.2: Dataset

## 3.2 Data Description

### 3.2.1 SARS

First Detected	November 2002
Medical Terminology	SARS-CoV
Total Length	29740
ATCG Percentage	T:30.76, A:28.48, G:20.80, C:19.96

Table 3.1: SARS

### 3.2.2 Human Coronavirus NL63

First Detected	January 2003
Medical Terminology	HCoV-NL63
Total Length	27553
ATCG Percentage	T:39.21,A:26.32,G:20.01,C:14.44

Table 3.2: Human Coronavirus NL63

### 3.2.3 Human Coronavirus HKU1

First Detected	January 2004
Medical Terminology	HCoV-HKU1
Total Length	29926
ATCG Percentage	T:40.10,A:27.83,G:19.04,C:13.01

Table 3.3: Human Coronavirus HKU1

### 3.2.4 MERS

First Detected	June 2012
Medical Terminology	MERS-CoV
Total Length	30119
ATCG Percentage	T:30.76,A:28.48,G:20.80,C:19.96

Table 3.4: MERS

### 3.2.5 Human Coronavirus 229E

First Detected	first detected 1965, its collected in 2016
Medical Terminology	HCoV-229E
Total Length	27271
ATCG Percentage	T:34.76,A:27.22,G:21.50,C:16.52

Table 3.5: Human Coronavirus 229E

### 3.2.6 COVID-19

First Detected	December 2019
Medical Terminology	SARS-CoV-2
Total Length	29903
ATCG Percentage	T:32.08,A:29.94,G:19.61,C:18.37

Table 3.6: COVID-19

### 3.2.7 Delta Variant

First Detected	October 2020
Medical Terminology	SARS-CoV-2 (B.1.617.2)
Total Length	29708
ATCG Percentage	T:32.16,A:29.86,G:19.64,C:18.34

Table 3.7: Delta Variant

### 3.2.8 Gamma Variant

First Detected	November 2020
Medical Terminology	SARS-CoV-2 P.1 (B.1.1.28.1)
Total Length	29858
ATCG Percentage	T:32.19,A:29.86,G:19.60,C:18.35

Table 3.8: Gamma Variant



### 3.2.9 Omicron Variant

First Detected	November 2021
Medical Terminology	SARS-CoV-2 (B.1.1.529)
Total Length	29684
ATCG Percentage	T:32.14,A:29.89,G:19.63,C:18.34

Table 3.9: Omicron Variant

## 3.3 Data Pre-Processing

### 3.3.1 Removing Unnecessary characters

The text-based format known as FASTA, which is used to describe nucleotide or protein sequences, is where our genome sequence data is kept. In this arrangement, the accession number and a description of the sequence come first on a line that starts with the ">" character.

We started the preparation stages by deleting the first line of raw text, which contained the accession number and description, in order to get the data ready for analysis. This guarantees that the genetic data will be the exclusive subject of our ensuing analysis.

Furthermore, we have eliminated the unnecessary newline characters that the FASTA file used to divide lines. This is an important step because, in order to make the data more readable, it is frequently divided into many lines. By eliminating these newlines, the genomic sequence is combined into a single, continuous string.

### 3.3.2 Feature and Label

As our goal is to predict the future genome sequence, we have set it according to table 10. The input data is the previous generation genome sequence and we are matching the sequence with the output label with the next generation genome sequence.

Feature	Label
SARS-CoV	HCoV-NL63
HCoV-NL63	HCoV-HKU1
HCoV-HKU1	MERS-CoV
MERS-CoV	HCoV-229E
HCoV-229E	SARS-CoV-2
SARS-CoV-2	SARS-CoV-2 (B.1.617.2)
SARS-CoV-2 (B.1.617.2)	SARS-CoV-2 P.1 (B.1.1.28.1)
SARS-CoV-2 P.1 (B.1.1.28.1)	SARS-CoV-2 (B.1.1.529)

Table 3.10: Feature and Label

### 3.3.3 Train Validation and Test set

We have split our dataset into 3 sets- train, validation and test. The training set is needed to train our model. The validation set is essential for improving model hyperparameters, preventing overfitting, and evaluating the model's ability to optimize our model. Finally the test dataset is for test our model after training to see its performance

Feature	Label
SARS-CoV	HCoV-NL63
HCoV-NL63	HCoV-HKU1
HCoV-HKU1	MERS-CoV
MERS-CoV	HCoV-229E
HCoV-229E	SARS-CoV-2
SARS-CoV-2	SARS-CoV-2 (B.1.617.2)

Table 3.11: Training Set

Feature	Label
SARS-CoV-2 (B.1.617.2)	SARS-CoV-2 P.1 (B.1.1.28.1)

Table 3.12: Validation set

Feature	Label
SARS-CoV-2 P.1 (B.1.1.28.1)	SARS-CoV-2 (B.1.1.529)

Table 3.13: Test set

### 3.3.4 One Hot Encoding

Through vectorization, the data is made ready for use by a neural network, which transforms the characters into numerical representations that may be used to train machine learning models. Because each character is represented as a binary vector via one-hot encoding, neural network designs can utilize it.

**Target and Input Sequences:** There are original genomic sequences as input texts, target texts have genomic sequences that have been modified.

**Unique Characters:** distinguishes between unique characters in genomic sequences that are both original and modified. Ensure a space character is included in every set as a padding.

### 3.3.5 Slicing

The method used is segmenting the original 30,000 base pair genome sequences into smaller groups of 300 base pairs each. This slicing method does two things: it reduces computing complexity and makes it possible for sequence-to-sequence models to analyze data more effectively.

The encoding and decoding processes are adapted accordingly to handle these shorter segments. By treating every 300-base pair slice as a separate input-output pair, the model may be trained to identify patterns and relationships within specific genomic areas. This method detects specifics at a smaller scale and enables a more detailed analysis of genetic data.

By segmenting the genome sequences, computational efficiency increases, enabling the application of sequence-to-sequence models on hardware with restricted resources. Furthermore, this approach was helpful for training models or doing studies when it's crucial to find a balance between granularity and computing practicality.

## 3.4 Working procedure of the model

We can think of our problem as a natural language processing problem because we can think of each nucleotide base in the genome sequence as a character of a word and we have one hot encoded all the unique characters which are A,T,G,C," ". If we treat genome sequence as a word we can state our problem as a word to word prediction problem in natural language processing. So we have prepared the data set in such a way that it can be trained on a model that is applicable in the natural language processing task.

In natural language processing there are various models for this type of word to word prediction problem or from a language to another language prediction problem. Among them, Seq2seq is a prominent model. Sequence to sequence model uses the neural network in its core and it has been used in various problems like text summarization, Language translation, Image captioning [12], Text Generation Language Modelling, Semantic Parsing, Question Answering, Abstractive Text Summarization, Speech Recognition etc. The main goal of a sequence to sequence model is machine translation.

Deep neural network only applicable to sizable labeled data sets where targets and inputs may be properly encoded using vectors of constant dimensions. With them, sequence to sequence mapping (like machine translation) is not possible [11]. This served as motivation for the development of encoder-decoder models, a framework capable of handling a variety of general sequence-to-sequence problems. Advanced sequence to sequence models use this architecture as the foundation for example Transformer models, BERT, GPT models, attention models etc are mentionable.

In our scenario, we are trying to predict future mutated genome sequences From the previous genome sequences. That's why we can think of our problem as a sequence to sequence prediction problem. Because of that in this paper we used a

seq2seq approach model to predict future mutated genome sequences from the previous genome sequences that are sorted in time series manner based on their mutation.

Long short term memory (LSTM) or Gated Recurrent Unit (GRU) which are the modified versions of Recurrent Neural Network (RNN) architectures are used for many kinds of sequence problems, which can be broadly categorized into five categories of one to one, one to many, many to one and two different varieties of many to many architecture. The figure:1 shows the mentioned categories.

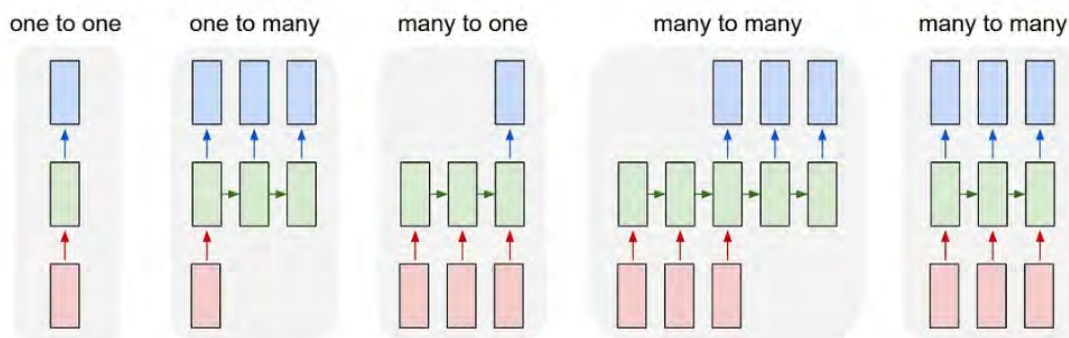


Figure 3.3: Categories of sequence problems

Before explaining the detailed architecture of the model, explaining the working mechanism of Neural Network and Recurrent Neural Network (RNN) is important because in our model we have used a modified version of RNN which is LSTM and also used a Dense layer.

### 3.5 Neural network

Neural networks are artificial intelligence models that replicate the way the human brain functions. Whereas all computations in a digital model work with ones and zeros, in a neural network, processing elements—the computer equivalent of neurons—are connected to generate new connections. The weights and connection configuration decide the output.[14] A biological brain is really just a massive collection of neurons. Each neuron receives inputs in the form of electrical and chemical impulses via its numerous dendrites, and it sends out messages via its axon (though there are some deviations to this behavior in more specialized contexts, such as multipolar neurons). At specialized connections known as synapses, axons connect with other neurons to transfer their output signals, allowing the neurons to repeatedly repeat the same process millions and millions of times [14].

Artificial neural networks come in a variety of forms. These networks are put into practice using a set of parameters and mathematical processes to determine the output [2]

One of the fundamental units of a neural network is the neuron which can be thought of as a processing unit. Based on the connections between neurons, neural network structures can be broadly separated into two groups: feed forward neural networks

and RNN. Feedback or backpropagation is present in a "recurrent neural network," as shown by synaptic connections from outputs to inputs, which could include connections with both, own and other neuron's inputs.

Feed-forward neural networks subdivided into two groups according to the quantity of layers they have: "single layer" networks and more intricate "multi-layer" networks. This layering difference is part of what gives neural networks their wide range of processing and learning from data capabilities [8].

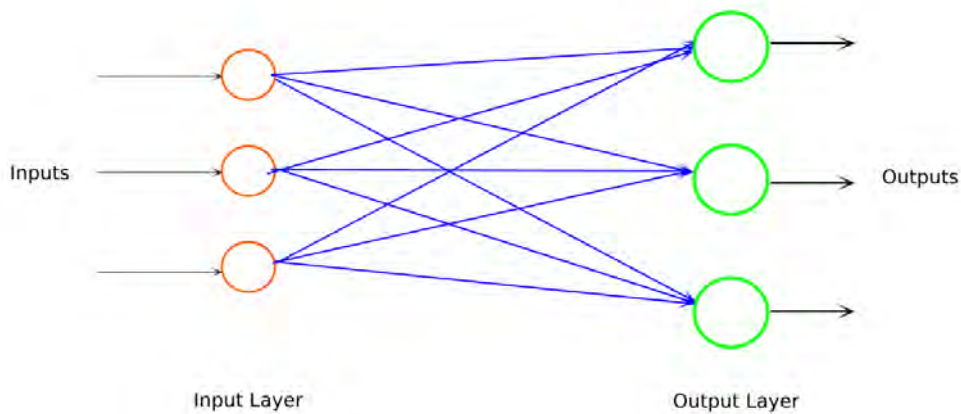


Figure 3.4: Single layer feed forward neural network

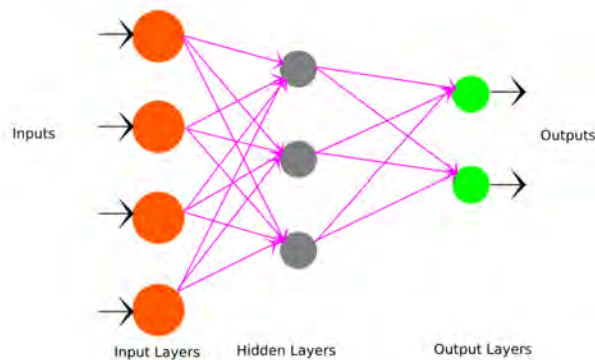


Figure 3.5: Multiple layer feed forward neural network

The purpose of hidden neurons is to act as an useful intermediary between the network's output and external input [6]. Higher-order statistics can be extracted by the network if there are one or more hidden layers. There are four input neurons, three hidden neurons, and two output neurons in the example shown in Figure 3. That's why the network is known as a 4-3-2 network.

## 3.6 Recurrent Neural Network (RNN)

In a 1986 article published in Nature, Rumelhart et al [1]. introduced the notion of RNNs in order to explain a novel learning process utilizing self-organizing neural networks.

When it comes to handling complexities for processing sequential data, RNN has been proven as one of the ideal options to the inherent constraints seen in classic neural networks, especially when compared to FeedForward Neural Networks (FNNs). RNNs are carefully designed to overcome the difficulties faced by sequential data, in contrast to FNNs, which analyze individual inputs in isolation through hidden layers, ignoring the underlying order and contextual links among distinct inputs. These specialized networks can capture the complex relationships between sequential inputs since they are designed to manage and evaluate sequential data effectively.

When compared to FNNs, RNNs have a significant advantage since they have the sophisticated potential to handle tasks that need sequential processing. Language modeling, machine translation, speech recognition, time series analysis, and other applications that need a thorough understanding of sequential relationships are examples of such tasks. FNNs struggle to succeed in these areas because of their intrinsic constraints, illustrating the necessity for more sophisticated designs such as RNNs to step in and provide a viable solution.

RNNs, in essence, are a strategic improvement over classic neural networks, providing a diverse and dynamic way to efficiently address the issues given by sequential data processing. Because of their ability to capture and use dependencies within sequences, they are a powerful tool in a variety of applications where understanding and modeling sequential interactions are critical.

Numerous modifications were suggested in addition to the SimpleRNN architecture to handle various use cases. Few advanced RNN models are GRU, LSTM, bidirectional RNN, attention models etc. [27].

A basic neural network with a feedback mechanism is a component of the SimpleRNN architecture, commonly referred to as SimpleRNN. Because of parameter sharing, which broadens the model's applicability to handle variable-length sequences, it can handle sequential data of varying length. RNNs share the same weights throughout a number of time steps, in contrast to FNNs, which have different weights for every input feature. In the figure below we have shown an unfold diagram of RNN. Which explains the proper working mechanism of a simple RNN throughout the time steps.

By gradually working on the data and updating hidden states for all timesteps here X is input and Y output. The network as a whole uses the same parameters U, V, and W. Here W stands for the weight, V creates the connection between the hidden and output layer, U controls the connection from the input layer to hidden layer where U is a weight parameter. By keeping the knowledge from the previous input in its current hidden state, this parameter sharing enables the RNN to handle sequential data more quickly and accurately while capturing temporal relationships.

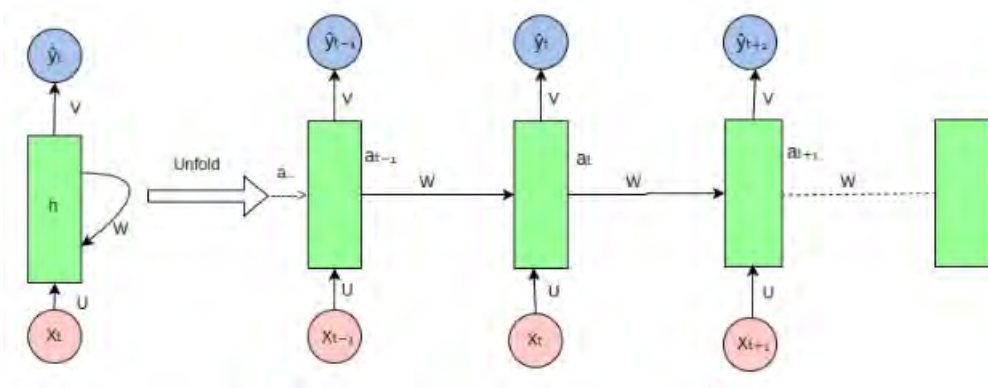


Figure 3.6: RNN Unfolded architecture. (Poudel, 2023) [30]

$a_t$  is the hidden state that gets calculated in every time step “t” by the given formula, which takes into account the model parameters :

$$a_t = f(a_{t-1}, x_t; \theta) \quad (1) \quad (3.1)$$

$$a_t = f(U \cdot X_t + W \cdot a_{t-1} + b) \quad (3.2)$$

The output of hidden layer is represented by  $a_t$ , and the input is  $x_t$ . A set of learnable parameters (weights and biases) is represented by  $\theta$ . The weight matrix governing the connections between the hidden layer and its input is represented by  $U$ ; the weight governing the connections between the hidden layer and itself (recurrent connections) is represented by  $W$ .

Formula to calculate output in time step :

$$\hat{y}_t = f(a_t; \theta) \quad - (2) \quad (3.3)$$

$$\hat{y}_t = f(V \cdot a_t + c) \quad (3.4)$$

Due to the sequential nature of forward propagation, parallelism is unable to minimize the runtime, making gradient computing a costly process. Prior to being utilized again in the back-propagation, the states calculated in the forward pass are saved. It is called back-propagation through time (BPTT) when the back-propagation method is used with RNN [9].

To determine output and loss during forward propagation, RNN performs the following computing processes.

$$a_t = U \cdot X_t + W \cdot a_{t-1} + b \quad (3.5)$$

$$a_t = \tanh(a_t) \quad (3.6)$$

$$\hat{y}_t = \text{softmax}(V \cdot a_t + c) \quad (3.7)$$

Following sequence processing, RNN produces a series of expected outputs,  $\hat{y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_t]$ . Next, the loss is calculated by comparing the actual target output ( $y$ ) with what was expected ( $\hat{y}$ ) and loss function is

$$L(y, \hat{y}) = \frac{1}{t} \sum (y_t - \hat{y}_t)^2 \quad (3.8)$$

After completing this step of forward propagation the model needs to work on backward pass.

### 3.7 Long Short Term Memory networks (LSTMs):

RNNs may theoretically handle long-term dependencies with no difficulty but these kinds of toy issues could be solved by a human carefully choosing settings though it appears that RNNs are unable to learn them in practice and it could be challenging [4]. That's why Long Short Term Memory networks were introduced which are capable of learning long-term dependencies which was introduced by Hochreiter & Schmidhuber (1997) [5].

One limitation that Recurrent Neural Networks (RNNs) encounter is that their backward information is usually limited to about ten timesteps, as has been noted in earlier studies [3]. This constraint results from the problem of the feedback signal blowing up or disappearing. LSTM has overcome this problem. A certain level of plausibility may be seen in LSTM networks [4], and depending on the complexity of the network's construction, they can learn across longer temporal sequences that go beyond 1,000 timesteps [7].

Moreover, the main drawbacks of recurrent neural networks are handling long range dependencies and the vanishing gradient problem. That's why LSTMs were introduced which stores necessary state information. For each time step using memory and drops the information which are irrelevant. By using this memory cell and gating mechanisms LSTMs selectively store and retrieve information over large periods which overcomes the long range dependencies problem of RNNs. LSTMs are a gradient-based technique that tackles the vanishing error problem [15].

LSTMs apply the idea of gates to simplify and effectively calculate operations using both long- and short-term memory. If long-term memory enters the forget gate then useless data is forgotten. Learn Gate is used when short-term memory

Remember Gate is used as an updated long-term memory by combining short-term memory. Utilize gate works as the updated short-term memory that predicts output for current events by using the long-term memory, short-term memory and Event. These are the main gates used in the LSTMs which are the feedforward neural networks. There are two layers in every neural network: the input layer and the output layer. All of the output neurons in each of these neural networks are connected to



the input neurons. Consequently, there are four fully connected layers in the LSTMs unit [10].

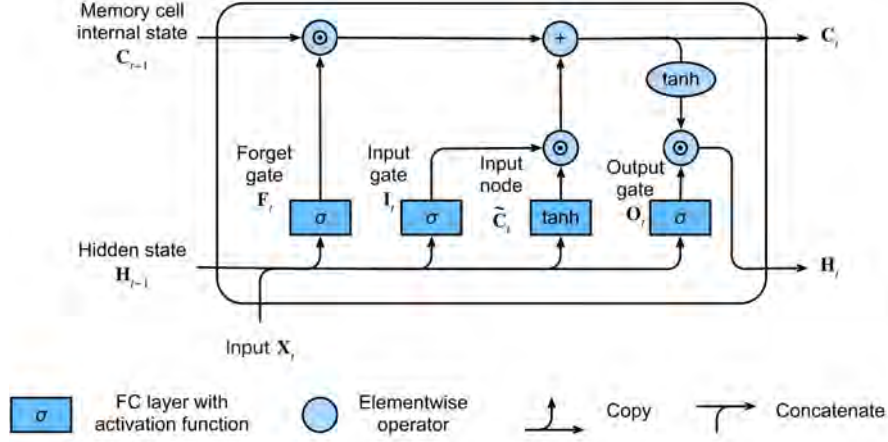


Figure 3.7: Architecture of a LSTM Unit [31]

LSTM architecture's gates equations:

$$I_t = \sigma(w_i[H_{t-1}, X_t] + b_i) \quad \text{— Represents the input gate.} \quad (3.9)$$

$$F_t = \sigma(w_f[H_{t-1}, X_t] + b_f) \quad \text{— Represents the forget gate.} \quad (3.10)$$

$$O_t = \sigma(w_o[H_{t-1}, X_t] + b_o) \quad \text{— Represents the output gate.} \quad (3.11)$$

Where:

$I_t$  — represents the input gate.

$F_t$  — represents the forget gate.

$o_t$  — represents output gate.

$\sigma$  — Represents the sigmoid function.

$[H_{t-1}, X_t]$  — Concatenation of  $H_{t-1}$  and  $X_t$ .

$w_x$  — Weight for the respective gate ( $x$ ).

$H_{t-1}$  — Output of the previous LSTM block (at timestamp  $t - 1$ ).

$X_t$  — Input at the current timestamp.

$b_x$  — Biases for the respective gates ( $x$ ).

Equations of cell state, candidate cell state, and final output:

$$\tilde{C}_t = \tanh(w_c[H_{t-1}, X_t] + b_c) \quad (3.12)$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot \tilde{C}_t \quad (3.13)$$

$$H_t = O_t \cdot \tanh(C_t) \quad (3.14)$$

Three vectors are fed into an LSTM unit. At the previous instant ( $t - 1$ ), the LSTM created two vectors that originate from the LSTM itself. The hidden state (H) and the cell state (C) are these. The external source is the third vector. This is the vector X that the LSTM received at moment t, also referred to as the input vector.

The cell state and hidden state vectors are influenced by the internal information flow that the LSTM controls via gates among the input vectors ( $C$ ,  $H$ , and  $X$ ). With vectors for the subsequent time step ( $t+1$ ) being impacted by this process, this regulation guarantees that the cell state operates as long-term memory and the concealed state as short-term memory.

The LSTM unit updates long-term memory (cell state,  $C$ ) by combining recent past knowledge (short-term memory,  $H$ ) with new external input (input vector,  $X$ ). The short-term memory (hidden state,  $H$ ), which also acts as the LSTM's output for a particular task signifying its evaluated performance, is then updated by the long-term memory. This is how the LSTMs work to evaluate performance.

### 3.8 SEQ2SEQ model

A seq2seq model consists of 2 major portions which are encoder and decoder. Together, these parts process input sequences and produce equivalent output sequences. The input sequence is sent into the encoder which then compresses it into a fixed-size vector known as the thought vector or context vector. This vector acts as the decoder's starting point and contains the important data from the input sequence. Next, by making one token prediction at a time, the decoder creates the output sequence.

In our model the encoder and decoder is developed by the LSTM model. The input Genome sequence is encoded and output in fixed length genome sequences. Then finally the output sequence is predicted by the decoder one nucleotide at a time, for each output time step obtained from the encoder output genome sequences.

First we will examine the working mechanism of the encoder layer. Each nucleotide in the genome sequence in the input sequence for the encoder was encoded as one-hot vectors which have the length of 5 as we have 'G', 'A', 'C', 'T', ' ' unique characters in the input sequence. Then we pass these one-hot-encoded sequences in the encoder LSTM model which has the 1000 units that refers to the dimensionality of the output space of our encoder LSTM model. The encoder LSTM returns output sequences from the encoder LSTM and it also returns the hidden states and cell state of all the cells in the encoder layer. We keep the final hidden and cell states output from the encoder model which we will pass to the decoder LSTM model in the next step.

In the decoder part we have used another LSTM model using the teacher forcing mechanism. The decoder input is the next mutated genome sequence's nucleotides one-hot encoded to binary vectors which has a length of 7. Also the decoder LSTM uses the final hidden state and cell state vectors of the encoder as the initial state for initializing the state of the decoder. This is how the decoder learns to generate the targets[ $t+1 \dots$ ] given the targets[ $\dots t$ ], depending on the order of input. Therefore, the final internal states of the encoder model are utilized as the starting point to generate the first character in the output sequence each time it encodes an input sequence. For this the number of cells in the encoder LSTM and decoder LSTM

layers must be equal.

At last, The decoder only passes the sequence of hidden states to the last dense layer that is used as the output layer to predict each nucleotide. In the dense layer we used the activation function softmax.

In our seq2seq model we have used the rmsprop as optimizer and for the loss function the categorical cross entropy loss has been used as we treated each nucleotide in the genome sequences as a separate class using one-hot-encoding.

```

Model: "model"
-----
Layer (type)                Output Shape                Param #   Connected to
-----
input_1 (InputLayer)        [(None, None, 5)]          0         []
input_2 (InputLayer)        [(None, None, 7)]          0         []
lstm (LSTM)                  [(None, 1000),             4024000  ['input_1[0][0]']
                        (None, 1000),
                        (None, 1000)]
lstm_1 (LSTM)                [(None, None, 1000),       4032000  ['input_2[0][0]',
                        (None, 1000),                'lstm[0][1]',
                        (None, 1000)]                'lstm[0][2]']
dense (Dense)                (None, None, 7)            7007     ['lstm_1[0][0]']
-----
Total params: 8063007 (30.76 MB)
Trainable params: 8063007 (30.76 MB)
Non-trainable params: 0 (0.00 Byte)

```

Figure 3.8: Architecture of the seq2seq model

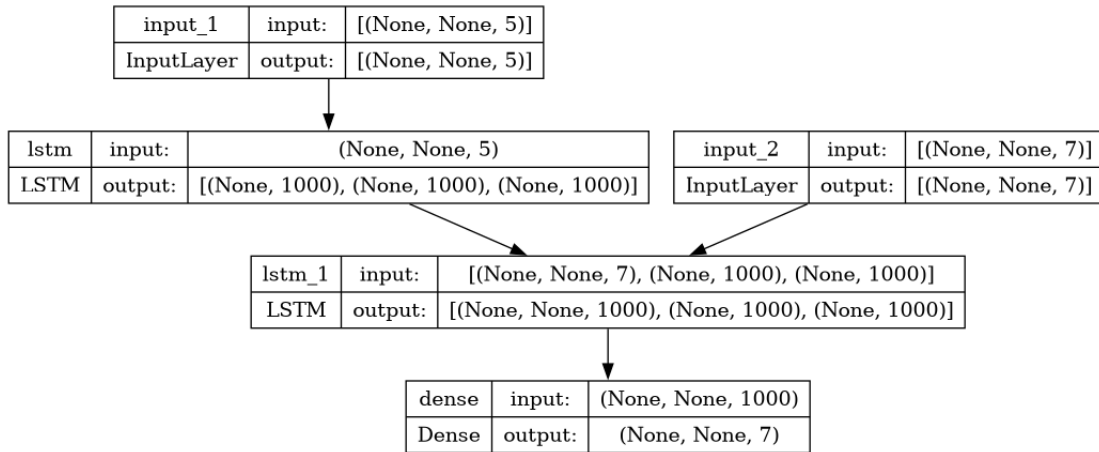


Figure 3.9: Seq2seq model architecture in our problem

### 3.9 Inference Model to generate mutated genome sequence

After training the model on the dataset it can be used for predicting mutations for future variants by giving the previous variants. However, the mentioned model cannot be used directly for recursive generation of individual nucleotides in the full genome sequence because it doesn't have the structure to generate one nucleotide at a time. So to resolve this issue a new model is needed for the prediction step of recursive generation of individual nucleotides.

That's why we will create an inference model and generate the prediction for future mutation genome sequences. Firstly, the encoder model takes input from the trained model and then it outputs the hidden and cell state tensors. Then we will develop the decoder model. For every nucleotide to be generated in the mutated sequence the encoder and decoder will be called repeatedly. On the initial call the hidden and cell states from the encoder will be used to construct the decoder LSTM layer produced as input to the model directly. The last hidden and cell state must be sent to the model on subsequent recursive calls to the decoder. Although these state values are already in the decoder because of the way the model was designed, we have to re-initialize the state for each call in order to obtain the final states from the encoder on the first call. In order to assign the hidden and cell states to a variable and use them on each future recursive call for the mutation generation of a particular input sequence the decoder must provide the hidden and cell states along with the predicted nucleotide on each call.

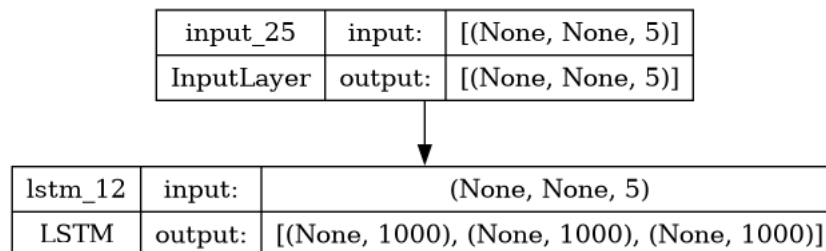


Figure 3.10: Encoder model architecture for inference model

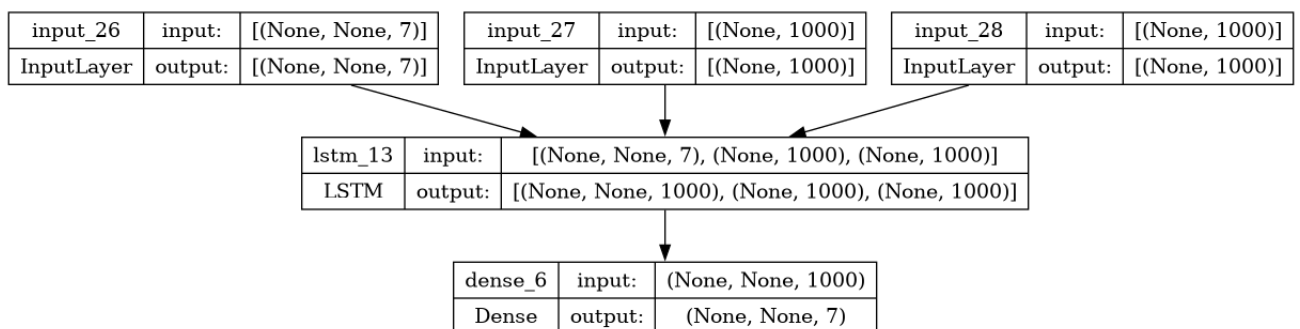


Figure 3.11: Decoder model architecture for inference model.

# Chapter 4

## Result and Discussion

After preprocessing, the dataset has been splitted into train, validation and test set and then we trained the model on the train dataset and used the validation dataset for validation. The metrics for evaluating the validation set is accuracy. As we have one-hot-encoded each nucleotide in the genome sequence which treats each position in the genome sequence as an unique class. So to calculate the accuracy of the model we will compare each position of the genome sequence with the predicted genome sequence.

However, we don't train the whole genome sequence at a time. Each genome sequence has been sliced in length of 300 and then we pass the genome sequence to train and validate. Secondly, we have loaded the model and then used the inference model to generate a genome sequence on test data and after it gets the prediction on test data we calculate the accuracy of the test dataset.

In this way we get accuracy for each 300 length of genome sequence separately. To calculate the overall validation accuracy we have taken the average of the best validation accuracy for each 300 length which gives 96.42% accuracy. At last 96.51% accuracy was achieved on average on the test dataset. We used 2000 epochs for both training and testing.

### 4.1 Validation and train graph

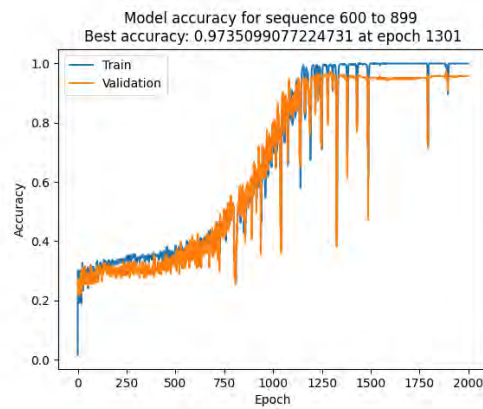
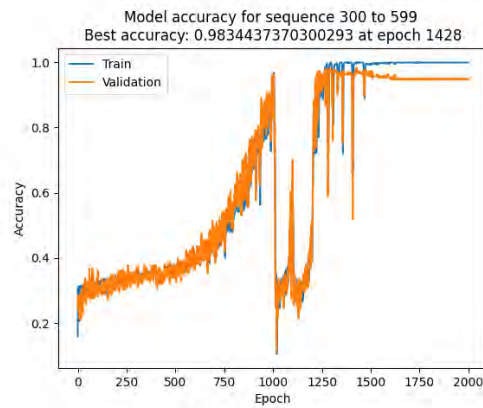
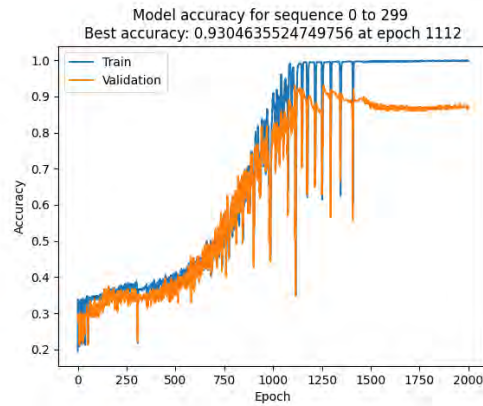
#### 4.1.1 Train and Validation accuracy graph

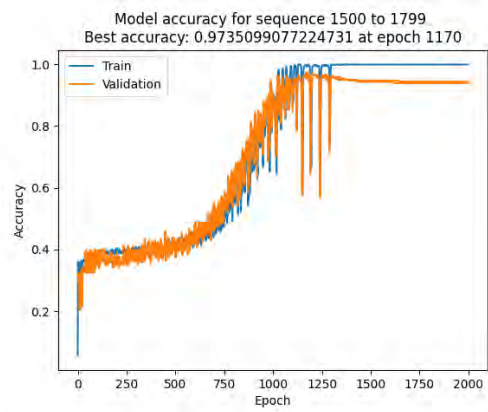
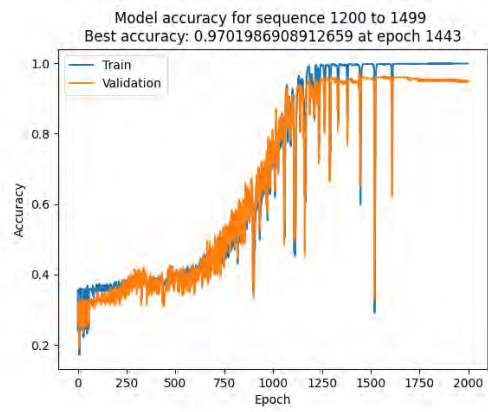
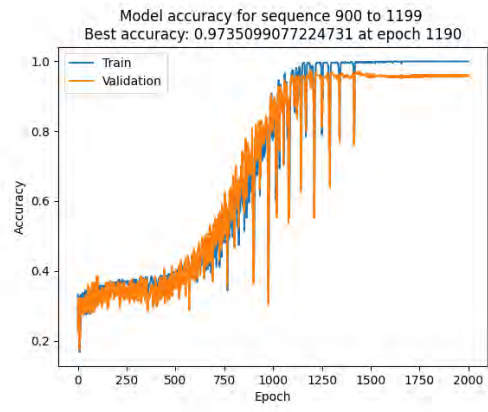
As our train and validation dataset has a total 101 number of slices for that we have shown the accuracy graph here for only 4 slices and the other train and validation accuracy graph has been given in the appendix.

We can observe the accuracy for the validation dataset is admirable. The model has been able to learn the important patterns for mutation that's why it's being able to predict the future mutated genome sequences with high accuracy.

However, we can see there are some high accuracy drops in the middle of the graph. After this accuracy drop the model again overcame the accuracy drop problem and gained satisfactory accuracy again. These drops might cause the learning rate value. Sometimes the high learning rate causes the model to overshoot the optimal weights

when getting trained. For that reason there are sometimes sudden drops in the accuracy. To resolve it more focus can be given to the hyperparameter tuning.





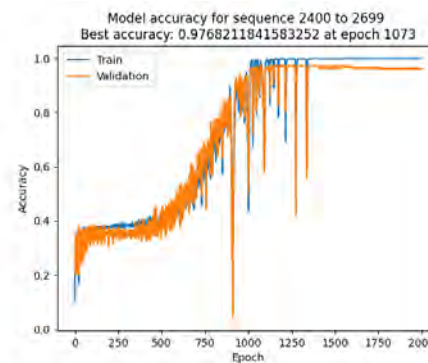
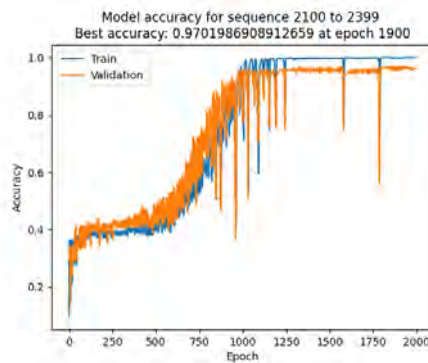
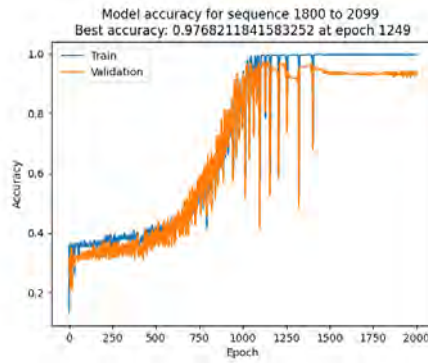


Figure 4.1: Few graphs of train and validation accuracy. The given graphs are for the genome sequence 0 to 2699

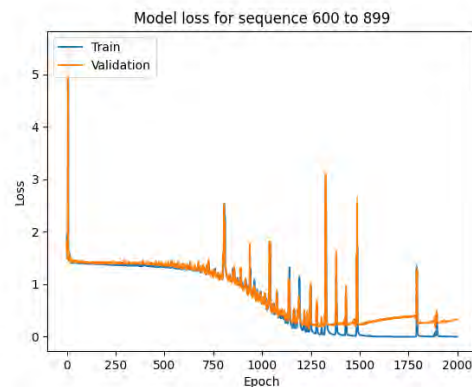
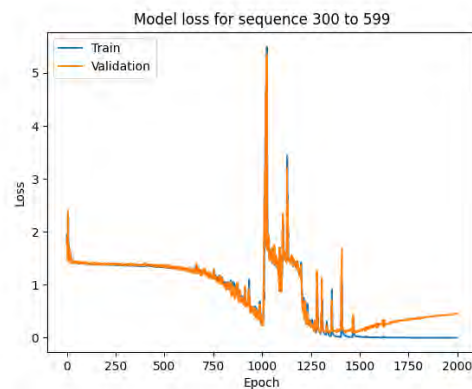
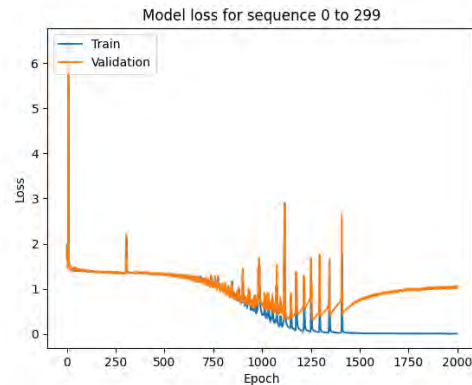
### 4.1.2 Train and Validation loss graph

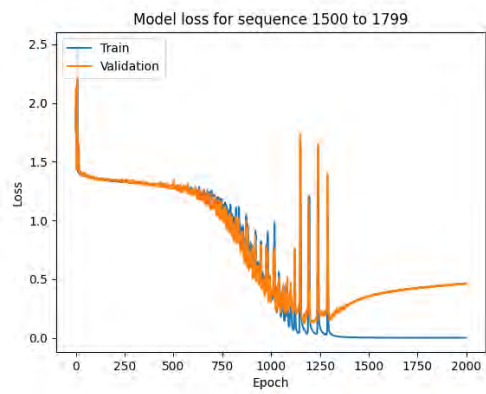
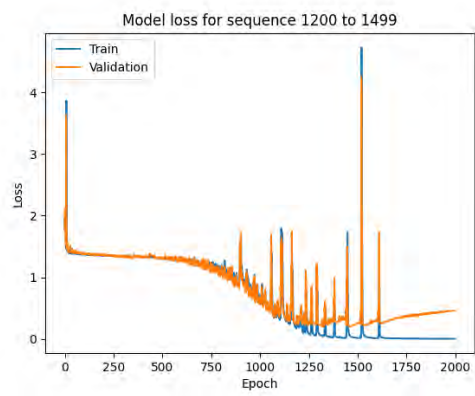
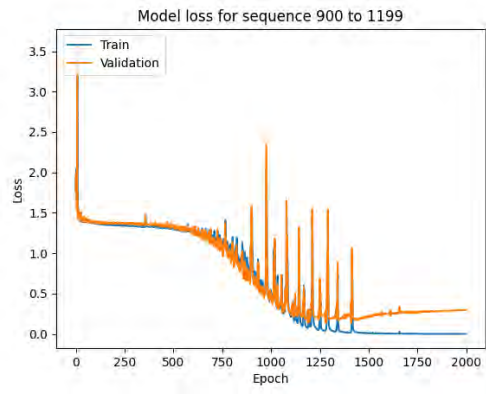
As our train and validation dataset has a total 101 number of slices for that we have shown the loss graph here for only 4 slices and the other train and validation loss graph has been given in the appendix.

If we look at the validation and train loss curve we can observe clearly that the model fits very well. It only gets the overfitting problem after 1250 epochs on average. But before that the model fits very well and gives satisfactory loss value along with admirable accuracy value.



However, in the validation and train loss graph we can see there are some sudden uprisings or spikes in the validation and training loss graph. This occurs for various reasons. Sometimes it causes a high learning rate. Often it occurs for vanishing gradients. If gradients vanish during the backpropagation, sudden spikes occur in the loss curve. By gradient clipping this issue can be solved. Moreover the optimizer can also cause these spikes as we used rmsprop optimizer in our model it might be the reason for occurrence of it. By finetuning more the model might resolve this problem.





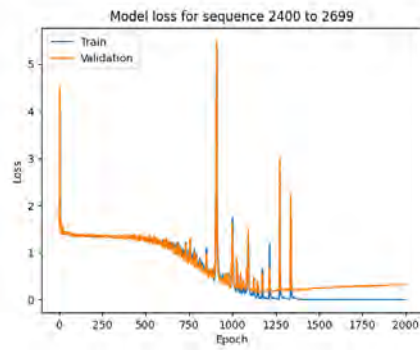
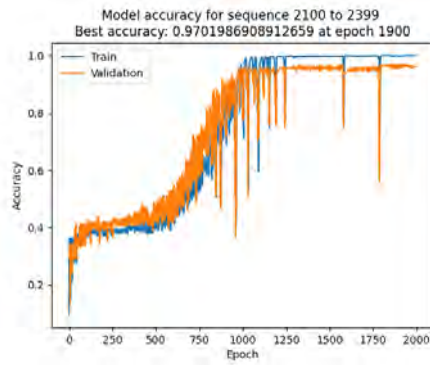
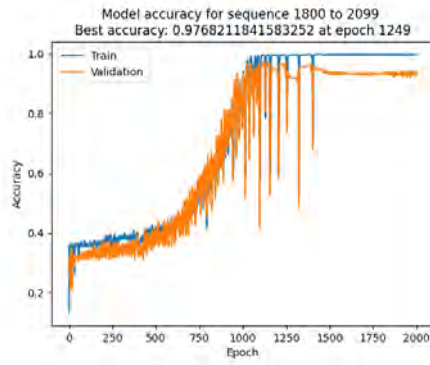


Figure 4.2: Few graphs of train and validation loss. The given graphs are for the genome sequence 0 to 2699

# Chapter 5

## Conclusion

Using a neural network to predict coronavirus mutation from genomic sequence is a difficult but effective method. The rapid evolution of the virus, the ongoing appearance of new strains, and the requirement for a thorough understanding of mutation patterns all provide challenges. Additionally, the complexity lies in efficiently encoding the genetic information, training the model to identify evolving patterns, and addressing the dynamic nature of viral mutations over time. Despite these difficulties, using Neural Networks is a promising way to learn more about possible viral variations and improve our capacity to adapt to its ever-changing nature. In our seq2seq model approach, it processes genome sequences using an encoder and decoder. It encodes nucleotides into one-hot vectors, passes them through an encoder LSTM, and utilizes the final states for decoding in a LSTM-based decoder. The model predicts mutated genome sequences, maintaining equal LSTM cells in both encoder and decoder layers. The output is generated through softmax activation, and training involves rmsprop optimization and categorical cross-entropy loss with one-hot encoding for nucleotides. The robustness of our results is grounded in the elevated validation accuracy percentage. This signifies the effectiveness of our model in accurately predicting the subsequent mutations by discerning positional changes within the genome sequence.

# Bibliography

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [2] J. A. Bradshaw, K. J. Carden, and D. Riordan, “Ecological applications using a novel expert system shell,” *Bioinformatics*, vol. 7, no. 1, pp. 79–83, 1991.
- [3] M. C. Mozer, “Induction of multiscale temporal structure,” *Advances in neural information processing systems*, vol. 4, 1991.
- [4] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. DOI: 10.1109/72.279181.
- [5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.
- [6] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.
- [7] R. C. O’Reilly and M. J. Frank, “Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia,” *Neural computation*, vol. 18, no. 2, pp. 283–328, 2006.
- [8] M. H. Sazli, “A brief review of feed-forward neural networks,” *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006.
- [9] M. C. Mozer, “A focused backpropagation algorithm for temporal pattern recognition,” in *Backpropagation*, Psychology Press, 2013, pp. 137–169.
- [10] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” *arXiv preprint arXiv:1402.1128*, 2014.
- [11] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.
- [12] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.
- [13] M. A. Salama, A. E. Hassanien, and A. Mostafa, “The prediction of virus mutation using neural networks and rough set techniques,” *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2016, pp. 1–11, 2016.
- [14] M. Islam, G. Chen, and S. Jin, “An overview of neural network,” *American Journal of Neural Networks and Applications*, vol. 5, no. 1, pp. 7–11, 2019.

- [15] R. C. Staudemeyer and E. R. Morris, “Understanding lstm—a tutorial into long short-term memory recurrent neural networks,” *arXiv preprint arXiv:1909.09586*, 2019.
- [16] I. Saha, N. Ghosh, D. Maity, N. Sharma, J. P. Sarkar, and K. Mitra, “Genome-wide analysis of indian sars-cov-2 genomes for the identification of genetic mutation and snp,” *Infection, Genetics and Evolution*, vol. 85, p. 104457, 2020.
- [17] A. Yang, W. Zhang, J. Wang, K. Yang, Y. Han, and L. Zhang, “Review on the application of machine learning algorithms in the sequence data mining of dna,” *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 1032, 2020.
- [18] M. S. Nawaz, P. Fournier-Viger, A. Shojaee, and H. Fujita, “Using artificial intelligence techniques for covid-19 genome analysis,” *Applied Intelligence*, vol. 51, pp. 3086–3103, 2021.
- [19] B. Wang and L. Jiang, “Principal component analysis applications in covid-19 genome sequence studies,” *Cognitive computation*, pp. 1–12, 2021.
- [20] European Centre for Disease Prevention and Control/World Health Organization Regional Office for Europe, *Methods for the detection and identification of SARS-CoV-2 variants: second update, August 2022*. Stockholm and Copenhagen: ECDC and WHO European Region, 2022.
- [21] F. Obermeyer, M. Jankowiak, N. Barkas, *et al.*, “Analysis of 6.4 million sars-cov-2 genomes identifies mutations associated with fitness,” *Science*, vol. 376, no. 6599, pp. 1327–1332, 2022.
- [22] P. Pushkar, C. Ananth, P. Nagrath, J. F. Al-Amri, D. A. Nayyar, *et al.*, “Mutation prediction for coronaviruses using genome sequence and recurrent neural networks,” 2022.
- [23] B. Saldivar-Espinoza, G. Macip, P. Garcia-Segura, *et al.*, “Prediction of recurrent mutations in sars-cov-2 using artificial neural networks,” *International Journal of Molecular Sciences*, vol. 23, no. 23, p. 14683, 2022.
- [24] S. Tasnim, K. H. Talukder, and A. Asfi, “Next mutation prediction of sars-cov-2 spike protein sequence using encoder-decoder based long short term memory (lstm) method,” *Khulna University Studies*, pp. 803–816, 2022.
- [25] A. Ullah, K. M. Malik, A. K. J. Saudagar, *et al.*, “Covid-19 genome sequence analysis for new variant prediction and generation,” *Mathematics*, vol. 10, no. 22, p. 4267, 2022.
- [26] D. S. Berman, C. Howser, T. Mehoke, A. W. Ernlund, and J. D. Evans, “Mutagan: A seq2seq gan framework to predict mutations of evolving protein populations,” *Virus Evolution*, vead022, 2023.
- [27] S. Das, A. Tariq, T. Santos, S. S. Kantareddy, and I. Banerjee, “Recurrent neural networks (rnns): Architectures, training tricks, and introduction to influential research,” *Machine Learning for Brain Disorders*, pp. 117–138, 2023.
- [28] V. Kakulapati, S. M. Reddy, S. S. D. Bhugubanda, and S. Naini, “Predictive analytics of genetic variation in the covid-19 genome sequence: A data science perspective,” in *Data Science for Genomics*, Elsevier, 2023, pp. 229–247.

- [29] M. S. Nawaz, P. Fournier-Viger, M. Aslam, W. Li, Y. He, and X. Niu, “Using alignment-free and pattern mining methods for sars-cov-2 genome analysis,” *Applied Intelligence*, pp. 1–24, 2023.
- [30] A. Poudel. “Rnn unfolded.” Medium. (2023), [Online]. Available: <https://medium.com/@poudelsushmita878/recurrent-neural-network-rnn-architecture-explained-1d69560541ef>.
- [31] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into deep learning*. Cambridge University Press, 2023.
- [32] A. UMAR, S. BHATTI, A. F. MEGHJI, N. A. MAHOTO, and S. KUMARI, “Analysis of frequent nucleotide patterns in covid-19 genome sequences using spm algorithms,”

# Google Drive link of all the Result Graphs

This Google Drive link contains all the validation accuracy and validation loss graphs generated by our model during the validation of our work:

[https://drive.google.com/drive/folders/1VequT7UES\\_955CdwsYgIsmRtOp8DTc\\_u?usp=drive\\_link](https://drive.google.com/drive/folders/1VequT7UES_955CdwsYgIsmRtOp8DTc_u?usp=drive_link)