Gene Mutations and Motifs Detection for Coronavirus in Biological Sequences of COVID-19 using Deep Learning Models

Submitted in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

Submitted by

Praveen Gugulothu

(Roll No. 719134)

Under the guidance of

Dr. Raju Bhukya

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA
October 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

THESIS APPROVAL FOR Ph.D.

This is to certify that the thesis entitled, Gene Mutations and Motifs Detection for Coronavirus in Biological Sequences of COVID-19 using Deep Learning Models, submitted by Mr. Praveen Gugulothu [Roll No. 719134] is approved for the degree of DOCTOR OF PHILOSOPHY at National Institute of Technology Warangal.

Examiner

Research Supervisor
Dr. Raju Bhukya
Dept. of Computer Science and Engg.
NIT Warangal
India

Chairman
Prof.Ch.Sudhakar
Dept. of Computer Science and Engg.
NIT Warangal
India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

CERTIFICATE

This is to certify that the thesis entitled, Gene Mutations and Motifs Detection for Coronavirus in Biological Sequences of COVID-19 using Deep Learning Models, submitted in partial fulfillment of requirement for the award of degree of DOCTOR OF PHILOSOPHY to National Institute of Technology Warangal, is a bonafide research work done by Mr. Praveen Gugulothu (Roll No. 719134) under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

Research Supervisor
Dr. Raju Bhukya
Associate Professor
Dept. of CSE
NIT Warangal
India

Place: NIT Warangal
Date: 28 October, 2024

DECLARATION

This is to certify that the work presented in the thesis entitled "*Gene Mutations and Motifs detection for Coronavirus in Biological Sequences of COVID-19 using Deep Learning Models*" is a bonafide work done by me under the supervision of Dr.Raju Bhukya and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Praveen Gugulothu
(Roll No. 719134)
Date: 28-10-2024

ACKNOWLEDGMENTS

It is with great pleasure that I acknowledge my sincere thanks and deep sense of gratitude to my supervisor Dr.Raju Bhukya for his invaluable guidance to complete the work. He always gave me ample time for discussions, reviewing my work and suggesting requisite corrections, which enabled me to attain my objective in time. He has provided me all kinds of inputs directly with his words, indirectly with his values not only to be a good teacher also to be a good human being. His sincerity and commitment to every aspect of the life has influenced me greatly. I want to inculcate all the great qualities of him for the rest of my life.

I extend my gratitude to all my Doctoral Scrutiny Committee members Prof.Ch.Sudhakar, Dr.U.S.N.Raju, Dr.E.Suresh Babu, and Dr.J.Praneetha for their insightful comments and suggestions during oral presentations.

I wish to express my thanks to the Head of the department Prof.R.Padmavathy Madam and all faculty members of the Computer Science and Engineering department for their continuous support and encouragement. I also express my thanks to Prof.Bidyadhar Subudhi, Director, NIT Warangal for his official support and encouragement.

I am eternally thankful to all faculty member of the department who have helped me directly and indirectly.

I would like to express my gratitude to my parents for their invaluable sacrifice for my education and trust they have, My gratitude to my family for their unconditional love, support and prayers for my success in achieving the goal. I am very much thankful to my family for all the support during the time I carried out my thesis work.

My mother and father have made so many sacrifices in their life by having a deep trust in me to see where I am today. Even in the most difficult financial situations at home, they backed me in pursuing higher studies and further in choosing teaching as a profession. Lastly, my deep sense of gratitude to my god for all the things that come into my life.

Praveen Gugulothu

Dedicated to

My Family & Teachers

ABSTRACT

In bioinformatics and computational biology, DNA Genome sequence analysis covers a broad range of research issues, such as identifying homology between sequences, recognition of intrinsic features, mutation detection, genetic diversity disclosure, and species evolution. Sophisticated sequencing technologies produce enormous DNA sequence data, thereby raising the difficulty of analysing sequences as well. The growth of genomic data is much faster compared to the sequence analysis rate. So, there is an enormous need for faster sequence analysis algorithms. Analysis of genome sequences is useful in disease detection, drug development, agriculture and forensics. Our solution to this problem is a Convolutional Neural Network (CNN) that can handle huge DNA sequences using Covid-19 feature extraction.

Given the fast spread of the disease, one of the world's primary concerns is detecting coronavirus disease 2019 (COVID-19). There have been over 1.6 million confirmed instances of COVID-19, and the disease is rapidly spreading to numerous nations throughout the world, according to recent figures. An analysis of the global incidence and distribution of COVID-19 is presented. We introduce a deep convolutional neural network (CNN) that can distinguish between the original (non-augmented) dataset and the augmented dataset that were both utilised for the assessment. A variety of COVID-19 datasets, including those for MERS-CoV, SARS-CoV, NL63, Alpha-CoV, BetaCoV-1, HKU1-CoV, and 229E-CoV, have been compiled by us from NCBI and GISAD. Each dataset is annotated with its accession number and contains nucleotides in FASTA format. In this study, we compiled a positive and negative dataset consisting of 1582 samples with varying genome sequence lengths. By using one-hot encoding, every categorical variable is transformed into its own feature with a binary value of either 1 or 0. Thus, in one-hot encoding, every nucleotide is represented by a four-dimensional one-hot vector; for example, the letters "A," "C," "G," and "T" are encoded as (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0),, and (0, 0, 0, 0), respectively. Using the top ten most sick coronavirus sequences as a guide, we trained the suggested CNN module to detect underlying patterns associated with the virus. Learned convolutional filters produce motif. The activation values for the 20th filter's entire sub-sequences are less

than 0.047075363 and close to 1.086 is the highest activation value is obtained.

Advanced Deep Learning Method for COVID-19 Point Mutation Rate Optimisation by Coot-Lion Preventing disease or tailoring treatment to an individual's needs both depend on an accurate diagnosis. Unfortunately, the processing time is greatly impacted by the enormous quantity of sequences, even though DNA sequence illness detection is safe. Consequently, computational approaches are suggested to enhance diagnostic precision and expedite the diagnostic procedure. Genetic disorders occur when an organism's DNA becomes aberrant as a result of mutations in exons. Our new Deep Quantum Neural Network (DQNN) called LBCA-based Deep QNN is built on the Lion-based Coot algorithm. It can forecast the COVID-19 virus using the DNA biological sequence pattern and the rates of point mutations. In this step, the genome sequences undergo feature extraction. This process extracts specific features from the genome sequences, such as CpG-based features and numerical mapping for integer and binary data. Additionally, numerical mapping is applied using the Fourier transform to generate features for skewness, kurtosis, and peak to average power ratio. To get the entropy feature, we also use K-mer extraction. We determined the K-group for point mutations in COVID-19 for both the 200- and 400-genome sequence learning sets, respectively.

Afterwards, we also focused on COVID-19 DNA sequence repeats for bi-character and tri-character types, among others, and put forward a DNA sequence clustering model called "ERSIT-GRU" (Exponential Robust Scaling-Identity Tanh-Gated Recurrent Unit) to detect COVID-19 DNA sequence repeats in large datasets. In order to address these challenges, such as the fact that the dataset is tiny, imbalanced, and has fasta quality issues, the dataset has been preprocessed in stages using multiple techniques in order to provide a useful training dataset. Consequently, computational approaches are suggested to enhance diagnostic precision and expedite the diagnostic procedure.Genetic disorders manifest in organisms when there is an aberration in their genetic composition as a result of exon mutations. The technique that uses the Trie data structure to forecast disease severity by counting the occurrences of repeat patterns in exons. Due to the tiny database, the suggested method can only forecast the condition of a small number of diseases, despite its effectiveness and speed in doing so based on pattern frequency. There is an immediate need to discover other patterns

that produce varied diseases in order to solve the problem of a small number of pathogenic patterns.

There is data in the genetic code that affects how fast and efficient translation is. In this extensive study of coronaviruses (CoVs) of both human and zoonotic origin, we compare and contrast their codon usage bias, relative errors in insertion and substitution, mutation rates in COVID-19, DNA motif sequence, size, feature extraction based on base frequency, dimer count, and feature extraction based on size. The evolutionary relationship between seven coronaviruses can be shown by the model Harris Hawks Optimisation (HHO) analysis, which we have presented. There have been many attempts to fix DNA-based errors using tandem repeats. Depending upon Age, symptoms, and chromosomes all have a role in the different patterns that correlate to normal, pre-mutated, and diseased frequencies. Tandem has identified the ATXN2, DMPK, ATN1, and JPH3 genes, among others, that are involved with disease state. The pattern frequency allows us to predict the disease's progress and treat it at an early stage. Proposed model reached highest Accuracy in terms of the various Parameters like Accuracy, Precision, Recall, F1 Score. The pattern frequency allows us to predict the disease's progress and treat it at an early stage.

Keywords: Repeats,point mutations, Tandem Repeat, Interpretable, Convolution neural network, Motif, Learned filters, Heatmap, Feature activation, Exons, Genes, Disease prediction, Pre-mutated, Mutations, ATXN2, Explainable, Coronavirus, 2019-nCoV, COVID-19, RSCU, HHO.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
U	Uracil
bps	base pairs
mRNA	messenger RNA
MR	Mutation rate
TM	total mutation t
ENC	effective number of codons
DQNN	Deep quantum neural network
tRNA	transfer RNA
RBP	RNA binding proteins
CAI	Codon adaptation index
ENC	Effective number of codons
RSCU	Relative synonymous codon usage
NMI	Normalized mutual information
CDs	Coding Sequences

CoVs	Coronaviruses
LBCA	Lion-based Coot algorithm
SVM	Support vector machines
HMM	Hidden markov model
RF	Random forest
ANN	Artificial neural networks
Sn	Sensitivity
Sp	Specificity
E-NJA	Entropy Neighbor-Joining Algorithm
AUC-ROC	Area under receiver operator characteristic curve
AUC-PR	Area under precision recall curve
Acc	Accuracy
DC	Dimer Count
BC	Base Count
ERSIT-GRU	Exponential Robust Scaling-Identity Tanh- Gated Recurrent Unit
FS	Fisher Score
CNN	Convolutional neural networks
RNN	Recurrent neural networks
DNN	Deep neural networks
LSTM	Long short term memory
NWJ-K-Means	Needleman-Wunsch Jaccard K-Means
SW	Sliding Window
HHO	Harris Hawks Optimizer
PWM	Probability Weight Matrix

Chapter 1

Introduction

Bioinformatics uses computer science, mathematics, statistics, and informatics to tackle biological problems, particularly those involving DNA, RNA, and protein sequences. It has recently emerged as a top multidisciplinary discipline. A dramatic uptick in the production of DNA sequences occurred with the introduction of new sequencing technology. Genomic data is expanding at a considerably quicker rate than it is being analysed. The cost of storing, processing, analysing, and transmitting the massive amounts of DNA sequence data is starting to become an obstacle. Numerous fields make use of DNA sequence analysis, including medicine, forensics, agriculture, and many more. Its primary use in forensics is in establishing paternity of a child and in identifying offenders using biospecimens. It finds utility in gene therapy, where it can replace defective genes with healthy ones, and in the medical field, where it may detect genes linked to certain hereditary or acquired disorders. The agricultural sector has made good use of DNA sequence analysis to create food crops and plants, as well as cattle with higher quality milk and meat.

Numerous areas of study encompass sequence analysis in bioinformatics and computational biology, including but not limited to: discovering mutations, revealing genetic diversity, uncovering inherent traits, discovering similarity between sequences, and studying species evolution. Clustering is one method that can be used to achieve sequence homology. It lays bare the underlying biological processes that give rise to species diversity as well as correlations, patterns, and hints to events in the past. In order to identify genes, their coding regions, and the roles they play in the body, DNA sequences must be annotated.

1.1 Preliminaries

Here, we present few definitions and technicalities of Deep learning, Machine learning and Harris Hawks Optimization that will be used throughout the thesis.

1.1.1 Biological Background

A cell is the fundamental building block of every living thing. Some creatures, like humans, have a complex network of cells, while others, like bacteria, have just one cell, and viruses, much less so, do not have any cells at all. Many organelles, including cytoplasm, nucleus, mitochondria, and others, come together to form cells. The human microbiome consists of 40 trillion bacteria and 37.2 trillion cells. In prokaryotes, the nucleus is not well developed, but in eukaryotic cells, membranes separate the nucleus and other organelles from the cell. Genomes, which comprise an organism's whole collection of DNA, are essential for the development and survival of all living things. Coding and non-coding DNA, as well as DNA from chloroplasts and mitochondria, make up the genome. An estimated 3.2 billion base pairs make up the human genome, which is comprised of 23 chromosomal pairs and contains an average of 20,000–25,000 genes [1]. Nucleic acid molecules (Nucleotides) made comprised of nitrogen bases, phosphate groups, and sugar molecules make up DNA and RNA. DNA is a double helix that is produced when nucleotides (A and G) pair with each other. A DNA strand is composed of a series of nucleotides. Both the paternal and maternal lines contribute to the two copies of DNA that are present in each cell.

The structural features of the nucleotide bases allow them to be classified as either pyrimidines or purines. In contrast to pyrimidines, which only have one ring with six nitrogen atoms, purines have two ring configurations, one with six nitrogen atoms and the other with five. The nucleotides adenine and guanine are considered purines, while cytosine, thymine, and uracil are considered pyrimidines. In DNA, you can find the bases adenine, cytosine, guanine, and thymine. The sole variation in RNA is the presence of Uracil rather than Thymine. The five nucleotide structures are illustrated in Figure 1.1

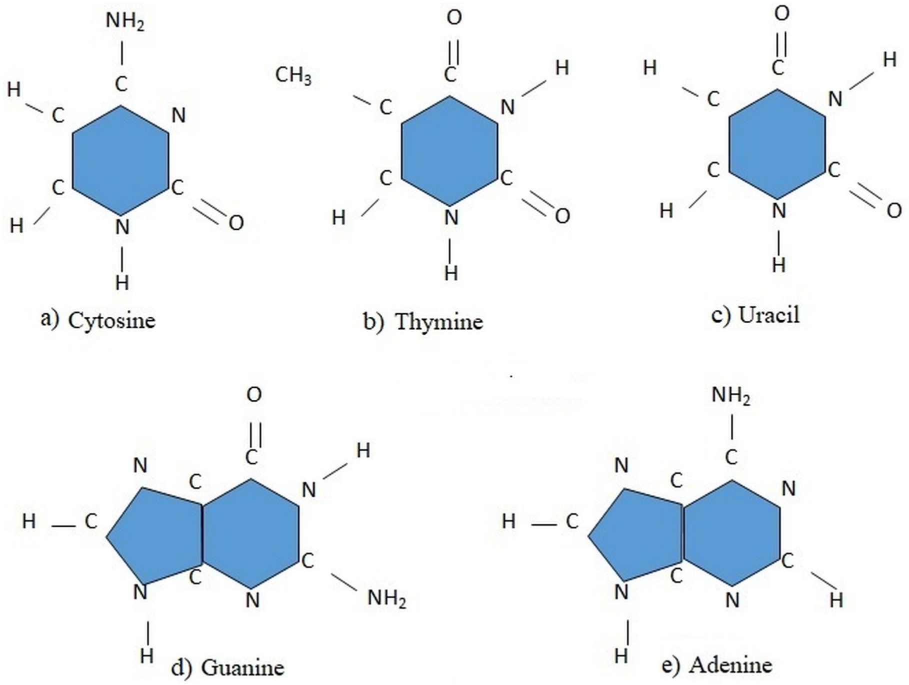

a) Cytosine b) Thymine c) Uracil d) Guanine e) Adenine

Figure 1.1: The nucleotides structure and Function of DNA.

Chromosomes, which are made up of DNA base pairs, are where genes are located. A gene is a functional product of a base pair sequence that can be an RNA molecule or, later on, a peptide. In the human genome, genes spread over 33.4% from start to stop codon, of which the protein coding sequences are only 3.66% [2]. The remaining gene space is occupied by non-coding regions, Introns, which separate adjacent exons from one another.

1.1.2 DNA Repeats

DNA repeats, which are patterns of nucleotides, are found in both prokaryotes and eukaryotic genomes in many copies. In tandem repetitions, the DNA patterns are directly next to each other; in other cases, they are dispersed throughout the genome. Repetition in the form of long diverse components is also seen in the human genome. Forensic and paternity testing uses of DNA fingerprinting can benefit from this variation in individuals. The degree to which different types of DNA repetitions influence phenotype varies among different types of repeats.

1.1.2.1 Tandem repeats

The replication of two or more base pairs adjacent to each other is a tandem repeat. These repeats are commonly correlated with non-coding DNA. In certain cases, the repetition of base pairs varies in number.

Example: ACTACTACTGTGTTTTGTGTGTGTGTAAAAAAGGGGGTGGGGT Consider a pattern (GT) of length two repeated twice from positions 10 and five times from position 18. in the same way, the three length pattern ACT repeated thrice from position 1.

1.1.2.2 Tandem repeats with interrupts

A tandem repeat with diverse continuation of base pairs is interrupted tandem repeat.

Example: ACACGTGTGTACACGTGTTCTCGTGTGT In the sequence it can be seen that "AC" is occurring at two different places and one more pattern "GT" is diverting the continuity of "AC". Hence GT is called the interrupt with tandem repeat. Tandem repeats with interrupts are shown in Table 1.1 Short tandem repeats is a special case of tandem repeat involving a repeated unit of 2 to 7 base pairs in length.

Table 1.1: DNA Tandem repeats with interrupt

Pattern	Start Index	End Index	Interrupt Pattern	Length of Interrupt
AC	5	10	GTGTGT	6
TG	10	23	TACACGTTCTCG	12
GT	11	14	ACAC	4
GT	19	22	TCTC	4

1.1.2.3 Mirror repeats

A mirror repeat is a sequence segment delimited on the bases of its center of symmetry on a single strand and identical terminal nucleotides. For example in the mirror repeat sequence ATCGTCCTGCTA the ATCGTC is the mirror image of CTGCTA.

1.1.2.4 Mirror repeats with interrupts

A mirror repeat with interrupt finds an interrupt between two mirror images. For example in the mirror repeat with interrupt sequence ATCGGCCTGCTA the ATCGGC is the mirror image with interrupt of CTGCTA.

1.1.2.5 Pairing repeats

In the DNA, nucleotides A and T complementary base pairs, similarly C and G. Pairing repeats are sequence of nucleotides in which a pattern is followed by its complementary pattern. Variable length pairing repeats are shown in Table 1.2.

Example: ACTGCTTTGTGACGTGCGACTGGTGACCTAATTTATTAAA In the sequence ACGTGC, TAATTTATTAAA are two pairing repeats in which ACG is followed by its complementary pattern TGC, similarly TAATTT is followed by ATTAAA

Table 1.2: DNA Variable length pairing repeats.

Pairing Repeat	**Pattern**	**Start Index**	**End Index**	**Complementary Pattern**	**Start Index**	**End Index**
ACGTGC	ACG	13	15	TGC	16	18
ACTGGTGACC	ACTGG	20	24	TGACC	25	29
TAATTTATTAAA	TAATTT	30	35	ATTAAA	36	41

1.1.2.6 Pairing repeats with interrupts

A pairing repeat with interrupt finds an interrupt between two complementary patterns. For example in the repeat CAATTTGTGAAA instead of T at position 9 for complementary nucleotide A there is an interrupt G occurred.

1.1.2.7 Inverted repeats

The inverted repeats are sequence of nucleotides in which a pattern is followed by its reverse complementary pattern.

Example: ACCTAGGTGAAAAAATTTTTC In the sequence ACCTAGGT, GAAAAAATTTTTC are two inverted repeats in which ACCT is followed by its reverse complementary pattern AGGT, similarly GAAAAAA is followed by TTTTTC.

1.1.2.8 Inverted repeats with interrupts

A inverted repeat with interrupt finds an interrupt between pattern and its reverse complementary pattern. For example in the repeat GAAAAAACTTTTC instead of T at position 8 for reverse complementary nucleotide A there is an interrupt C occurred.

1.1.3 DNA Mutations

Any alteration to the DNA sequence is known as a mutation. A genetic ailment develops in a person when there is an anomaly in their genetic makeup. Any number of mutations, from those resulting from the addition or deletion of chromosomal sets to those affecting only a single nucleotide might cause a genetic disorder. Some diseases are passed down via generations of parents, while others develop as a result of mutations brought about by commonplace habits and environmental factors, such as smoking, poor nutrition, lack of exercise, and so on. A variety of complicated human diseases include genetic components. Autosomal dominant (AD), X-linked recessive (XR), X-linked dominant (AD), and Y-linked holandric (HR) illnesses are the several types of single gene disorders that result from DNA sequence abnormalities. Mutations in repeats can also cause some neurological diseases. As an example, the neurological disorder known as Huntington's disease (HD) is passed down through generations in families by the autosomal dominant pattern of gene expansion at the CAG trinucleotide in the first exon of the HD gene. The frequency of the CAG repeat, which is quite polymorphic, often falls between the usual range of 6 to 37 in healthy individuals. The CAG frequency range on HD patient genes varies from 30 to 180 due to mutations.

1.1.4 Point Mutations

Any alteration, addition, or deletion of a single base pair in a genome is called a point mutation. Point mutations are often harmless, although they can modify gene expression or encode proteins in unexpected ways, among other functional effects.

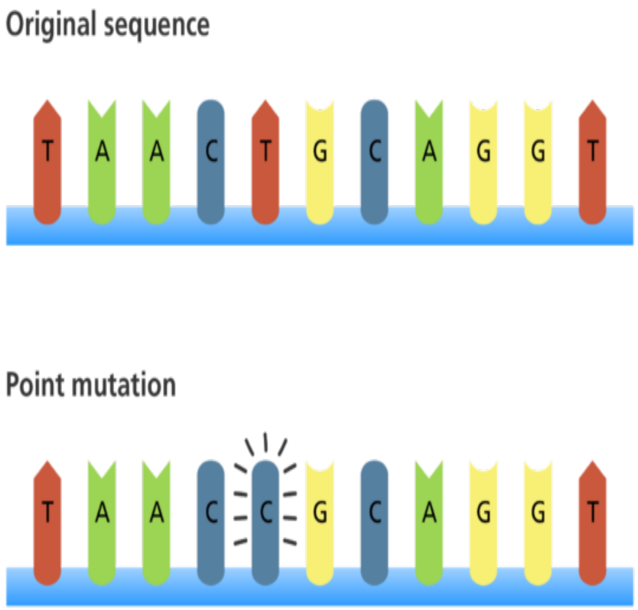


Figure 1.2: DNA sequence Point mutation with single character 'C' change.

While most point mutations are benign, they can also have various functional consequences, including changes in gene expression or alterations in encoded proteins. Different types of Point mutations can be silent, missense, or nonsense mutations, as shown in Table:1.3

Table 1.3: DNA different types of Point mutations

Type	**Description**	**Example**	**Effect**
Silent	mutated codon codes for the same amino acid	CAA (glutamine) → CAG (glutamine)	none
Missense	mutated codon codes for a different amino acid	CAA (glutamine) → CCA (proline)	variable
Nonsense	mutated codon is a premature stop codon	CAA (glutamine) → UAA (stop)	usually serious

1.1.5 Coronaviruses

In the family of coronaviruses (CoVs), there are two subgroups: common human CoVs and other human CoVs. Common human CoVs like NL63, OC43, and HKU1 are less dangerous to humans, while other human CoVs like SARS-CoV, MERS-CoV, and SARS-CoV-2 are more dangerous because they originated in animals and were transmitted to humans [17]. Upper respiratory tract infections caused by human coronaviruses were initially identified in children in the 1960s. A virus originally known as B814 was reclassified as 229E after its 1965 discovery in adults [18]. Beta coronavirus, which infects both people and animals, was discovered later in OC43. In November 2002, SARS-CoV was initially identified in the Chinese province of Guangdong. The World Health Organisation (WHO) reported that 800 persons had died from this infectious disease out of an estimated 8,000 cases. After that, in 2004, researchers discovered NL63, a different coronavirus that primarily affects younger children and causes mild to moderate lower respiratory system infections [19]. A genetic difference between HKU1 and OC43 and other coronaviruses was discovered in 2005 [20]. As a zoonotic virus, MERSCoV was discovered in 2012. The World Health Organisation reports that approximately 35 of infected individuals have passed away as a result of MERS-CoV. A new coronavirus, COVID-19, that is similar to SARS-CoV, started wreaking havoc in December 2019 and will continue to infect over 200 nations and 6.4 million people until 05-06-2020. All three coronaviruses—MERS, SARS-CoV, and SARS-CoV-2—are zoonotic, meaning they came from animals and then infected people. Congenital obstructive vaginosis viruses can infect just specific host cell types. Parasite (virus) host specificity refers to the variety and quantity of host species utilised by parasites. The molecular basis for host specificity suggests that in order for a virus to interact, a surface molecule known as a viral receptor must be present on the host's surface. Different coronaviruses are host-specific for humans and a wide variety of other animals, including pigs, dogs, camels, and bats. Permissive cells allow the host receptors to enter and utilise the reproductive machinery of human cells [21]. Host receptors for 2019-nCoV, SARS-CoV, and NL63 in humans are Angiotensin-Converting-Enzyme 2 (ACE2), Dipeptidyl Peptidase IV (DPP4), Aminopeptidase N (hAPN), N-acetyl-9-O-acetylneuraminic acid (Neu5, 9Ac2), OC43, and HKU1 [24]. Inside a protein shell termed Nucleocapsid (N), coupled with two membrane proteins, namely Membrane (M) and Envelope (E), and one glycoprotein Spike (S) [25], are positive-stranded DNA (or RNA) genomes seen in coronaviruses. Initiating the infection, the S protein binds the virion to the host cell's receptor (ACE2) and is susceptible to mutations on the receptor interface that protect the host immune system [26, 27]. In addition to its role as a channel for ion transport, the E protein creates holes between proteins and lipids [28]. In addition to aiding in viral manufacturing, the M protein is critical for improving viral RNA transcription [29]. The N protein, which is highly conserved and versatile, regulates host cells by intoxicating their machinery and interacts with the M protein during virion assembly [30]. Viruses that replicate genetic material rely on enzymes and proteins found in host cells to construct their own DNA, which is then translated into messenger RNA. The process of protein and enzyme synthesis within the host cell is regulated by the viral mRNA.

1.1.6 Technical Background

1.1.6.1 Trie data structure

An effective data structure for retrieving information is the trie. The search complexity can be optimised using trie. The complexity of searching for a key in a trie data structure is O(key length). Key values of varying lengths are particularly well-suited to trie indexing. The trie used links to infer the values of keys rather than store them directly. Only by using a portion of the key value can trie support multi-way branching, not the full key.

1.1.6.2 Convolutional neural networks

CNNs are multilayer feed-forward neural networks, which learn to map a fixed size input (images, One hot encoded genome sequences) to a fixed size output (probability for each of several classes) [3]. The CNN architecture consist of multiple layers, typically, one input and a number of hidden and one output layers. Each layer contains a number of neurons and each neuron consist of various parameters (weights) [4]. To go from one to the next layer, a weighted sum of their inputs from the previous layer is computed and the result is passed through a non-linear function. The non-linear activation function plays a key role after convolution to understand CNN. The three commonly used non-linear activation functions are Sigmoid, tanh, and ReLU. The most popular non-linear function is a ReLU, simply a half-wave rectifier (f(z) = max(0,z)), which learns much faster than other activation functions like tanh(z) and sigmoid(1/1+exp(-z)) as it involves simple mathematical operations [5],

[6]. Moreover, non-saturation of gradient is indeed biggest boon of ReLU, which speed up the convergence of stochastic gradient descent than other activation functions [7]. In classification problems, the output layer uses the softmax to calculate the probability for each class. CNNs are NNs with one or more convolution layers, which contains a number of filters, sliding over one hot encoded sequence to detect patterns as features. In CNN, the weights are stored in filters shared over different positions.

1.2 Motivation, Aim & Objectives

1.2.1 Motivation

Genome sequence analysis involves using computer approaches to investigate and comprehend the properties, structure, function, evolution, and characteristics of DNA, RNA, and proteins. The complexity of analysing sequences has grown in tandem with the massive amounts of DNA sequence data generated by modern sequencing technology. Genomic data is expanding at a far quicker rate than sequence analysis. That being said, quicker techniques for sequence analysis are desperately needed. Disease detection, medication development, agriculture, and forensics can all benefit from genomic sequence analysis. Finding sequence homology, intrinsic feature identification, mutation discovery, genetic diversity disclosure, and species evolution are just a few of the many research topics encompassed by sequence analysis in bioinformatics and computational biology.

Clustering and classification allow us to obtain genomic sequence homology. When processing data from genomic sequences, clustering is a crucial step. When comparing sequences, most of the current tools rely on alignment-based methods, which are tedious and time-consuming. When it comes to quick clustering, alignment-free methods work well. Though they are vulnerable to large-size sequences, state-of-the-art approaches have been used to cluster tiny genome sequences of different species. Like in other genomes, the majority of human and other species' DNA is composed of repetitive sequences. The structural and functional functions, locations, lengths, and numbers of covid-19 repetitive DNA types all play a role in how significant each type is. It is still a mystery to biologists where exactly these DNA sequences are located on the chromosome and whether or not they should be conserved. It is still difficult to detect their position and identify new repeated sequences because of how variable they are. We circumvented this issue by investigating the function of repetitions in complicated illness initiation in humans and their type, structure, and control. The role of repetitive DNA, especially tandem repeats, is critical in genetics. It turns out that the extremely repetitive regions found in the X and Y chromosomes are really located in other human chromosomes or genomes.

Many cancers arise from diseases or alterations in genes that regulate growth. The ability to accurately forecast splice signals is fundamental for many biological and medical fields, including gene regulation, alternative splice events, human illness diagnosis, and medication discovery. Nevertheless, due to its enormous size and intricate structure, identifying those borders is no easy feat. Understanding the nucleotide relationships, dependencies, and properties in the Covid-19 environment is crucial for accurate splice boundary detection.

Any change to the sequence of nucleotides in a gene is called a mutation. Motif frequency in genomic sequences can be altered by mutation. Protein abnormalities caused by mutations in the coding regions (exons) can cause complicated illnesses. On the one hand, there are hereditary diseases; on the other, mutations caused by commonplace lifestyle choices and environmental factors, such as smoking, poor nutrition, lack of exercise, and so on, are the root causes of many problems. One of the most difficult challenges in personalised medicine is disease status prediction for complicated human diseases using genomic data. Coronaviruses are among the many human diseases caused by viruses; they penetrate host cells, interact with host molecules, and may disrupt the normal function of host cells, which can lead to cancer and other devastating illnesses. In order to comprehend complicated viral infections such as NL63, MERS-CoV, Ebola, etc., novel viral genome prediction is essential. There is data in the genetic code that affects how fast and efficient translation is. The proteome is defined in large part by translation elongation, and diseases can result from mistakes in proteins. In order to compare the new coronavirus (2019-nCoV) with other human coronaviruses (CoVs), it is necessary to conduct a codon-level analysis of the virus. This necessitates an extensive and comparative study of different zoonotic and human-associated COVs concerning codon usage bias, relative synonymous codon usage (RSCU), codon frequencies, mutation bias, the effective number of codons (ENC), and slow di-codon and codon proportions.

1.2.2 Aim

Improved genome annotation, analysis, classification, and illness detection can be achieved by the methods developed in this dissertation, which utilise machine learning and deep learning to uncover and understand hidden patterns of mutations.

1.2.3 Objectives

The main objectives of this dissertation are stated as follows:

- To develop a Exploring Coronavirus Sequence Motifs through Convolutional Neural Network for Accurate Identification of Covid-19.
- Coot-Lion Optimized Deep learning Algorithm for COVID-19 Point mutation rate prediction using Genome Sequences.
- DNA Sequence Clustering and ERSIT-GRU(Exponential Robust Scaling-Identity Tanh- Gated Recurrent Unit) for Repeat Detection in COVID-19 Prediction.
- Genome-wide analysis for Tandem Repeat and Substitution Errors to detect Covid-19 using Harris Hawks Optimization.

We describe COVID-19 illnesses in this thesis. The use of deep learning and machine learning for disease pattern extraction from DNA sequences, with predictions including DNA point mutations, DNA repeats, motif sizes with filter identifiers, dataset sizes, feature extraction, frequency calculations, and so on. Optimisation Strategies for Harris Hawks.

1.3 Overview of the Contributions of the Thesis

In this section, an overview of the chapter-wise contributions of the thesis is presented. Each subsection presents a summary of the contributions of the chapters.

Proposed Work 1: Exploring Coronavirus Sequence Motifs through Convolutional Neural Network for Accurate Identification of Covid-19.

Technique: To identify Covid-19 datasets in DNA sequences of various types of Corona-Virus. An alignment-free method for SARS-CoV-2 classification utilising complementary DNA—DNA produced by the single-stranded RNA virus—is described in this paper. In this study, we collected data from 1582 samples, including both positive and negative datasets with genomic sequences of varying lengths. It must be able to efficiently process huge DNA sequences while improving clustering accuracy and speed by utilising all of the information in the sequences. We used a new method called convolutional neural networks (CNNs) to extract unique top DNA motif subpatterns from massive DNA sequences. [8]. One hot encoding uses binary vectors to represent categorical variables. The first step is to convert the category values to integers. Then, a binary vector with the values 0 and 1 is used to represent each number. By employing 10-fold cross-validation, we were able to evaluate the classifiers' efficacy using the training dataset and metrics like accuracy and F-measure. We investigate and display physiologically significant features (motifs) automatically learned by CNN using the learnt filters in order to solve the neural networks' decision-making process gap.

1. We applied a method called Convolutional Neural Network (CNN).
2. Our method generates to identify top motifs Pattern diseases in a covid-19 with filter Id.
3. One hot encoding to extract the features for DNA sequence.
4. It improve the system's accuracy and solve, prepossessing feature extraction diseased pattern.

Proposed Work 2: Coot-Lion Optimized Deep learning Algorithm for COVID-19 Point mutation rate prediction using Genome Sequences.

Technique: Proposed Technique:A Deep quantum neural network (DQNN) based on the Lion-based Coot algorithm (LBCA-based Deep QNN) is employed to predict COVID-19

DNA Point mutation. A change in the sequence of bases in DNA is called mutation[9]. The contributions of this work are listed below.

- We proposed an Coot algorithm (LBCA-based Deep QNN) method to identified Point mutation in covid-19.
- We extracted DNA CpG-based features CGp1 = P(C) + P(G).
- DNA Point mutation rate is calculated for each Covid-19 datsets.
- Our technique has been tested on Comparative investigation of LBCA DQNN in terms of (a) mutation rate, (b) testing accuracy (c) TPR and FPR.

Proposed Work 3: DNA Sequence Clustering and ERSIT-GRU (Exponential Robust Scaling-Identity Tanh- Gated Recurrent Unit) for Repeat Detection in COVID-19 Prediction):

Technique: Proposed a ERSIT-GRU for a novel repeats in covid-19 from huge DNA sequences in Covid-19.

Direct repeat: The Direct Repeats are nucleotide sequences present in multiple copies within the same pattern in the genome sequence. Example ACGACGACGACGACGAC-GACG is direct repeat wherein the sequence ACG is repeated many times in a DNA sequence [10].

Mirror repeat :DNA Mirror repeat is a sequence Imperfect DNA mirror repeats (IMRs) are less than 100 percent symmetrical. Example ACGTGTCCACGTCGT is a Mirror repeat wherein the sequence TGCTGCACCTGTGCA is reverseively. The contributions of this work are listed below.

- DNA K-mer Repeat Sequence Identification is done such as Uni-character, Bi-character, Tri-character, Reversion Inversion.
- Direct Repeat feature extracted from DNA Sequence using GRU.
- Mirror Repeat feature extraction from DNA Sequence using GRU.
- DNA repeat strings used to create the repeat set.

- Proposed Model The ERSIT-GRU takes 58056ms less time to execute the results, when we compared with existing models such as GRU, LSTM, and RNN techniques.

Proposed Work 4: Genome-Wide Analysis for Tandem Repeat and Substitution Errors to detect Covid-19 using Harris Hawks Optimization.

Technique: An efficient algorithmic framework for tandem repeat, motif, and mutation rate detection in COVID-19 is proposed in this article. Our approach finds key DNA motifs and brief tandem insertion/substitution mistakes with mutation rates ranging from mild to high. Sequence Length, size, frequency base, and dimer features are extracted. In order to uncover the evolutionary relationship between seven coronaviruses, we conducted an analysis using the proposed model Harris Hawks Optimisation (HHO). [11]. Multiple efforts that focus on error correction for DNA-based tandem repeats, Insertion Error, Deletion Error, 2-substition Error, 3-substition Error. The contributions of this work are listed below.

- Tandem Repeat storage proposed feature extraction technique, the size of the sequence is expressed in terms of KB (Kilo Bytes) Size(S) = Length(S)/1024 .

- The feature ”Dimer Count (DC)” refers to the number of occurrences of all possible combinations of Dinucleotides in the Genome sequences. Dinucleotides = AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG,GT, TA, TC, TG, TT DC(S) = Number of occurrences of individual Dinucleotides.

- Most Repeat Pattern Count (MRP):Most Repeat Pattern Count refers to the number of occurrences of MRP in the Genome Sequence.

- Mutation rate (MR) is calculated by the following formula: MR = TM/TNB x 100 (where TM is the total mutation taking place in the two sequences and TNB is the total number of nucleotides.)

- Proposed model reached highest Accuracy in terms of the various Parameters like Accuracy, Precision, Recall, F1 Score.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows, Chapter 2, is about Literature Survey, this chapter describes the recent state-of-the-art works on large DNA sequence clustering for finding homology between different Covid-19 sequences, by applying Machine learning, Harris Hawks Optimizer (HHO) and Deep learning Techniques applied for predicting viral DNA sequences based on extracted features different types of Repeats and Mutations. Next chapter, i.e., Chapter 3, Presents a Exploring Coronavirus Sequence Motifs through Convolutional Neural Network for Accurate Identification of Covid-19. Chapter 4, presents a Coot-Lion Optimized Deep learning Algorithm for COVID-19 Point Mutation rate Prediction using Genome Sequences. Next, chapter 5, presents a DNA Sequence Clustering and ERSIT-GRU for Repeats Detection in COVID-19 Prediction. Chapter 6, introduces a Genome-Wide Analysis for Tandem Repeat and substitution Errors to detect Covid-19 using Harris Hawks Optimization. Finally, chapter 7 summarizes the work presented in this thesis and mentions future directions of research related to these problems.

Chapter 2

Related Work

In this chapter, a brief survey of the literature related to the contributions made in this thesis is given. The field of bioinformatics and computational biology aims to investigate various research concerns like sequence analysis, different genome signals prediction and disease prediction. Toward this end, many works exists to address these concerns. However, there are some limitations and trade offs. All these approaches are majorly categorized into alignment based, alignment free, probabilistic and machine learning, and deep learning based methods. An alignment based approach needs base-by-base comparisons to obtain similarity score. "Alignment free methods make use of pattern frequency, length of the common sub-string and the number of word matches for sequence similarity search. Machine learning methods initially construct a set of features, then perform feature reduction for effective feature set, and finally these features are used for prediction or classification. Recently deep learning is emerged as prominent technique in sequence-based bioinformatics because of its weight sharing and automatic feature extraction mechanism.

2.1 Alignment based methods

In alignment based approach, the similarity is measured by obtaining the score from FASTA [12] or BLAST [13]. Popular sequence clustering tools like CD-HIT [14], DNACLUST [15], and UCLUST [16] follow greedy algorithms, which may not guarantee an optimal solution. Alignment based methods like Tophat [17], use reads from DNA sequence data for splice site prediction. The detection of viral sequences in human bio-specimens is generally performed by using BLAST [13]. The sequences are compared to known publicly available databases and classify the sequences based on the similarity index. Metagenomic datasets contain divergent virus sequences so there is no similarity at all among known database sequences. As a result, many of the virus sequences produced from sequencing technologies are categorized as “unknown” by the NCBI BLAST [18], [19]. The most popular alignment-based techniques for viral genome classification are REGA [20], [21], USEARCH [16], and SCUEAL [22]. Another tool for virus sequence detection within metagenomic sequence datasets is HMMER3 [23], which uses profile Hidden Markov Models by comparing with vFams [24] database, vFams, a database with viral family proteins was designed by Multiple Sequence Alignments (MSA) from all RefSeq viral proteins. HMMER3 detects homological viral sequences more effectively but not highly divergent ones [25] because it depends on the reference database VFams. MeShClust [26] uses mean shift algorithm to cluster the DNA sequences. The ability of MeShClust is to cluster DNA sequences with high accuracy even though the sequence similarity parameter provided by the user is not very accurate. All these alignment based methods purely depend on the alignment score between the viral sequence being classified and the reference dataset. The summary of existing alignment based methods are shown in Table 2.1. The major drawbacks include, the classification performance depends purely on the selection of one of the several initial alignments and hyper-parameters. For larger genomes, sequence alignment is not preferred due to high computational complexity, even though it gives better results. These methods are expensive and their performance is unstable for divergent regions of the genome.

2.2 Alignment free methods

Alignment-free methods have been used in sequence similarity searches, clustering, classification, and more recently in phylogenetics. The pattern based measure is one of the most commonly used alignment free methods as explained in [27], [28]. In this method, a short string of length(l) is used to generate n-dimensional vectors by mapping each sequence. Pearson correlation distance [29], Euclidean distance [30] assesses the similarity between two vectors. Normalized local histograms are obtained by using the frequencies of four bases(A, C, G, and T), in turn, which are used as features for sequence clustering effectively [31]. Hierarchical and Partitional clustering [32] are two major genome clustering algorithms based on the result format. The former algorithm, generates a set of partitions, which forms a cluster hierarchy, and the latter obtain partitions by optimizing certain clustering criteria. Hierarchical approaches may produce good clustering results, but they are complex and need high computational time and memory for large data sets [33]. Where as, partitional algorithms are simple and best-suited for large genome sequences [34]. Hierarchical algorithm generates nested series of clusters while partitional algorithms produces flat clusters. BlastCLUST [35], a hierarchical clustering approach, measures sequence similarity based on the BLAST [13] score and generates clusters of linear topology. The limitation of BlastCLUST is that its performance reduces with the increase in input size. CD-HIT-EST [36], is also a widely used partitional algorithm to cluster DNA sequences. It aligns sequences by using the frequency of identical motifs between them. Even though, CD-HIT-EST's performance is better than BlastCLUST, in most of the cases both algorithms are generating clusters with only one sequence [37]. K-means is a partitional clustering algorithm that partition the sequences into some clusters. K-means is used in [38] to cluster Hepatitis B Virus (HBV) sequences into two groups. In which, the first group HBV sequences are virulent than the second cluster sequences. Although MeShClust [26] has shown superior performance in terms of cluster quality with other related tools its ability to generate training data for classification is not suitable for longer sequences. MeShClust2 [26], on the other hand, generates semi-synthetic sequence pairs, avoiding alignment algorithms, with known mutation rates. The method [39], uses the fuzzy integral with the markov chain to cluster the DNA sequences by taking into account the occurrence frequencies of of DNA sequence of all possible nucleotide pairs. The summary of state-of-the-art alignment free methods are shown in Table 2.1. However, alignment free methods have consistently shown poorer efficiency in quantifying low-abundance and small-scale sequences.

Table 2.1: Summary of selected alignment based and alignment free methods for sequence analysis tasks.

Article's Reference	**Clustering/Prediction**	**Technique**	**Performance Measures**	**Datasets Used**	**Result**
Alcantara et al. [[20]]	Alignment based clustering	phylogenetic-based tool-REGA	sensitivity and specificity	HIV-1 pol sequences	96.6, 96.8
Edgar et al. [[16]]	Alignment based clustering	UBLAST, USEARCH	similarity	Pfam, Rfam	87.1-99.5%
Pond et al. [[22]]	Alignment based Viruses prediction	Evolutionary Algorithm	-	HIV-1	Accurate phylogenetic breakpoint map
Bzhalava et al. [25]	Alignment based Viruses prediction	HMMER3 algorithm	-	Metagenomic sequencing	Viruses sequences are identified
Saw et al. [39]	Alignment free DNA clustering	Fuzzy integral algorithm	AUC-ROC	30 Coronavirus , Bacteria genomes	0.841, 0.926
James et al. [[26]]	Alignment free DNA clustering	K-mer counts with mean shift algorithm	Purity, NMI	Bacterial dataset	0.741, 0.630
James et al. [[26]]	Alignment free DNA clustering	Mean shift algorithm	Silhouette, NMI	Synthetic	1.0, 0.59
Mendizabal-Ruiz et al. [[40]]	Alignment free DNA clustering	K-means clustering algorithm	Time (secs)	KEGG, COXI	0.770
Bustamam et al. [38]	Alignment free DNA clustering	K-means clustering algorithm	Min-max normalization	hepatitis B virus sequences	2 clusters
Aleb et al. [41]	Alignment free DNA clustering	Improved K-means Algorithm	Execution Time (ms)	DNA sequences	375
Wei et al. [42]	Alignment free Gene clustering	mBKM with DMk	(F-score, Time)	NCBI, HOVERGEN, HOGENOM, HOMOLENS	(0.8080, 6.875), (0.9645, 1.844), (0.9143, 2.375), (0.9587, 1.328)
Sperisen et al. [34]	Alignment free DNA, Protein clustering	Random sampling, Bootstrap	Blosum62/-12/-1	(Prokaryotic lyases), (SH2 containing proteins)	(0.969, 0.887), (0.689, 0.735)
Li et al. [36]	Alignment free DNA, Protein clustering	Cd-hit and cd-hit-est	Time	NCBI-nr with more than 3.2 million proteins	8 hours
Bzhalava et al. [43]	Alignment free Gene clustering	GMAP program	Sensitivity, JI Index	Homo sapiens	0.995, 0.995

2.3 Probabilistic and Machine learning Methods

In probabilistic methods, consensus patterns are taken into consideration for estimating position-related probabilities by calculating the likelihood of candidate sites, and find the underlying relationship between the nucleotides around DNA mutations, Repeat sites regions. Many probabilistic models have proposed to increase the predictive power for instance taking advantage of Markov Models [44], [45], Random Forest [46], Bayesian Networks [47], and Support Vector Machines (SVM) [48]. In SpliceIT [48] positional probabilistic descriptions of various orders are created and a pool of candidate characteristics is produced. Each feature's discriminative power is evaluated and the most informative features are selected using either collection of positional features or pruning with examination of principal features. On probabilistic parameters, the SVM classifier is trained. Machine learning algorithms learn how to make predictions based on genome data and have a range of emerging bioinformatics applications. The conventional machine learning methods have been successfully used for biological prediction problems based on DNA or protein sequences [49]. To predict exact diseased pattern in covid-19, they focus on learning complex features from consensus di-nucleotides AG/GT before passing it to a classifier. Various machine learning methods, including Hidden Markov Model [50], [51], SVM [52], [53], [54], [55], [56], [57], Random Forests [46], and Bayesian Networks [47] have been developed for splice signal classification. Various computational methods are the possible alternative solutions for the prediction of potential DNA and types of Repeats and mutations [58]. PWMs are used to indicate the sequence specificity of a protein and are easy to interpret to identify mutation occurs in DNA diseased pattern [15]. Sophisticated computational techniques (like machine learning) have achieved ample performance in capturing sequence specificities [59], [60]. Various supervised learning and unsupervised machine learning algorithms have been developed to solve types of Diseased pattern and DNA mutations problems.

Supervised learning methods need label information to infer model training and help to predict whether the transcription factors are bound or unbound to specific regions. Many methods exist for predicting binding sites based on supervised learning includes discriminative maximum conditional likelihood [61], support vector machines[62], and random forest[63]. Unsupervised techniques generally cluster the bounded and unbounded genome sequences separately based on their bond to certain transcription factors. Hierarchical mixture models [56], and hidden markov models [64], [29] are unsupervised methods. Hence, all these probabilistic and machine learning methods mostly use a three step process: first construct the set of features, then perform feature reduction for effective feature set, and finally feed these features to a machine learning model for better prediction or classification. The performance of probabilistic and machine learning algorithms purely depends on the constructed features [65]. However, features are extracted by domain experts, the prediction performance remains undefined and unknown. So, the state-of-the-art machine learning methods still confront many issues, like their incapability to obtain useful information from raw DNA data [3], [66], the challengeable discovery of splice signal patterns, overfitting, and underfitting. Disease prediction is performed by patient care datasets analysis [67], DNA patterns analysis [68], [67], [69] and DNA patterns analysis [68], [70]. Several methods are proposed to classify viral metagenomic sequences [71]. VirSorter [72], a probabilistic tool to predict novel viruses in microbial genome data with and without reference. VirFinder [73], machine learning model to identify viral contigs based on k-mer frequency. The summary of existing probabilistic and machine learning methods are shown in Table 2.2.

The existing recommendation like system ViraPipe [43] used an artificial neural network and random forest by using relative synonymous codon usage frequency to improve the classification of metagenomic data into a virus and non-virus sequences. The performance of machine learning algorithms purely depends on the constructed features. Extraction of these numerical features is a tedious job. Moreover, there is a high chance of missing effective information. So, the state-of-the-art machine learning methods still confront many issues like their incapability to obtain useful information from raw DNA data and the challengeable discovery of various signal patterns. However, features are derived by domain experts, the predictive performance remains uncertain and unknown.

Table 2.2: Summary of selected machine learning based methods for Covid-19 and viral DNA sequence classification tasks.

Article's Reference	**Classification/Prediction**	**Technique**	**Performance Measures**	**Datasets Used**	**Result**
Sonnenburg et al. [56]	Covid-19	SVM	AUC-PR	GWH SARS-CoV-2/2019-nCoV	0.5412/0.5469
Bari et al. [74]	Covid-19	SVM	Sensitivity, Specificity, Accuracy, AUC-ROC	NN269 MERS-CoV / SARS-CoV	0.7740, 0.8716, 0.9339, 0.9790/ 0.8798, 0.9719, 0.9525, 0.9830
Lee et al. [75]	Covid-19	Boltzmann machines	F1 score	GWH HCov-229E SARS-CoV-2r/ Donor	0.753/0.816
Chen et al. [76]	Covid-19	SVM	Accuracy, Mcc, Sensitivity, Specificity, AUC-ROC	Human Covid-19 splicing Acceptor/ Donor sites	88.73, 77.89, 94.24, 83.07, 95.18/ 87.71, 75.46 89.56 85.86 92.39
Zhang et al. [51]	Covid-19	Markov models with SVM	Sensitivity, Specificity, Global Accuracy	HS^3DHCoV-OC4/ Donor	0.9024, 0.8757, 0.8879/ 0.9306, 0.9131, 0.9210
Malousi et al. [48]	Covid-19	Gaussian SVM	Sensitivity, Specificity, AUC-ROC	HS^3D, AT HCoV-OC4/ Donor	(93.56, 92.20, 0.751), (90.72, 90.82, 0.712)/ (94.96, 92.97, 0.782), (93.68, 93.61, 0.783)
Wei et al. [57]	Covid-19	SVM	Accuracy, AUC-ROC	HS^3D SARS-CoV/ Donor	92.29, 0.973/ 92.85, 0.976
Meher et al. [46]	SARS-Covid-19	RF,SVM, ANN	True positive rate	HS^3D Donor	0.984, 0.935, 0.892
Amgarten et al. [47]	Covid-19	Bayesian networks	False negative rate and False positive rate	Multiple-exon human genes	0.78,0.77,0.88
Amgarten et al. [71]	SARS-CoV-2 Bacterial sequence	RF	Recall, Specificity, Accuracy, F1-score	Metagenomic bins	0.91, 0.89, 0.935, 0.942
Roux et al. [72]	Viral sequence	RF and ANN	Precision, Recall	Microbial sequence data	99.82%, 98.99%
Ren et al. [73]	Viral sequence	k-mer frequency and Machine learning	AUC-ROC	Mixed metagenome sequence	0.76 to 0.97
Bzhalava et al. [43]	HCoV-OC4 Viral sequence	RF and ANN	AUC-ROC	Metagenomic sequence	0.79

2.4 Deep learning methods

Due to technological advances, deep learning is emerged as a prominent technique that has demonstrated outstanding results in the field of bioinformatics, genomics and computational biology. It is a prominent technique, has achieved record-breaking results in computer vision [77], speech recognition [55], natural language processing [78], [79], Image processing [7], [80], [81] and sequence-based bioinformatics [82], [83]. Deep learning applications in the bioinformatics, genomics and computational biology mostly concentrate in (i) genome sequencing and analysis [84], [80], [85], [86] (ii) classification of DNA [87], [88], chromatin [89], polyadenylation [90], and (iii) protein structure prediction [80], [91], [92], [93]. Several architectures based on CNN and RNN have been developed for splice sites and other signals such as Repeats, mutations branch points and polyadenylation prediction. Convolutional neural networks is one of the most successful architecture for genome sequence analysis in widely used deep learning architectures. To address the limitations of existing machine learning algorithms, several CNN based models are developed for splice sites and other signals such as transcription factor binding sites [94], [49], branch points [95] prediction. The automatic feature extraction and weight sharing mechanism of CNN has made it suitable for several sequence analysis tasks such as splice signal classification and pattern recognition. Various state-of-the-art CNN models for splice junctions prediction includes SpliceRover [4], iSS-CNN [96], DNN [97], DeepSS [98], RepeatsFinder [99], and RepeatsDeep [100]. RepeatsRover [4] an end-to-end learning model for classification of true/pseudo Mutationsand utilized deepLIFT [61] to visualize the patterns. The performance of iSS-CNN [96] and DNN [97], were evaluated on HS^3D acceptor and donor datasets. DeepSS [98] contains two modules, DeepSS-C for splice site prediction and DeepSS-M for downstream analysis for pattern prediction.

RepeatsFinder [99], randomly extracted exons to generate acceptor and covid-19 datasets and predict both canonical and non-canonical Mutations. Repeats2Deep [100] accurately recognize Mutations for those datasets on which the model was not trained. Disease diagnosis is performed by image analysis [101], [102] medical statistical datasets [81] and pattern matching computational techniques [103].

This model predicts cell specific enhancer activities and also the impact of SNPs on binding without ChIP-Seq data. KEGRU a bidirectional Gated Recurrent Unit (GRU) architecture for automatic feature extraction and classification. Initially, DNA sequences are split into k-mers of predefined length with slide window. Each k-mer is converted to a vector by using word2vec algorithm then passed as input to the model for prediction. DeeperBind [104] is a hybrid architecture (LSTM+CNN) for prediction of protein binding specificities concerning to variable length DNA probes. DeepGRN [105] model combines CNN and RNN with attention mechanism to predict Mutations TFBS on the ENCODE-DREAM in vivo challenge datasets. iDeepE [106] combines global and local CNNs to interpret variable length RNA-protein binding sites. DanQ [85] a convolutional and recurrent deep neural network model capture long term dependencies between sequence patterns to comprehend underlying semantics to improve prediction. FactorNet [81] is a hybrid model which leverages on variety of features like genome annotations, expressions and signal data to computationally attribute missing binding sites. The summary of state-of-the-art deep learning methods are shown in Table 2.4.

Although CNN and RNN methods have shown reasonable performance for classification and prediction problems, most of them show a limited degree of interpretability as deep neural networks are criticized for their black-box nature. The reasoning power of the existing models is very limited, so there is significant room to achieve high prediction performance and to improve the reasoning capability. A relatively large clinical dataset from 380 Covid-19 diagnosed patients was used to train/test the models. Evaluating a series of conventional classifers for predicting outcomes using patients' clinical data only, and investigating strategies to select a set of proper clinical labels from the pool of clinical data for the classifcation of imbalance data. An optimal data pre-processing is a critical initial step, prior to the initiation of training process, with possible boosting impacts on the overall performance of a model. A variety of pre-processing strategies can be chosen based on the type of data and/or algorithms used.

Table 2.3: Summary of selected deep learning based methods for Covid-19/Mutations and viral sequence prediction tasks.

Article's Reference	**Classification/Prediction**	**Technique**	**Performance Measures**	**Datasets Used**	**Result**
Zuallaert et al. [4]	Covid-19	CNN	AUC-PR, Sensitivity, Specificity, Accuracy, AUC-ROC	(GWH), (NN269) SARS-CoV-2/ 2019-nCoV	(0.5960/0.6194), (0.9077, 0.9739, 0.9612, 0.9899/0.9011, 0.9674, 0.9535, 0.9829)
Xu et al. [107]	Covid-19	Sparse Auto-Encoder	Accuracy, Mcc, Sensitivity, Specificity, AUC-ROC	Human splicing Acceptor/ Donor sites	91.11, 82.24, 90.14, 92.11, 96.28/ 90.56, 81.56, 90.09, 91.04, 95.66
Tayara et al. [96]	Covid-19	CNN	Accuracy, Mcc, Sensitivity, Specificity, AUC-ROC	Human splicing 2019-nCoV Acceptor/ Donor sites	93.57, 87.19, 95.16, 91.94, 98.11/ 96.66, 93.32, 97.23, 96.06, 99.26
Naito et al. [97]	Covid-19 Splice sites	DNN	Sensitivity, Specificity, Global Accuracy, Mcc	HS^3D Acceptor/ Donor	96.42, 93.34, 94.57, 89.87/ 97.88, 95.36, 96.27, 93.33
Du et al. [98]	Covid-19 Splice sites	CNN	(AUC-ROC, AUC-PR)	HS^3D, CE, NN269 Acceptor/ Donor	(98.79, 94.28), (99.56, 98.18), (99.34, 97.32)/ (99.02, 95.93), (99.47, 97.88), (98.43, 93.97)
Wang et al. [99]	Covid-19 Splice sites	CNN	AUC-ROC, AUC-PR	DM Acceptor/ Donor	98.14, 96.34/ 98.66, 96.73
Albaradei et al. [100]	Covid-19 Splice sites	CNN	AUC-ROC	HS^3D, AT, DM, CE 2019-nCoV Acceptor/ Donor	98.69, 98.31,98.16, 99.49/ 99.10, 98.69, 96.56, 99.48
Fabija et al. [108]	Viral sequence	CNN	F1-score	Dengue, Hepatitis B&C, HIV-1, Influenza A	0.85 to 1.00
Tampuu et al. [109]	Viral sequence	CNN	AUC-ROC	Metagenomic assembled contigs	0.923
Alipanahi et al. [94]	DNA, RNA proteins	DNN	AUC-ROC	Raw DNA sequences	0.475 to 0.955
Ren et al. [110]	Viral sequence 2019-nCoV	CNN	AUC-ROC	metavirome	0.93 to 0.98
Chen et al. [111]	TFBS	Interpretable DNN	AUC-PR	ChIP-Sequence	0.344 to 0.883
Pan et al. [106]	RBP	Local and Global CNN	AUC-ROC	RBP-24 and RBP-47	0.758 to 0.979
Hassanzadeh et al. [112]]	DNA Covid-19	LSTM	AUC-ROC	PBM experiment data	0.90

2.4.1 Harris Hawks Optimization (HHO) Method

Harris Hawks Optimization (HHO) is a swarm optimization approach capable of handling a broad range of optimization problems[11]. HHO is a popular swarm based, gradient free optimization algorithm with several active and time-varying phases of exploration and exploitation. Whale Optimization Algorithm (WOA)[16] and the Grey wolf optimizer (GWO)[17]. This technique employs the flight patterns of hawks to produce (near)-optimal solutions, enhanced with feature selection, for challenging classification problems for covid-19 [113]. For solving feature selection, Feature Extraction problems, this study presents a hybrid binary version of Harris Hawks Optimization algorithm (HHO). The Harris hawk optimizer (HHO) [8] was an attempt to reach not only better performance but also low-cost and efficient operators within a new stochastic optimizer we focused on the Genome wide analysis and developments of the recent well established robust optimizer Harris hawk optimizer (HHO) [10] as one of the most popular swarm-based techniques. Harris hawks optimization technique is to reduce the required computational cost while maintaining optimal outcomes. Harris hawks optimization (HHO) using Harris' hawk's behavior is based on the coordinated behavior and hunting of Harris hawks in nature. This algorithm is a new type of hunting and hunter algorithm. The HHO algorithm is still relatively new and has not been tested sufficiently on real-world problems. In this research, therefore, it is applied to the multilevel image segmentation of chest images of COVID-19 patients. Its segmentation results are then analyzed and compared against those obtained by the HHO method. The authors in this research claim that the use of metaheuristic algorithms in image segmentation domain lowers the amount of computations required to locate the best threshold configuration. Harris hawk's optimization is a population-based swarm intelligence algorithm. It mimics the hunting strategy of Harris hawks, which is mathematically modeled to address different optimization problems.

Table 2.4: Summary of selected Harris Hawks Optimization based methods for Covid-19 viral genome sequence prediction tasks.

Article's Reference	Classification/Prediction	Technique	Performance Measures	Datasets Used	Result
Zuallaert et al. [114]	Covid-19	HHO	AUC-PR, Sensitivity, Specificity, Accuracy, Recall Precision	, Covid19+/Covid-19-ve	(87.860/60.6194), (5077, 88739, 70.12, 80.88/70.9011, 60.96, 80.95, 90.98)
liao et al. [115]	Tandem Repeats	Covid-19 HHO	Accuracy, Mcc, Sensitivity, Specificity, AUC-ROC	SARS-CoV-2	81.11, 72.24, 60.14, 62.11, 96.28/ 90.56, 81.56, 90.09, 91.04, 95.66
ffazal et al. [116]	Tandem Repeats	CNN	Accuracy, Mcc, Sensitivity, Specificity, AUC-ROC	Alpha-CoV/ MERS-CoV	63.47, 74.29, 85.26, 81.44, 78.91/ 86.76, 83.32, 86.33, 86.06, 86.16
An et al. [117]	A tandem-repeat dimeric RBD protein-based covid-19	DNN	Sensitivity, Specificity, Global Accuracy, Mcc	HS^3D Acceptor/ Donor	96.42, 83.44, 74.67, 88.86/ 87.85, 85.26, 86.37, 73.23
Du et al. [118]	A deep population reference panel of tandem repeat variation	CNN	(AUC-ROC, AUC-PR)	HS^3D, CE, / NN269	(98.79, 94.28), (99.56, 98.18), (99.34, 97.32)/ (99.02, 95.93), (99.47, 97.88), (98.43, 93.97)
Khalid. [119]	Tandem Repeats Feature extraction	RNN	(Accuracy, -Precision)	HS^3D, CE, / NN269	(65.55, 71.54), (89.16, 88.21), (79.14, 77.22)/ (89.01, 76.13), (89.15, 91.33), (88.00, 91.22)
fang et al. [120]	DeepRepeat	RNN	AUC-ROC	HS^3D, HKU1-CoV NL63-CoV/	98.69, 98.31,98.16, 99.49/ 99.10, 98.69, 96.56, 99.48
arslan2 et al. [121]	Viral sequence	DQNN	F1-score	Dengue, Hepatitis B&C, MERS-COV, Influenza A	0.75 to 1.00
Tampuu et al. [122]	DeepSymmetry: using 3D convolutional networks for identification of tandem repeats	ML	AUC-ROC	NL63, Merscov-2	0.623
li et al. [115]	DNA, RNA proteins	HHO	AUC-ROC	Raw DNA sequences	0.875 to 0.855
Hanaka et al. [123]	Genome-wide identification of tandem repeats associated with splicing	RF	AUC-ROC	metavirome	0.83 to 0.88
Don et al. [119]	Tandem repeats with interrupt	Lstm	Accuracy PR	HKU1-CoV, 229E-CoV	0.143 to 0.282
Pan et al. [124]Tandem Repeats in DNA Sequence	RBP	Local and Global CNN	AUC-ROC	RBP-24 and RBP-47	0.758 to 0.979
Hassanzadeh et al. [125]	Tandem Repeat Polymorphisms	LSTM	Accuracy	Precision experiment data	0.70

2.5 Summary

In this chapter, some of the existing works related DNA sequence clustering, various regulatory signals, pattern frequency based disease prediction, viral genome prediction, and translation, mutation rate, evolutionary relationships of corona viruses are discussed. Most of the existing alignment based techniques handles only small sequences of DNA data and are time consuming. To overcome the limitations in alignment based approach a few alignment free methods are introduced. A survey on different alignment free methods have been presented. Moreover, an exhaustive survey on probabilistic and machine learning techniques is performed. Discussed how the performance of machine learning algorithms purely depends on the constructed features. Further, a survey on state-of-the-art deep learning techniques for various sequence analysis tasks, and the limitations of those techniques have been included. Further, a survey on state-of-the-art deep learning techniques for various sequence analysis tasks, drug prediction tasks, Covid-19 and the limitations of those techniques have been included. For the purpose of classification, a multi-class SVM classifier has been considered, as SVM is the most widely classifier in the field of ML and DNA pattern recognition. The number of training DNA pattern, batch size, DNA pattern size, learning rate and number of model parameters have a significant impact on the performance of a neural network. We done the work on Covid-19 Classification, Point Mutation rates in covid-19, top Motif's feature extraction and feature selection, Repeats in covid-19 were identified.

Chapter 3

Exploring Coronavirus Sequence Motifs through Convolutional Neural Networks for Accurate Identification of Covid-19

In order to accurately identify Covid-19, we provide in this chapter a broad and robust CNN model called DeepCoV. Our CNN-based model successfully captures significant patterns of human coronaviruses by utilising convolution and weight-sharing techniques. Differentiating new coronaviruses from existing ones and finding important patterns for accurate identification of new coronaviruses are the two main goals of the suggested method. The following is a synopsis of the work's main contributions.

1. It is a convolutional neural network (CNN) architecture comprising two fully connected layers, three sequential convolutional layers, and a max-pooling layer. In order to reduce noise, this layer-wise learning.

2. Fully linked layers resolve corona prediction by pattern interactions, whereas convolutional and pooling techniques find predictive patterns from corona virus sequences.

3. The suggested model is able to detect distinct patterns associated with coronaviruses by employing a collection of learnt filters of convolutional layer. We can learn more about the basics of coronavirus reproduction from these patterns that were retrieved.

3.1 Materials and Methods

In order to accurately identify COVID-19, the suggested method relies on a convolutional neural network (CNN) model that necessitates a large dataset of samples. The 2019 Novel Coronavirus Resource (2019nCoVR) maintained by the National Centre for Bioinformation in China is a major repository for various coronaviruses [126].

3.1.1 Dataset Collection

Results for this study were culled from datasets sourced from a number of databases, such as CNCB/NGDC, GISAID, NCBI, and NMDC. Covid-19 is fully integrated with all of these data centres. It also provides visualisation tools for the results of genome variation analyses using all of the collected Covid-19 strains and compiles a wide variety of relevant material for scientific dissemination, including scientific literatures, news, and popular pieces. This collection contains many human coronaviruses, such as Sars-cov-2, NL63-CoV, HKU1-CoV, AlphaCoV, BetaCoV-1, MERS-CoV, and 229E-CoV. COVID-19 negative sequences include HKU1-CoV, AlphaCoV, BetaCoV-1, MERS-CoV, and 229E-CoV, whereas COVID-19 positive sequences comprise SARS-CoV-2. We utilised 592 genome sequences from different human coronaviruses in our investigation, together with the 1000 SARS-CoV-2 sequences. Our focus is on the fact that, unlike SARS-CoV-2, all human coronavirus genome sequences are freely available for download.

The datasets $\mathbb{S}$ of coroaviruses can be formulated as below

$$\begin{cases} \mathbb{S} = \mathbb{S}^{\mathbb{P}} \cup \mathbb{S}^{\mathbb{N}} \\ \mathbb{S}^{\mathbb{P}} = \mathbb{S}^{P}_{Cov-2} \\ \mathbb{S}^{\mathbb{N}} = \mathbb{S}^{N}_{AlphaCoV} \cup \mathbb{S}^{N}_{BetaCoV-1} \cup \mathbb{S}^{N}_{MERS-CoV} \cup \mathbb{S}^{N}_{NL63-CoV} \\ \cup \mathbb{S}^{N}_{HKU1-CoV} \cup \mathbb{S}^{N}_{229E-CoV} \end{cases} \tag{3.1}$$

Table 3.1: The characteristics of human coronavirus whole genome sequences

S.No.	Types of human CoVs	The number sequences	Label
1	SARS-CoV-2	1000	1
2	Alpha-CoV	888	0
3	BetaCoV-1	640	0
4	MERS-CoV	341	0
5	NL63-CoV	654	0
6	HKU1-CoV	745	0
7	229E-CoV	247	0

where $\mathbb{S}^{\mathbb{P}}$, $\mathbb{S}^{\mathbb{N}}$ are COVID-19 positive and negative samples of all coronavirus datasets. Table 6.7 shows the characteristics of various coronaviruses. The majority of coronaviruses are negative (0), while the main one, SARS-CoV-2, is seen as positive (1). Preprocessing was applied to the acquired data to ensure that only high-quality data was utilised. We wrote some Python scripts to filter out genomic sequences sharing an accession number in order to get rid of duplicates. In order to preserve the genetic signature encoded in dinucleotide frequencies, sequences that contained any nucleotides other than A, T, C, and G were excluded from consideration for each species.

3.1.2 Problem Definition

The SARS-CoV-2 gene sequence motif determination and coronavirus gene localization can be formalised as a two-class classification problem. Binary (two-class) or multi-class classification problems are the most common ways to express prediction challenges in bioinformatics and computational biology. In order to construct an effective bioinformatics classifier, one must be competent in designing biological sequence problems. Here, identification of SARS-CoV-2 can be established as a binary classification problem, with dataset $\mathbb{D}$ of N samples $\mathbb{D}$ = $\{x_i, y_i\}_{i=1}^{N}$, x_i indicates feature set, that could be considered as a 4 X N dimensional matrix. DNA sequences contain four bases: Adenine (A), Guanine

(G), Cytosine (C) and Thymine (T). A, G, C, T is the sequence that these four base pairs form. The one-hot vectors [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], and [0, 0, 0, 1] can be used to represent these base pairs, in that order. For all sequences, the class label is 0, except for SARS-CoV-2, which has a value of 1. Due to their similarity to other Coronavirus sequences, conventional examination of SARS-CoV-2 sequences may yield erroneous results. Among the several Coronavirus gene sequences, the primary goal of this study is to accurately predict the SARS-CoV-2 gene sequence. Additionally, we discovered SARS-CoV-2 sequence-induced common patterns.

3.1.3 Convolutional Neural Networks

Every typical convolutional neural network (CNN) consists of four layers: a convolutional layer, a non-linearity layer, a max-pooling layer, and a fully connected layer [127]. Convolutional neural networks (CNNs) have proven to be highly effective in picture categorization, computer vision, and NLP [128]. Topic categorization and sentiment analysis are only two examples of the text-related problems that have found solutions with their help. Genetic sequences display patterns of sequential letters that are not separated from one another, in contrast to text data that contains gaps between words. The words that comprise these sequences are formed by combining the four nucleotides A, G, C, and T. As mentioned in Ref., one-hot vectors can be transformed into 2D matrices and used to describe DNA sequences. In this case, the DNA sequences are characterised as 2D matrices using a CNN layer [129]. One compelling reason for including CNNs in proposed DeepCoV, is that they are quick and effective at representing text or sequences [130]. As a result, we use a deep learning algorithm CNN, to classify SARS-CoV-2 genes and other Coronavirus genes.

3.2 Proposed Apporach

A fully linked layer, a pooling (down-sampling) layer, and an output layer follow each convolution in DeepCoV. When training, the proposed architecture takes as input a raw binary matrix-embedded DNA sequence with corresponding A, G, C, and T positions that is N size-long and NxN sized. One hot encoding method is used to transform the text into numerical information because neural networks can only process numerical data and the input is sequences of nucleotides [94].

A one-dimensional convolutional layer including thirty-two equally sized learnable filters, each seven units wide, constitutes the top layer of the architecture. In order to find patterns in the input one hot encoded matrix, each layer filter uses a convolution operation similar to the sliding window. The approach involves mapping each sub-matrix at every nucleotide position in order to detect motifs around the target SARS-CoV-2 sequences. Motifs are commonly found subsequences in sequences and are often crucial in identifying authentic and fake Covid-19 sequences. The activation function known as the Rectifier Linear Unit (ReLu) is used by the convolutional layer. If the activation value x is greater than 0, the relevant motifs from the current position have progressed to the next level in the convolutional layer, resolving the issue of vanishing gradients. This is indicated by the activation function ReLu, f(x) = Max(0,x). The connected theme is removed as uninteresting if its value is 0. As the positive number grows, so does the probability that the presented sequence is a genuine Covid-19 sequence.

We utilised the 100-nucleotide sequence length of the SARS-CoV-2 dataset as an example; this methodology is also applied to the sequences of other Corona viruses. After receiving the 100 x 4 input matrix with binary encoding, the first 1D convolutional layer sends it to the 7-size feature detector/filter within the layer. With just one filter to train on, the neural network can only understand a single feature. We need to specify 32 feature detectors (filters) if we want to gather 32 distinct features. With N being the padded sequence length, K the length of the filters, F the total number of filters, and P the padding, the convolutional layer generates a unique matrix of size (N - K + 2P +1) X F. With a three-padding P, the window iteratively traverses the data for 100 steps, yielding an output matrix of size 100 X 32 that is in perfect correspondence with the input matrix.

After removing half of the values from the previous layer, the pooling layer replaces them with the maximum value and moves a 2-length window over the encoded sequence, producing 50 X 32. By combining smaller samples, we can detect more interesting features and cut the hidden layers in half. Additional precaution against overfitting is taken

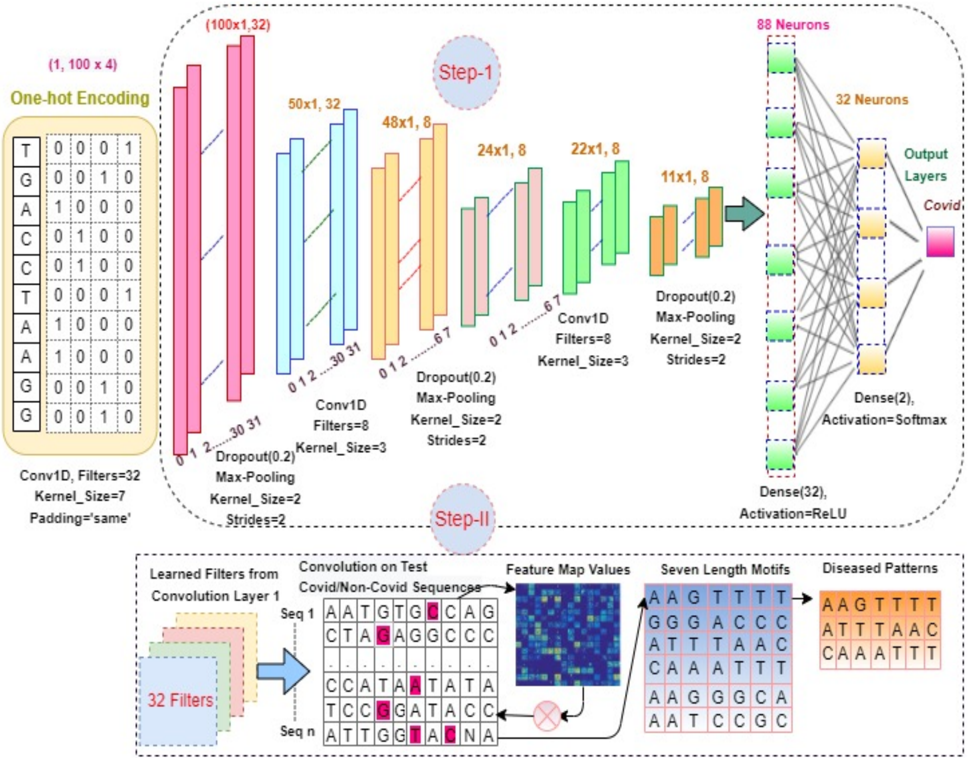


Figure 3.1: DeepCov Architecture: Step-I: showing stacked CNN layers with Maxpooling and dense layers. Step-II: Shows the procedure to extract diseased patterns from learned convolutional networks

by including a dropout layer with a 0.2 probability. This leads to the loss of 20The convolution operation is a vital step in CNN. The first convolutional layer convolves the one-hot encoded input with 32 filters which are slide across the input genome. The filter of size 7 is stride one position at a time and the padding is set to be as 'same' to preserve the actual size(300) of the input. These learned filters used to identify the particular patterns as features in the DNA sequence. In each convolution operation, the encoded input genome convolves with a number of k filters F={f_1,f_2, ... f_K}, and biases B = {b_1,b_2, ... b_K} are added, and each filter generates separate feature map M_k^l[131].

$$M_k^l = b_k^{l-1} + \sum_{i=1}^{N_{l-1}} (f_{ik}^{l-1} \circledast M_{ik}^{l-1}) \tag{3.2}$$

where f_{ik}^{l-1} is learned filter weights at previous layer l-1, M_k^l is the value after convolution operation. The non-linear activation transformation $\sigma(.)$ is applied to feature maps and the same process repeated to all convolution layers.

$$Y_k^l = \sigma(M_k^l) \tag{3.3}$$

Equation (2) provides the ReLU activation, which is used element-wise to build feature maps after the convolution layer using the max(0,z) operation. “A 0.2-dropout-rate dropout layer precedes the ReLU activation layers; this layer provides regularisation and reduces over-fitting by randomly dropping 20

$$Z_k = max(Y_{1,k}, Y_{2,k}, ..., Y_{n,k}) \tag{3.4}$$

The initial convolutional layer, in conjunction with the max-pooling and dropout layers, is responsible for extracting the global characteristics. Similar to the first convolution, the second and third layers are supplemented by max-pooling and dropout layers that extract local features in the same sequence. In the table 3.2, the detailed structure of the proposed model is shown.

Table 3.2: The detailed model structure of the proposed approach.

Step	Operation	Output Dimension
Input Layer	One-hot encoding	300x5
Convolutional Layer 1	Conv1D(32,7)	300 x 32
	Activation(ReLU)	300x32
	Dropout(0.2)	300x32
	Max-pooling1	150 x 32
Convolutional Layer 2	Conv1D(8,4)	147 x 8
	Activation(ReLU)	147 x 8
	Dropout(0.2)	147 x 8
	Max-pooling1	73 x 8
Convolutional Layer 3	Conv1D(8,3)	71 x 8
	Activation(ReLU)	71 x 8
	Dropout(0.2)	71 x 8
	Max-pooling1	35 x 8
Flatten step	Flatten	280 x 1
Dense Layer1	Dense(32)	32 x 1
	Activation(ReLU)	32 x 1
	Dropout(0.2)	32 x 1
Dense Layer2	Dense(2)	32 x1
	Activation(Softmax)	2x1
Output Layer	Classification	Probabilities

The model’s classifier receives the output of the final pooling layer after all stacking layers have finished processing it and converted it to a dimensional vector. This section consists of two thick layers, the first of which has 32 neurons and the second of which contains 2 neurons. Two thick layers are separated by a dropout layer. The softmax activation function, which is present in the final dense layer, generates two probabilities: one for the true target classes and one for the erroneous target classes. The following is a mathematical representation of a softmax function:

$$S(Z) = \frac{e^{Z_i}}{\sum_i e^{Z_i}} \tag{3.5}$$

Finally, the genome sequence is classified as viral/non-viral type based on the output probability. The categorical-cross entropy loss function is given in the equation(1). After every epoch the filter weights are updated to minimize the loss function.We used Keras [132], a minimalistic, highly modular neural network library, written in Python, in our implementation of the network.

Two more 1D convolutional layers, max-pooling, and dropout are added so the model can understand more complex features. In the first convolutional layer, there are eight filters with four-unit kernel sizes; in the second and last convolutional layers, there are eight filters with three-unit kernel sizes. To decrease the likelihood of overfitting, a two-layer pooling layer is added after each convolutional layer with a dropout probability of 0.2. Upon completion of all layer blocks, the final CNN features maps are smoothed out. The novel coronaviruses are distinguished from other coronaviruses using a probability distribution across binary output classes that are generated by a softmax classifier using all the integrated information.

3.2.0.1 Hyper parameter tuning and summary of output parameters

To prevent overfitting caused by an exploding gradient problem, the dropout layers are employed. Just like with other parameters, the dropout rate is found by running hyperparameter optimisation. We thought of employing a random search technique to choose an optimal set of model hyper parameters instead of manually looking for them. An optimisation process includes defining a search space. The goal of optimisation is to select a vector that provides the best possible performance to the model, in terms of accuracy or error rate, for that particular model. By use of random sampling, we were able to optimise the following two convolutional layer parameters: batch size, optimisation method, dropout, number of fully connected layers, number of filters, and filter length. We tweaked the settings for the third convolutional layer by hand. All dropout layers were fine-tuned to a single rate throughout the random search. The model validation set was used to determine the optimal set of parameters after numerous combinations were considered. The hyper parameters search space for first and second convolution layers are listed in Table 3.2, and the parameters which are the best set found. The Table 3.2 displays the many parameters and dimensions that were utilised to train the proposed DeepCoV model. It provides a concise overview of the model training process's inputs and results. It further demonstrates the layer-to-layer transfer of these vectors together with many other factors such as activation units, dropouts, and maxpooling layers.

3.2.1 K-Fold Cross Validation

We used a cross validation technique to verify that the DeepCoV model accurately predicts the SARS-CoV-2. By utilising a cross validation approach, issues such as over fitting and selection bias can be mitigated, and valuable information about the proposed model's ability to generalise to unknown data can be gleaned. For *k* fold cross validation the whole dataset is partitioned into *k* complimentary subsets, performing the analysis on one subset called as testing set and training other *(k-1)* subsets. The results of the validation are averaged over k separate tests and sets of trials. To achieve a balanced bias-variance trade-off, the value of k must be carefully chosen. The researchers in this study estimated the levels of bias and moderate variation by experimental means using 10-fold cross validation.

3.3 Results Analysis

The performance and discriminative capabilities of the DeepCoV model are compared to existing benchmark machine learning models in this section [133]. We have already examined CNN architecture at different levels in order to explore the effects of different designs on network analysis. We started with a simple model with one (Convolution + Pooling) and two fully linked layers. We then added more layers to increase its complexity and tested it at four different depths to see how well it performed. At 3* (Convolution + Pooling) depth, we achieved the best results for the datasets that were evaluated for the experiment.

3.4 Experimental Setup

We utilised a desktop computer with a Core i7 3.2 GHz CPU, 16 GB of RAM, and a GeForce GTX Titan X GPU with 12 GB of DDR5 RAM to conduct the trials for the suggested model. Dataset overviews, data pre-processing, cross-validation, and training parameters for the proposed model are all found in this section.

3.4.1 One hot encoding

One hot encoding technique is used to convert DNA sequence to numerical vector as neural networks handle only numerical data. Specifically, consider a DNA sequence S with n bases S = $\{b_1, b_2, b_3, ..., b_n\}$, $S_i \epsilon$ {A, C, G, T, N}. The encoded vector is stored as an array (M) of size nx5 as the following:

$$A_{i,j} = \begin{cases} 1, & \text{if } S_{i-1} = j^{th}\text{base in (A, C, G, T, N).} \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

The number of rows is equal to the length of the DNA sequence and the columns are equal to unique number (five) of bases.

3.4.2 Performance Measures

When it comes to binary classification, the most popular model evaluation parameter, accuracy, could be misleading. When used in isolation from other performance metrics when dealing with imbalanced data. Consequently, classifiers may be biassed towards the dominant class, rendering classification ineffective. Consequently, the suggested deep learning model is evaluated and compared using the following metrics produced from a confusion matrix: An often-used metric for two-class classification problems, the area under the receiver operating characteristic (AUC) curve, is used in the proposed model's performance evaluation. Finding the optimal threshold values using these curves is a well-known capability. However, the datasets we have are biassed since SARS-CoV2 and other coronaviruses are not evenly distributed. Addressing such an imbalanced problem is better illustrated using AUC PR curves [134]. Since there is a higher chance of biassed performance evaluation in the event of an imbalanced task, the precision-recall curve is added. The AUC ROC curve displays the TPR and FPR, or True Positive Rate and False Positive Rate, respectively. The True Positive Rate (TPR) is the proportion of true positives found from all candidate positives, whereas the False Positive Rate (FPR) is the proportion of false negatives that are incorrectly classified as true positives. On the AUC PR curve against TPR, you can see precision, which is the proportion of correctly predicted positive classes to total positive classes. The equations for these various measurements are as follows.

$$Acc = \frac{TP + TN}{TP + FN + FP + FN} \tag{3.7}$$

$$FPR = \frac{(FP)}{(TN + FP)} \tag{3.8}$$

$$TPR/Recall = \frac{(TP)}{(TP + FN)} \tag{3.9}$$

$$Precision = \frac{(TP)}{(TP + FP)} \tag{3.10}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.11}$$

The FP, FN, TP, and TN values indicate the number of false positive, false negative cases and True positive and True negative cases of whether a given sequence belongs to novel Coronavirus or not.

3.4.3 DeepCoV Model Interpretablity

There are two main goals that this study aims to achieve. To start, we check if a sequence is SARS-CoV2 by using stacked convolutional neural network (CNN) layers with Maxpooling and dense layers. To get sick patterns out of learnt Convolutional Networks, the second step is to follow a process. Many problems with biological sequence categorization have been solved in recent years by using Machine Learning approaches with complex internal implementations. Classification efficiency has been the subject of much research and development. However, before a computational model can be accepted, its classification accuracy must be carefully verified, and the user must have a good grasp of the principles behind it. While the revised prediction method achieves far higher accuracy, the model interpretability supporting the algorithm's prediction is noticeably lacking. The ability to understand the correct mathematical structure (how a mathematical model achieves classification) and conduct some downstream analysis (which biological processes are triggered by what genomic features hidden in biological sequences) is what makes model interpretability so important. Attempts to establish a connection to biological differences are thwarted by the computer models' opaque machine learning methods and the model's complex learned decision rules, which are difficult to understand. Here, we used the activations and feature map values of the filters to extract SARS-CoV2 illness patterns from the learnt convolution networks.

3.4.3.1 Procedure to extract Motifs

We trained the proposed CNN module on the Coronavirus dataset (included in the dataset section) to detect underlying motifs that are positively associated with having coronavirus sequence motifs. The architecture of this module is the same as that of the part before it refer Figure 3.1. One key difference is that this model incorporates all sequences from the dataset into its convolutional and rectification stages. A trained convolutional neural network (CNN) works by first identifying which subsequence fragments were most activated by the first convolutional layer and then using a position weight matrix (PWM) metric to extract a motif from these fragments. For the purpose of identifying possible local subsequence properties, every L-length convolutional filter is matched exactly against all possible 4 X L one-hot encoded input sub-matrices at every point. It is acknowledged as a measure of the positional subsequence's contributions to exercise a prediction function, since a high value determined by Relu indicates a strong contribution to being a true novel coronavirus sequence.

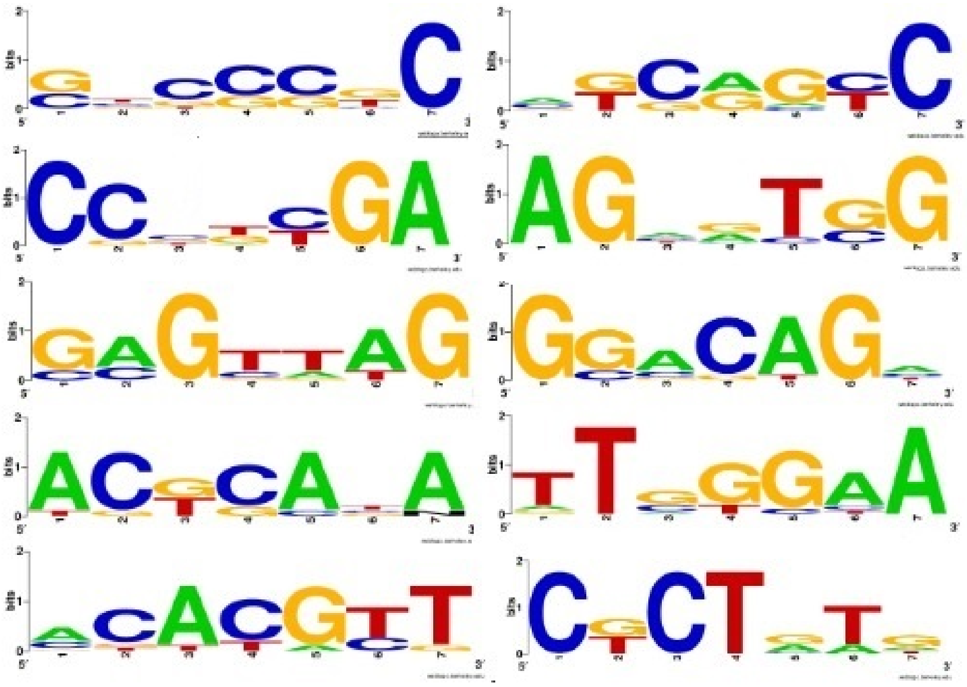


Figure 3.2: Top Ten Motifs generated from learned convolutional filters.

3.4.3.2 Motif discovery for Covid-19

Table 3.3: Patterns that motivate Covid-19 for AlphaCoV, BetaCoV, MERS-CoV-2, NL63-CoV HKU1-CoV, 229E-CoV.

TSS Pattern	Repeated in No. of Filters	Filter Numbers	Average Activity Value	FSS Pattern	Repeated in No. of Filters	Filter Numbers	Average Activity Value
CCCAGGG	30	1-2,4-32	0.2426	GAGGGGG	27	1-6,8-30	0.2881
CCCAGCT	29	1-2,4-29,31	0.1295	CAGGGAG	25	2,8-12,14-32	0.2363
CTGCAGA	18	16-17,21-32	0.1728	CCTGAGC	15	12-15,17-24,30-32	0.1499
TCCAGGG	13	5-11,13-15,18-20	0.1400	AGGTGGG	11	4-8,11,13-17	0.2674
CCCAGGC	12	5-10,24-27,30,32	0.1964	TGCCCAG	10	1-3,5-11	0.2603
CCCAGGA	12	12-23	0.1432	GCTGCAG	10	7-8,20,25-29,31-32	0.2189
*CAGGTGG	22	11-32	0.2061	CCTGCAG	10	19,23,25-32	0.1977

*Neutral Patterns, present in both true and false splice site genome sequences.

Take the pattern CCCAGGG as an example; it appears 64 times and has been duplicated in 31 filters (1-2, 4-32). Based on their recurrence in various filters, the motifs are arranged in descending order. True and false Covid-19 share the same themes (CAGGTGG, GTGAGTG), but we disregard them since they have no predictive power. Figure shows that the interpretable CNN framework Inter Covid-19 can predict true and false sites by extracting high-level information. 3.2. Table 3.4 displays the patterns retrieved from the Covid-19 TSS and FSS datasets using the second approach. It displays the most common

Table 3.4: Top-ten filters activation means with motif frequency for Covid-19 TSS and FSS datasets.

Rank	**Covid-19 True dataset**				**Covid-19 False dataset**			
	Filter Number	Filter Mean Activation Value	Pattern	Pattern Frequency	Filter Number	Filter Mean Activation Value	Pattern	Pattern Frequency
1	8	0.3688	GGTGAGT	82	30	0.3416	GTGAGTG	266
2	6	0.3366	GCAGGTG	32	20	0.3274	GTGAGTG	206
3	7	0.3342	GGTGAGT	83	26	0.3269	GTGAGTG	266
4	2	0.3252	GGTGAGT	54	7	0.3266	GGTGGGT	266
5	21	0.3218	GGTGAGT	151	5	0.3257	GTGGGTG	67
6	32	0.3210	GGTGAGT	207	14	0.3238	GGTGGGT	144
7	5	0.3186	GGTGAGT	81	28	0.3226	GTGAGTG	266
8	6	0.2090	GGTGAGT	81	27	0.3158	GTGAGTG	266
9	15	0.3064	GGTGAGT	154	9	0.3126	GTGAGTG	266
10	14	0.1899	GGTGAGT	112	21	0.3126	GTGAGTG	206

pattern from each filter, which is one out of 10 that are extremely similar between TSS and FSS. The fact certain motifs appear with the greatest activation value in nearly all filters is evidence of their redundancy. If a motif is redundant, it means the motif is very important for determining the real and false splice locations. If the activation value of the subsequence at a specific site (ReLU = max(0; x)) exceeds a threshold for all subsequences and positions, then the subsequence subunit is kept. In this research, the cutoff is 50

3.4.3.3 Motifs Filter Analysis

Using DeepCoV, we may not only analyse motifs, but also test the limits of the learnt convolve filters in the first convolutional layer. The function of the filter is to recognise motifs. From the local sequence context, it can learn SARS-CoV2 illness patterns and identify candidate motifs. One measure of a filter's ability to detect relevant motifs is the frequency with which they appear in sequence windows; this is also known as filter activity. One way to measure the filter activity value for a group of sequences is to take the average of the mean activations. First, we activate every consecutive sequence by overlapping each k X 4 filter on L - k +1 subsequences for one sequence, where L is the length of the sequence and K is the length of the filter. The average of all subsequences is then used to determine the final activity value.

The capacity of the filter and the influence of the motif on selecting diseased patterns increase with increasing filter activity value. The variation of each filter activity, which

Table 3.5: Mean/Highest activation for top ten filters of the first convolutional layer derived from DeepCoV model

Rank	Filter	Motifs	Mean Activation	Highest Activation
1	20	CCGTTGA	0.047075363	1.0865947
2	2	CTCCCGC	0.110824939	0.9866006
3	23	GAGTTAG	0.034365958	0.8609414
4	31	ACGCATA	0.077731049	0.8048915
5	4	ACACGTT	0.032601854	0.8019278
6	24	GTCGGCC	0.053501801	0.75201905
7	18	AGAGTCG	0.019740433	0.73908246
8	19	GGACAGC	0.052045315	0.7305308
9	25	TTGGGAA	0.029710728	0.7282068
10	8	CGCTGTG	0.030981092	0.72371507

exemplifies the activation fluctuations range for all sequences, is also calculated. Table 3.5 lists the top ten activities, with the 20th filter's highest activity value on the Coronavirus dataset being around 0.0470. Overall, mean and greatest filter activity both drop at the same time. The activation values for the 20th filter's entire subsequences are less than 0.047075363 and close to 1.086 is the greatest activation value.

3.4.4 Performance Analysis of DeepCoV model with other existing benchmark methods

The DeepCoV model's capability was compared to that of known baseline machine learning models like SVMs, Naive Bayes, K-NN, and Random Forest methods that classify Covid-19 sequences by utilizing various methods for feature extraction proposed by Hilal Arslan [133]. The selection of discrete features is an important step in improving recognition accuracy based on the properties of the COVID-19 virus. Arslan [133] proposed the use of CpG island features [135] based on the assumption that SARS-CoV-2 has a inviable vacancy of CpG [136].

For a proper comparison, the DeepCoV method is subjected to 10-fold cross-validation. The outcomes demonstrate that, across all performance criteria, the proposed solution has outperformed competing approaches. Table 3.6 shows, Precision, Recall, F-Measure, Ac-

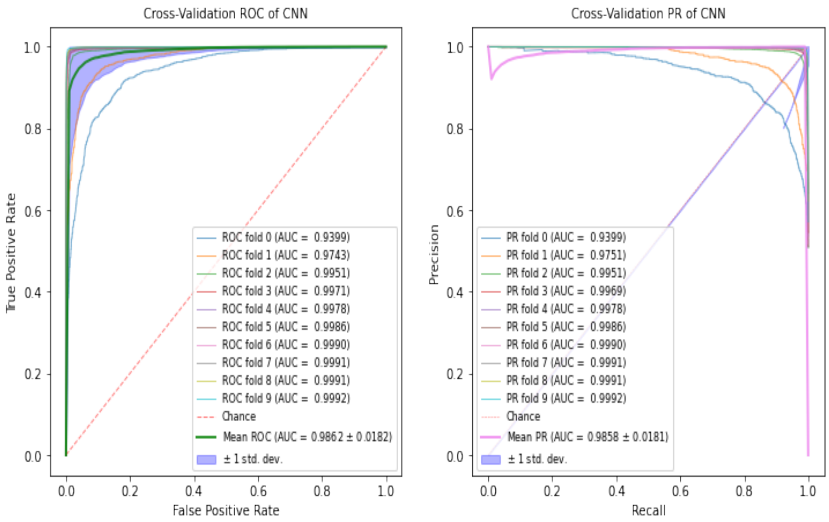


Figure 3.3: Ten fold average AUC ROC and AUC PR values of DeepCoV model.

Table 3.6: Results Comparison with Existing techniques

Method	Precision	Recall	F-Measure	Accuracy
Support Vector Machine	0.869	0.873	0.868	0.87
Naive Bayes	0.882	0.879	0.87	0.88
K-Nearest Neighbor	0.927	0.926	0.926	0.92
Random Forest	0.93	0.90	0.91	0.93
(DeepCoV) (Proposed Method)	**0.98**	**0.97**	**0.99**	**0.96**

curacy values of DeepCov methods in comparison with existing baseline methods. The proposed model has achieved an accuracy of 96% nearly showing an improvement of 5% when compared with Random Forest method [133]. As discussed earlier only considering accuracy as an evaluation metric can be deceptive, to overcome this issue we also evaluated AUC ROC and AUC PR which has given 98.62% and 98.58%, respectively shown in Figure 3.3 it demonstrates the DeepCoV model's ability to discriminate.

3.4.5 Filter ability and visualization

Filters are trained vector of weights, plays a crucial role in pattern (motif) detection for classification problems. Along with pattern analysis, we also determine the potential of the convolve filters by Deepcov-CNN in the first convolution layer. The visualization of CE acceptor and donor learned filters is shown in Figure 3.4, by using displayr. The X-axis of heatmaps shows the position of each nucleotide in learned filter weight matrix and Y-axis shows the nucleotides (A, C, G, T, N).

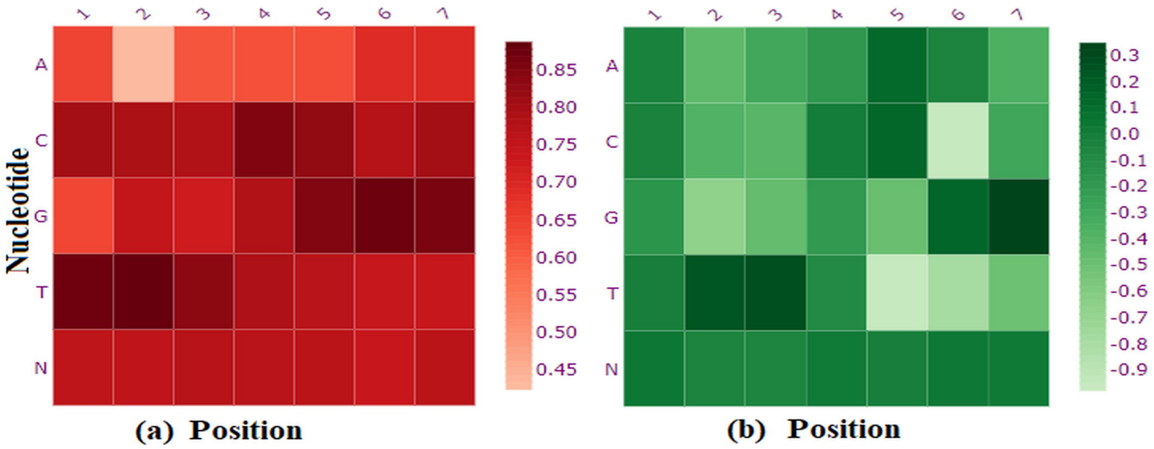


Figure 3.4: Graphical representation (Heat Map) of CE (a) Covid-19(TSS) (b) Covid-19(FSS) learned filters.

A filter's learning weights, displayed visually as a heatmap, indicate the relative relevance of nucleotides at each place in the filter. for that reason, the patterns are more important in establishing the veracity of the COVID-19. Darker shades of green and red indicate a higher concentration of that nucleotide at that specific location.

3.4.6 Summery

In this chapter, we presented DeepCoV, a general-purpose and robust CNN model for precise Covid-19 detection. Our convolutional neural network (CNN) model successfully identifies significant top pathogenic motif patterns in human coronaviruses by utilising weight sharing and convolution". To respond swiftly to a viral outbreak, like COVID-19, it is crucial to understand the genetic sequence of the virus, as discussed in this chapter. Finding new coronaviruses is difficult since SARS-CoV-2 and other coronaviruses are so similar. Comparing the similarities between the SARS-CoV-2 virus and other similar and well-known viruses is crucial for determining if a DNA sequence is of SARS-CoV-2 virus or not. In this research, we propose DeepCoV, an understandable model for accurate SARS-CoV-2 prediction using deep neural networks, to circumvent these shortcomings.

Unlike other models, ours uses a convolutional architecture to identify sequence motifs of biological significance by combining convolution and pooling processes. In order to better understand the regulatory mechanisms that SARS CoV-2 uses to control gene expression, DeepCoV finds motifs that identify whether a given sequence is SARS CoV-2 and also probable trends. There is further dispute regarding the interpretability of convolutional neural networks since they are said to be opaque.

Chapter 4

Coot-lion Optimized Deep Learning Algorithm for COVID-19 Point Mutation rate Prediction using Genome Sequences

Here, we introduce a model that uses genome sequences to forecast the spread of carbon virus type 19. Feature mining involves applying specific traits to genome sequences; these traits may include CpG-based features, numerical mapping (intger and binary), and numerical mapping (using the Fourier transform to create features for skewness, kurtosis, and peak-to-average power ratio). Diseases caused by the coronavirus, more often known as COVID-19, can range from the common cold to more severe respiratory illnesses. A newly identified coronavirus, COVID-19, has just emerged, causing pneumonia and other severe diseases. Nevertheless, in order to prevent health risks, it is critical to distinguish between the favourable possibilities as soon as possible.

4.1 Materials and Methods

To forecast the spread of COVID-19, this chapter makes use of a Deep Quantum Neural Network (DQNN) trained on the Lion-based Coot algorithm (LBCA-based Deep QNN), which accounts for genetic sequences. “Here, the genome sequences undergo feature extraction, which involves extracting specific features (such as CpG-based features) and numerical mapping (such as integer and binary) from the genome sequences. The numerical mapping is then applied using the Fourier transform to generate features (such as Peak to average power ratio, skewness, and kurtosis). Furthermore, the entropy feature is extracted using the K-mer extraction method. The Bray-Curtis distance and the Deep Belief Network (DBN) are used to accomplish the feature fusion. Finally, COVID-19 predictions are made using a deep quantum neural network. In order to train Deep QNN, the Lion-based Coot algorithm (LBCA) is utilised. The Coot algorithm and the Lion optimisation algorithm (LOA) are combined to produce the suggested LBCA. Point mutation is used for the examination of COVID-19 prediction. Testing accuracy was 0.941, True Positive Rate (TPR) was 0.931, and False Positive Rate (FPR) was 0.869, demonstrating that the proposed LBCA-based Deep QNN performed better than current prediction methods.

4.1.1 Motivation

Several models are proposed to anticipate the spread of COVID-19 by analysing genome sequences; nevertheless, these methods do not yield more accurate predictions.

4.1.2 Problem Definition

Diseases ranging from the common cold to severe respiratory ailments are caused by human pathogens known as COVID-19. Conventional approaches to vaccine development neglected to assess CpG motifs and similarity features. Due to the rapid mutation rate of SARS-CoV-2, genomic sequences are particularly useful for tracking coronavirus genes, which change constantly as the disease advances from person to person. We built a model for COVID-19 to forecast the pattern of disease using genomic sequences because early detection is critical for preventing the development of the disease.

4.1.3 Challenges

The issues confronted by priorly devised COVID-19 prediction models with genome sequences are listed below.

- In [121], a technique is devised for predicting COVID-19, considering the genome sequences. However, this technique did not examine the CpG motifs and similarity features for developing the vaccines.
- An AI-based model is utilized for learning the interesting data from the genome sequences of COVID-19. However, this technique failed to handle genome sequences of varying lengths [133].
- In [137], the DeepCOVID-19 identification pipeline is devised for predicting COVID-19 using the genome sequences. However, examining model efficacy for other critical bioinformatics tasks is not explored.
- The training of CNN and its accuracy depends on the datasets utilized. For more complicated data, the neural network can require going deep in order to discover more features to categorize precisely and fail to handle noisy data[138].
- Some studies have looked at the treatment for thoroughly examining the disease in view of COVID-19's rapid spread. The genetic closeness of COVID-19, however made the discovery process challenging.

4.2 Proposed approach

We laid out the suggested model and its assessment criteria here. This work aims to provide a model for predicting COVID-19 point mutations using DNA genome sequences. Here, the genome sequences undergo feature mining, which yields characteristics including CpG based features, integer and binary numerical mapping, and numerical mapping by applying Fourier transform to create skewness, kurtosis, and peak to average power ratio. Furthermore, the entropy feature is mined using the K-mer extraction. Using the Bray-Curtis distance and DBN, the features are fused. Before finishing, the COVID-19 forecast is made using Deep QNN. In order to train Deep QNNs, LBCA is used. The LBCA is created by merging the Coot algorithm with LOA. Examination of point mutations in the prediction of COVID-19 is carried out. The main contribution includes

- Proposed LBCA-based Deep QNN for COVID-19 prediction: In this work, an LBCA-based Deep QNN is used for predicting COVID-19, considering the DNA genome sequences to detect whether diseased is coivd-19 or not by using Deep QNN.
- Proposed LBCA: Deep QNN is trained with LBCA, which is formed by integrating with Point mutation rate is calculated for 200 and 400 genome sequence with the hekp of proposed model.

The viral infection caused many illnesses involving cancer and COVID-19. Dangerous illness develops when viruses infect cells and disrupt the host's normal process. The model's explainability is low, despite the fact that automated feature mining produces multiple strategies. The purpose is to present a method for COVID-19 prediction based on genomic sequences. Using hybrid optimisation and genomic sequences, we aim to provide a deep model for COVID-19 prediction. Features extracted from genomic sequences constitute the initial stage of the procedure. These features include CpG-based features [121] and numerical mapping using the Fourier transform to extract characteristics such as Peak to Average Power Ratio, Skewness, and Kurtosis. In addition, the K-mer extraction is done to extract the entropy feature. The feature fusion is done using Bray-Curtis distance and DBN [11]. Finally, the COVID-19 prediction is done using DQNN [100]. The Deep QNN is trained with LBCA. The proposed LBCA is devised by combining the Coot algorithm [11] and LOA [139]. The analysis of COVID-19 prediction with respect to mutation point is performed [95]. The COVID-19 prediction model with LBCA-based DQNN is exposed in figure 4.1.

4.2.1 Parameter tuning in training model

As the model is learned, its hyperparameters are fine-tuned, and the optimal parameter values are selected by minimising validation loss. Hyperparameters that have been fine-tuned include batch size, dropout probability, epochs, strides, filter size, number of filters,

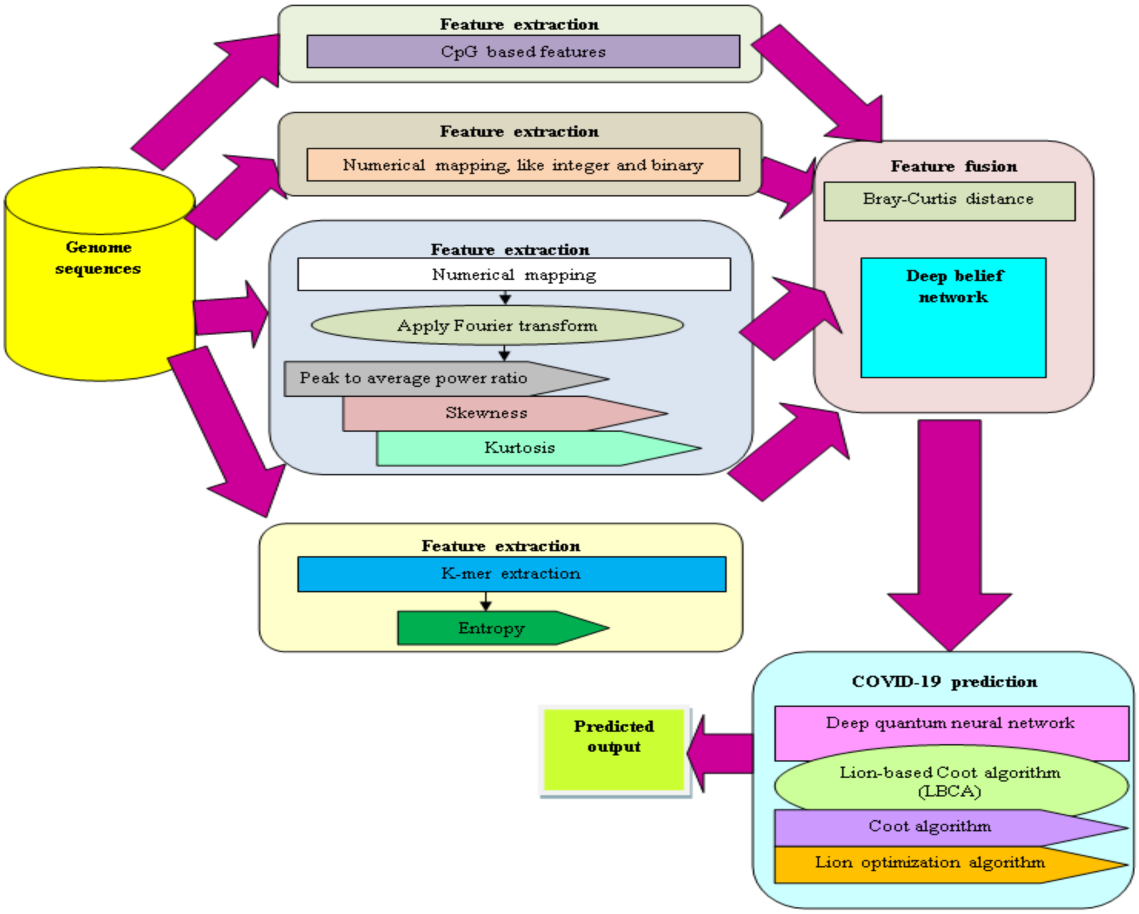


Figure 4.1: COVID-19 prediction model with LBCA-based DQNN

and convolution layers. All imbalanced datasets have 10 epochs, while all balanced datasets have 50. Each of the three convolutional layers has a different filter size and number of filters: (7, 4, 3). 4.1

Table 4.1: Experimental setup parameters for proposed method

Parameters	Values
Number of Filters	3, 7, 4
Number of layers	8, 32
Model	Sequential
activation	relu
optimizer	adam
loss	mae
metrics	accuracy
Epoch	100
learning_rate	0.01

4.2.2 Acquisition of data

Assume a database Q with various samples of data and is formulated by,

$$Q = \{q_1, q_2, q_3, ...g_g, ..q_h\} \tag{4.1}$$

where, h is total data and q_g is g^{th} data, of size $M \times N$

4.2.3 Mining of features

The input data q_g with size $M \times N$ is termed input. Process optimisation is achieved by the use of numerical mapping, entropy K-mer extraction, peak-to-noise ratio, skewness, and kurtosis, as well as CpG-based feature mining. Discovering the essential features with data is the goal of achieving each feature. The characteristics comprise data that needs to be there in order to finish the job, but which doesn't necessarily need to be looked at in detail. Dealing with massive amounts of data is beneficial. It also helps with decreasing the amount of data.

4.2.3.1 CpG based features

The explanation regarding the CpG based features [121] is examined below. The initial CpG based feature CpG_1 is evaluated by summing ratio C and ratio G in which the ratio of nucleotide is evaluated by splitting the occurrences of frequency considering nucleotide to the length of the sequence S . It is mathematically represented by,

$$CpG_1 = ratioC + ratioG \tag{4.2}$$

where, CpG_1 indicates first CpG based features, $ratioC$ is obtained by evaluating the ratio of C nucleotide in sequence S and $ratioG$ by evaluating ratio of G nucleotide in sequence S. The second CpG based feature CpG is evaluated by taking the ratio of CG with respect to $ratioC \times ratioG$. To enhance the prediction of COVID-19, a combination of CpG based and similar features is performed. It is mathematically represented as,

$$CpG_2 = \frac{ratioCG}{(ratioC \times ratioG)} \tag{4.3}$$

where, $ratioCG$ is obtained by computing ratio of CG nucleotides in sequence S. Hence, the CpG-based features is expressed as Y_1.

4.2.3.2 Numerical mapping

Numerical mapping, like binary and integer, is performed. Here, convert the biological sequence into the numerical sequence.

a) **Binary representation** The binary technique [140] is extensively utilized for numerical mapping to convert DNA sequences. It splits the complete sequence of DNA into four subsequences $\{X_A, X_G, X_T, X_c\}$ of the original length. The existence of associated nucleotide in position is expressed by binary '1' or '0'. Considering the DNA sequence, the quadruple dimensionality of generated subsequences of binary of generated binary subsequences can elevate computational overhead. The binary representation is modelled as,

$$f(x) = \begin{cases} 1, & \text{If nucleotidex exists at } k^{th} \text{ position.} \\ 0, & \text{Otherwise;where } x \in A, G, C, T. \end{cases} \tag{4.4}$$

b) **Integer method** The illustration considering the integer method[140] is demonstrated below. DNA sequences are encoded with an integer method considering the real or integer values. The outcome obtained represents a discrete value signal. The integer technique provides values of integers that relies in $\{1, 2, 0.3\}$ to nucleotides C, A, T, G. This type of model reveals that A is higher compared to T and G and is also higher compared to C. Another integer technique allocates $C = 3, A = 1, T = 4, G = 2$ values for the boding biological bar. It is also utilized for mapping nucleotides C, A, T, Gas. The integer method encodes the genome sequences with integer or real values. The integer representation is expressed by 3, 2, 3,

$2, 3, 0, 3, 2, 1, 1, 2$. The numerical mapping-based features is expressed as $\{1, 2, 0.3\}$. The integer method encodes the genome sequences with integer or real values. The integer representation $GAGAGTGACCA$ is expressed by $3, 2, 3, 2, 3, 0, 3, 2, 1, 1, 2$. The numerical mapping-based features is expressed as Y_2.

4.2.3.3 Skewness, kurtosis and peak to average power ratio

Calculations are made for the characteristics of skewness, kurtosis, and peak-to-average power ratio [141]. While processing signals and images, the Discrete Fourier Transform (DFT) is used to produce characteristics based on the Fourier technique. The DFT of the signal having a length $He \in D^f$ at frequency r are given by,

$$F[r] = \sum_{f=0}^{H-1} e[f] e^{j\frac{2\pi}{f} rf}, r = 0, 1, \ldots, H-1 \tag{4.5}$$

where H refers signal length such that $-1 \leq r \leq 0$, refers to frequency.
To enumerate DFT, the fast Fourier transform (FFT) is utilized, which represents the highly effective process for evaluating the DFT with time series. Numeric modelling must be utilized to map the GSP model's genomic data. The feature extraction is considered in each Fourier to transform representation and adapts PAPR, skewness, and kurtosis. Here, the PAPR is formulated by,

$$PAPR = \frac{max_{0 \leq r \leq H-1}(P[r])}{\frac{1}{H}\sum_{r=0}^{H-1} P[r]} \tag{4.6}$$

The skewness, kurtosis and PAPR-based features is expressed as Y_3, Y_4, Y_5.

4.2.3.4 Entropy K-mer extraction

Entropy refers to the uncertainty measure that is linked using a probabilistic experiment. A K-mer method[141] is utilized to produce a probabilistic model. Here, the mapping of each sequence with the frequency of neighboring bases produces statistical data. Decomposing a sequence into its K-mers for analysis allows this set of fixed-size chunks to be analysed rather than the sequence, and this can be more efficient. K-mers are very useful in sequence matching.

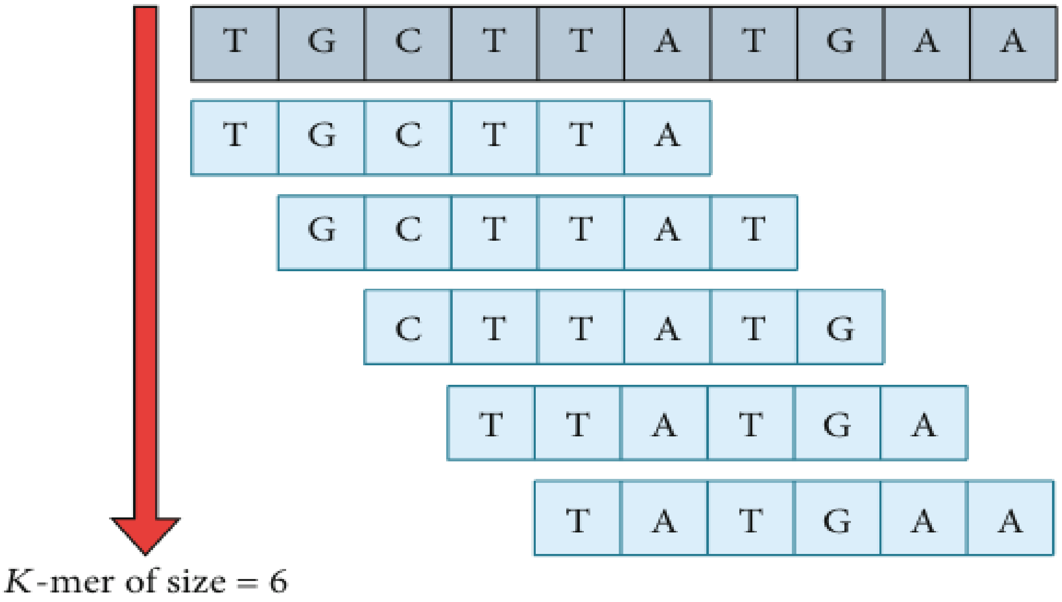


Figure 4.2: DNA Sequence data encoding using the K-mer step by step process technique.

4.3 Results and Discussion

4.4 Experimental setup

To perform the experiments of the proposed model, we used a desktop computer (Core i7 3.2 GHz CPU, 16 GB RAM) with GeForce GTX TITAN X GPU equipped with 12 GB of DDR5 RAM. This section contains an overview of datasets, pre-processing of data, cross-validation and the parameters used in training the proposed model.

4.4.1 Dataset description

The projected COVID-19 prediction scheme utilized the NCBI virus dataset available at [NCBI Labs](https://www.ncbi.nlm.nih.gov/labs). Here, the dataset has various kinds of human coronavirus genomes or species. However, this research considered only some specific species, such as HCov-NL63, HCov-229E, HCov-HKU1, HCov-OC43, MERS-

CoV, SARS-CoV, SARS-CoV-2, and RaTG13. Moreover, all the selected species come under the target class of COVID-19. In this dataset, each sequence contains more than 1000 genome sequences. In the first analysis, we used 200 genome sequences from each file. Now, we are using 400 genome sequences from each dataset. 4.2 shows the types of human coronavirus genomes processed in this research. First, we performed the analysis using fewer sequences and now we have improved the analysis using more sequences.

Table 4.2: The number of covid-19 and non covid-19 samples in human metagenomic datasets.

Species	Genus	Number of entires	Target class
HCov-NL63	Alpha coronavirus	200	COV19-
HCov-229E	Alpha coronavirus	200	COV19-
HCov-HKU1	Alpha coronavirus	200	COV19-
HCov-OC43	Alpha coronavirus	200	COV19-
MERS-CoV	Beta coronavirus	200	COV19-
SARS-CoV	Beta coronavirus	6	COV19-
SARS-CoV-2	Beta coronavirus	200	COV19+
RaTG13	Beta coronavirus	1	COV19-

4.4.2 One hot encoding

One hot encoding technique is used to convert DNA sequence to numerical vector as neural networks handles only numerical data. Specifically, consider a DNA sequence S with n bases S = $\{b_1, b_2, b_3, ..., b_n\}$, $S_i \epsilon$ {A,C,G,T, N}. The encoded vector stored as an array (M) of size nx5 as the following:

$$A_{i,j} = \begin{cases} 1, & \text{if } S_{i-1} = j^{th} \text{base in (A, C, G, T, N).} \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

The number of rows is equal to the number of bases and number of columns are equal to different number of bases.

4.4.3 K-fold cross-validation

Cross-validation is a statistical approach used to evaluate the quality of the proposed model, which also helps to avoid underfitting and overfitting. In k-fold cross-validation, the dataset is randomly divided into k, approximately equal size folds or groups. Iteratively, one fold at a time is treated as a test set and remaining k-1 folds used for model training. We follow a representative tactic to choose k values as 5 and 10 to evaluate the proposed model on different benchmark datasets.

4.5 COVID'19 prediction with proposed LBCA-based DQNN

The prediction of COVID'19 is performed with LBCA-based DQNN. The DQNN training is performed using LBCA and is devised by integrating LOA and Coot algorithm. The DQNN model and training steps of LBCA is shown in Algorithm 4.1. This section provides a detailed explanation about anticipated $LBCA_DQNN$'s results and its discussion in relation to COVID-19 forecast. We have evaluated our method with benchmark eight datasets.

4.5.1 Evaluation metrics

LBCA_DQNN is assessed based on the metrics, such as testing accuracy, TPR, and FPR. Testing accuracy: It refers to the proportion of exact predictions to the overall count of input values, and is illustrated as,

$$A_a = \frac{\alpha_{tp} + \alpha_{tn}}{\alpha_{tp} + \alpha_{tn} + \beta_{fp} + \beta_{fn}} \tag{4.8}$$

where, α_{tp} indicates the true positive, α_{tm}depicts the true negative, β_{fp} specifies the false positive and β_{fn}indicates the false negative.

TPR: TPR or sensitivity is defined as the correctly identified affected patients, which

Algorithm 4.1 Algorithm Steps of LBCA

```
Input: Z,V,M(q)
Output:M(q+1)
Initialize population
Evaluate fitness
if (rand < S) then
    Z, Z_1 and Z_3 are random vectors
    else
     Z,Z_1, and Z_3 are random vectors
for (i=1toS); S-number of coots
 Compute u using Eq. (39)
if (rand > 0.5) then
    Update coot position by Eq. (48)
     else
    if (rand > 0.5) then
        Update coot location
        else
        Update position of coot
for number of leaders
if (rand < 0.5) then
    Update location of leader by an equation that satisfies Z_4 < 0.5
    else
    Update the leader's location by the expression that satisfies condition Z_4 < 0.5
Return best solution
end
```

is expressed as,

$$A_{sen} = \frac{\alpha_{tp}}{\alpha_{tp} + \beta_{fn}} \tag{4.9}$$

FPR: FPR or specificity is expressed as the correctly identified unaffected patients, which is indicated as,

$$A_{spe} = \frac{\alpha_{tn}}{\alpha_{tn} + \beta_{fp}} \tag{4.10}$$

4.5.2 Assessment based on Confusion matrix

The confusion matrix is expressed as the matrix form of predicted output, and it portrays the overall evaluation measure of a proposed model. Thus, the confusion matrix of the newly modelled LBCA_DQNN scheme for COVID-19 prediction with point mutation as shown in figure 4.3.

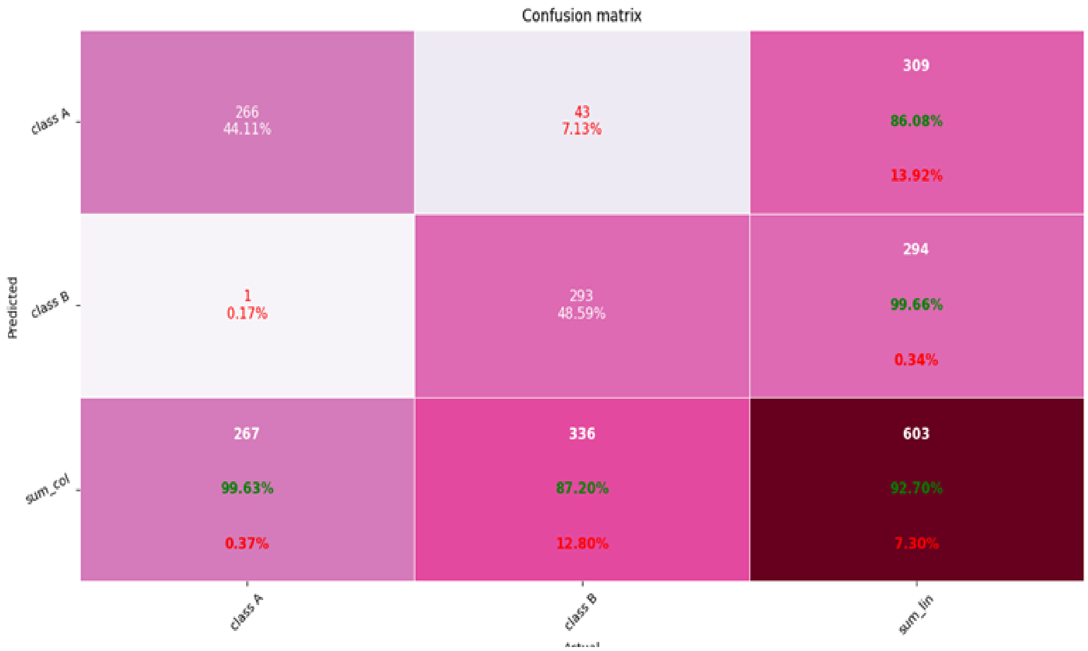


Figure 4.3: Confusion matrix measurement for group-A and group-B

The confusion matrix has two classes: class A and class B. Here, the predicted percentage of class A is 44.11% and the predicted percentage of class B is 48.59%. Thereby, the overall percentage of predicted outcomes is 92.70%.A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

4.5.3 Performance assessment of LBCA_DQNN scheme for COVID-19 prediction with Point mutation rate

In this section, The performance of the devised LBCA_DQNN scheme for COVID-19 prediction is assessed by changing the epoch with various neurons based on the efficiency computation metrics.

4.5.3.1 Performance assessment with respect to epoch using 200 genome sequences

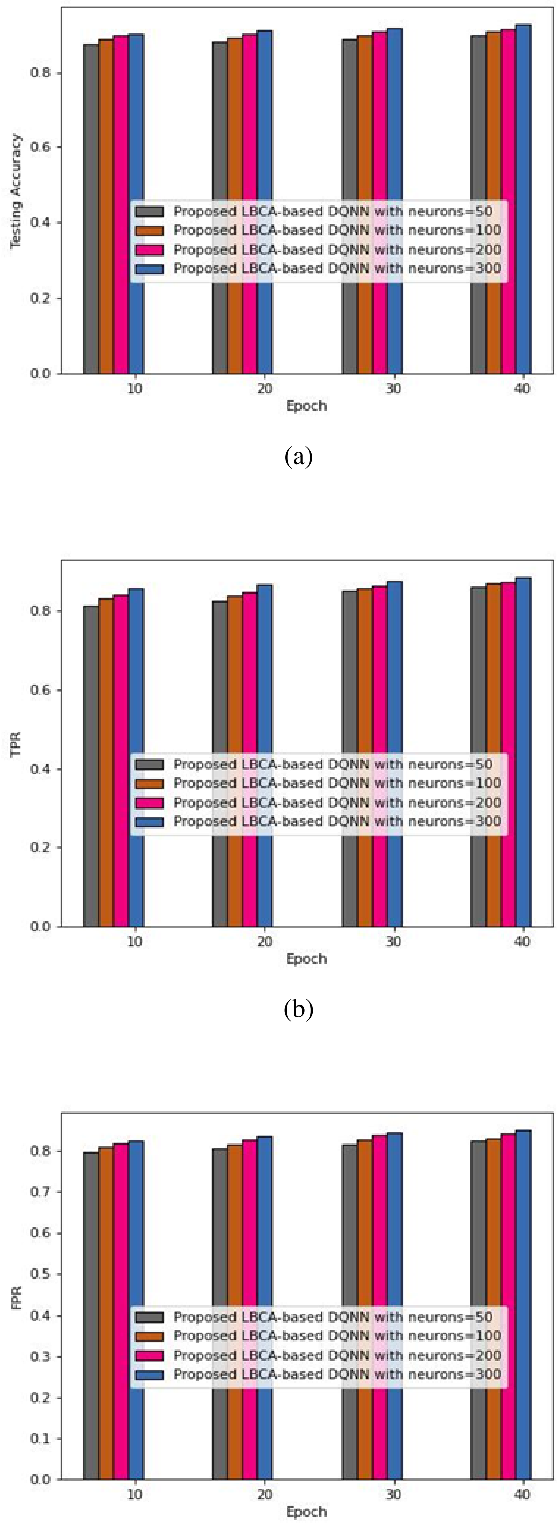
Figure 4.4a shows the performance investigation of devised LBCA_Deep QNN scheme based on testing accuracy with varying neuron count and epoch. Here, the testing accuracy of LBCA_Deep QNN is 0.896, 0.908, 0.914 and 0.926 when the count of neurons is 50, 100, 200 and 300, and the epoch size is 40. The TPR of the LBCA_Deep QNN scheme is computed by varying the epoch size and neurons, and is given in figure 4.4b. The TPR of LBCA_Deep QNN is 0.860, 0.869, 0.872 and 0.885 when the epoch is 40 and the neuron is from 50 to 300. The FPR attained by the LBCA_Deep QNN while selecting the epoch is from 10 to 40 and the neuron is from 50 to 300 with four dissimilar count variations is given in figure 4.4c. Here, the FPR of LBCA_Deep QNN is 0.824 for the neuron is 50, 0.832 for the neuron is 100, 0.843 for neuron is 200 and 0.851 for neuron is 300.

4.5.4 Performance assessment for epoch using 400 genome sequences

Figure 4.5a shows the performance investigation of devised LBCA_Deep QNN scheme based on testing accuracy with varying neuron count and epoch. Here, the testing accuracy of LBCA_Deep QNN is 0.917, 0.927, 0.937, and 0.941 when the count of neurons is 50, 100, 200 and 300, and the epoch size is 40. The TPR of the LBCA_Deep QNN scheme is computed by varying the epoch size and neurons, and is given in figure 4.5b. The TPR of LBCA_Deep QNN is 0.908, 0.916, 0.927, and 0.931 when the epoch is 40 and the neuron is from 50 to 300. The FPR attained by the LBCA_Deep QNN while selecting the epoch is from 10 to 40 and the neuron is from 50 to 300 with four dissimilar count variations is given in figure 4.5c. Here, the FPR of LBCA_Deep QNN is 0.837 for the neuron is 50, 0.847 for the neuron is 100, 0.858 for neuron is 200 and 0.869 for neuron is 300.

4.5.5 Algorithmic methods for assessing the performance of LBCA_Deep QNN

Various optimization algorithms used for assessing the efficacy of LBCA_Deep QNN is Competitive Swarm Optimization (CSO) +DQNN[142]. Optimization algorithm (ROA)+DQNN



(a)

(b)

(c)

Figure 4.4: Performance investigation in terms of a) Testing accuracy b) TPR c) FPR

[142], LOA +DQNN [143] and Coot algorithm +DQNN [51].

4.5.5.1 Algorithmic assessment

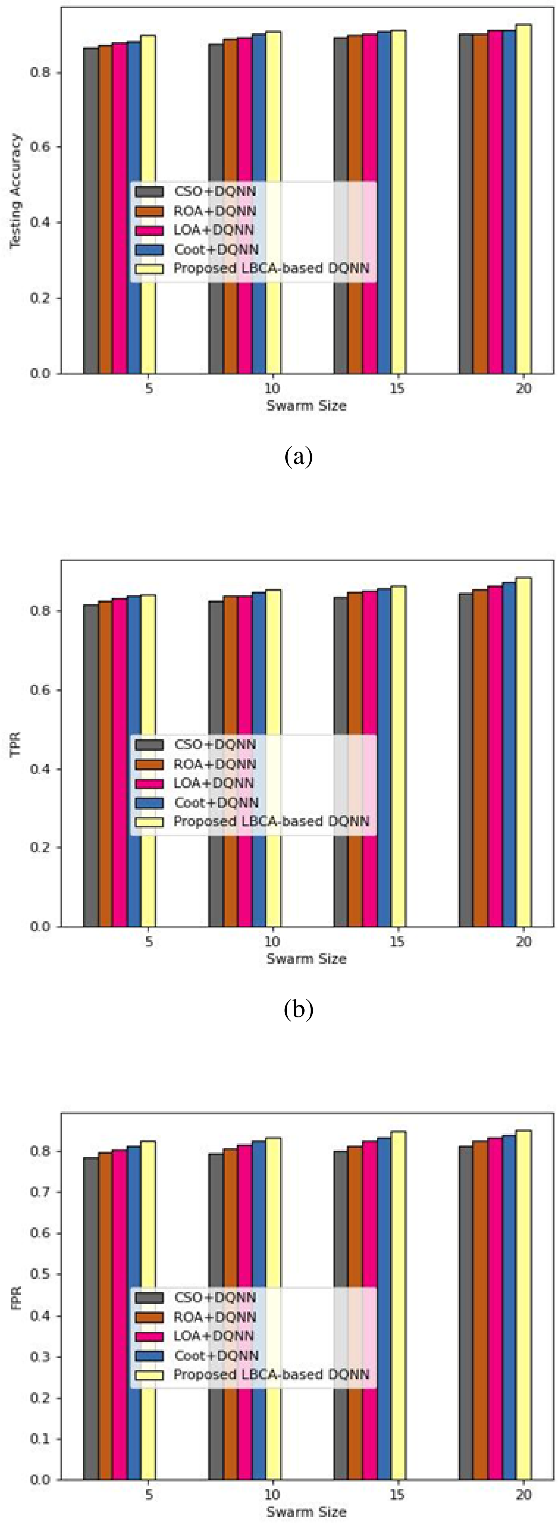
A hybrid optimization model and DQNN-based comparative graph of several optimization algorithms are shown in Figure 4.5a. The CSO+DQNN, ROA+DQNN, LOA+DQNN, and Coot+DQNN in this case each reached the testing accuracy of 0.899, 0.900, 0.909, and 0.912 when the swarm size was 20, while the LBCA_DQNN developed obtained the testing accuracy of 0.926. Also, the LBCA_Deep QNN's improved performance in terms of testing accuracy is 2.93%, 2.82%, 1.84%, and 1.54%. The TPR of the LBCA_DQNN for COVID-19 prediction is shown in Figure4.5b. When the swarm size is 20, the TPR for CSO+DQNN, ROA+DQNN, LOA+DQNN, Coot+DQNN, and LBCA_DQNNis 0.845, 0.853, 0.864, 0.873, and 0.885. Moreover, LBCA_DQNNhas improved by 4.47%, 3.62%, 2.37%, and 1.35%. Figure 4.5c depicts the LBCA DQNN's FPR graph. The FPR for CSO+DQNN, ROA+DQNN, LOA+DQNN, Coot+DQNN, and LBCA_Deep QNN is 0.813, 0.824, 0.835, 0.840, and 0.851 respectively. As a result, the LBCA_Deep QNN's improved performance is 4.5%, 3.0%, 1.89%, and 1.289%.

4.5.6 Comparative Methods for Point Mutation rates with Swarm Genome sequence size

The efficiency of devised optimal deep learning for COVID-19 detection is validated by comparing it with conventional COVID-19 prediction techniques, such as KNN+CpG [121], AKOM [144] and CNN+LSTM [144], RNN-based LSTM [100], intelligent computing model [145].

4.5.6.1 Comparative analysis for learning set using 200 genome sequences

Figure 4.6a shows the analysis of the learning set with mutation rate. When the learning set is from 60 to 90, then the mutation rate is reached 1.995, 1.987, 1.979, and 2.592, correspondingly. Figure 4.6b demonstrates the analysis graph testing accuracy for LBCA_DQNN. Here, the testing accuracy of KNN+CpG is 0.806, AKOM is 0.854, CNN is 0.897 RNN-



(a)

(b)

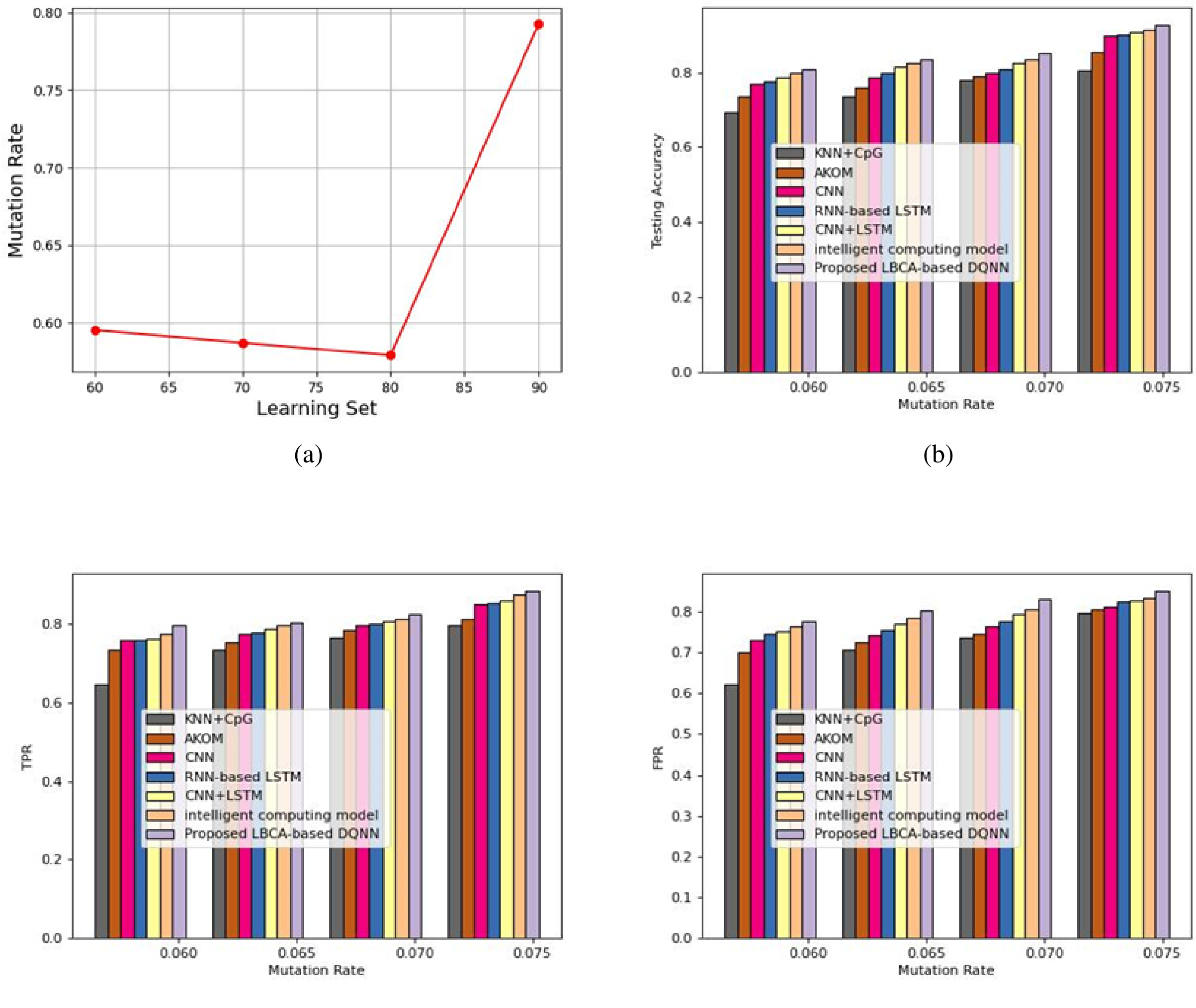
(c)

Figure 4.5: Performance investigation in terms of a) Testing accuracy b) TPR c) FPR

based LSTM is 0.900, CNN+LSTM is 0.907, and intelligent computing model is 0.916, whereas the testing accuracy of LBCA_DQNN is 0.927 while the mutation rate is 0.075. Thus, the progressed performance of LBCA_Deep QNN is 13.07%, 7.89%, 3.26%, 2.912% 2.16%, and 1.18%. Figure 4.6c the TPR of LBCA_DQNN for COVID-19 prediction. The TPR of KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, intelligent computing model and LBCA_Deep QNN is 0.798, 0.815, 0.852, 0.856, 0.861, 0.875 and 0.885 when the mutation rate is 0.075. In addition, the percentage improvement of LBCA_DQNN is 9.914%, 7.991%, 3.711%, 3.27%, 2.757%, and 1.12%. The FPR attained by the LBCA_Deep QNN is given in figure 4.6d. Here, the FPR of LBCA_Deep QNN is 0.851, whereas the FPR of KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, and intelligent computing model is 0.798, 0.807, 0.813, 0.825, 0.827, 0.836, and 0.852, correspondingly when the mutation rate is 0.075. Moreover, the percentage improvement of LBCA_DQNN is 6.358%, 5.317%, 4.503%, 3.16%, 2.930%, and 1.87%.

4.5.6.2 Comparative analysis for K-group using 200 genome sequences

Figure 4.7a shows the analysis of the K-group with mutation rate. When the K-group is from 6 to 10, the mutation rate reaches 2.067, 1.995, 1.989, and 2.700, correspondingly. Comparative graph of devised model based on testing accuracy. Here, when the mutation rate is 0.075, the KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, and intelligent computing model and LBCA_Deep QNN achieved the testing accuracy of 0.806, 0.858, 0.879, 0.887, 0.909, 0.913 whereas the devised LBCA_DQNN acquired the testing accuracy of 0.930. Besides, the progressed performance of LBCA_Deep QNN corresponding to testing accuracy is 13.36%, 7.75%, 5.48%, 4.62%, 2.281%, and 1.82%. Figure4.7b TPR of LBCA_DQNN for COVID-19 prediction. The TPR of KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, and intelligent computing model and LBCA_Deep QNN is 0.800, 0.808, 0.822, 0.836, 0.860, 0.865 and 0.895 when the mutation rate is 0.075. In addition, the percentage improvement of LBCA_DQNN is 10.69%, 9.791%, 8.155%, 6.59%, 3.955%, and 3.35%. The FPR graph of LBCA_DQNN is shown in figure 4.7c. The FPR of KNN CpG is 0.780, AKOM is 0.793, CNN is 0.807, RNN-based LSTM is 0.814, CNN LSTM is 0.837, intelligent computing model is 0.847.Here, by modifying the K-group so



(a)

(b)

(c)

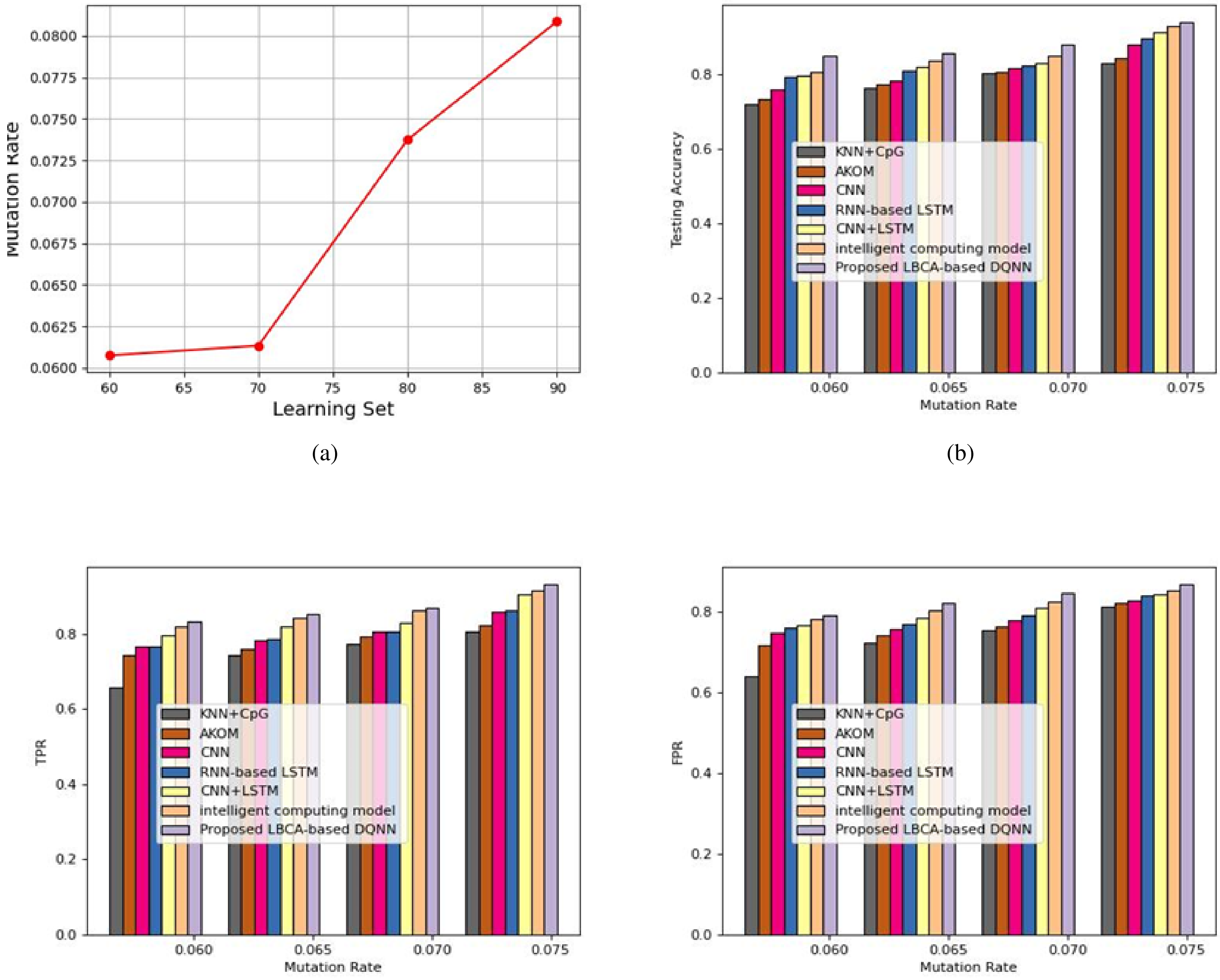
(d)

Figure 4.6: Comparative investigation of LBCA_DQNN in terms of a) Mutation rate b) Testing accuracy c) TPR d) FPR

that the COVID-19 prediction scheme achieved a testing accuracy of 0.943. shows the comparative graph of devised model based on testing accuracy.

4.5.6.3 Comparative analysis with respect to learning set using 400 genome sequences

Figure 4.7a shows the analysis of the learning set with mutation rate. When the learning set is from 60 to 90, then the mutation rate is reached 0.061, 0.061, 0.074, 0.081, correspondingly. Figure 4.7b demonstrates the analysis graph of testing accuracy for LBCA_DQNN. Here, the testing accuracy of KNN+CpG is 0.832, AKOM is 0.846, CNN is 0.880, RNN-based LSTM is 0.898, CNN+LSTM is 0.915, and intelligent computing model is 0.930, whereas the testing accuracy of LBCA_ DQNN is 0.927 while the mutation rate is 0.941. Figure 4.7c shows the TPR of LBCA_DQNN for COVID-19 prediction. The TPR of KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, intelligent computing model and LBCA_Deep QNN is 0.805, 0.822, 0.860, 0.863, 0.907, 0.917, and 0.931 when the mutation rate is 0.075.The FPR attained by the LBCA_Deep QNN is given in figure 4.7d. Here, the FPR of LBCA_Deep QNN is 0.851, whereas the FPR of KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, and intelligent computing model is 0.814, 0.823, 0.830, 0.841, 0.844, 0.852, and 0.869, correspondingly when the mutation rate is 0.075. The frequency of mutants will increase linearly with time. Thus an accurate estimate of phenotypic mutation rate requires a long intervals between frequency measurements and these experiments typically last for hundreds of generations. The frequency of mutants will increase linearly with time. Thus an accurate estimate of phenotypic mutation rate requires a long intervals between frequency measurements and these experiments typically last for hundreds of generations.



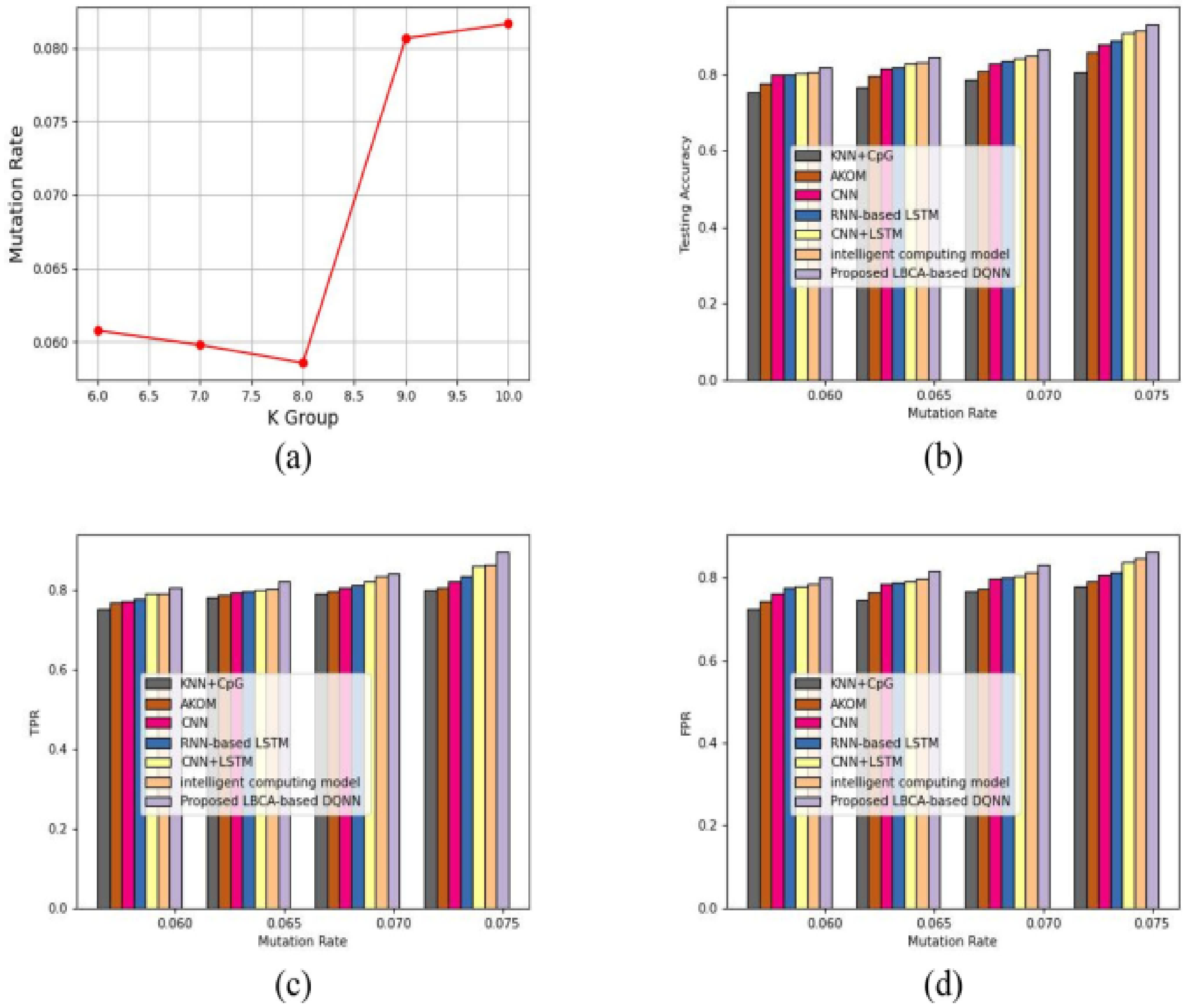
(a)

(b)

(c)

(d)

Figure 4.7: Comparative investigation of LBCA_DQNN in terms of a) Mutation rate b) Testing accuracy c) TPR d) FPR



(a) (b) (c) (d)

Figure 4.8: Comparative investigation of LBCA_DQNN in terms of a) Mutation rate, b) Testing accuracy, c) TPR, d) FPR

4.5.6.4 Comparative analysis for K-group using 400 genome sequences

The study of the K-group with mutation rate is shown in 4.7a. The mutation rate reaches 0.063, 0.063, 0.075, 0.083, 0.085, respectively, when the K-group is between 6 and 10. The comparative graph of the model developed based on testing accuracy is shown in 4.7b. When the mutation rate was 0.075, the testing accuracy for the KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, intelligent computing model, and LBCA_ Deep QNN was 0.819, 0.871, 0.892, 0.900, 0.922, and 0.927, respectively, while the developed LBCA_ DQNN got the testing accuracy of 0.943. Also, the LBCA_ Deep QNN's improved performance in terms of testing accuracy is 13.14%, 7.63%, 5.40%, 4.55%, 2.22%, and 1.69%. The TPR of the LBCA DQNN for COVID-19 prediction is shown in 4.7c. When the mutation rate is 0.075, the TPR of KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, intelligent computing model, and LBCA_Deep QNN is 0.815, 0.823, 0.838, 0.862, 0.887, 0.892, and 4.7d0.922. Moreover, the LBCA DQNN showed improvements of 11.60%, 10.84%, 9.11%, 6.50%, 11.38%, and 10.84% respectively. The LBCA DQNN's FPR graph. When the mutation rate is 0.075, the FPR of the LBCA_Deep QNN is 0.851, whereas the FPRs of the KNN+CpG, AKOM, CNN, RNN-based LSTM, CNN+LSTM, and intelligent computing model are, respectively, 0.785, 0.798, 0.854, 0.869, 0.864, 0.871, and 0.895.

4.5.7 Comparative discussion Results with other models

The comparison of the LBCA_DQNN scheme based on evaluation metrics is shown in Table 4.3 through adjusting the training set and K-group using 200 and 400 genome sequences, respectively. In this case, the devised method with more sequences improved the prediction performance by adjusting the K-group so that the COVID-19 prediction scheme obtained testing accuracy of 0.943, TPR of 0.922, and FPR of 0.895, respectively. At 0.819, 0.871, 0.892, 0.900, 0.922, and 0.927, the testing accuracy for traditional techniques including KNN+CpG, AKOM, CNN, CNN+LSTM, and LBCA_Deep QNN was assessed. The TPR was recorded at 0.815, 0.823, 0.838, 0.862, 0.887, and 0.892, while the FPR was measured at 0.785, 0.798, 0.854, and 0.8. In addition to increasing convergence speed, being simple, scalable, and efficient, the LBCA-based DQNN that was designed provides an excellent balance between exploration and exploitation. We must determine the effective point mutation target size in order to convert point mutation rates to a per-base-pair mutation rate.

4.5.8 Time complexity

The suggested method's temporal complexity analysis is shown in Table 4.4 alongside comparisons to other methods, including CNN, RNN-based LSTM, CNN+LSTM, KNN+CpG,

Table 4.3: Performance evaluation proposed model with other existing models on covid-19

Dataset	Variations	Metrics	KNN + CpG	AKOM	CNN	RNN-based LSTM	CNN + LSTM	intelligent computing model	Proposed LBCA-based DQNN
200	Learning set	Testing accuracy	0.806	0.854	0.897	0.900	0.907	0.916	0.927
200	Learning set	TPR	0.798	0.815	0.852	0.856	0.861	0.875	0.885
200	Learning set	FPR	0.798	0.807	0.813	0.825	0.827	0.836	0.852
200	K-group	Testing accuracy	0.806	0.858	0.879	0.887	0.909	0.913	0.930
200	K-group	TPR	0.800	0.808	0.822	0.836	0.860	0.865	0.895
200	K-group	FPR	0.780	0.793	0.807	0.814	0.837	0.847	0.862
400	Learning set	Testing accuracy	0.832	0.846	0.880	0.898	0.915	0.930	0.941
400	Learning set	TPR	0.805	0.822	0.860	0.863	0.907	0.917	0.931
400	Learning set	FPR	0.814	0.823	0.830	0.841	0.844	0.852	0.869
400	K-group	Testing accuracy	0.819	0.871	0.892	0.900	0.922	0.927	0.943
400	K-group	TPR	0.815	0.823	0.838	0.862	0.887	0.892	0.922
400	K-group	FPR	0.785	0.798	0.854	0.869	0.864	0.871	0.895

AKOM, and intelligent computing model. The time complexity of the proposed LBCA-based DQNN is 0.216 sec, while that of the currently used techniques is 0.387 sec for KNN+CpG , 0.358 sec for AKOM, 0.309 sec for CNN, 0.265 sec for RNN-based LSTM, 0.225 sec for, CNN+LSTM, 0.208 sec for intelligent computing model. Data used for testing was used for training. Therefore, the percentage of correctly classified records is defined as accuracy

Table 4.4: Time complexity for different algorithms with proposed model

Method	Time (sec.)
KNN+CpG	0.387
AKOM	0.358
CNN	0.309
RNN-based LSTM	0.265
CNN+LSTM	0.225
intelligent computing model	0.208
Proposed LBCA-based DQNN	0.216

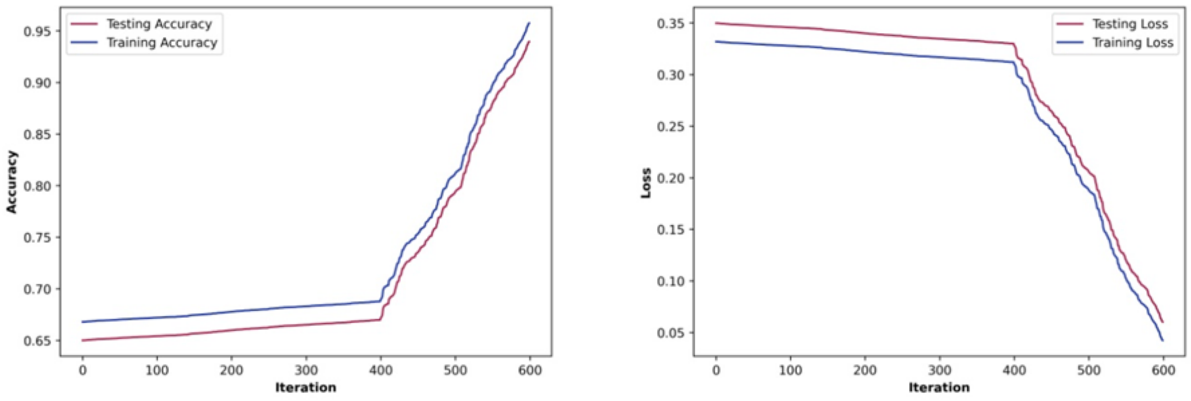
$$\text{Time complexity} = O(M(q)n \times mq) \tag{4.11}$$

Where, $M(q)$ is the population size, $n \times m$ is the dimension, and q is the iteration.

Table 4.5: Comparison proposed algorithm with other algorithm analysis

Metrics	**CSO + DQNN**	**ROA + DQNN**	**LOA + DQNN**	**Coot + DQNN**	**Proposed LBCA-based DQNN**
Testing accuracy	0.899	0.900	0.909	0.912	0.926
TPR	0.845	0.853	0.864	0.873	0.885
FPR	0.813	0.824	0.835	0.840	0.851

From the analysis using tables 4.3 and 4.5, The effectiveness of the prediction scheme allowed the anticipated COVID-19 prediction scheme to achieve greater performance. Given that the LBCA algorithm is used to optimise the expected outcome, the DQNN model produced the better prediction. The goal of the LBCA is to improve the outcome by optimising the projected one. In addition, the LBCA algorithm took into account the benefits of both the Coot method and the Lion optimisation algorithm.



(a) (b)

Figure 4.9: Loss and Accuracy Curves (Training and Validation)

4.5.9 Experimentation tests on Training and validation

A set of data that is kept apart from your training data is referred to as validation data. When training a network, it is used to test how well it would function with data that hasn't been explicitly used to train it. The accuracy graph is shown in Figure4.9a. At 100th iteration, the testing accuracy and training accuracy of the proposed LBCA-based DQNN are 0.655 and 0.672, respectively". The proposed LBCA-based DQNN's loss curve is displayed in Figure 4.9b. At 100th iterations, the training loss and testing loss of the proposed LBCA-based DQNN are 0.345 and 0.327, respectively.

4.6 Summary

This chapter introduces a strategy for COVID-19 prediction based on genomic sequences. The most important thing that this study did was to use hybrid optimisation and genome sequencing to build a deep Coot lion that could forecast the spread of COVID-19. Three steps make up this suggested model: extracting features, calculating the point mutation rate, and predicting the spread of COVID-19. At the outset, Genome sequences undergo feature extraction, which yields characteristics such as CpG based features, integer and binary numerical mapping, and numerical mapping derived from the Fourier transform, which includes features like as skewness, kurtosis, and peak to average power ratio. Furthermore, the entropy feature is extracted using the K-mer extraction method. In order to merge the features, the Bray-Curtis distance and DBN are employed. Before finishing, the COVID-19 forecast is made using Deep QNN. We train Deep QNN with LBCA. The suggested LBCA is a hybrid of the Coot and Lion optimisation algorithms. The COVID-19 prediction is examined in relation to the mutation spot. A test precision of 0.941, a true positive rate of 0.931, and a false positive rate of 0.869 all point to the proposed LBCA-based Deep QNN outperforming older methods. In the future, other sophisticated real-time patient data will also be considered when using genomic sequences to predict the spread of COVID-19. The correlation between COVID-19's dispersion across cities and nations and environmental factors including humidity and terrain will be studied. By combining useful features with Machine learning and Parallel computing methods, future research may investigate the elements influencing the recovery status of COVID-19 patients in more depth.

Chapter 5

DNA Sequence Clustering and ERSIT-GRU for Repeat Detection in COVID-19 Prediction

In this chapter, the Deoxyribo Nucleic Acid (DNA) sequences are collected, and the Sliding Window (SW) process is performed. Next, it identified the different sequences in the data, which are grouped utilizing the Needleman-Wunsch Jaccard K-Means () algorithm. For every group, the sequence Association score is computed. The genome tree is constructed based on this score utilizing the Entropy Neighbor-Joining Algorithm (E-NJA). Subsequently, feature extraction is performed, and using the Fisher Score (FS) approach, the optimal features are selected. Meanwhile, utilizing one-hot coding, the initially aligned data is encoded. Lastly, for predicting DNA Repeats variants in Covid-19, the encoded vector and the optimal features are given to the Exponential Robust Scaling-Identity Tanh-Gated Recurrent Unit (ERSIT-GRU). The experimental evaluation revealed that the proposed model is found to be more efficient compared to the prevailing approaches.

5.1 Introduction

Severe pneumonia is caused by the rapid spread of Covid-19, which is also estimated to generate a high impact on the healthcare system [9]. “By investigating chest X-ray images, Computed Tomography (CT) scans, and whole genome sequences, the traditional approaches identified CoV. Since molecular approaches can easily track CoV genes, they have attracted attention recently and produced more satisfactory outcomes than CT scan [121]. Hence, for corona disease prediction, Genomic sequencing by making molecular genetics diagnoses is hugely important. The total sum of an organism's complete genetic potential, which is stored as an encoded sequence consisting of Thymine-T, Guanine-G, Adenine-A, and Cytosine-C, is named the genome of an organism [146]. Individuals with 35 to 41 repeats might develop corona disease with mild symptoms, whereas it is almost not possible for individuals with 29 to 34 repeats for developing the disease [147]. In addition, as the mutation is the basic process, which results in the emergence of a CoV's new variant, the SARS-CoV-2 virus has mutated continuously similar to any other virus since its emergence [11], [148], [149]. From insertions, deletions, and alterations, the virus genome can mutate into diverse variants [150]. But, owing to its mutational adaptation and modification in its genomic islands, there are massive complications in designing a prediction system [96]. Hence, for analyzing the variants of CoV in an individual, identifying the repeats and mutations in the genome sequence helps. For the prediction of CoV, the prevailing studies used Deep Learning (DL) models [151]. However, in the model, only the common type of CoV is considered and the multiple new variants are neglected. Hence, for overcoming this, the work proposed ERSIT-GRU-based multi-variants of CoV prediction based on DNA Repeats genome sequence analysis.

5.1.1 Problem Definition

Several issues in existing models are: In existing models, CoV was predicted with unstructured big gene data, which reduced the prediction accuracy. There are only CoV2, SARS, and MERS CoV prediction were concentrated; but, the other variants were neglected in the literature. The extraction of mismatched features from different genes was led by the feature extracted directly from the genome sequence.

For solving these issues, a reliable CoV prediction model is developed in the proposed approach, and its contributions are:

- The data are converted to a structured format with the ENJA technique to improve accuracy.
- Using the ERSIT-GRU model, different variants of Repeats CoV are predicted.
- Sequences are identified and grouped using NWJ-K-Means to perform reliable feature extraction.

5.2 Proposed Approach

The proposed a genome variant-based disease prediction with DNA sequences using the ERSIT-GRU technique is shown the figure:

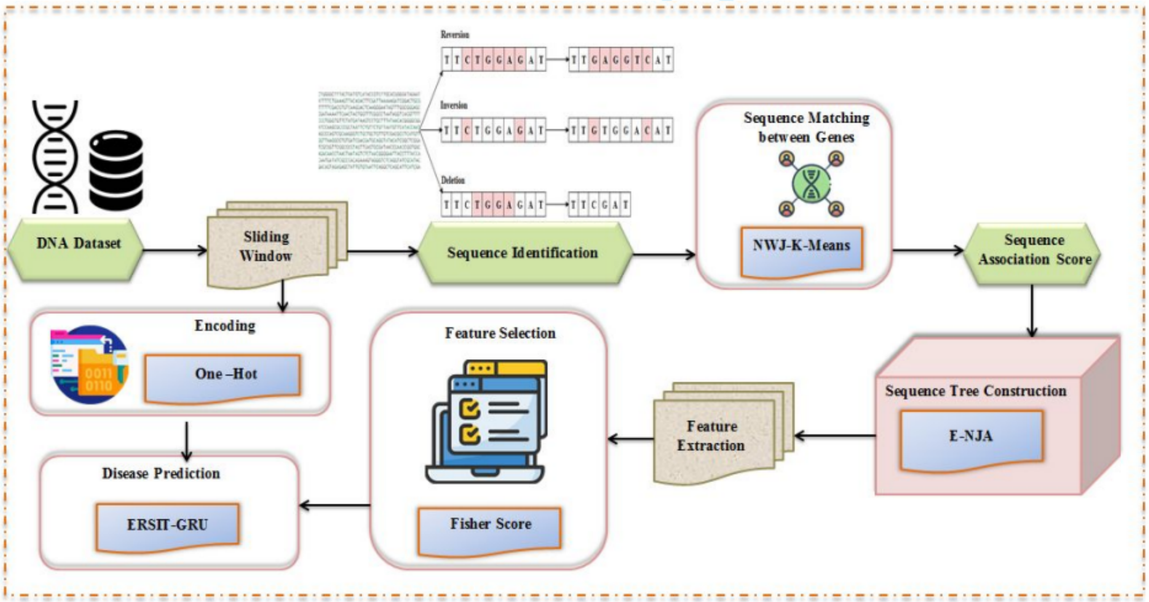


Figure 5.1: Global prediction model for repeats in COVID-19 pandemic with Proposed model

5.2.1 Sliding Window

By collecting the DNA data of various people, the proposed work begins. The collected data is expressed as,

$$\Im = t_1, t_2, t_3, ...t_n \tag{5.1}$$

Here, the gene sequence of the n^{th} person is depicted as t_n . The SW process is performed in the input data ($\Im$) for reducing the temporal complexity. Moving a window of a specific length across the genome sample by sample and computing the statistic over the data in the window are involved in the SW technique. For each input sample, the output is the statistic over the window of the current sample and the prior sample. Lastly, the sequences are processed and are denoted as $\Im_{SW}$.

5.2.2 DNA Sequence K-Mer Identification

In this, to find the occurrence of exactly the same nucleotide in the same position in aligned sequences $\Im_{SW}$, sequence identification is performed. Same Uni-character, same Bi-character, same Tri-character, same number of characters, Reversion, Inversion, Substitution, Duplication, Insertion, Tandem, mirror repeats, and Deletion are identified and the sequences (S) are expressed as,

Here, the q^{th} sequence identified is depicted as s_q . The sequence identification process is specified in Table 5.1.

Table 5.1: Global analysis of repetitive DNA from Various DNA Repeat Sequences

Sequence Name	Sequences	Resulted Sequences
Uni-character	TTCTGGAGAT	A, T, G, C
Bi-character	TTCTGGAGAT	AA, TT, GG, CC
Tri-character	TTCTGGAGAT	AAA, TTT, GGG, CCC
Reversion	TTCTGGAGAT	TTGAGGTCAT
Inversion	TTCTGGAGAT	TTGTGGACAT
Substitution	TTCTGGAGAT	TTGACGAGAT
Duplication	TTCTGGAGAT	TTCTGGAGGAGAT
Insertion	TTCTGGAGAT	TTCTGGAAGTGAT
Deletion	TTCTGGAGAT	TTCGAT
Tandem	TTCTGGAGAT	TTCTTCTTCTGGAGAT
Mirror	TTCTGGAGAT	TTCTGGAGATTAGAGGTCTT

5.2.3 Sequence Matching

After sequence identification, using the NWJ-K-Means algorithm, all the sequences are grouped. This sequence matching is for grouping the related genes. The K-Means algorithm executes effective clustering on big data. However, it has trouble with data having varying sizes and densities. For solving this, the Needleman–Wunsch algorithm selects the centroids and the Jaccard index is also used. Primarily, using the Needleman–Wunsch algorithm approach, the sequences (S) form a number of clusters(k) , and the cluster centroids are selected. The Needleman–Wunsch algorithm initializes the sequences in the form of a matrix , which is formed with one column and row added to the length of the sequence. Next, the matrix is filled with a fundamental scoring scheme. The score is 1 if two nucleotides of the sequence at i^{th} and j^{th} position of the matrix are the same. Otherwise, the score is filled as -1. This matrix filling with maximum score process is explained as,

$$M_{i,j} = max\{M_{i-1,j-1} + \delta_{i,j}, M_{i,j-1} + \epsilon, M_{i-1,j} + \epsilon\} \tag{5.2}$$

Here, the mismatch score is depicted as , and the gap score is given as . Lastly, for acquiring the appropriate matching, a traceback step is performed. The sequence with the maximum score is selected as the centroid (σ). Currently, based on the similarity, the other sequences are assigned to the centroids, which is calculated utilizing the Jaccard index (λ) as,

$$\lambda(S, \sigma) = \frac{|S \cap \sigma|}{|S| + |\sigma| - |S \cap \sigma|} \tag{5.3}$$

Till the convergence is attained, repeat the steps. Hence, the grouped sequences are shown as,

$$K = \{k_1, k_2, k_3, ..., k_z\} \tag{5.4}$$

Here, the z^{th} number of the cluster is depicted as k. The NWJ-K-Means' pseudo-code is represented as in algorithm5.2:

5.2.4 Sequence Tree Construction

Here, to get accurate features from the structural data, a sequence tree is constructed using the E-NJA based on the score (A_{sc}). For constructing phylogenetic trees with less time, NJA is used. However, the undesirable features in it often assign negative lengths to some branches. Hence, in conventional NJA, the Entropy technique is used. ENJA utilizes a

Algorithm 5.2 Pseudo-code

```
Require: : Identified Sequences (S)
Ensure: Matched Sequences (k)
  Begin
  Initialize the Identified Sequences (S) and the number of clusters
  while $i = 1 to q$ do
      Elect the cluster centre using NW technique
  initiate centroids ($\sigma$)
  for each remaining data do
      Compute similarity using Jaccard index
          $$\lambda(S,\sigma) = \frac{|S \cap \sigma|}{|S| + |\sigma| - |S \cap \sigma|}$$
      Assign $S_i$ to $k$ with minimum similarity
  Return number of clusters
  End
```

distance matrix(d_{ij}) grounded on which the association scores are merged to construct a tree for every gene. The criterion value (C_{ij})for acquiring the dimension is described as,

$$C_{ij} = (h-2)d_{ij} - d_i - d_j \tag{5.5}$$

Here, the number of genes is signified as h. The value (C_{ij}) creates a new node that depicts the structure's root. Next, using the entropy technique , the branch lengths (h_{iu}, h_{ju}) are estimated as,

$$E(h) = \sum \frac{1}{2} P(h_i - h_{iu} + \frac{1}{2} \log(h_j - h_{ju}) \tag{5.6}$$

Here, P refers to probability. For every gene sequence, the tree is constructed based on the lengths.

5.2.5 Feature Extraction

In this phase, the features, namely Pattern length, Mean, Standard deviation, Correlation, Entropy, MRP Count, ORF Count, Palindrome count, Palindrome Threshold, Occurrences, etc., are extracted from the trees. The extracted features($\Re$)are expressed as,

$$\Re = \{r_1, r_2, r_3, ..., r_p\} \tag{5.7}$$

Here, the p^{th} feature is symbolized asr_p . The output generated from the ERSIT-GRU phase is passed to algorithm, to find the frequency of each type of repeat. They are used to create CSV (Comma-Separated Value) files containing features for different lengths (30, 40, and 50) of repeats. The CSV file contains nine columns, the first column has the filename along with chromosome number, and the remaining eight columns have frequencies of different repeats corresponding to that particular chromosome. In some chromosomes, if specific repeats are not found, then the corresponding column value is set to zero.

5.2.5.1 Algorithm for feature extraction

In the algorithm, fN is the name of a file and freq is the number of lines in that file which means the number of particular repeats present in that file. Hashmap M stores all file names along with corresponding frequency. gK is a key extracted from Map M containing sequence name sN and repeat number rN. rC is a value of a particular file extracted from Map M. hash map stores sN and all rCs corresponds to that sN.5.3

5.2.6 Feature Selection

Here, for reducing the training time, the optimal features from are selected using the FS technique. Features with the highest FS are selected by the approach, which also returns a projection matrix of indicators. The FS process is described as,

$$FS(\Re) = v\{(\overrightarrow{m_1})(\overrightarrow{m_2} + \tau\rho)^{-1}\} \tag{5.8}$$

Here, the total number of features is signified as v,τ is a regularization parameter, the perturbation term is depicted as ρ, the between-class scatter and total scatter matrix are symbolized as $\overrightarrow{m_1}$ and $\overrightarrow{m_2}$, respectively. Hence, the important features $\Re_{im}$ are selected.

Algorithm 5.3 Feature Extraction()

Input: All files f generated by reduce function for all sequences.

Output: A CSV file f_{CSV}, contains sequence name and frequency of all repeats as columns.

```
for each file $f_i$ do
    $fN$ ←fileName($f_i$)
    $freq$ ←numberOfLines($f_i$)
    add tuple <$fN$,$freq$> to a Map<String, String> $M$
Create hashMap<String,ArrayList> $hM$
for each tuple <$fN$,$freq$> $t$ of $M$ do
    $gK$ ←getKey($t$)
    Split $gK$ into sequenceName($sN$) and repeatNumber($rN$)
    if $hM$ is not contains($sN$) then
        add <$sN$, $arrayList()$> to $hM$
    $rC$ ←getValue($t$)
    add $rC$ value of all repeats to the corresponding arrayList() of $sN$
    add <$sN$, arrayList()> to $hM$
for each entry $r$ in $hM$ do
    add a row $r + 1$ into a CSV file $f_{CSV}$
return $f_{CSV}$
```

5.2.7 Encoding

In the meantime, here, One-hot encoding is done in the data $\Im_{sw}$. Since DL models can only interpret numerical data, this technique transforms the categorical values into numerical values. In this technique, a new variable is created for each level of categorical feature. Each term is mapped with a binary variable with either 0 or 1, where 0 signifies the absence and 1 depicts the availability of that category. This encoding process is expressed as,

$$O(\Im_{sw}) = \{(1, if \Im_{sw} \in category, 0, otherwise) \tag{5.9}$$

Hence, the strings in the data $\Im_{sw}$. are processed and an encoded result $O(\Im_{sw})$. is acquired

5.2.8 DNA Repeats Prediction

Lastly, to predict the new variants, $O(\Im_{sw})$ and $\Re_{im}$ are given to the ERSIT-GRU classifier. The two inputs are merged and depicted as K. To allow the model to learn and train faster, GRU employs gating mechanisms. It has a problem with gradient explosion and overfitting. To solve this, the Exponential-centric Robust Scaling is included in the prevailing GRU. In addition, to overcome the slow convergence rate, identity Tanh activation is used. Figure 5.2 displays the architecture of ERSIT-GRU.

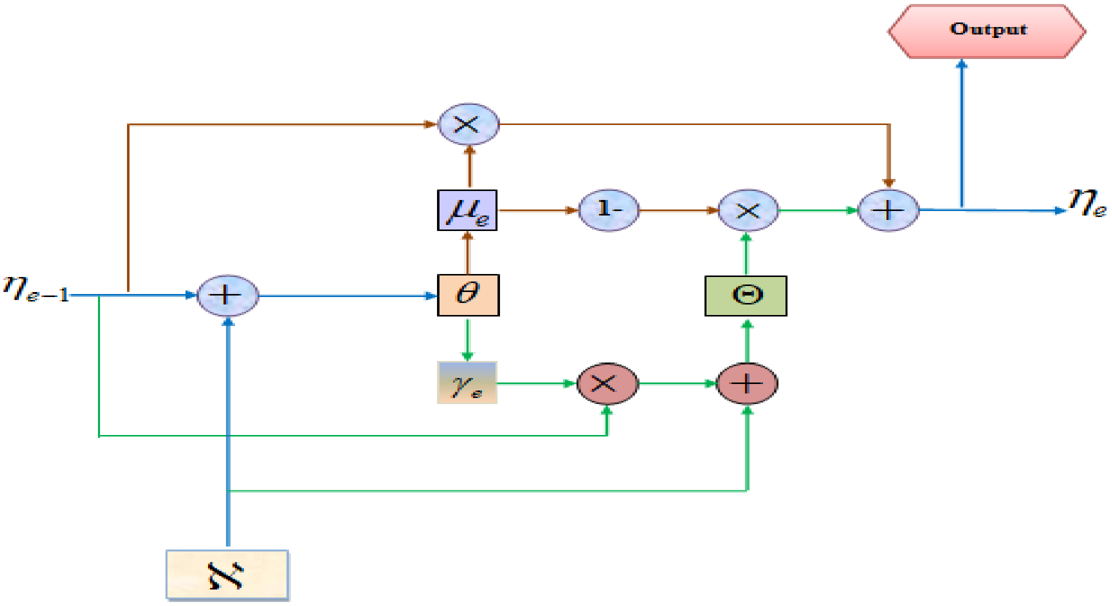


Figure 5.2: ERSIT-GRU

In ERSIT-GRU, each gate takes two inputs at each time (current input and previous hidden state). Two primary gates operations are given below:

5.2.9 Clustering based on features

Clustering is applied to thousands of DNA sequences and similar ones are grouped together based on patterns present in those sequences. Clustering performed directly on actual DNA sequences is a time-consuming process because of the large length of sequences. Different types of repeats are extracted and the frequency of each type of repeat is calculated for every chromosome sequence. Thus, by considering frequencies of repeats as features, time complexity can be greatly reduced. These features are then used for sequence clustering with the ERSIT-GRU approach such as K-means. The eight types of repeats, we considered in the proposed approach are defined as follows:

5.2.9.1 Repeats Feature generation process

Definition1.a: Tandem Repeat ($\mathcal{T}$) is a pattern in a DNA sequence S defined over Σ = {A, C, G, T} based on the following properties:

i $\mathcal{T} = \Upsilon_1\Upsilon_2\Upsilon_3\ldots,$
ii $|\Upsilon_1| = |\Upsilon_2| = |\Upsilon_3| = \ldots$, and $\Upsilon_1 = \Upsilon_2 = \Upsilon_3 =\ldots$, and
iii $3 \leq |\Upsilon_i| \leq 30, \forall i \geq 1$.

Short Tandem Repeat is defined in the same way as Tandem Repeat but the only difference is that $1 \leq |\Upsilon_i| \leq 2, \forall i \geq 1$.

Definition1.b: Tandem Repeat with interrupt ($\grave{\mathcal{T}}$) is a pattern in DNA sequence S defined over Σ = {A, C, G, T} based on the following properties:

i $\grave{\mathcal{T}} = \ddot{\Upsilon}_1\ddot{\Upsilon}_2\ddot{\Upsilon}_3\ldots,$
ii $\ddot{\Upsilon}_i \neq \ddot{\Upsilon}_j$, (differing in one or more nucleotides), and $|\ddot{\Upsilon}_i| \neq |\ddot{\Upsilon}_j|$ (if insertion or deletion of a nucleotide(s))(or) $|\ddot{\Upsilon}_i| = |\ddot{\Upsilon}_j|$ (if updation of a nucleotide), and
iii $3 \leq |\ddot{\Upsilon}_i| \leq 30, \forall i \geq 1$.

Short Tandem Repeat with interrupt is defined in the same way as Tandem Repeat with interrupt but the only difference is that $1 \leq |\ddot{\Upsilon}_i| \leq 2, \forall i \geq 1$.

Definition2.a: Mirror Repeat (♏) is a pattern in DNA sequence S defined over Σ= {A, C, G, T}, where ♏= ⩕∂⩕^{R}, |⩕| = |⩕^{R}|, ⩕ = (⩕^{R})^R, and $|\partial|$ = 0 (or) 1.

Definition2.b: Mirror Repeat with interrupt (♍)is a pattern in DNA sequence S defined over Σ= {A, C, G, T}, where ♍= ⩕∂⩕^{R}, |⩕| = |⩕^{R}|(if updation of a nucleotide), |⩕| $\neq$ |⩕^{R}|(if insertion or deletion of a nucleotide), ⩕ $\neq$ (⩕^{R})^R, and $|\partial|$ = 0 (or) 1.

Definition3.a: Pairing Repeat (Þ) is a pattern in DNA sequence S defined over Σ = {A, C, G, T}, based on the following properties:

i Þ=$\alpha\beta$, $|\alpha| = |\beta|$ (≥ 1)
ii if α_i = A then β_i = T and vice versa, $\forall i \geq 1$, and

iii if α_i = C then β_i = G and vice versa, $\forall i \geq$1.

Definition3.b: Pairing Repeat with interrupt (þ) is a pattern in DNA sequence S defined over Σ = {A, C, G, T}, based on the following properties:

i þ=$\gamma\delta$, |γ| = |δ| = n ($\geq$ 1)
ii if γ_i = A then $\delta_i \neq$ T and vice versa, $\exists i \geq$1, and
iii if γ_i = C then $\delta_i \neq$ G and vice versa, $\exists i \geq$1.

Definition4.a: Inverted Repeat (£) is a pattern in DNA sequence S over Σ = {A, C, G, T}, defined based on the following properties:

i £=$\mu\nu$, |μ| = |ν| = n ($\geq$ 1)
ii if μ_i = A then $\nu_{\text{n-i+1}}$ = T and vice versa, $\forall i \geq$1, and
iii if μ_i = C then $\nu_{\text{n-i+1}}$ = G and vice versa, $\forall i \geq$1.

Definition4.b: Inverted Repeat with interrupt ($\jmath$) is a pattern in DNA sequence S defined over Σ = {A, C, G, T}, based on the following properties:

i $\jmath$ =$\varphi\psi$, |φ| = |ψ| = n ($\geq$ 1)
ii if φ_i = A then $\psi_{\text{n-i+1}} \neq$ T and vice versa, $\exists i \geq$1, and
iii if φ_i = C then $\psi_{\text{n-i+1}} \neq$ G and vice versa, $\exists i \geq$1.

5.3 Computational Complexity Analysis

The time complexity of the proposed method in different phases is estimated in the following subsections.

5.3.1 ERSIT-GRU for DNA repeats

In ERSIT phase, to check different types of repeats and to write them into a file, the complexity involved is Time complexity for one gradient step for sequence O(T d2h + T dhdi) length T, dhdi=hidden state and input dim. In number of time steps it takes for a neuron in hash table to vanish is achieved by assuming that the probability of either obtaining a negative or a positive contribution to its activation at step t is 1. The computation of ht does not involve any matrix multiplications between previous hidden-state ht-1, and the permutation operator can be applied in O(dh). However, since b does not require any information from previous states, it can be applied in parallel to all time steps, thus greatly reducing the total runtime. It assumes that the number of hidden units in each layer is O(di).

5.3.2 Generation features

To generate a CSV file, which contains the frequency of seven type of repeats in each chromosome of differen species, the time complexity is O(N*M*C)+O((N^2)*X)+O((N^2)*X/R)+ O(N*X *D) = O(N*M*C+((N^2)*X)+N*X*D), where C is maximum number of bytes in a line of a file, M is number of lines in a file, N is total files, X is maximum size of a file, R is number of repeats per file, D is total digits in repeats.

5.3.3 Clustering and overall complexity

K-Means is used to cluster genome sequences for evolutionary relationships among different biological species. The computational(Time) complexity of the K-Means algorithm is $\theta(n*ɗ*ɨ*ƙ)$, where $n = |S|$ is the number of genome sequences, ɗ=$|R|$ is a different type of repeats, ɨ= $|I|$ is number of iterations and ƙ=$|K|$ is number of clusters. In Covid-19-Kmeans the different types of repeats are constant (ɗ=8). So, the worst-case time complexity of K-Means is $\theta(n*ɨ*ƙ)$. The overall time complexity of our proposed methods are O(N*S) + O(N*M*C+((N^2)*X)+N*X*D) + $\theta(n*ɨ*ƙ)$.

5.4 Results and Discussion

Here, the proposed system's experimental outcomes are examined, which were executed on the working platform of PYTHON.

5.4.1 Dataset description

From the National Center for Biotechnology Information (NCBI) database, the CoV genome dataset was gathered. The data consisted of the gene sequence of people with the features of HCoV-NL63, SARS2-Cov2, HCoV-HKU1, MERS-CoV, HCoV-OC43, HCoV-229E, and

SARS-CoV. The links for the dataset used are given below:
https://www.ncbi.nlm.nih.gov/nuccore/NC_002645.1?report=fasta
https://www.ncbi.nlm.nih.gov/nuccore/NC_019843.3?report=fasta
https://www.ncbi.nlm.nih.gov/nuccore/NC_004718.3?report=fasta

5.4.2 Analysis of mutational bias in different CoVs

We performed mutation analysis of three coronavirus (NL63, MERS-CoV, and 2019-nCoV) DNA sequences and examine the transformations of slow codons and non-slow codons due to mutations. A sample of mutations found in the various DNA sequences are shown in Table 5.2. We identified and analyzed transition, transversion, silent, missense and nonsense mutations at codon level in CDs that reveals the genetic diversity of various CoVs. The mutation rate in 2019-nCoV is very less compare with MERS-CoV and NL63. In 2019-nCoV, we pointed out silent and missense mutations whereas in other two CoVs nonsense mutations are also recognized. In MERS-CoV and NL63 silent mutations are very high compare with 2019-nCoV. The mutation rates in 2019-nCoV DNA strains of human collected from countries like USA, Greece, Brazil, and Srilanka have higher than the China, India, and South Africa. Due to point mutations at codon level the transformation of slow codons to non-slow codons found to be high that may impact the protein synthesis rate. The results provide evidence for genetic diversity and fast evolution of new coronaviruses.

5.4.3 Performance analys on each k-mer DNA sequence

In this segment, the proposed algorithms' performance in contrast to the prevailing algorithms is discussed.

The common sequences used among different datasets are shown in figures 5.3, 5.4, 5.5, 5.6, 5.7, 5.8. The figures depict several sequences achieved for varying lengths of descriptors. Uni-character, Bi-character, Tri-character, Reversion, Inversion, Substitution, Duplication, Insertion, Tandem repeats, Mirror repeats, and Deletion are the major se-

Table 5.2: Mutations found in different biological DNA sequence 2019-nCoV, MERS-CoV, and NL63.

CoV Type	**# Transitions: Transversion Mutations**	**Silent Mutations**	**Missense Mutations**	**Nonsense Mutations**	**CoVs-Strain**
2019-nCoV	1 : 2	-	GTC(11082)→CTC, TTC(28143)→CAC	-	CHN/Yunnan-01/2020_MT049951
	5 : 0	GTG(18059)→TTT	CCT(17746)→CTT, TAT(17857)→TGT etc.	-	USA/WA3-UW1/2020_MT163719
	1 : 0	-	GCT(14407)→TTT	-	ZAF/R03006/2020_MT324062
	3 : 0	GTT(18125)→GTC	CCT(14407)→CCT, GCT(14785)→AAT	-	GRC/10/2020_MT328032
	2 : 2	AAC(14804)→TTT, GGT(17246)→CGG	GCA(11082)→TTA, GGT(26143)→GTT	-	BRA/SP02cc/2020_MT350282
	1 : 0 2 : 1	- -	CCT(14407)→CTT AGT(1396)→AAT, GTA(11082)→TTA	- -	IND/GBRC1/2020_MT358637 TWN/CGMH-CGU-05/2020_MT370518
	2 : 3	-	AGT(1396)→AAT, GTA(11082)→TTA etc.	-	LKA/COV38/2020_MT371047
MERS-CoV	41 : 11	CTT(776)→CCG, AAC(1832)→CCA etc.	CAT(749)→CAG, AAA(1453)→AAA etc.	-	HCoV-EMC_MH306207
	43 : 5	AGA(3275)→AGG, GTC(12683)→ATT etc.	TTT(541)→TAT, CAT(749)→CAG etc.	-	HCoV-EMC_MH013216
	57 : 14	CCC(1832)→CCA, AGA(3275)→AGG etc.	CAT(749)→CAG, TCG(1381)→TTG etc.	CAG(13395)→TAG, GAG(23553)→TAG etc.	HCoV-EMC_MH454272
	57 : 14	CCC(1832)→CCA, CTG(6285)→TTG	CTA(652)→CAA, CAT(749)→CAG etc.	CAG(13395)→TAG, CAA(29850)→TAA	2366_MH432120
	57 : 14	CTG(7554)→TTG, CTC(8501)→CTT	CTA(1903)→CCA, TTG(2773)→TCG etc.	GAG(23553)→TAG, CAA(29850)→TAA etc.	2363_MH395139
NL63	42 : 4	TGC(12974)→TGT, GCC(13352)→GCT etc.	ATT(17433)→GTT, TCT(17620)→TTT etc.	GAA(20799)→TAA	Haiti-1/2015_KT266906
	67 : 12	TGT(14591)→TGC, CTC(14672)→CTT etc.	TTT(414)→CTT, TAT(2373)→CAT etc.	GAA(20799)→TAA, TTG(21478)→TAG	UF-1/2015_KT381875
	70 : 12	GAA(12902)→GAG, TGC(12974)→TTT etc.	CCC(7740)→TTT, CGT(9159)→TGT etc.	GAA(20799)→TAA, TTT(21478)→TAG	UF-2/2015_KU521535
	57 : 14	GAA(12902)→GAG, TGC(12974)→TGT	AAA(12902)→GAG, TGC(12974)→TCT etc.	AAA(20799)→TAA, TTC(21478)→TAG	UF-2/2015_KX179500
	21 : 46	CTT(16560)→TTG, AAA(16616)→AAG etc.	AGT(13293)→TGT, GAT(14627)→GAA etc.	-	UNKNOWN-CS124012_CS124012

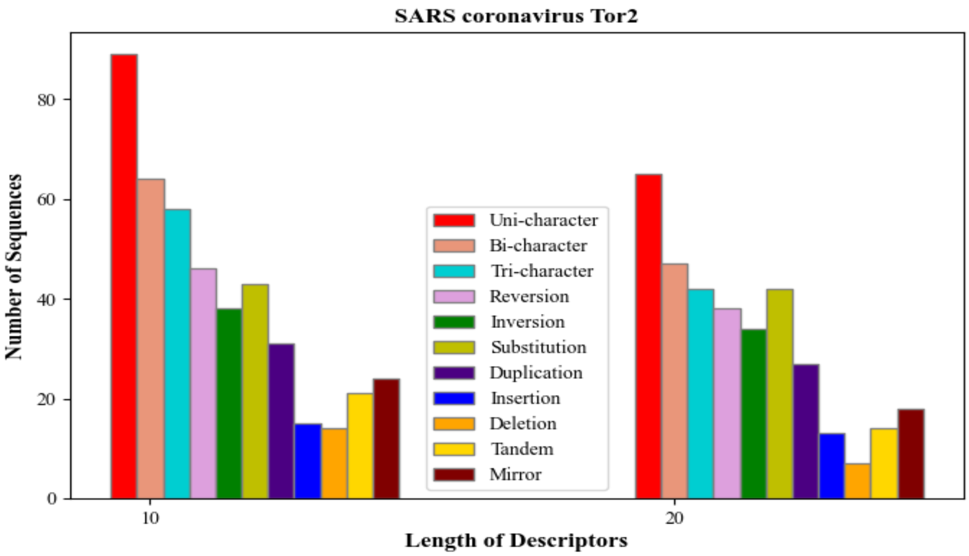


Figure 5.3: Repeat Sequence Illustration of SARS CoV Tor2

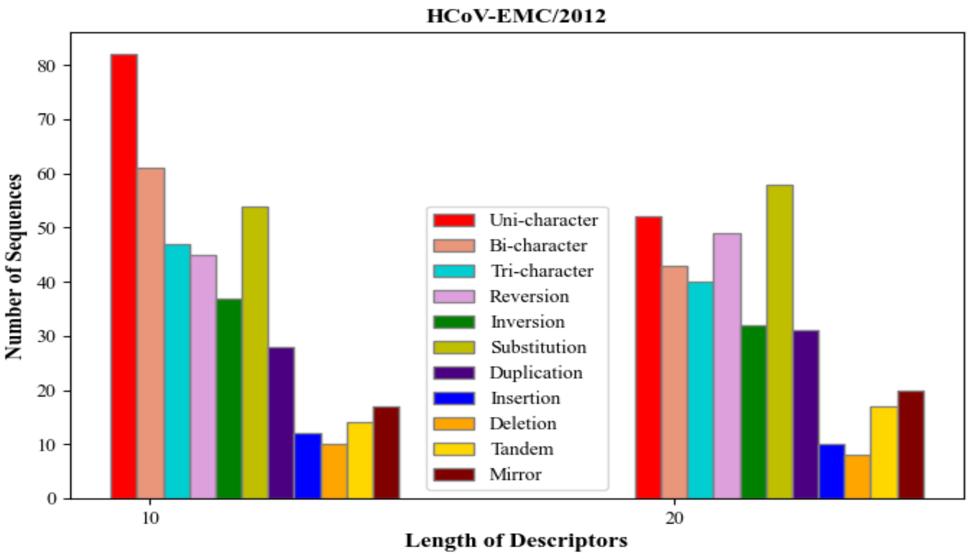


Figure 5.4: Repeat Sequence Illustration of HCoV-EMC 2012

quences considered. Thus, the gene structure of DNA can be well interpreted using such figures.

Figure 5.9 shows that when compared to Bidirectional-LSTM (Bi-LSTM), LSTM, and Recurrent Neural Network (RNN) classifiers, the GRU achieved superior accuracy (96.47%) and precision (93.48%). However, the ERSIT approach in the GRU improved

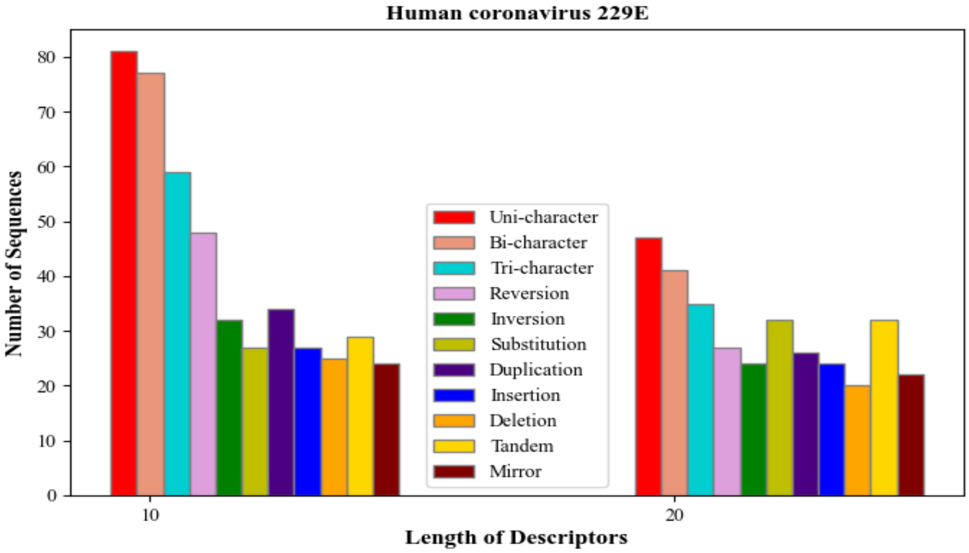


Figure 5.5: Repeat Sequence Illustration of 229E

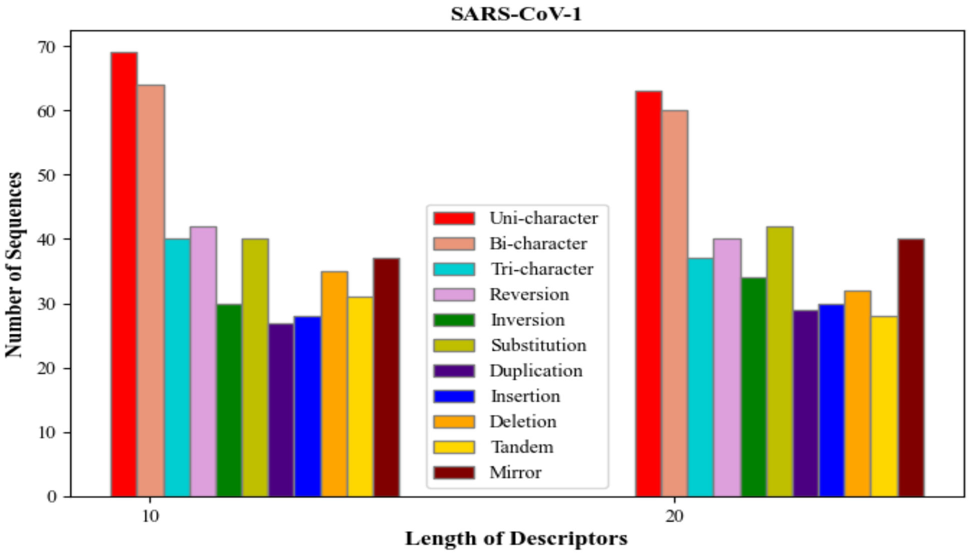


Figure 5.6: Sequence Illustration of SARS-CoV1

the performance by 1.68%, 3.36%, and 2.84% than the traditional GRU. This exhibited the ERSIT-GRU's reliability on the proposed CoV prediction.

The F-Measure, specificity, and Negative Predictive Value (NPV) achieved by the ERSIT-GRU are 96.12%, 96.48%, and 96.32%, which was a higher performance when compared to the baseline GRU, BiLSTM, LSTM, and RNN approaches. Hence, Figure 5.10 proves

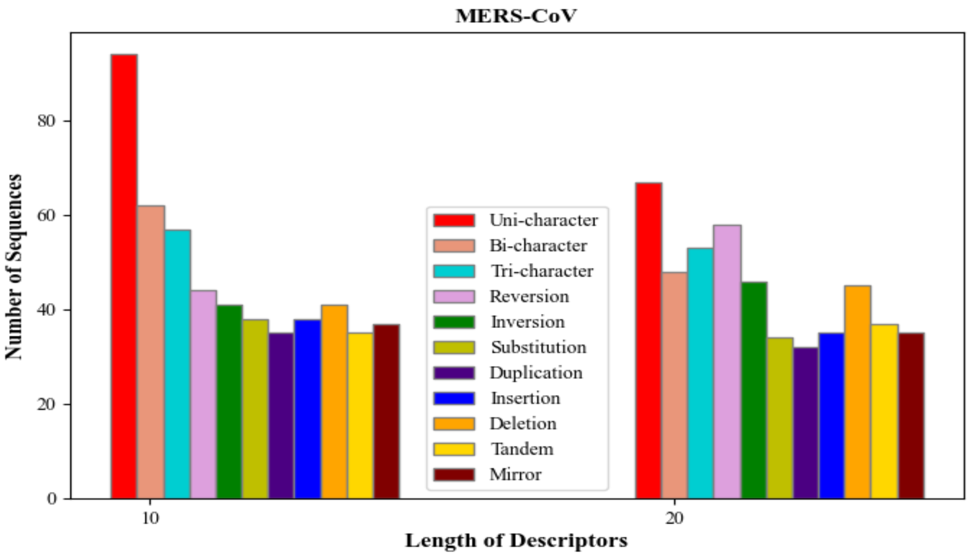


Figure 5.7: Repeat Sequence Illustration of MERS- CoV

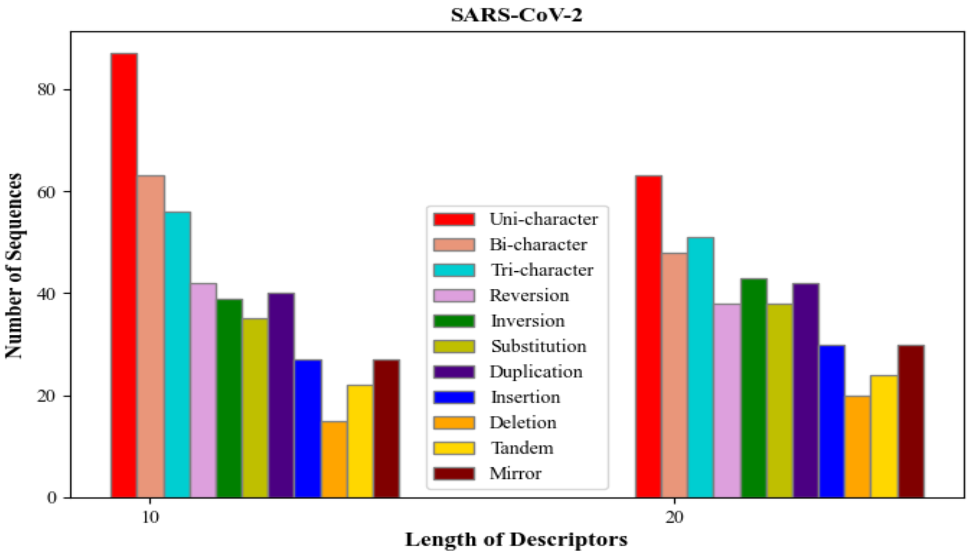


Figure 5.8: Repeat Sequence Illustration of SARS-COV-2

that the proposed technique correctly detects the normal and the new variants of CoV.

Table 5.3 displays the time taken to train the proposed and prevailing algorithms for disease prediction. The ERSIT-GRU takes 5805ms, 16203ms, and 21006ms less time than the GRU, LSTM, and RNN approaches. This proves that in the proposed scheme, the training time was less.

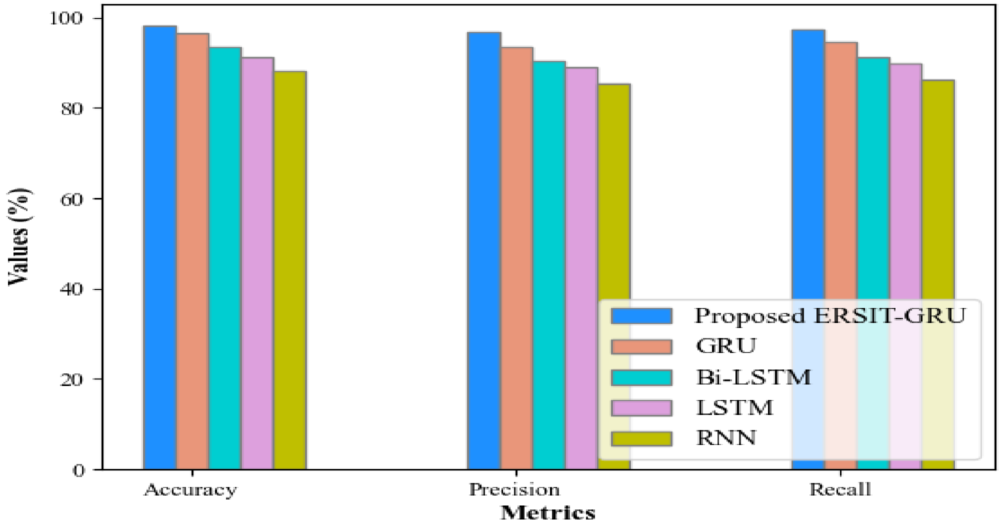


Figure 5.9: Accuracy, precision, and recall analysis

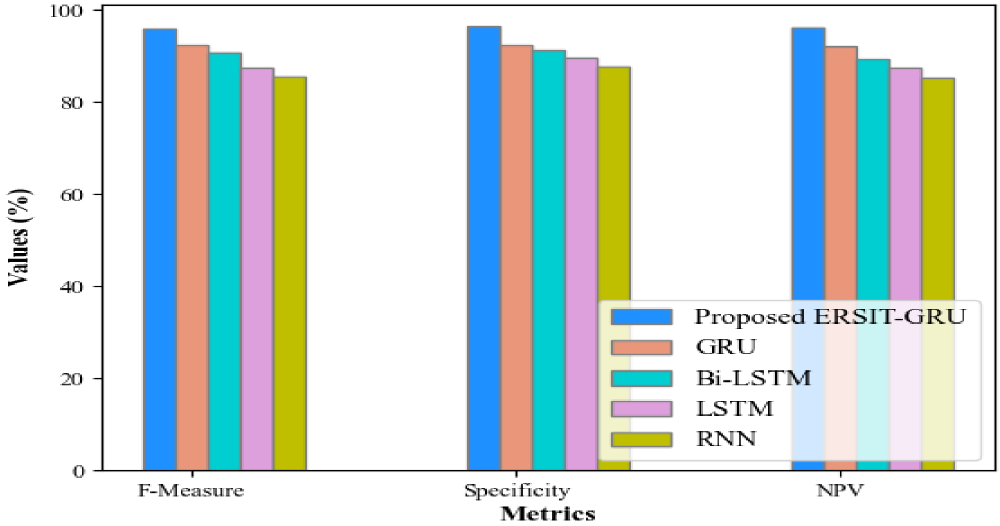


Figure 5.10: Experimental results outcome of F-Measure, specificity, and NPV

Figure 5.11 analyzes the Area Under the Curve (AUC) that signifies the quality of the predicted outcomes by the proposed and the prevailing classifiers. In this, the proposed model attained an AUV of 0.98, which shows its dominance over the other classifiers.

In Figure 5.12 , the Receiver Operating Characteristics (ROC) curve displays the correctness of separating the diseased and the normal classes by the proposed and conventional

Table 5.3: Training time during execution process for different model with proposed model

Techniques	Training time (ms)
Proposed ERSIT-GRU	77451
GRU	83256
Bi-LSTM	87457
LSTM	93654
RNN	98457

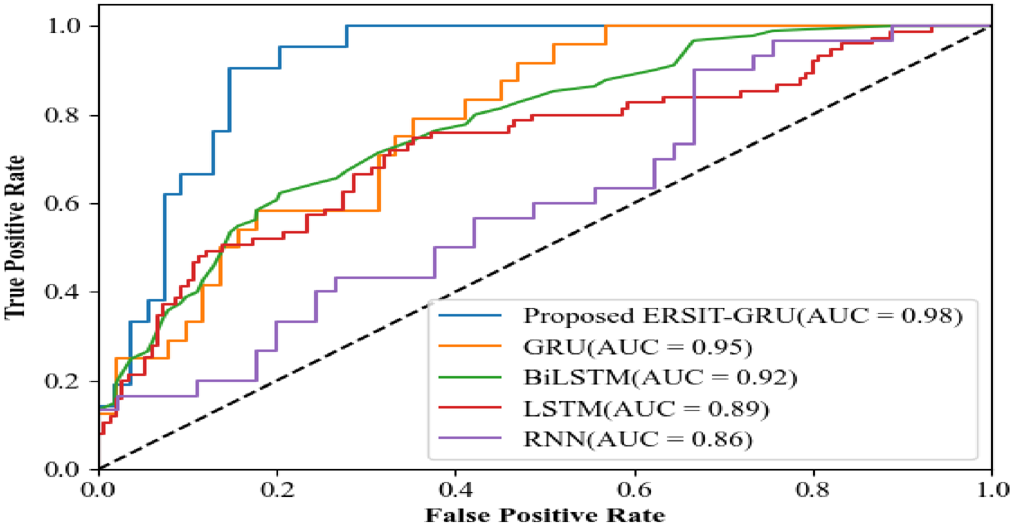


Figure 5.11: EAUC outcome for different models repeats in covid-19

techniques. The ROC characteristics achieved by ERSIT-GRU are 3.06% higher than the GRU model. Thus, the accurate prediction of the disease and normal class by the ERSIT-GRU technique is proved.

Table 5.4: Experimental results of error rate, FPR, FNR

Techniques	Error Rate (%)	FPR (%)	FNR (%)
Proposed ERSIT-GRU	**1.89**	**3.26**	**3.75**
GRU	3.78	7.15	8.12
Bi-LSTM	7.48	12.48	11.84
LSTM	13.48	17.34	16.84
RNN	17.65	23.27	22.48

The error rate, False Positive Rate (FPR), and False Negative Rate (FNR) of the ERSIT-GRU and other traditional classifiers are displayed in Table 5.4. In this, the RNN shows

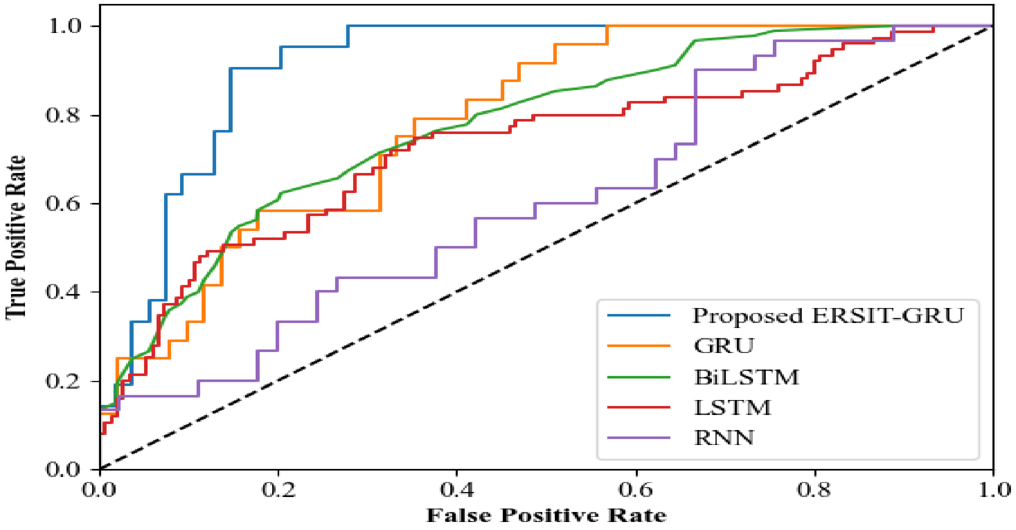


Figure 5.12: ROC analysis for different models with proposed model

higher FPR and FNR values; this makes it less reliable for disease prediction. However, when analogized to other algorithms, the ERSIT-GRU achieved less error rate (1.89%), FPR (3.26%), and FNR (3.75%).

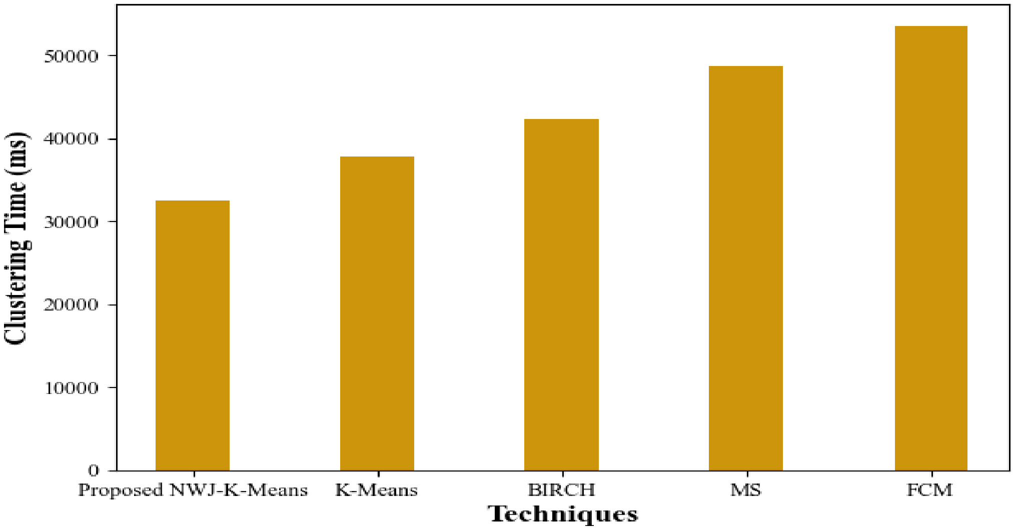


Figure 5.13: Clustering time analysis for different models

The clustering time of the proposed NWJ-K-Means (Needleman-Wunsch Jaccard K-Means) and prevailing K-Means, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Mean-Shift (MS), and Fuzzy-C-Means (FCM) are revealed in 5.13. Among existing algorithms, the K-Means achieved less clustering time (37845ms). However, the NWJ-K-Means attained less time, which makes it more suitable for sequence clustering.

Table 5.5: Sequence Tree Construction Time (STCT) of the proposed ENJA

Techniques	STCT (ms)
Proposed E-NJA	4574
NJA	8654
BST	11245
AVL	14542
Splay	17658

Table 5.5 depicts the Sequence Tree Construction Time (STCT) of the proposed ENJA and conventional NJA, Binary Search Tree (BST), Adelson-Velskys and Landis (AVL), and Splay. The STCT of the E-NJA is 4080ms, 6671ms, and 13084ms lower than the NJA, AVL, and Splay approaches. This exhibits the ENJA's time efficiency during sequence tree construction.

5.4.4 Comparative analysis

Here, the proposed scheme's accuracy is analyzed with the existing schemes of [152], [153], [154]. Figure 5.14 clearly shows that the proposed system predicted the new variants of the CoV centered on the genome sequence analysis with a higher accuracy rate (98.15%). However, the prevailing models in [152], [153], [154] predicted the disease variants with 2.05%, 0.19%, and 16.85% less accuracy than the proposed model". This clearly proved the efficiency of the proposed model gives the best model compared with other existing models. In the future will also take into account other advanced real-time patient data. We will investigate the relationship between temperature, humidity, and topography and COVID-19's distribution over cities and countries. Future studies may further look at the factors impacting the recovery status of COVID-19 for diffrent types of DNA repeats.

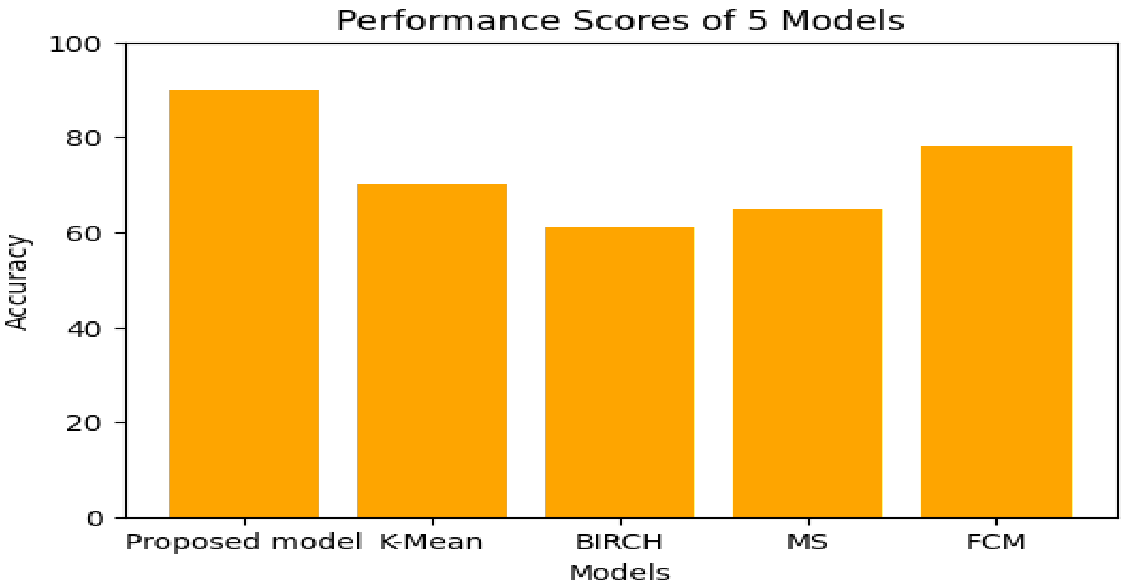


Figure 5.14: Comparative analysis of existing models with Proposed model

5.5 Summary

In this chapter, an ERSIT-GRU model is proposed for predicting new Repeats variants of CoV. Here, and E-NJA techniques were proposed for sequence grouping and sequence tree generation. Next, using the ERSIT-GRU algorithm, the disease type was classified. On the DNA dataset, the proposed approaches were experimentally evaluated. During the experimental analysis, the proposed ENJA and NWJ-K-Means performed their operations in 4574ms and 32564ms. Subsequently, the proposed ERSIT-GRU achieved higher accuracy, precision, recall, f-measure, NPV, and specificity with less training time (77451ms). Lastly, the proposed model's superiority is proved by the comparative analysis. The sequences of the gene are analyzed manually in this work, which is a time-consuming process. Hence, in the future, the unsupervised approach can be included in the proposed system for enhancing Coronavirus repeat disease prediction.

Chapter 6

Genome-wide Analysis for Tandem Repeat and Substitution Errors to Detect Covid-19 using Harris Hawks Optimization

Here, we assess the efficacy of our model, Harris hawks optimisation (HHO), by measuring its accuracy, F1-score, recall, precision, and specificity, among other indicators.Our Model stands out from the others because it achieves a 97 accuracy rate, which is higher than any previous work on the highly diverse COVID-19 dataset. The experimental results show that our method outperforms the competing algorithms. Research has demonstrated that the COVID-19 genome has the Tandem Repeat pattern AATCC more frequently than any other pattern, making the proposed methodologies essential options for identifying disease Tandem repeat patterns in the SARS-CoV-2 genes.

6.1 Introduction

Repeated DNA sequences that are next to one another in a genome are known as tandem repeats (TRs). Deoxyribose nucleic acid (DNA) is the building block of all life on Earth. It contains the instructions for creating and maintaining life. The instructions are expressed

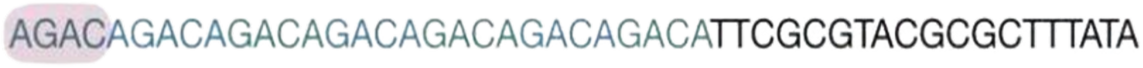


Figure 6.1: Short Tandem Repeat sequence of AGAC

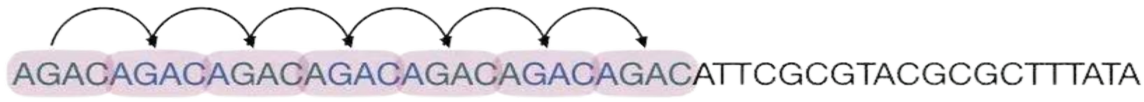


Figure 6.2: Located one after another, Short Tandem Repeats

using materials that are generated from DNA or RNA. Adenine (A), Cystosine (C), Guanine (G), and Thymine (T) are the four letters that make up DNA, which is also called the nucleotide bases [155]. The four nucleotide bases are organised in what are known as genome sequences. The DNA sequence is another name for the DNA nucleotides that make up the genome. The sample Genome sequence is AGCGTTGATCGTTGACGAGA. Bioinformatics' most important subfield, dealing with the study of living organisms, is genome sequence analysis. Repeating DNA strands of varying pattern sizes make up the human genome [156], [120]. When the number of repeats in a TR area grows in succeeding generations, this process is called an expansion of TRs [157]. Mutation in Genome sequences. The term "mutation" is used to describe alterations to the DNA sequence, such as the addition, deletion, or replacement of nucleotide bases[158]. SARS-CoV-2 is a member of the Coronaviridae family, which also includes MERS-CoV and SARS-CoV-1 [159], [160]. All humans have STRs of variable length at specific known locations in the genome (known as loci). The technique of DNA profiling involves analyzing DNA evidence to identify potential culprits, as every person's DNA is unique.

The DNA sequence may be accidentally duplicated and inserted after the original substring in a mistake known as a tandem duplication [120]. As an example, we may get ACGCGT from ACGT. The substring being copied, 2 in the previous example, is then

AGACAGACAGACAGACAGACAGACAGACATTCGCGTACGCGCTTTATA

Figure 6.3: Short Tandem continuously Repeated in the sequence

duplicated. There have been studies on both fixed-length duplication[161], [117] and bounded-length duplication [11], [117], where the duplication length is constrained from above [162] suggested error-correcting codes for duplications with a maximum length of 3, which is the most relevant case to this research [115]. Demonstrated that these codes had an asymptotically optimum rate. Any time a symbol is added to, removed from, or replaced in a sequence, is known as an edit event. The literature has examined substitution errors in conjunction with fixed-length duplication errors, such as those that occur only in the inserted copies (representing the copying mechanism's noisiness during duplication) [163], [164] and those that can happen anywhere in the string [165]. We focus our attention on fixing mistakes that may emerge from channels with many short-duplication faults (those with a length of little more than three) and a single edit error (that can happen anywhere in the string). Taking into consideration a single editing mistake gives significant insights about the interactions between to explore the general case of t edit mistakes, as well as edit and duplication errors. The input ACG may be transformed into the following basic example using this channel: ACG → ACCCG →ACTCG → ACTACTACTCG →ACTCTACTACTCG. The underlined copies indicate duplication, and the symbol T is a consequence of copies of the substitution C → T. An infinite portion of the output word may be impacted by the mistakes since an indefinite number of duplication's are conceivable; for instance, the replacement symbol may appear several times. But we prove that the harmful consequences of the mistakes may be contained by building and preparing the channel's output appropriately, based on the notion that brief tandem duplication's [165]. To start, we'll build error-correcting codes that can fix a replacement and a number of short duplicates. We will next demonstrate that by changing deletion and insertion mistakes to substitution errors, the same code may fix an edit error and an unlimited number of duplicates. The DNA sequence of a pattern that repeats perfectly and sequentially is called an exact tandem repeat. Several techniques have been devised to identify such repetitions. Some examples of such work include a programme that can identify pre-specified patterns and an efficient parallel algorithm provided by Tandem repeat [163]. Additionally [117] presents a vectorizable technique. Since events like mutations, insertions, and deletions will make the copies defective, most repetitions are approximate rather than accurate. Approximate tandem repeats (ATRs) detection is therefore a major area of concentration in the present study. The pattern size of microsatellites, which are also called short tandem repeats (STRs), is limited to 1–6 bp, and their length is less than 150 bp. In the human genome, there are more than 10,000 STR sequences that have been published so far. Analysis of short tandem repeats (STRs) has garnered significant interest in bioinformatics and forensics since the late 90s. Segments that have two or more nearly identical copies of a nucleotide sequence are called tandem repeats [117]. The repeat unit (or sequence pattern) TGGCA occurs three times throughout a STR, and the pattern breadth may be anything from two to sixteen base pairs (bp). One use of STR is genetic fingerprinting [166]. Their link to hereditary illnesses has also been the subject of recent studies [167]. For instance, a trinucleotide pattern known as CAG may cause an explosion in the copy number, leading to disorders like Huntington's disease that impact muscular coordination. We expand upon STR by introducing SAR, which permits gaps between adjacent repetition units and generalises it. Mutations and mistakes in genetic modifications during evolution are common causes of such inter-unit insertions.

Multiple errors occurred within one strand: We concentrate on fixing mistakes that may result from channels having one unconstrained substitution error and several short duplications, or duplications of length no more than three [168], [169], [118]. "When analyzing the general case for tandem substitution mistakes, it is useful to consider a single substitution error since it gives key insights into the interplay between substitution and duplication errors. If we take ACG as an example of an input and change it to ACTCTACTCG, we can see that the symbol T appears because of several instances of the substitution C→ T. There is no limit to the number of potential duplications, therefore the wrong symbol may appear several times in the output word and mistakes may impact an unlimited number of segments.

In recent years, illnesses have been linked to atypical STR enrichment patterns. Dinucleotide GA repeats of lengths 13 to 16 (in the SH2D2A gene) are more common in MS patients compared to controls, according to research by [8]. When comparing the allelic frequency distribution of CpG-CA repeat lengths between controls various diseases like Alzheimer's patients, another study10 found a change. In a similar vein, Myers11 con-

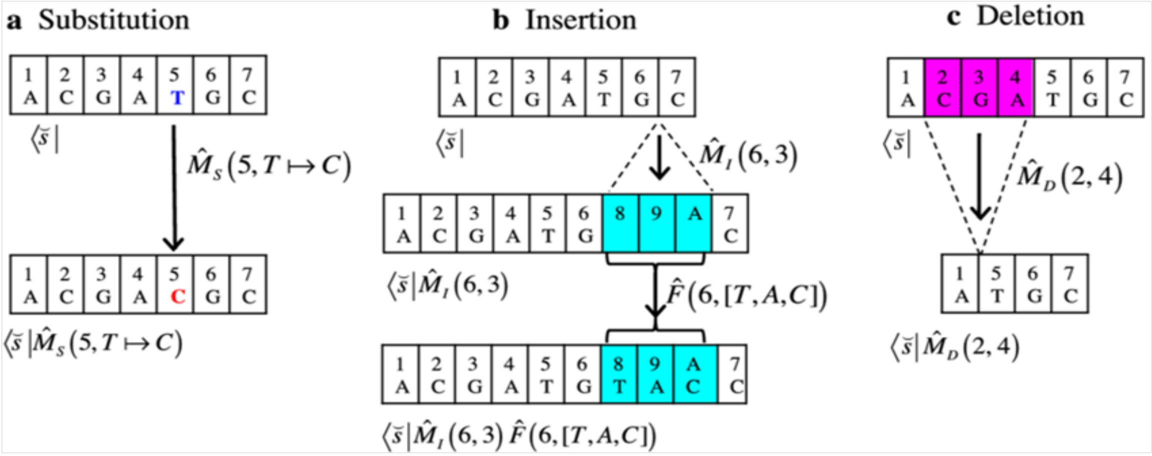

a Substitution
b Insertion
c Deletion
1 A 2 C 3 G 4 A 5 T 6 G 7 C
$\langle \tilde{s}|$
$\hat{M}_S(5, T \mapsto C)$
1 A 2 C 3 G 4 A 5 C 6 G 7 C
$\langle \tilde{s}|\hat{M}_S(5, T \mapsto C)$
$\hat{M}_I(6,3)$
1 A 2 C 3 G 4 A 5 T 6 G 8 9 A 7 C
$\langle \tilde{s}|\hat{M}_I(6,3)$
$\hat{F}(6,[T,A,C])$
1 A 2 C 3 G 4 A 5 T 6 G 8 T 9 A A C 7 C
$\langle \tilde{s}|\hat{M}_I(6,3)\,\hat{F}(6,[T,A,C])$
$\hat{M}_D(2,4)$
1 A 5 T 6 G 7 C
$\langle \tilde{s}|\hat{M}_D(2,4)$

Figure 6.4: a. A substitution operator, $\hat{M}$ S : T $\rightarrow$ C. The residues before and after the substitution are in boldface in blue and red, respectively. b. An insertion operator, $\hat{M}$ I 6;3, and a fill-in operator, $\hat{F}$ 6; A;C 1/2. The inserted sites are shaded in cyan. c. A deletion operator, $\hat{M}$ D 2;4

firmed that the gene typically contains the trinucleotide sequence CAG around 20 times, but that the onset of Huntington's disease requires an approximate doubling of repetitions to 40 or more. For example, in SCA[170] hexa nucleotide GGCCTG repeats often vary from 5 to [171], [172] for SCA12 trinucleotide CAG repeats typically range from [173], and for SCA10 pentanucleotide ATTCT repeats typically range from [174]. Viewers interested in comprehensive summaries of the existing literature on STRs may consult the cited works [174], [175]. According to these findings, the frequency of di- and tri-nucleotide repeats varies across genes associated with health and sickness. Therefore, STR frequency patterns might serve as genetic identifiers[163], examined in dinucleotide frequency trends in whole-genome sequences from over 1300 bacterial species. This analysis was conducted in quite recently. The AC, AG, CA, CT, GA, GT, TC, and TG dinucleotides are more consistently found in different genomes than in other dinucleotides, which differ significantly among species [117] conducted an independent analysis of 22 coding markers often used in DNA species categorization. Trinucleotides ATG, TAA, TAG, and TGA were part of these characteristics; the last three trinucleotides acted as stop codons. Characteristics that are effectively utilised in C. The authors of the study claim that Elegans isn't always a good fit for humans. Prior research indicates that ATA, ATT, and TAT are secondary structures, whereas TTA is preferred by alpha helix secondary structures. On the other hand, trinucleotide AAT thrives in beta bulges, which develop when a beta sheet's typical hydrogen bonding is disrupted. Protein synthesis can be stopped by trinucleotides such as TAA, TAG, and TGA. Our initial examination of the approach suggests that genes associated with neurological illness do not randomly exhibit an AA, AT, TA, TG, or TT enrichment pattern. We show that the trinucleotides in issue are more prevalent in genes that encode families of neurological diseases compared to all human genes. Protein misfolding causes the creation of beta sheets and the breakdown of alpha helices, as previously shown in this and other significant findings[176]. Some neurological illnesses have also been linked to gene alterations that cause premature termination codons [177], because these dinucleotide sequences are part of the trinucleotide sequences that are involved in protein misfolding and are preferred in secondary structures [177]. Given that trinucleotide repeat patterns[178] may represent possible genetic traits in neurological disease family genes, we chose to examine their frequency distribution in relation to premature termination codon mutations and, by extension, certain neurological illnesses.

6.1.1 Preliminary Definitions and Notations

Here you will find definitions of the terms and symbols that will be utilised throughout the project. However, that section provides a more comprehensive definition of the language that is relevant to each component. In our work, we exclusively handle trinucleotide sequences. Other forms of repeating sequences include hexa-, penta-, tetra-, and mononucleotide repeats. The terms repeat unit (RU) and STR are often used interchangeably. Each part of a STR is most commonly referred to as a trinucleotide (RU).In addition, because an RU possesses three nucleotides of a STR-dependent genetic characteristic, there are 64 possible pairings (43= 64). The RC of an RU is the total number of times it shows up in the subsequent parts. A repetition sum (RS) is produced for every RU by adding up the RCs [178]. Additional classifications for RCs include maximum repetition count (max RC), lowest repeat count (min RC), and most frequent (most RC), all of which indicate different repeat frequencies[179]. Lastly, a repetition is the term used to describe the growth of an RU based on the RC. Here we will use a gene sequence as an example to show the terminology and their relationships as shown6.4 . There are 64 nucleotides in the sequence, and two RUs, CAT and AAG, are present[180] due to their more than three repetitions. There are a number of recommended cutoffs for the repetition count in the literature; in this case, we are using the four-point minimum that Lai and Sun[181] proposed for humans. Furthermore, further statistical investigation on human and neurological genes revealed that almost all repetition units had a minimum RC of four. Thus, RU CAT at positions i = 0, i = 21 and i = 34 have RCs of 6, 4 and 4, respectively. In addition, RC ranges from 4 to 6, with 4 being the most common. The result of combining RU CAT with the RCs is a 12-repeat sequence and an 18-repeat sequence (CATCATCATCATCAT-CAT), (CATCATCATCAT). Next, RU CAT's RS is 14. It may be inferred that this gene exhibits higher amounts of RUCAT (indicating strong CAT expression) and lower levels of RU AAG (has low expression levels of AAG).

6.2 Materials and Methods

In this section, we collected datasets and introduced various parameters.

6.2.1 Data collection

One of the most important sources of diverse coronaviruses is the 2019 Novel Coronavirus Resource (2019nCoVR) maintained by China's National Centre for Bioinformation [181]. The information used in this study comes from several sources, such as CNCB/NGDC, GISAID, NMDC, NCBI, [182]. These data Centers are all fully integrated with 2019nCoVR (NGDC). In addition, it provides visualization tools for the results of genome variation studies using all of the collected 2019-nCoV strains and compiles a wide variety of relevant material for scientific dissemination, including scientific literature, news, and media stories. The set contains a large number of human coronaviruses, such as [183], SARS-COV-2, NL63-CoV, HKU1-CoV, AlphaCoV, BetaCoV-1, MERS-CoV, and 229E-CoV. With the 1000 viral sequences[184], we supplemented our analysis with 592 genome sequences from other human coronaviruses. All human coronavirus genome sequences, apart from SARS-CoV-2, are available for download[117]. To guarantee that balanced and unbalanced datasets are prepossessed is done. To eliminate duplicate sequences, [117] we created some Python scripts to remove genomic sequences with the same accession number. For each Tandem Repeats, sequences containing any additional nucleotides outside A, T, C, and G were disregarded since their presence would obscure the genetic signature encoded in dinucleotide frequencies, datasets collected from NCBI. The details of data sets are shown in the table 6.1www.ncbi.nlm.nih.gov/nuccore/1798174254, https://gisaid.org/CoV2020. The reference genomes and test sequences of various CoVs for evaluating the mutational bias are collected from NCBI. There are 13 human slow codons (ACC, AGT, CAT, CCC, CGC, CTC, GAT, GCC,GGT, GTC, TCC, TGT, TTT) [185]. By the combinations of these 13 slow codons, a total of 169 slow di-codons were formed. For example, by combing ACC, AGT, and CAT we can form six slow di-codons ACCAGT, AGTACC, ACCCAT, CATACC, AGTCAT, and CATAGT. Two consecutive slow codons can reduce the translation rate extremely.

Table 6.1: Data Sets for Tandem repeats in covid-19 for both Positive and Negative

SlNo.	strains	No. of Samples	Name of the Strain
1	HCov-229E	30737	Human coronavirus Common
2	HCoV-NL63	27055	Human coronavirus Infection
3	HCoV-OC43	29604	Mild Lower Respiratory Infection
4	MERS-CoV	27553	Middle East Respiratory Syndrome
5	SARS-CoV	30119	Severe Acute Respiratory Symdrome
6	2019-nCoV	29711	Novel Coronavirus acute respiratory disease
7	SARS-CoV-2	29797	COVID-19+

6.3 Problem Definition

As a binary issue, we can find the sequence motifs of the SARSCoV2 gene and where it is located among other coronaviruses. The majority of prediction tasks in bioinformatics and computational biology can be conveyed as classification problems, like binary (two-class) or multi-class classification tasks[176]. Designing a biological sequence problem is a skill that requires for building an efficient bioinformatics classifier. Here, identification of SARS-CoV-2 can be established as a binary classification problem, with dataset D of N samples $D = D = x_i, y_i N_i = 1$, x_i indicates feature set, that could be considered as a 4 X N dimensional matrix. The four nucleotides that make up DNA sequences are adenine (A), guanine (G), cytosine (C), and thymine (T) [186]. These four base pairs make up the sequence A, G, C, T. These base pairs can be represented by the one-hot vectors [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], and [0, 0, 0, 1] respectively. The class label yi is 1 for SARSCoV2 and 0 for all other sequences. These sequences are similar to the other sequences in the Coronavirus family , classical analysis of them may produce inaccurate results. Among the several Coronavirus gene sequences, the primary goal of this study is to accurately predict the SARS-CoV-2 gene sequence. We also found the frequent patterns which are induced in the SARSCoV2 sequences.

6.3.1 Standard Harris Hawks Optimization (HHO) Swarm intelligence

This optimization algorithm is invented by the inspiration of Harris' Hawks hunting mechanism. One notable predatory bird that still lives in somewhat stable populations in the southern part of Arizona, USA, is the Harris' Hawk [11], [8]. The two primary steps of HHO, like other meta-heuristics, are diversification (exploration) and intensification (exploitation). This structure is reminiscent of the way Harris hawks adapt their assault to different prey conditions [135].

6.3.1.1 Exploration phase

Based on the prey's escape energy E, the HHO algorithm may switch between exploration and exploitation [129]. It is possible to define the mathematical model for the prey's energy using the equation

$$E = 2E_0(1 - t/T) \tag{6.1}$$

E_0 varies at random within the interval with each iteration, where E_0 is a variable in HHO, T is the total number of sequences, and t is the number of errors that happened in each sequence. A linear relationship between number of iterations and the reduction of E is required for this.

6.3.1.2 Intensification phase (exploitation)

Harris hawks drop down unexpectedly to get their prey. On the other hand, the victim may easily get away from the danger. Here is how r reflects the prey's chance of evading attack: Escape capability =successfully DNA Changes , Un successfully DNA Changes

6.4 Proposed Approach

In this section, we introduced the proposed model architecture, and evaluation metrics.

6.4.1 Proposed Architecture

The stream of bytes vector of size NxN that integrated an N-length Genetic code with locations for the strands A, G, C, and T serves as the basis for this architecture. A one-hot encoding approach is used to transform the input DNA sequences to numerical information since HHO can only interpret numerical data. This method, as described in reference [187], is used to transform the text into numerical data that can be processed by the network. The features in the suggested Frequency-based Feature [188] are divided into the following categories according to the characteristics of the extraction techniques: characteristics according to storage Based on Base(s) Frequency and Features.

6.4.2 Features based on storage

Different species have different genome sizes and lengths, which will be useful for categorization. The length and size of the genome are the two characteristics that make up this category. Both of these characteristics are connected to one another in some way. DNA storage is potentially less expensive, more energy-efficient and longer lasting. Studies show that DNA properly encapsulated with a salt remains stable for decades at room temperature and should last much longer in the controlled environs of a data center.

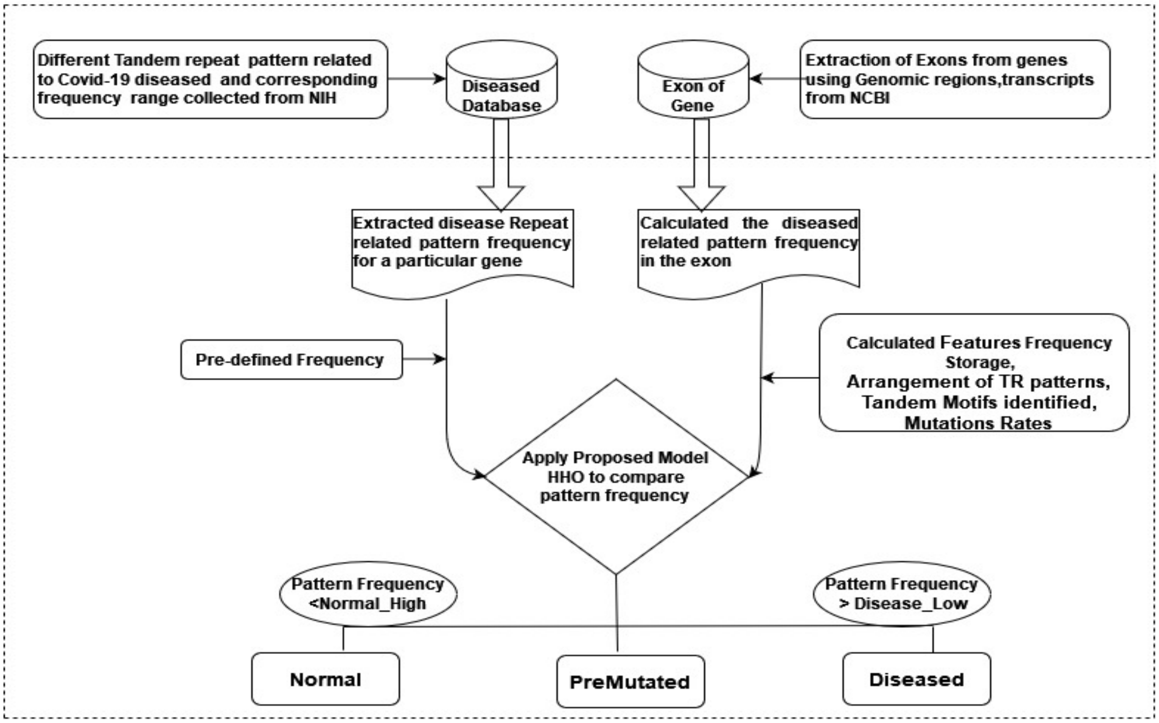

Different Tandem repeat pattern related to Covid-19 diseased and corresponding frequency range collected from NIH
Diseased Database
Exon of Gene
Extraction of Exons from genes using Genomic regions,transcripts from NCBI
Extracted disease Repeat related pattern frequency for a particular gene
Calculated the diseased related pattern frequency in the exon
Pre-defined Frequency
Calculated Features Frequency Storage, Arrangement of TR patterns, Tandem Motifs identified, Mutations Rates
Apply Proposed Model HHO to compare pattern frequency
Pattern Frequency <Normal_High
Pattern Frequency > Disease_Low
Normal
PreMutated
Diseased

Figure 6.5: Proposed Architecture for Normal, PreMutated, Diseased

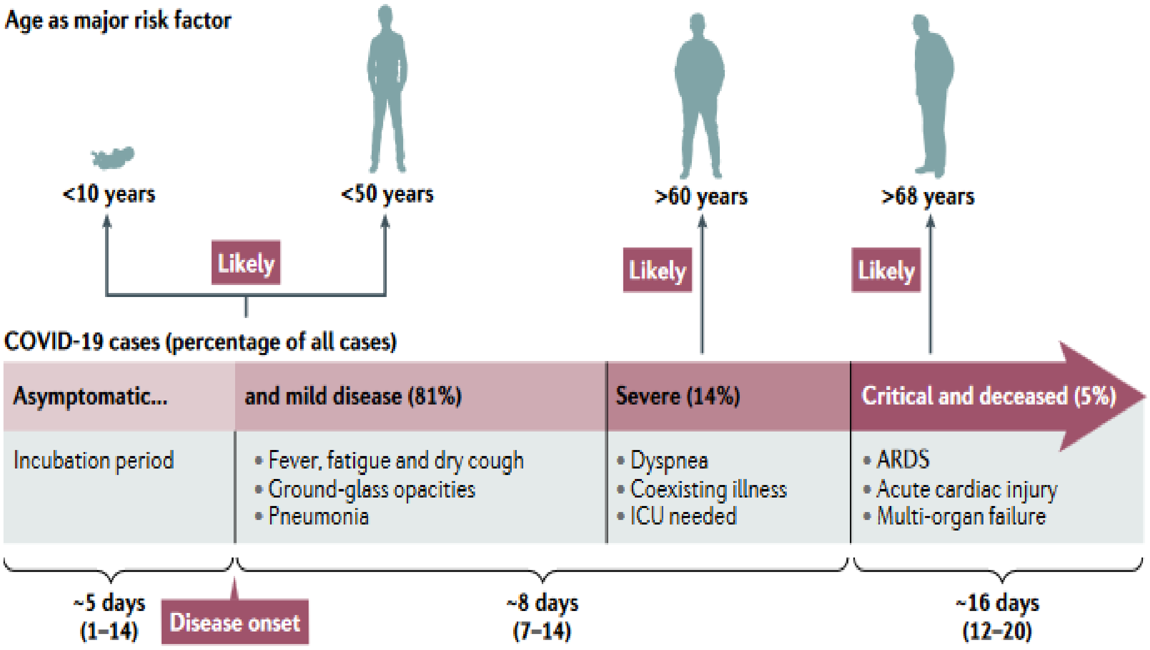


Figure 6.6: Clinical features of COVID-19. Typical symptoms of coronavirus disease 2019 (COVID-19) are fever, dry cough and fatigue and in severer cases dyspnea, causes Many infections, in particular in children and young adults.

6.4.3 Length

Assume if 'S' is the sequence of the genome and 'B' is a base in that sequence. The input Genome sequence's length is referred to as length. A human DNA can have up to 500 million base pairs with thousands of genes.

$$Length(S) = Totalnumberofbases(B) \tag{6.2}$$

6.4.4 Size

The sequence of nucleotides makes up a genome. The length of the sequence is directly proportional to its size when represented in terms of bytes as each base requires 1 byte for storage. The suggested feature extraction method uses Equation 3 to convert sequence size to KB (Kilo Bytes)(1 Kilo Byte =1024 Bytes).

$$S_i = Length(S)/1024 \tag{6.3}$$

Table 6.2: Dataset Size (KB)length for tandem repeats in Covid-19

S.No.	Features	HCoV-229E	HCoV-NL63	HCoV-OC4	MERS-CoV	SARS-CoV	2019-NCoV	SARS2-CoV2
1	Length	30737	27055	29604	27553	30119	29711	29797
2	Size (in KB)	30	26	29	27	29	29	29

6.4.5 Features based on Base(s) Frequency

N Count, Base Count, Dimer Count, and Codon Count are all traits that fall into this category.

N Count: Genome sequences consist of a Non template base 'N' in addition to the four nucleotide bases(A,C,G and T). The feature N Count (NC) refers to the number of occurrences of Non-template base (N) in the Genome sequence as given in equation 6.4

$$(S) = numberofoccurencesofBaseN \tag{6.4}$$

BaseCount: The number of times each nucleotide base (A, C, G, and T) appears in the sequence of a genome is called the base count. The BC(S) is then calculated in the following way: Bases = A,C,G,T.

Dimer Count: A dimer is formed when two separate bases come together. The Dimer Count (DC) feature counts how so many times each potential dinucleotide combination appears in the genome sequences.There are total of 16 Dimers in a Genome sequence. Dinucleotides ={ AA, AC, AG, AT, CA, CC, TT, CC, TT,}.

Codon Count: To produce a codon, three separate bases must come together.which is the total number of trinucleotide pairings found in the sequence. There are total of 64 codons in a Genome sequence. Codons={ Number of all possible combinations of trinucleotides }

Codons={ AAA, AAC, AAG, AAT, CCA, CCC, CCG, CCT, TTG, TTT }.

6.4.6 Features based on arrangement of patterns

Most Repeat Pattern Count (MRP): The genome sequence has several instances of MRP, which is referred to as the Most Repeat Pattern Count. Set to 4, the MRP count indicates the frequency of the most repetitive tetra nucleotide pattern inside the sequence.

Tetra nucleotides = AAAA, AAAC, AAAG, AAAT, CCCC, CCCA, CCCG..., TTTT

MRP(S) = Number of occurrences of most repeat tetra nucleotide pattern.

6.4.7 Exon extraction

Genes are the basic physical and functional units, act as instructions for creating a protein and are composed of Exons and Introns. Exons are the protein-coding regions of the gene; they are interleaved with non-coding regions called introns. Exon detection is crucial for proper disease detection and diagnosis. The existing tools for exon detection are Genome Scan [189], Genscan server [190], Gene finder [112], Augustus server [191], Ensembl [192], National Center for Biotechnology Information (NCBI) gene table [193], and spidey

Table 6.3: Tandem Repeat for Base count and Dimer count

S.No.	Features	HCoV-229E	HCoV-NL63	HCoV-OC4	MERS-CoV	SARS-CoV	2019-NCoV	SARS2-CoV2
K-mer-Base -Count								
1	A	7458	7985	7412	7458	7986	9854	8547
2	C	7724	9854	8231	8741	9471	7414	7654
3	G	7584	7123	7321	7064	7145	7954	7658
4	T	9851	6854	8744	9888	8744	8574	8414
K-mer-Dimer -Count								
5	AA	1243	1244	2415	2724	2912	1811	2144
6	AC	1471	2144	1247	2144	2021	1877	1421
7	AG	1244	1022	2214	1054	1035	877	988
8	AT	655	688	621	654	1111	587	501

[194]. Genome Scan [189] is a program for identifying the exon-intron structures of genes in genomic DNA sequences from a variety of organisms, with a focus on human and other vertebrates. NCBI gene table generates the index range and length of exons in the gene sequence and also contains index range and length of coding regions as shown in Table 6.4.

Table 6.4: Tandem Repeat Gene for mRNA NM_0020232.1 of AFF2 gene

Exon Range	**Coding Range**	**Exon**	**Coding**	**Intron**
1-528	482-528	526	47	150854
151383-151515	151383-151515	133	133	9776
161292-162152	161292-162152	861	861	147107
309260-309304	309260-309304	45	45	27726
337031-337117	337031-337117	87	87	5232

6.4.8 Cross-validation

Cross-validation is a model quality evaluation method, which is better than the residual evaluation approach, useful to avoid overfitting and underfitting. K-fold cross-validation randomly divides the dataset samples into k, approximately equal size folds or groups. Iteratively, one fold at a time treated as a test set and the model is trained on remaining k-1 folds. We follow a representative tactic to choose k values like 10 to evaluate the HHO model on different human Covid-19 datasets experiments. We also evaluated the proposed model with Leave-One-Experiment-Out (LOEO) cross-validation. We repeat the LOEO approach, by using one serum metagenomic experiment as a test set and the remaining four

Table 6.5: Different patterns and its corresponding genes, chromosomes and Normal, Premutated and Diseased frequency ranges.

Chromosome Number	Gene Name	Pattern	Normal Low	Normal High	Premutated Low	Premutated High	Diseased Low	Diseased High	Disease Name
12	ATN1	CAG	6	35	36	47	48	80	DRPLA
4	HTT	CAG	10	35	36	39	40	NL	HD
6	ATXN1	CAG	4	39	40	50	51	NL	SCA1
12	ATXN2	CAG	0	30	31	32	33	NL	SCA2
14	ATXN3	CAG	12	43	44	52	53	NL	SCA3
8	CACNA1A	CAG	4	18	19	19	20	23	SCA6
3	ATXN7	CAG	4	17	28	33	34	NL	SCA7
6	TBP	CAG	25	42	43	48	49	66	SCA17
6	TBP	CAA	25	42	43	48	49	66	SCA17
X	AR	CAG	0	36	37	37	38	NL	SMBA
5	PPP2R2B	CAG	0	42	43	50	51	NL	SCA12
14	PABPN1	GCN	0	10	11	11	12	17	OPMD
19	DMPK	CTG	5	37	38	49	51	NL	DM1
3	CNBP	CCTG	0	25	26	74	75	1100	DM2
X	AFF2	CCG	4	40	50	200	200	NL	FRAX-E
9	FXN	GAA	5	33	34	65	66	1000	FRDA
X	FMR1	CGG	5	40	55	200	201	NL	FXS
16	JPH3	CTG	6	28	29	43	44	59	HDL2
16	JPH3	CAG	6	28	29	43	44	59	HDL2
13	KLHL1	CGG	0	49	50	70	80	1300	SCA8
22	ATXN10	ATTCT	10	32	280	850	851	4500	SCA10
20	NOP56	GGCCTG	3	14	15	649	650	NL	SCA36
21	C9orf72ÿ	GGGGCC	0	30	31	39	40	NL	Atony

experiments are used to train the model. We also evaluated the model by considering 5 serum experiments as test set and remaining 4 metagenomic covid datasets experiments are used to train the model.

6.4.9 Disease database design for Tandem repeats

We have collected different patterns related to Covid-19 diseases (HCov-229E, HCoV-NL63, and HCoV-OC4, MERS-CoV, 2019-nCoV etc.) from National Institute of Health genetics home reference [195], which contains only 21 out of 1480 genes that have pattern frequency-based disease information. Additionally from various existing works [196], [70], [197], [198], [199], [200], and from some standard genome projects [201] also gathered. The collected pre-mutated and diseased frequency ranges of a gene pattern corresponding to specific chromosome for accurate disease diagnosis is stored in Table 6.5. The algorithm 6.1 reads the received inputFile (.xls) that contains patterns and its corresponding frequency

Algorithm 6.1 CreateDB()

Input: inputFile that contains patterns and its frequencies.
Output: Disease database as a Hashmap.

```
workbook = getWorkbook(File(inputFile))
sheet = workbook.getSheet(0)
numRows = sheet.getRows()
for each row $\epsilon$ numRows do
    cell = sheet.getCell(row,col)
    extract gene and pattern using cell.getContents()
    key = geneName + - + Repeat pattern
    extract Chromosome number, All ranges, Disease using cell.getContents()
    value = append(All frequency ranges,Chromosome number,Disease name)
    Disase Data Base.put(key,value)
```

ranges. It generates Disease database as an output. Disease database is a hashmap, which is in the form key-value pairs. The key is <gene name, pattern> and the value is frequency ranges. The different methods (getContents(), getRows() etc.,) used to handle the data in workbook. Now the Create Data Base(filePath) reads the database and stores it in hashmap in the form key-value pairs. The key is Tandem Repeat gene name and pattern and the value is frequency ranges.

6.4.10 Tandem Repeat Disease Prediction in genes for covid-19

In all the exons corresponding to the particular gene, the frequency of those special patterns are calculated by updating pointers and counters. Algorithm 6.3 maintains maximum pattern length plus one pointers to point different levels of Trie based on nucleotides in the exons.

6.4.11 Trie Construction for Tandem Repeat Disease Patterns

Constructed a Trie data structure for all the patterns stored in the disease database. The Construct Trie algorithm is used for the construction of a Trie 6.2. This Trie is used to find the frequency of multiple disease patterns in exons very fast by using pointers. The Trie data Structure for the diseased patterns are as shown in 6.7. If all characters of pattern have been processed, i.e., there is a path from root for characters of the given pattern, then print all indexes where pattern is present. To store indexes, we use a list with every node that stores indexes of suffixes starting at the node.

Algorithm 6.2 Construct Trie(pattern, pattern Index, root)

Input: Pattern, pattern index and rpointer.

Output: Generates a Trie data structure for the given input Tandem repeat patterns.

```
length = pattern.length()
node = root
for each level $\epsilon$ length do
    index = getIndex(pattern.charAt(level))
    if node.children[index] == null then
        node.children[index]=new Trie Node()
    node=node.children [index]
    if level == length - 1 then
        node.pattern Index = pattern Index
    node.is End Of Word=true
```

A trie pattern for tandem is a rooted tree where each edge is labeled with a symbol and the string concatenation of the edge symbols on the path from the root to a leaf gives a unique word (k-mer) X. We label each leaf with a set of T VNTRs that contain corresponding k-mer.

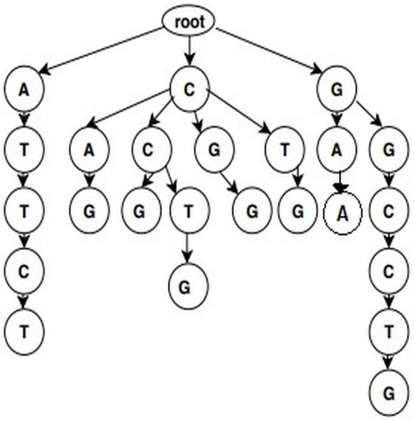


Figure 6.7: Trie for Tandem repeat disease-related patterns.

Algorithm 6.3 Proposed Algorithm for Tandem Repeat Disease Prediction in genes

Input: DNA Short Tandem Repeats Sequence from NCBI.

Output: TR Error types Feature Extraction, Base frequency, Predict the tandem repeat disease(s) status Normal, PreMuted, Diseased.

```
rPointer = new Trienode
keys[] = {CAG, CTG, CCG, GAA, GGG, CGG, CCTG, ATTCT, GGCCTG, GGGGCC}
kmerInds[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
for each i $\epsilon$ keys.length  do
    ConstructTrie(keys[i], kmerInds[i], rPointer)
TrieNode[] q = TrieNode[maxKmerSize+1]
q[0] = rootPointer
for each index $\epsilon$ keys.length do
    kmerCounts[index] = 0
diseaseDatabaseFile= path(DiseaseDb1.xls)
DiseaseDB<String, DiseaseData> = GenerateDB(disease DatabaseFile).readFile()
File[] dirs = path(Exons_NCBI).listFiles is (Directory)
for each k $\epsilon$ dirs.length do
    if dirs[k] $\neq$ null then
        for each file $\epsilon$ dirs[k].listFiles() do
            String[] filepath = file.getPath().split(\\)
            geneName = filepath[4]
            exonPath = filepath[5]
            exonName=geneExt[0]
            for each temp $\epsilon$ reader.readLine $\neq$ null do
                list.add(temp)
            minLength = Integer.MAXVALUE
            for each index $\epsilon$ keys.length do
                key = geneName + - + keys[index]
                if DiseaseDB.containsKey(key) then
            hasDisease = false
            hasPremuted = false
```

```
for each index1 $\epsilon$ list.size() & !hasDisease do
    if list.get(indexl).length() < minLength then
        continue
    for each index2 $\epsilon$ maxKmerSize do
        Tq[index2] = null
    for each index $\epsilon$ keys.length do
    exon = list.get(indexl)
    for each index2 $\epsilon$ exon.length do
        updatePointers(Tq, exon.charAt(index2))
        updateCounts(Tq, kmerCounts)
    for each index $\epsilon$ keys.length do
        checkFreqInDB(index)
if !hasDisease & !hasPremuted then
    diseaseStatus.add(geneName +- + exonName+ : + - Normal)
reader.close()
```

6.5 Computational Complexity Analysis

The time complexity of proposed method in different phases is estimated in the following subsections.

6.5.1 Extraction of Tandem Repeats

In xploration phase phase, to check different type of repeats and to write them into a file, the complexity involved is O(Td2h+Tdhdi) , where L is maximum length of a sequence, X is size of key, and P is pattern length. In Partition phase, to process the key, it costs O(X). In Extraction phase phase, the time complexity involved is O(Td2h+Tdhdi) which is a combination of reduce function O(K*(P+X)) and the maximum number of patterns(K=O(NF(L-P))) in output file, where N is number of key-value pairs and F is maximum input files. The worst case time complexity of Harris Hawks optimization (HHO) algorithm. is O(N*F*(L-P)*(P+X)). If P and X are constants, then worst case complexity is O(NFS). If F,P,X,m,p,r are constants, then worst case time complexity to find different features using HHO is O(N*L).

6.5.2 Generation Features

Disease database design: The worst case time complexity to design disease database is O($\Im$), where $\Im$ is the number of diseases. The time taken by the methods of disease database module is O(1).

Construction of Trie for various patterns: The time complexity to insert and search for various patterns in Trie depends on the maximum length pattern. So, the worst case time complexity is O(ℓ), where ℓ is the maximum length pattern of all the patterns.

K-mer counter for the exons and overall time complexity: The worst case time complexity to update pointers and to find different k-mers count is O($\mathscr{L}$e$\mathscr{M}$), where $\mathscr{L}$ is the length of the exon, e is the number of exons for each gene and $\mathscr{M}$ is the maximum k-mer size. To diagnose the diseases, the time complexity is ($\mathscr{K}$e$\mathscr{M}$), $\mathscr{K}$ is the maximum number of key k-mers. The overall time complexity of proposed algorithm is (($\mathscr{L}$+$\mathscr{K}$)e$\mathscr{M}$). If $\mathscr{L}$ value is greater than $\mathscr{K}$, then the time complexity is O($\mathscr{L}$e$\mathscr{M}$). Finally the overall worst case time complexity is O($\mathscr{L}$e).

6.6 Discussion and Results

6.6.1 Performance Metrics Mesures

The proposed CNN model was assessed by using two popular classification performance metrics i.e., AUC-ROC and AUC-PR. To calculate these metrics precision (Prec), Specificity (Sp), Recall or Sensitivity (Sn),True Positive Rate(TPR), False Positive Rate(FPR), and Accuracy (Acc) are required.

$$Prec = \frac{TP}{TP + FP} \tag{6.5}$$

$$TPR(or)Sn = \frac{TP}{TP + FN} \tag{6.6}$$

$$Sp = \frac{TN}{TN + FP} \tag{6.7}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.8}$$

$$FPR = 1 - Sp \tag{6.9}$$

The TP, TN, FP, FN are the number of true positive, true negative, false positive, and false negative values respectively. Accuracy, sensitivity and specificity are sensitive to the dataset class distribution, because there are very less viral sequences than non-viral.The majority of samples would have a greater effect on the curve than the minority, which may result in bias. On the other hand, for the class of imbalanced problems, a precision-recall curve is largely used as it does not accept false positives and false negatives, so there is no probability of effect of majority samples, thus providing sufficient evaluation.

6.6.2 Tandem Repeat Gene prediction results

The genes are collected from NCBI [193] database. The gene contains multiple mRNAs that contain exons of that particular gene. The mutations in the exon sequence could lead to the production of abnormal proteins leading to complex diseases. We introduce a HHO pattern matching approach in this work to predict the disease because of the number of mutations. The gene DMPK contains 11 Tandem Repeat mRNAs (NM_004409.5, NM_004943.2, NM_175865.5, NM_001081560.3, NM_001288764.2, NM_001288766.2, NR_147192.1, NM_001081562, NM_001081563.2, and NM_001288765.1). Due to mutations one of the exons in the mRNA NM_001081563.2 the CTG frequency increased to 47 and NM_ 175875.5 CTG count became 59. So, the mRNAs NM_001081563.2 and NM_175875.5 are pre-mutated and diseased respectively. The genes ATN1, ATXN2 and JPH3 contains 8, 16 and 37 mRNAs respectively. In one of the exons of ATN1 gene's mRNA(XM_0152938 81), the CTG pattern frequency increased to 61 due to point mutations, which leads to DRPLA (phenomena which causes brain disorder). The JPH3 gene's mRNAs NM_001271604, NM_001271605 are pre-mutated due to the increase in the frequency of tri-nucleotide CTG and mRNA NM_020655 affected to Covid-19 disease because of increase in CAG repeats. In the gene ATXN2 most of the mRNAs are diseased because of mutations in the corresponding exons. The detailed results are shown in Figure 6.8. Different types of samples, including tandem repeats, prostate secretion, serum, and cervical tissues, were used to train our model's divergent metagenomic contig sequences. The unknown test dataset, which was not part of the training set, is used by the trained HHO to predict viral sequences. When used to forecast viral sequences from fresh samples, the suggested model performs admirably. The HHO model was trained using fourteen datasets from human metagenomic experiments. The filters transform into learnt filters once training is complete; this allows the filter to automatically adjust its weights to their optimal values. Our computationally-based, step-by-step technique allows us to identify the underlying patterns that drive viral sequence prediction. If your dataset is balanced, then AUC-ROC is the way to go. If it's imbalanced, then AUC-PR is the way to go. Biassed measurements in AUC-ROC could be the consequence of a majority sample having a greater influence on the curve than a minority sample. Big AUR-ROC numbers might not always mean accurate classifications. However, false positives and false negatives are not taken into account by the AUR-PR curve. Due to the lack of bias caused by majority-over-minority sampling, it is a superior metric for evaluating performance. The proposed model achieves the 0.988 and 0.989 AUC-PR values on human metagenomic and human serum datasets respectively shown in 6.11. In contrast, the pattern ACACACA is the most repeated and occurred in 27 filters(6-32) with frequency 16110, which predicts the sequence as non-viral. The motifs (AAAAAAA, AAAGAAA, TTTTTTT) are skipped, which are present in both viral and non-viral sequences because the influence of those patterns is neutral in viral prediction. Each pattern frequency is compared with disease database, which outputs one of the following: Normal, Premutated and Disease Occurred. The graphical illustration of the disease prediction system. The continuous development of these methods leads to a reduction in the number of errors appearing in the encoding and decoding processes.

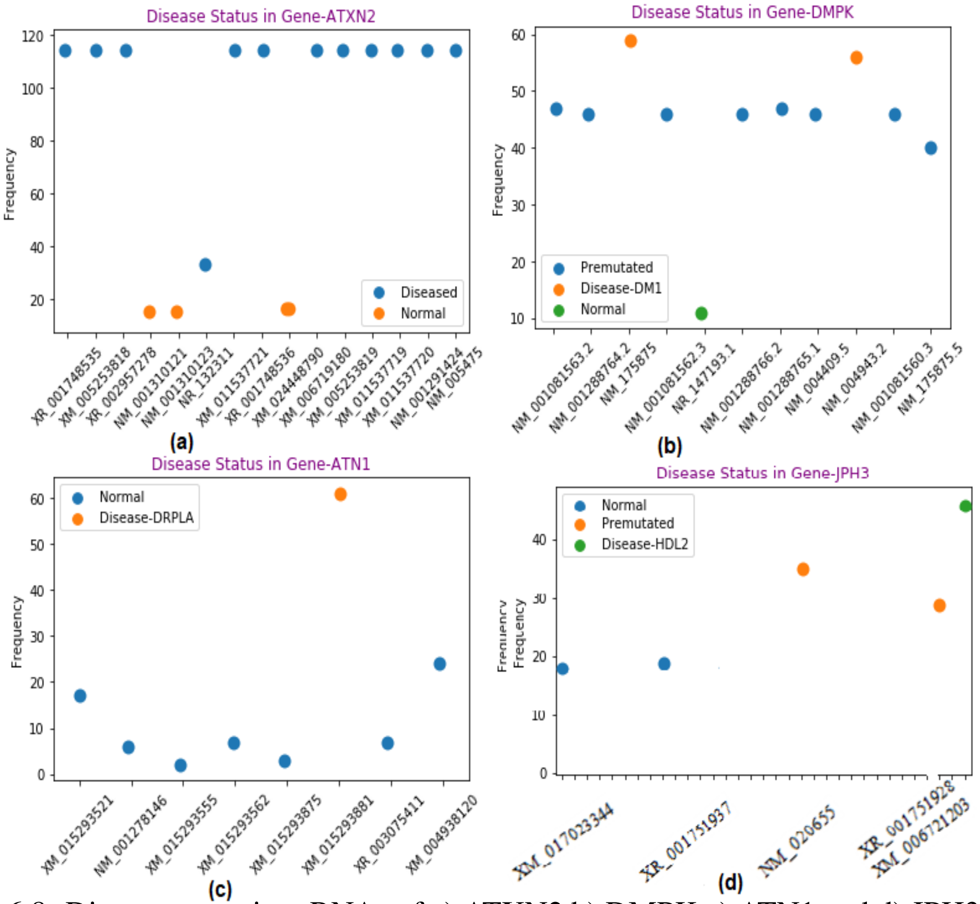


Figure 6.8: Disease status in mRNAs of a) ATXN2 b) DMPK c) ATN1 and d) JPH3 genes.

The proposed method finds the frequencies of all the exons and detected whether the exon is pre-mutated or diseased as shown in Table 6.6. The results in Figure 6.9 shows time comparison for different mRNA sequences of various genes to diagnose nucleotide Tandem repeat diseases. The proposed HHO multi-string pattern matching algorithm runs on multiple genes with increasing patterns, it gives better execution performance than sequential and GPGPU based HHO pattern matching algorithm. Even though, the patterns are increased it gives better speed up against sequential and parallel based different Models.Top four Mutations rate is calculated for each Nuclotide present in Covid-19 for Tandem repeats top diseased pattern”. This is clear that all rates have a common factor of having the high mutation rate of T and A. But there is a significant increase in the mutation rate compared to other mutations. This clearly indicates that this virus is some changes in T and A as shown 6.10. In the gene ATXN2 most of the mRNAs are diseased because of mutations in the corresponding exons.

Table 6.6: Disease status in Tandem repeat various mRNAs of a) ATXN2 b) DMPK c) ATN1 and d) JPH3 genes.

Gene	**mRNA**	**Pattern**	**Frequency**	**Disease Status**	**Disease Name**
ATN1	XM_015293881.2	CAG	61	Diseased	DRPLA
ATN1	XR_003075411.1	CAG	17	Normal	-
HTT	NM_002111.8	CAG	123	Diseased	Huntington disease
HTT	NR_045414.1	CAG	15	Normal	-
ATXN1	XR_001748535.1	CAG	109	Diseased	Cerebellum, spinal cord and brainstem related disorder
ATXN1	NM_006877.4	CAG	10	Normal	-
ATXN2	NM_001357857.2	CAG	114	Diseased	Short term memory problem
ATXN2	NM_001310121.1	CAG	11	Normal	-
ATXN3	NR_028457.1	CAG	103	Diseased	Memory loss
CACNA1A	NM_001357857.1	CAG	20	Diseased	Loss of coordination in their arms, muscles, tremors
CACNA1A	NM_001127221.1	CAG	18	Normal	-
ATXN7	NM_000333.3	CAG	36	Diseased	Macular degeneration, upper motor neuron
ATXN7	XM_024453841.1	CAG	5	Normal	-
ATXN10	NM_013236.4	ATTCT	4	Normal	-
TBP	NM_003194.5	CAG	48	Pre-mutated	Spinocerebellar ataxia type17
TBP	NM_002793.4	CAG	8	Normal	-
AR	NM_000044.6	CAG	112	Diseased	Disorder of muscle movement nerve cells
PPP2R2B	XR_002956249.1	CAG	44	Pre-mutated	Head and hand tremor, akinesia
PPP2R2B	NM_181674.2	CAG	17	Normal	-
PABPN1	NM_004643.3	GCN	0	Normal	-
DMPK	NM_001081563.2	CTG	47	Pre-muted	Affects skeletal and smooth muscle
DMPK	NM_175875.5	CTG	59	Diseased	Affects skeletal and smooth muscle
DMPK	NR_147193.1	CTG	11	Normal	-
CNBP	NM_001127193.2	CCTG	15	Normal	-
AFF2	NM_002025.4	CCG	60	Pre-mutated	Impairs thinking ability and cognitive functioning
AFF2	NM_001170628.1	CCG	2	Normal	-
FXN	NM_000144.5	GAA	109	Diseased	Affects nerves which causes movement problems
FMR1	NM_001185081.2	CGG	35	Normal	-
JPH3	NM_001271605.2	CTG	35	Diseased	Emotional, movement and cognitive abnormalities
JPH3	XR_001751940.1	CTG	11	Normal	-
JPH3	NM_001271604.3	CTG	29	Pre-muted	Huntington disease
KLHL1	NM_001286725.1	CGG	3	Normal	-
NOP56	XR_001754267.1	GGCCTG	0	Normal	-

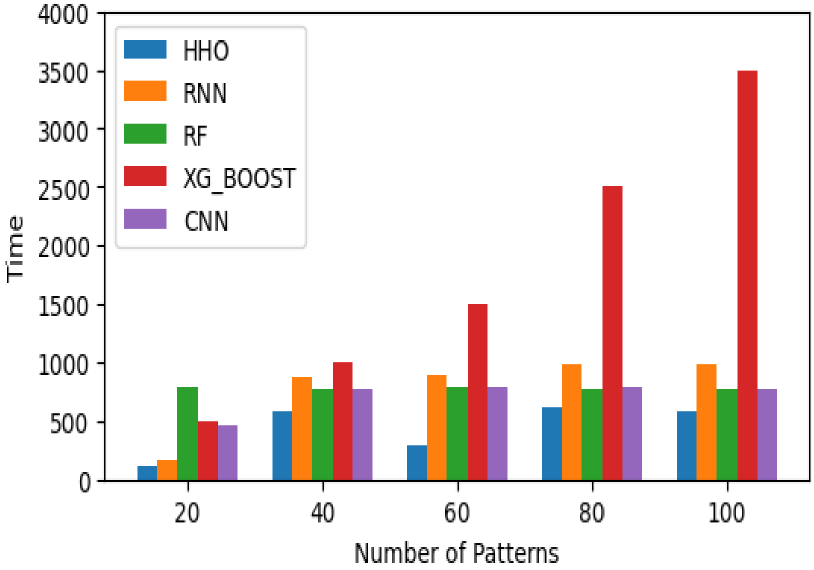


Figure 6.9: Time comparison of differnt models with Proposed HHO Tandem Repeat pattern matching.

Table 6.7: Top-1 motifs with the highest activation value from each filter extracted from 19 metagenomic experiments, which act as features to predict Tandem Repeat viral genomes.

Filter Id	Viral Pattern	Mean Activation	Activation Value	Filter Id	Viral Pattern	Mean Activation	Activation Value
12	ACGACCC	0.689984	1.4799685	9	GCTGTTT	0.69609371	1.2921874
14	AGCAGAG	0.692365	1.2847302	28	GATTTGA	0.58795887	1.1759177
24	GGGATCG	0.778848	1.3576957	3	GCGAGGT	0.58052796	1.1610559
21	TACGGGG	0.673739	1.3474776	7	TCAGGTC	0.57200754	1.1440151
4	AGGCGGG	0.652724	1.3054485	19	AAAGTCT	0.55970358	1.1194072
10	AAAAATT	0.650489	1.3009783	23	GTCCTGA	0.55340659	1.1068132
18	AGTACGA	0.649528	1.2990574	31	CCGTTAT	0.54953909	1.0990782
1	CTTTTTT	0.644374	1.2887475	14	ATGGAGT	0.54869616	1.0973923
20	GTCACTC	0.634448	1.2688965	22	GAGAAAA	0.54356331	1.0871266
2	ACCTCTG	0.632819	1.2656392	8	ATATGTG	0.54122984	1.0824597
26	CACAGTG	0.630168	1.2603375	17	TTGAAAT	0.52796107	1.0559222
5	TGAGCTC	0.924386	1.2487631	29	CGTGCCC	0.52501839	1.0500368
32	CTAGGCT	0.708599	1.2171972	27	TAACGTC	0.51818478	1.0363696
30	TGGGCCG	0.703728	1.2074571	13	GATCCTA	0.48595235	0.9719047
6	TCACATC	0.403138	1.2062765	25	ATGAGAG	0.46512297	0.9302594
12	TCACAAC	0.497368	1.2062765				

Table 6.8: The patterns extracted by most of the filters with a threshold as an average of all activations, which act as features to predict viral and non-viral genomes from human metagenomic datasets.

Viral Pattern	Frequency	Repeated in No. of Filters	Filter Numbers	Non-Viral Pattern	Frequency	Repeated in No. of Filters	Filter Numbers
TAAAAAA	4229	28	1-3,7-19,21-32	ACACACA	16110	27	6-32
AAAAAAA	1783	23	7-29	GAAAAAA	8113	26	6-31
TTTTTTT	1036	20	1-20	AAGAAAA	4282	11	1-5, 7-12
AAAAAAA*	6915	32	1-32	CACACAC	11133	5	13-16,32
AAAAAAA*	4796	32	1-32	TTTTTTT*	13978	13	20-32

*Neutral patterns, present in viral and non-viral genome sequences

6.6.3 Comparative results of the Proposed Model with the existing models

The HHO model's capability was compared to that of known baseline ML models like Random forests, Naive Bayes, RNN, XGBoost and HHO that classify Covid-19 in Short Tandem Repeat sequences by utilizing various methods for feature extraction proposed by Hilal Arslan[121]. HHO algorithm for finding the best features subset A 99.88 accuracy is obtained by using a moderate Tandem repeat datasets for Covid-19. Further more, the HHO algorithm's searching power was used to construct a novel suggested algorithm compared with other existing algorithms 6.9 compares the HHO techniques to the baseline methods in terms of accuracy, recall, precision, and F-measure.method[121].

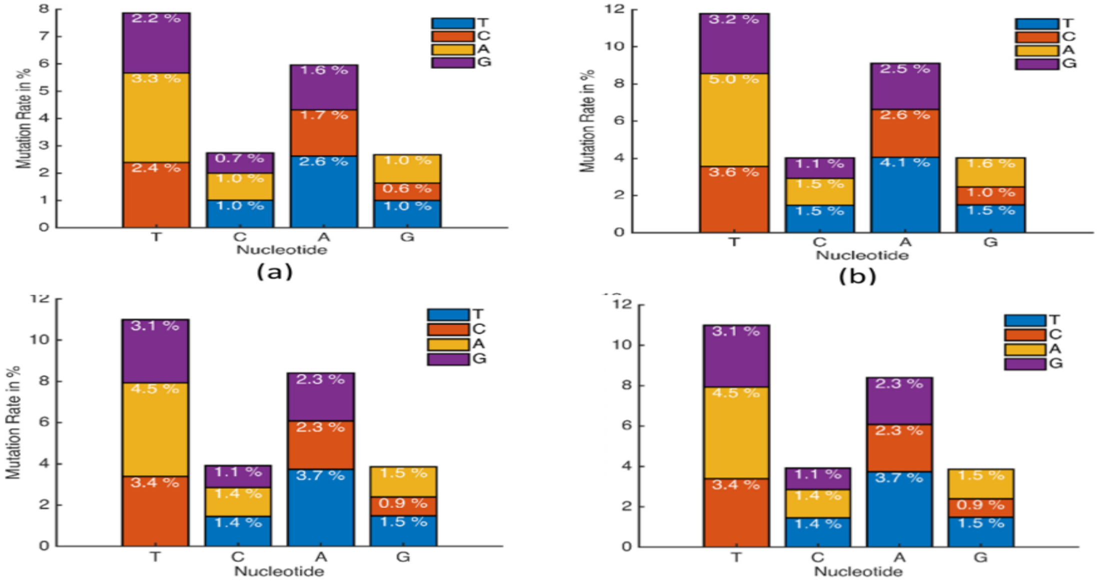


Figure 6.10: Top 4 Mutation Rates for a) HCov-229E b) HCoV-NL63 c) HCoV-OC4 d) MERS-CoV.

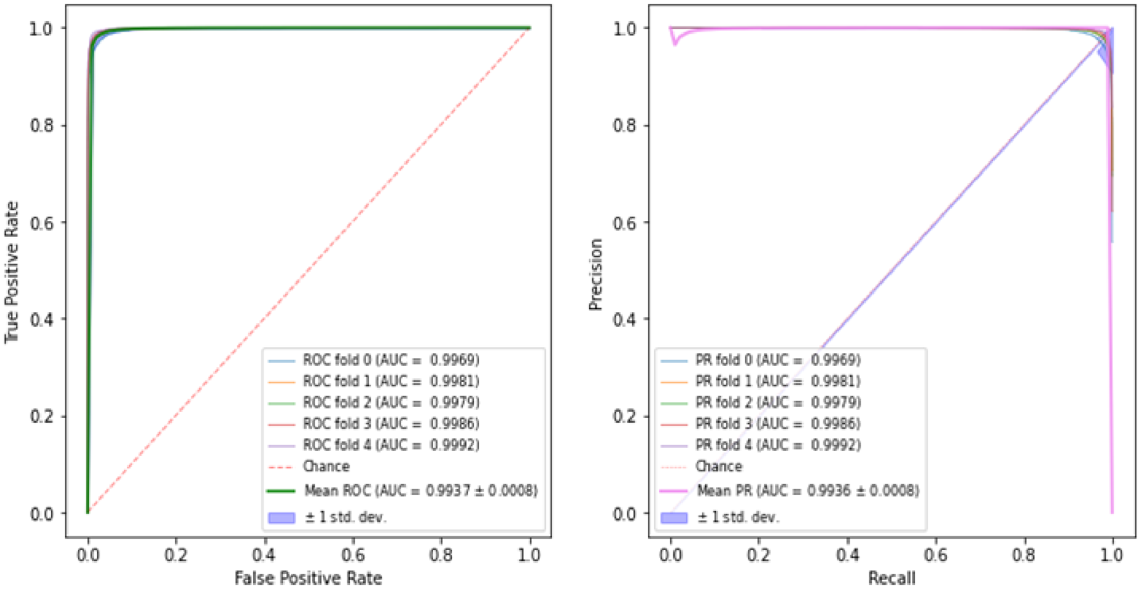


Figure 6.11: The ROC and Precision-Recall curves for (a) human metagenomic Tandem Repeat dataset (b) human-Cov serum dataset 5-fold cross-validation is used.

Table 6.9: Evaluation of metrics measurements accuracy, precision, recall, F1-score,

Sl.No.	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
1	RNN	89.56	79.22	86.33	9.11
2	NB	93.56	65.36	90.11	15.12
3	RF	84.61	68.55	87.23	9.36
4	XGBoost	74.95	71.35	79.36	11.54
5	CNN	88.36	71.33	89.99	8.55
6	**HHO**	**99.88**	**99.75**	**98.91**	**18.55**

The results are compared with existing models our proposed models give best results and achieved good accuracy, Precision, Recall, F1 Score.

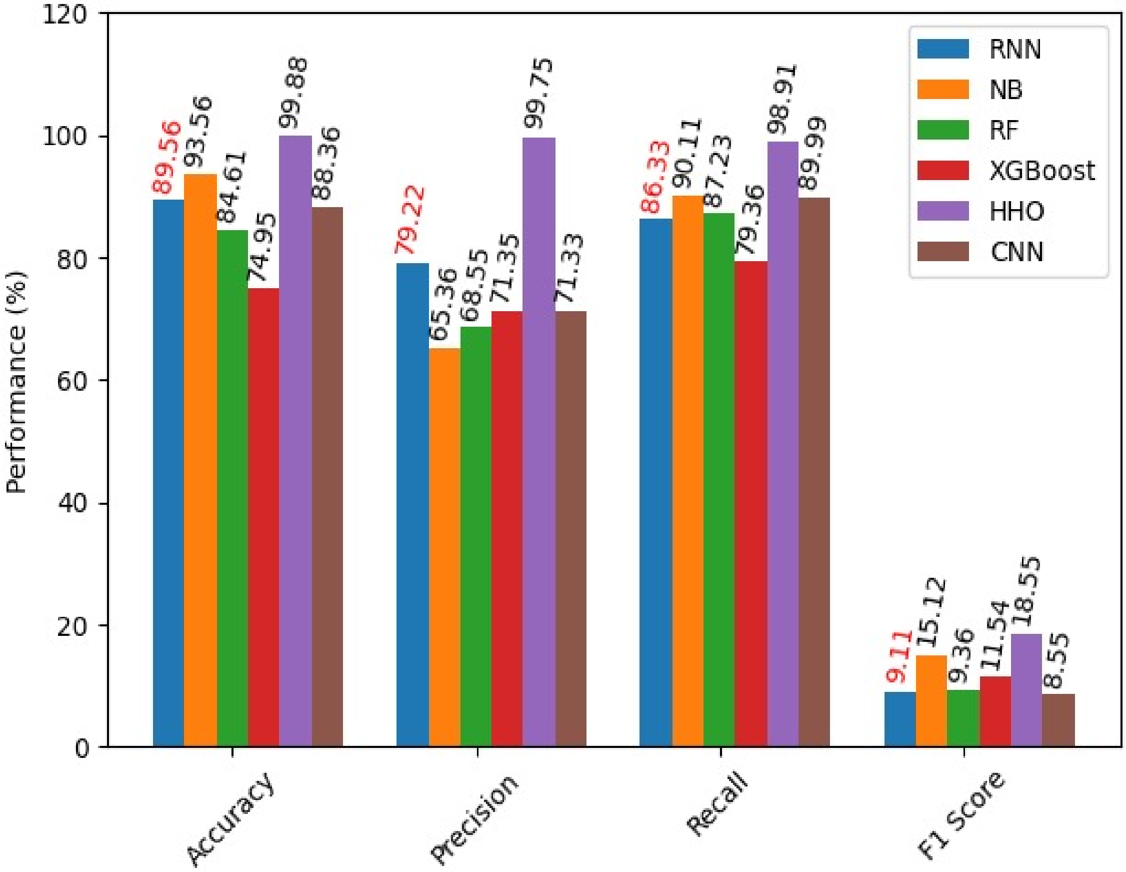


Figure 6.12: Performance comparison of different models with Existing proposed

6.7 Summary

Here, we introduced Harris hawks optimisation, or HHO. Depending on the mutation rate, the HHO approach can move between exon extraction, features based on base(s) frequency, disease database creation for tandem repeats, and transitioning from exploration to exploitation. It can also switch between behaviours involving DNA short tandem repeats (STR) mistakes. Model for optimising Harris Hawks for accurate prediction of SARS-CoV-2 symptoms. For each nuclotide, HHO uses and analyses mutations tate to make it interpretable, unlike other approaches. This model stands out because it uses an HHO design to incorporate DNA sick pattern motifs that are low, moderate, high, and biologically significant. Insight into the methods by which SARS CoV-2 regulates gene expression and the ability to identify Tandem repeat sequences are both provided by these patterns. Although HHO are successful, they are frequently criticised for not being easily interpretable. Consequently, the topic of building a database and a Trie for tandem pattern disorders in COVID-19 is also covered in this paper.

Chapter 7

Conclusion and Future Scope

Here, we present the summary of contributions made in this thesis and mention the open problems triggered out of the study.

7.1 Conclusions

In addition to detecting the predisease condition, the suggested method is significantly more efficient, which could lead to the early identification of complicated disorders. In complicated illness investigation, the prediction of the viral genome is a crucial task. Important features, possible viral mutation sequences, and virally-associated sequences can all be extracted by the suggested CNN framework. The mechanisms of viral illnesses can be better understood by observing repeating patterns. An accurate mutation rate estimation for COVID-19 enables us to comprehend genetic variants and compare other COVs. Using deep convolution neural networks, the suggested method may forecast the likelihood that DNA mutations produce an increased frequency of DNA diseases. Classification of Omicron virus variants using deep learning approaches can also anticipate patterns associated with the new coronavirus.

7.2 Future Scope

Here are a few of the open sequence analysis problems that this study brought to light. To improve prediction using recurring pattern pathogenic themes, the suggested study uses convolutional neural networks (CNNs) to automatically extract features in chapter 3. Additionally, the suggested model has the potential to be expanded to anticipate various genomic signals, such as locations of polyadenylation, pupylation, protein-protein interactions, and so on. Chapter 4 presents the suggested work which is optimised for Coot-Lion. We used a deep learning technique to forecast the point mutation rate for both the positive and negative datasets. A more accurate prediction of which DNA mutations are causal for a given disease is possible with the help of the improved model. In order to increase the disease database, a deep learning algorithm is used to extract viral patterns. Additionally, using any viral or bacterial genome, a generalised deep-learning model may be trained to anticipate different illness patterns. Major disorders can result from DNA sequence repeats, as discussed in chapter 5, which can be misidentified as regulating genome patterns. Moving forward, the suggested ERSIT-GRU learning models can be enhanced to detect various Repeats and the corresponding patterns. In Chapter 6, we analysed the mutation rates of different coronaviruses and assessed different metrics that show how important Tandem Repeat is in 2-deletion error rates. In addition, the novel Omicron coronavirus can be analysed using HHO methods for pattern prediction and medication discovery.AI-Enhanced Mutation Analysis, Real-Time Genome Sequencing and Analysis, Predictive Modelling of Mutation Evolution Identifying common mutation patterns can inform the development of targeted antivirals that remain effective against a broad range of variants, reducing the risk of resistance. Advanced machine learning models, particularly deep learning (DL) and transformer-based architectures, can help detect and predict mutations' effects faster. These innovations in mutation detection and analysis can make COVID-19 monitoring more efficient and effective, improving global response to future variants and enhancing readiness for other viral threats. We demonstrated that the RNN based RF generate valid novel Repeats.

Bibliography

[1] International Human Genome Sequencing Consortium et al. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931, 2004.

[2] Jennifer Harrow, Adam Frankish, Jose M Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, et al. Gencode: the reference human genome annotation for the encode project. *Genome research*, 22(9):1760–1774, 2012.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[4] Jasper Zuallaert, Fréderic Godin, Mijung Kim, Arne Soete, Yvan Saeys, and Wesley De Neve. Splicerover: interpretable convolutional neural networks for improved splice site prediction. *Bioinformatics*, 34(24):4180–4188, 2018.

[5] Anil Jain, Ruud Bolle, and Sharath Pankanti. Introduction to biometrics. In *Biometrics*, pages 1–41. Springer, 1996.

[6] Anil K Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric template security. *EURASIP Journal on advances in signal processing*, 2008:1–17, 2008.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Mohammad Hashem Ryalat, Osama Dorgham, Sara Tedmori, Zainab Al-Rahamneh, Nijad Al-Najdawi, and Seyedali Mirjalili. Harris hawks optimization for covid-19 diagnosis based on multi-threshold image segmentation. *Neural Computing and Applications*, 35(9):6855–6873, 2023.

[9] RG Babukarthik, V Ananth Krishna Adiga, G Sambasivam, D Chandramohan, and JJIA Amudhavel. Prediction of covid-19 using genetic deep learning convolutional neural network (gdcnn). *Ieee Access*, 8:177647–177666, 2020.

[10] Abubakr S Issa, Yossra H Ali, and Tarik A Rashid. Enhanced harris hawks optimization (ehho) for detecting covid-19 using chest x-ray. In *2022 2nd International Conference on Advances in Engineering Science and Technology (AEST)*, pages 361–365. IEEE, 2022.

[11] Sidra Abbas, Gabriel Avelino Sampedro, Mideth Abisado, Ahmad Almadhor, Iqra Yousaf, and Seng-Phil Hong. Harris-hawk-optimization-based deep recurrent neural network for securing the internet of medical things. *Electronics*, 12(12):2612, 2023.

[12] William R Pearson and David J Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[13] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[14] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

[15] Mohammadreza Ghodsi, Bo Liu, and Mihai Pop. Dnaclust: accurate and efficient clustering of phylogenetic marker genes. *BMC bioinformatics*, 12(1):271, 2011.

[16] Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[17] Cole Trapnell, Lior Pachter, and Steven L Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, 2009.

[18] Davit Bzhalava, Johanna Ekström, Fredrik Lysholm, Emilie Hultin, Helena Faust, Bengt Persson, Matti Lehtinen, Ethel-Michele de Villiers, and Joakim Dillner. Phylogenetically diverse tt virus viremia among pregnant women. *Virology*, 432(2):427–434, 2012.

[19] Jessica M Labonté and Curtis A Suttle. Previously unknown and highly divergent ssdna viruses populate the oceans. *The ISME journal*, 7(11):2169–2177, 2013.

[20] Luiz Carlos Junior Alcantara, Sharon Cassol, Pieter Libin, Koen Deforche, Oliver G Pybus, Marc Van Ranst, Bernardo Galvao-Castro, Anne-Mieke Vandamme, and Tulio De Oliveira. A standardized framework for accurate, high-throughput genotyping of recombinant and non-recombinant viral sequences. *Nucleic acids research*, 37(suppl_2):W634–W642, 2009.

[21] Andrea-Clemencia Pineda-Peña, Nuno Rodrigues Faria, Stijn Imbrechts, Pieter Libin, Ana Barroso Abecasis, Koen Deforche, Arley Gómez-López, Ricardo J Camacho, Tulio de Oliveira, and Anne-Mieke Vandamme. Automated subtyping of hiv-1 genetic sequences for clinical and surveillance purposes: performance evaluation of the new rega version 3 and seven other tools. *Infection, genetics and evolution*, 19:337–348, 2013.

[22] Sergei L Kosakovsky Pond, David Posada, Eric Stawiski, Colombe Chappey, Art FY Poon, Gareth Hughes, Esther Fearnhill, Mike B Gravenor, Andrew J Leigh Brown,

and Simon DW Frost. An evolutionary model-based algorithm for accurate phylogenetic breakpoint mapping and subtype prediction in hiv-1. *PLoS computational biology*, 5(11), 2009.

[23] Jaina Mistry, Robert D Finn, Sean R Eddy, Alex Bateman, and Marco Punta. Challenges in homology search: Hmmer3 and convergent evolution of coiled-coil regions. *Nucleic acids research*, 41(12):e121–e121, 2013.

[24] Peter Skewes-Cox, Thomas J Sharpton, Katherine S Pollard, and Joseph L DeRisi. Profile hidden markov models for the detection of viruses within metagenomic sequence data. *PloS one*, 9(8), 2014.

[25] Zurab Bzhalava, Emilie Hultin, and Joakim Dillner. Extension of the viral ecology in humans using viral profile hidden markov models. *PloS one*, 13(1), 2018.

[26] Benjamin T James, Brian B Luczak, and Hani Z Girgis. Meshclust: an intelligent tool for clustering dna sequences. *Nucleic acids research*, 2018.

[27] Qi Dai, Xiaoqing Liu, Yuhua Yao, and Fukun Zhao. Numerical characteristics of word frequencies and their application to dissimilarity measure for sequence comparison. *Journal of theoretical biology*, 276(1):174–180, 2011.

[28] Takuyo Aita, Yuzuru Husimi, and Koichi Nishigaki. A mathematical consideration of the word-composition vector method in comparison of biological sequences. *BioSystems*, 106(2-3):67–75, 2011.

[29] G Fichant and Christian Gautier. Statistical method for predicting protein coding regions in nucleic acid sequences. *Bioinformatics*, 3(4):287–295, 1987.

[30] B Edwin Blaisdell. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*, 83(14):5155–5159, 1986.

[31] Kin On Cheng, Ngai Fong Law, and W-C Siu. Clustering-based compression for population dna sequences. *IEEE/ACM transactions on computational biology and bioinformatics*, (1):1–1, 2017.

[32] Guozhu Dong and Jian Pei. Classification, clustering, features and distances of sequence data. In *Sequence data mining*, pages 47–65. Springer, 2007.

[33] Xiaolin Hao, Rui Jiang, and Ting Chen. Clustering 16s rrna for otu prediction: a method of unsupervised bayesian clustering. *Bioinformatics*, 27(5):611–618, 2011.

[34] Peter Sperisen and Marco Pagni. Jacop: a simple and robust method for the automated classification of protein sequences with modular architecture. *BMC bioinformatics*, 6(1):216, 2005.

[35] I Dondoshansky and Y Wolf. Blastclust (ncbi software development toolkit) bethesda: Ncbi. 2002.

[36] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[37] Ernesto Picardi, Flavio Mignone, and Graziano Pesole. Easycluster: a fast and efficient gene-oriented clustering tool for large-scale transcriptome data. In *BMC bioinformatics*, volume 10, page S10. BioMed Central, 2009.

[38] Hengki Muradi, Alhadi Bustamam, and Dian Lestari. Application of hierarchical clustering ordered partitioning and collapsing hybrid in ebola virus phylogenetic analysis. *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 317–323, 2015.

[39] Andrzej Zielezinski, Susana Vinga, Jonas Almeida, and Wojciech M Karlowski. Alignment-free sequence comparison: benefits, applications, and tools. *Genome biology*, 18(1):1–17, 2017.

[40] Gerardo Mendizabal-Ruiz, Israel Román-Godínez, Sulema Torres-Ramos, Ricardo A Salido-Ruiz, Hugo Vélez-Pérez, and J Alejandro Morales. Genomic signal processing for dna sequence clustering. *PeerJ*, 6:e4264, 2018.

[41] Essam Abdellatef, Nabil A Ismail, Salah Eldin SE Abd Elrahman, Khalid N Ismail, Mohamed Rihan, and Fathi E Abd El-Samie. Cancelable multi-biometric recognition system based on deep learning. *The Visual Computer*, 36:1097–1109, 2020.

[42] Dan Wei, Qingshan Jiang, Yanjie Wei, and Shengrui Wang. A novel hierarchical clustering algorithm for gene sequences. *BMC bioinformatics*, 13(1):174, 2012.

[43] Zurab Bzhalava, Ardi Tampuu, Piotr Bała, Raul Vicente, and Joakim Dillner. Machine learning for detection of viral sequences in human metagenomic datasets. *BMC bioinformatics*, 19:1–11, 2018.

[44] Mihaela Pertea, Xiaoying Lin, and Steven L Salzberg. Genesplicer: a new computational method for splice site prediction. *Nucleic acids research*, 29(5):1185–1190, 2001.

[45] Jagath C Rajapakse and Loi Sy Ho. Markov encoding for detecting signals in genomic sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):131–142, 2005.

[46] Prabina Kumar Meher, Tanmaya Kumar Sahu, and Atmakuri Ramakrishna Rao. Prediction of donor splice sites using random forest with a new sequence encoding approach. *BioData mining*, 9(1):4, 2016.

[47] Te-Ming Chen, Chung-Chin Lu, and Wen-Hsiung Li. Prediction of splice sites with dependency graphs and their expanded bayesian networks. *Bioinformatics*, 21(4):471–482, 2005.

[48] Andigoni Malousi, Ioanna Chouvarda, Vassilis Koutkias, Sofia Kouidou, and Nicos Maglaveras. Spliceit: a hybrid method for splice signal identification based on probabilistic and biological inference. *Journal of biomedical Informatics*, 43(2):208–217, 2010.

[49] Jian He, Xuemei Pu, Menglong Li, Chuan Li, and Yanzhi Guo. Deep convolutional neural networks for predicting leukemia-related transcription factor binding sites from dna sequence data. *Chemometrics and Intelligent Laboratory Systems*, 199:103976, 2020.

[50] Mario Stanke and Stephan Waack. Gene prediction with a hidden markov model and a new intron submodel. *Bioinformatics*, 19(2):215–225, 2003.

[51] Quanwei Zhang, Qinke Peng, Qi Zhang, Yanhua Yan, Kankan Li, and Jing Li. Splice sites prediction of human genome using length-variable markov model and feature selection. *Expert Systems with Applications*, 37(4):2771–2782, 2010.

[52] Abdul KMA Baten, Bill CH Chang, Saman K Halgamuge, and Jason Li. Splice site identification using probabilistic parameters and svm classification. In *BMC bioinformatics*, volume 7, page S15. Springer, 2006.

[53] Abdul KMA Baten, Saman K Halgamuge, and Bill CH Chang. Fast splice site detection using information content and feature reduction. *BMC bioinformatics*, 9(S12):S8, 2008.

[54] Sven Degroeve, Yvan Saeys, Bernard De Baets, Pierre Rouzé, and Yves Van De Peer. Splicemachine: predicting splice sites from high-dimensional local context representations. *Bioinformatics*, 21(8):1332–1338, 2005.

[55] Prabina Kumar Meher, Tanmaya Kumar Sahu, AR Rao, and SD Wahi. Identification of donor splice sites using support vector machine: a computational approach based on positional, compositional and dependency features. *Algorithms for molecular biology*, 11(1):16, 2016.

[56] Sören Sonnenburg, Gabriele Schweikert, Petra Philips, Jonas Behr, and Gunnar Rätsch. Accurate splice site prediction using support vector machines. In *BMC bioinformatics*, volume 8, page S7. Springer, 2007.

[57] Dan Wei, Huiling Zhang, Yanjie Wei, and Qingshan Jiang. A novel splice site prediction method using support vector machine. *Journal of Computational Information Systems*, 9(20):8053–8060, 2013.

[58] Zhen Chen, Pei Zhao, Fuyi Li, Tatiana T Marquez-Lago, André Leier, Jerico Revote, Yan Zhu, David R Powell, Tatsuya Akutsu, Geoffrey I Webb, et al. ilearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of dna, rna and protein sequence data. *Briefings in bioinformatics*, 21(3):1047–1057, 2020.

[59] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[60] Shuichiro Makigaki and Takashi Ishida. Sequence alignment using machine learning for accurate template-based protein structure prediction. *Bioinformatics*, 36(1):104–111, 2020.

[61] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

[62] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[63] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[64] Tracy L Meiring, Anna T Salimo, Beatrix Coetzee, Hans J Maree, Jennifer Moodley, Inga I Hitzeroth, Michael-John Freeborough, Ed P Rybicki, and Anna-Lise Williamson. Next-generation sequencing of cervical dna detects human papillomavirus types not detected by commercial kits. *Virology journal*, 9(1):164, 2012.

[65] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular systems biology*, 12(7):878, 2016.

[66] Haohan Wang and Bhiksha Raj. On the origin of deep learning. *arXiv preprint arXiv:1702.07800*, 2017.

[67] Min Chen, Yixue Hao, Kai Hwang, Lu Wang, and Lin Wang. Disease prediction by machine learning over big data from healthcare communities. *Ieee Access*, 5:8869–8879, 2017.

[68] Rui Liu, Jiayuan Zhong, Xiangtian Yu, Yongjun Li, and Pei Chen. Identifying critical state of complex diseases by single-sample-based hidden markov model. *Frontiers in genetics*, 10:285, 2019.

[69] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Aidong Zhang, and Jing Gao. Personalized disease prediction using a cnn-based similarity learning method. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 811–816. IEEE, 2017.

[70] Sandeep U Mane and Ketaki H Pangu. Disease diagnosis using pattern matching algorithm from dna sequencing: a sequential and gpgpu based approach. In *Proceedings of the International Conference on Informatics and Analytics*, page 58. ACM, 2016.

[71] Deyvid Amgarten, Lucas PP Braga, Aline M da Silva, and João C Setubal. Marvel, a tool for prediction of bacteriophage sequences in metagenomic bins. *Frontiers in genetics*, 9:304, 2018.

[72] Simon Roux, Francois Enault, Bonnie L Hurwitz, and Matthew B Sullivan. Virsorter: mining viral signal from microbial genomic data. *PeerJ*, 3:e985, 2015.

[73] Jie Ren, Nathan A Ahlgren, Yang Young Lu, Jed A Fuhrman, and Fengzhu Sun. Virfinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, 5(1):69, 2017.

[74] Brett E. Pickett, Eva L. Sadat, Yun Zhang, Jyothi M. Noronha, R. Burke Squires, Victoria Hunt, Mengya Liu, Sanjeev Kumar, Sam Zaremba, Zhiping Gu, Liwei Zhou, Christopher N. Larson, Jonathan Dietrich, Edward B. Klem, and Richard H. Scheuermann. ViPR: an open bioinformatics database and analysis resource for virology research. *Nucleic Acids Research*, 40(D1):D593–D598, 10 2011.

[75] Christophe Legendre, Gerald C Gooden, Kyle Johnson, Rae Anne Martinez, Winnie S Liang, and Bodour Salhia. Whole-genome bisulfite sequencing of cell-free dna identifies signature associated with metastatic breast cancer. *Clinical epigenetics*, 7(1):100, 2015.

[76] Wei Chen, Peng-Mian Feng, Hao Lin, and Kuo-Chen Chou. iss-psednc: identifying splicing sites using pseudo dinucleotide composition. *BioMed research international*, 2014, 2014.

[77] Markus Hoffmann, Hannah Kleine-Weber, Nadine Krüger, Marcel A Mueller, Christian Drosten, and Stefan Pöhlmann. The novel coronavirus 2019 (2019-ncov) uses the sars-coronavirus receptor ace2 and the cellular protease tmprss2 for entry into target cells. *BioRxiv*, 2020.

[78] Li Deng and Roberto Togneri. Deep dynamic models for learning hidden representations of speech features. In *Speech and audio processing for coding, enhancement and recognition*, pages 153–195. Springer, 2015.

[79] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[80] Sai Zhang, Jingtian Zhou, Hailin Hu, Haipeng Gong, Ligong Chen, Chao Cheng, and Jianyang Zeng. A deep learning framework for modeling structural features of rna-binding protein targets. *Nucleic acids research*, 44(4):e32–e32, 2016.

[81] Jihye Lim, Jungyoon Kim, and Songhee Cheon. A deep neural network-based method for early detection of osteoarthritis using statistical data. *International journal of environmental research and public health*, 16(7):1281, 2019.

[82] Raju Bhukya and Achyuth Ashok. Gene expression prediction using deep neural networks. *INTERNATIONAL ARAB JOURNAL OF INFORMATION TECHNOLOGY*, 17(3):422–431, 2020.

[83] Zaheer Ullah Khan, Farman Ali, Izhar Ahmed Khan, Yasir Hussain, and Dechang Pi. irspot-spi: Deep learning-based recombination spots prediction by incorporating secondary sequence information coupled with physio-chemical properties via chou's 5-step rule and pseudo components. *Chemometrics and Intelligent Laboratory Systems*, 189:169–180, 2019.

[84] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 2016.

[85] Daniel Quang and Xiaohui Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, 44(11):e107–e107, 2016.

[86] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[87] Riccardo Rizzo, Antonino Fiannaca, Massimo La Rosa, and Alfonso Urso. A deep learning approach to dna sequence classification. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 129–140. Springer, 2015.

[88] Saed Khawaldeh, Usama Pervaiz, Mohammed Elsharnoby, Alaa Eddin Alchalabi, and Nayel Al-Zubi. Taxonomic classification for living organisms using convolutional neural networks. *Genes*, 8(11):326, 2017.

[89] Bojian Yin, Marleen Balvert, Davide Zambrano, Alexander Schönhuth, and Sander Bohte. An image representation based convolutional network for dna classification. *arXiv preprint arXiv:1806.04931*, 2018.

[90] Xin Gao, Jie Zhang, Zhi Wei, and Hakon Hakonarson. Deeppolya: a convolutional neural network approach for polyadenylation site prediction. *IEEE Access*, 6:24340–24349, 2018.

[91] Jesse Eickholt and Jianlin Cheng. Dndisorder: predicting protein disorder using boosting and deep networks. *BMC bioinformatics*, 14(1):88, 2013.

[92] Matt Spencer, Jesse Eickholt, and Jianlin Cheng. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(1):103–112, 2014.

[93] Sheng Wang, Shunyan Weng, Jianzhu Ma, and Qingming Tang. Deepcnf-d: predicting protein order/disorder regions by weighted deep convolutional neural fields. *International journal of molecular sciences*, 16(8):17315–17330, 2015.

[94] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[95] Iman Nazari, Hilal Tayara, and Kil To Chong. Branch point selection in rna splicing using deep learning. *IEEE Access*, 7:1800–1807, 2018.

[96] Hilal Tayara, Muhammad Tahir, and Kil To Chong. iss-cnn: Identifying splicing sites using convolution neural network. *Chemometrics and Intelligent Laboratory Systems*, 188:63–69, 2019.

[97] Tatsuhiko Naito. Human splice-site prediction with deep neural networks. *Journal of Computational Biology*, 25(8):954–961, 2018.

[98] Xiuquan Du, Yu Yao, Yanyu Diao, Huaixu Zhu, Yanping Zhang, and Shuo Li. Deepss: Exploring splice site motif through convolutional neural network directly from dna sequence. *IEEE Access*, 6:32958–32978, 2018.

[99] Ruohan Wang, Zishuai Wang, Jianping Wang, and Shuaicheng Li. Splicefinder: ab initio prediction of splice sites using convolutional neural network. *BMC bioinformatics*, 20(23):652, 2019.

[100] Somayah Albaradei, Arturo Magana-Mora, Maha Thafar, Mahmut Uludag, Vladimir B Bajic, Takashi Gojobori, Magbubah Essack, and Boris R Jankovic. Splice2deep: An ensemble of deep convolutional neural networks for improved splice site prediction in genomic dna. *Gene: X*, page 100035, 2020.

[101] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: Achievements and challenges. *Journal of digital imaging*, pages 1–15, 2019.

[102] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. In *Classification in BioApps*, pages 323–350. Springer, 2018.

[103] Guangming Zhu, Bin Jiang, Liz Tong, Yuan Xie, Greg Zaharchuk, and Max Wintermark. Applications of deep learning to neuro-imaging techniques. *Frontiers in neurology*, 10:869, 2019.

[104] Fabian Hausmann and Stefan Kurtz. Deepgrp: engineering a software tool for predicting genomic repetitive elements using recurrent neural networks with attention. *Algorithms for Molecular Biology*, 16:1–13, 2021.

[105] Jakub M Bartoszewicz, Anja Seidel, and Bernhard Y Renard. Interpretable detection of novel human viruses from genome sequencing data. *BioRxiv*, 2020.

[106] Xiaoyong Pan and Hong-Bin Shen. Predicting rna–protein binding sites and motifs through combining local and global deep convolutional neural networks. *Bioinformatics*, 34(20):3427–3436, 2018.

[107] Zhao-Chun Xu, Peng Wang, Wang-Ren Qiu, and Xuan Xiao. iss-pc: identifying splicing sites via physical-chemical properties using deep sparse auto-encoder. *Scientific reports*, 7(1):1–12, 2017.

[108] Anna Fabijańska and Szymon Grabowski. Viral genome deep classifier. *IEEE Access*, 7:81297–81307, 2019.

[109] Ardi Tampuu, Zurab Bzhalava, Joakim Dillner, and Raul Vicente. Viraminer: Deep learning on raw dna sequences for identifying viral genomes in human samples. *PloS one*, 14(9), 2019.

[110] Jie Ren, Kai Song, Chao Deng, Nathan A Ahlgren, Jed A Fuhrman, Yi Li, Xiaohui Xie, and Fengzhu Sun. Identifying viruses from metagenomic data by deep learning. *arXiv preprint arXiv:1806.07810*, 2018.

[111] J Chen, H Liu, J Yang, and K-C Chou. Prediction of linear b-cell epitopes using amino acid pair antigenicity scale. *Amino acids*, 33(3):423–428, 2007.

[112] Guanxiong Zhang, Jian Shi, Shiwei Zhu, Yujia Lan, Liwen Xu, Huating Yuan, Gaoming Liao, Xiaoqin Liu, Yunpeng Zhang, Yun Xiao, et al. Diseaseenhancer: a resource of human disease-associated enhancer catalog. *Nucleic acids research*, 46(D1):D78–D84, 2018.

[113] Hossam Magdy Balaha, Eman M El-Gendy, and Mahmoud M Saafan. Covh2sd: A covid-19 detection approach based on harris hawks optimization and stacked deep learning. *Expert systems with applications*, 186:115805, 2021.

[114] Omar Fitian Rashid, Zulaiha Ali Othman, Suhaila Zainudin, and Noor Azah Samsudin. Dna encoding and str extraction for anomaly intrusion detection systems. *IEEE Access*, 9:31892–31907, 2021.

[115] Xingyu Liao, Juexiao Zhou, Bin Zhang, Xingyi Li, Xiaopeng Xu, Haoyang Li, and Xin Gao. Deep learning enhanced tandem repeat variation identification via multimodal conversion of nanopore reads alignment. *bioRxiv*, pages 2023–08, 2023.

[116] Sarah Fazal, Matt C Danzi, Isaac Xu, Shilpa Nadimpalli Kobren, Shamil Sunyaev, Chloe Reuter, Shruti Marwaha, Matthew Wheeler, Egor Dolzhenko, Francesca Lucas, et al. Rexprt: a machine learning tool to predict pathogenicity of tandem repeat loci, 2024.

[117] Yaling An, Shihua Li, Xiyue Jin, Jian-bao Han, Kun Xu, Senyu Xu, Yuxuan Han, Chuanyu Liu, Tianyi Zheng, Mei Liu, et al. A tandem-repeat dimeric rbd protein-based covid-19 vaccine zf2001 protects mice and nonhuman primates. *Emerging microbes & infections*, 11(1):1058–1071, 2022.

[118] Helyaneh Ziaei Jam, Yang Li, Ross DeVito, Nima Mousavi, Nichole Ma, Ibra Lujumba, Yagoub Adam, Mikhail Maksimov, Bonnie Huang, Egor Dolzhenko, et al. A deep population reference panel of tandem repeat variation. *Nature communications*, 14(1):6711, 2023.

[119] Donya A Khalid and Nasser Nafea. A deep neural network model for paternity testing based on 15-loci str for iraqi families. *Journal of Intelligent Systems*, 32(1):20230041, 2023.

[120] Li Fang, Qian Liu, Alex Mas Monteys, Pedro Gonzalez-Alegre, Beverly L Davidson, and Kai Wang. Deeprepeat: direct quantification of short tandem repeats on signal data from nanopore sequencing. *Genome biology*, 23(1):108, 2022.

[121] Hilal Arslan. Covid-19 prediction based on genome similarity of human sars-cov-2 and bat sars-cov-like coronavirus. *Computers & Industrial Engineering*, 161:107666, 2021.

[122] Israel Pagán, Edward C Holmes, and Etienne Simon-Loriere. Level of gene expression is a major determinant of protein evolution in the viral order mononegavirales. *Journal of virology*, 86(9):5253–5263, 2012.

[123] Kohei Hamanaka, Daisuke Yamauchi, Eriko Koshimizu, Kei Watase, Kaoru Mogushi, Kinya Ishikawa, Hidehiro Mizusawa, Naomi Tsuchida, Yuri Uchiyama, Atsushi Fujita, et al. Genome-wide identification of tandem repeats associated with splicing variation across 49 tissues in humans. *Genome Research*, 33(3):435–447, 2023.

[124] Sudhir Kumar, Glen Stecher, Michael Li, Christina Knyaz, and Koichiro Tamura. Mega x: molecular evolutionary genetics analysis across computing platforms. *Molecular biology and evolution*, 35(6):1547–1549, 2018.

[125] Anthony J Hannan. Tandem repeat polymorphisms: modulators of disease susceptibility and candidates for 'missing heritability'. *Trends in genetics*, 26(2):59–65, 2010.

[126] Wen-Ming Zhao, Shu-Hui Song, Mei-Li Chen, Dong Zou, Li-Na Ma, Ying-Ke Ma, Ru-Jiao Li, Li-Li Hao, Cui-Ping Li, Dong-Mei Tian, et al. The 2019 novel coronavirus resource. *Yi chuan= Hereditas*, 42(2):212–221, 2020.

[127] Dau Phan, Giang N Ngoc, Favorisen R Lumbanraja, Mohammad R Faisal, Bahriddin Abipihi, Bedy Purnama, Mera K Delimiyanti, Mamoru Kubo, and Kenji Satou. Combined use of k-mer numerical features and position-specific categorical features in fixed-length dna sequence classification. *Journal of Biomedical Science and Engineering*, 10(8):390–401, 2017.

[128] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[129] Hanyu Zhang, Che-Lun Hung, Meiyuan Liu, Xiaoye Hu, and Yi-Yang Lin. Ncnet: Deep learning network models for predicting function of non-coding dna. *Frontiers in genetics*, 10:432, 2019.

[130] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[131] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *arXiv preprint arXiv:1905.03554*, 2019.

[132] Inc Github. Github, 2016.

[133] Hilal Arslan. Machine learning methods for covid-19 prediction using human genomic data. *Multidisciplinary digital publishing institute proceedings*, 74(1):20, 2021.

[134] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[135] Loıc Ponger and Dominique Mouchiroud. Cpgprod: identifying cpg islands associated with transcription start sites in large genomic mammalian sequences. *Bioinformatics*, 18(4):631–633, 2002.

[136] Yong Wang, Jun-Ming Mao, Guang-Dong Wang, Zhi-Peng Luo, Liu Yang, Qin Yao, and Ke-Ping Chen. Human sars-cov-2 has evolved to reduce cg dinucleotide in its open reading frames. *Scientific Reports*, 10(1):1–10, 2020.

[137] Caleb Joseph Othieno, Siham Sikander, and Xing Su. Ahsan nawaz 1*, xing su 1*, muhammad qasim barkat 2, sana asghar3, ali asad4, farwa basit 5, shahid iqbal6, hafiz zahoor 7 and syyed adnan raheel shah 8. *System level Interventions, Prevention Strategies, Mitigation Policies and Social Responses During COVID-19 That Improve Mental Health Outcomes: Evidence From Lower-and Middle-Income Countries (LMICs)*, 2022.

[138] Chandra Mohan Dasari, Santhosh Amilpur, and Raju Bhukya. Exploring variable-length features (motifs) for predicting binding sites through interpretable deep neural networks. *Engineering Applications of Artificial Intelligence*, 106:104485, 2021.

[139] Hilal Arslan and Hasan Arslan. A new covid-19 detection method from human genome sequences using cpg island features and knn classifier. *Engineering Science and Technology, an International Journal*, 24(4):839–847, 2021.

[140] Yasin Kaya and Ercan Gürsoy. A mobilenet-based cnn model with a novel fine-tuning mechanism for covid-19 infection detection. *Soft Computing*, 27(9):5521–5535, 2023.

[141] Naima Bousnina, Sanaa Ghouzali, Mounia Mikram, Maryam Lafkih, Ohoud Nafea, Muna Al-Razgan, and Wadood Abdul. Hybrid multimodal biometric template protection. *Intell Autom Soft Comput*, 27(1):35–51, 2021.

[142] Raju Hazari and Parimal Pal Chaudhuri. Analysis of coronavirus envelope protein with cellular automata model. *International Journal of Parallel, Emergent and Distributed Systems*, 37(6):623–648, 2022.

[143] Maziar Yazdani and Fariborz Jolai. Lion optimization algorithm (loa): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, 3(1):24–36, 2016.

[144] Bihter Das. An implementation of a hybrid method based on machine learning to identify biomarkers in the covid-19 diagnosis using dna sequences. *Chemometrics and Intelligent Laboratory Systems*, 230:104680, 2022.

[145] Santhosh Amilpur and Raju Bhukya. Edeepssp: explainable deep neural networks for exact splice sites prediction. *Journal of Bioinformatics and Computational Biology*, 18(04):2050024, 2020.

[146] Si Chen, Lih-Yuan Deng, Dale Bowman, Jyh-Jen Horng Shiau, Tit-Yee Wong, Behrouz Madahian, and Henry Horng-Shing Lu. Phylogenetic tree construction using trinucleotide usage profile (tup). In *BMC bioinformatics*, volume 17, pages 117–130. BioMed Central, 2016.

[147] Xiangao Jiang, Megan Coffee, Anasse Bari, Junzhang Wang, Xinyue Jiang, Jianping Huang, Jichan Shi, Jianyi Dai, Jing Cai, Tianxiao Zhang, et al. Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. *Computers, Materials & Continua*, 63(1):537–551, 2020.

[148] Pratichi Basak, Saurabh De, Mallika Agarwal, Aakarsh Malhotra, Mayank Vatsa, and Richa Singh. Multimodal biometric recognition for toddlers and pre-school children. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 627–633. IEEE, 2017.

[149] Md Shahadat Hossain, AQM Sala Uddin Pathan, Md Nur Islam, Mahafujul Islam Quadery Tonmoy, Mahmudul Islam Rakib, Md Adnan Munim, Otun Saha, Atqiya Fariha, Hasan Al Reza, Maitreyee Roy, et al. Genome-wide identification and prediction of sars-cov-2 mutations show an abundance of variants: Integrated study of bioinformatics and deep neural learning. *Informatics in Medicine Unlocked*, 27:100798, 2021.

[150] Fabienne FV Chevance and Kelly T Hughes. Case for the genetic code as a triplet of triplets. *Proceedings of the National Academy of Sciences*, 114(18):4745–4750, 2017.

[151] Juliana Carneiro Gomes, Aras Ismael Masood, Leandro Honorato de S. Silva, Janderson Ferreira, Agostinho AF Júnior, Allana Lais dos Santos Rocha, Letícia Castro, Nathália RC da Silva, Bruno JT Fernandes, and Wellington Pinheiro dos Santos. Optimizing the molecular diagnosis of covid-19 by combining rt-pcr and a pseudo-convolutional machine learning approach to characterize virus dna sequences. *BioRxiv*, pages 2020–06, 2020.

[152] Amin Ullah, Khalid Mahmood Malik, Abdul Khader Jilani Saudagar, Muhammad Badruddin Khan, Mozaherul Hoque Abul Hasanat, Abdullah AlTameem, Mohammed AlKhathami, and Muhammad Sajjad. Covid-19 genome sequence analysis for new variant prediction and generation. *Mathematics*, 10(22):4267, 2022.

[153] Muthulakshmi Murugaiah and Murugeswari Ganesan. A novel frequency based feature extraction technique for classification of corona virus genome and discovery of covid-19 repeat pattern. *Brazilian Archives of Biology and Technology*, 64:e21210075, 2022.

[154] Feiming Huang, Lei Chen, Wei Guo, Xianchao Zhou, Kaiyan Feng, Tao Huang, and Yudong Cai. Identifying covid-19 severity-related sars-cov-2 mutation using a machine learning method. *Life*, 12(6):806, 2022.

[155] Jie Cui, Fang Li, and Zheng-Li Shi. Origin and evolution of pathogenic coronaviruses. *Nature Reviews Microbiology*, 17(3):181–192, 2019.

[156] Stuart G Siddell, John Ziebuhr, and Eric J Snijder. Coranaviruses, toroviruses and arteriviruses. In *Topley and Wilson's microbiology and microbial infections*, pages 823–856. Edward Arnold, 2005.

[157] Torsten Thomas, Jack Gilbert, and Folker Meyer. Metagenomics-a guide from sampling to data analysis. *Microbial informatics and experimentation*, 2(1):3, 2012.

[158] Ali M Zaki, Sander Van Boheemen, Theo M Bestebroer, Albert DME Osterhaus, and Ron AM Fouchier. Isolation of a novel coronavirus from a man with pneumonia in saudi arabia. *New England Journal of Medicine*, 367(19):1814–1820, 2012.

[159] Yuefei Jin, Haiyan Yang, Wangquan Ji, Weidong Wu, Shuaiyin Chen, Weiguo Zhang, and Guangcai Duan. Virology, epidemiology, pathogenesis, and control of covid-19. *Viruses*, 12(4):372, 2020.

[160] Arash Keshavarzi Arshadi, Julia Webb, Milad Salem, Emmanuel Cruz, Stacie Calad-Thomson, Niloofar Ghadirian, Jennifer Collins, Elena Diez-Cecilia, Brendan Kelly, Hani Goodarzi, et al. Artificial intelligence for covid-19 drug discovery and vaccine development. *Frontiers in Artificial Intelligence*, 3:65, 2020.

[161] Pardeep Garg, Sunildatt Sharma, and Sanjeev Narayan Sharma. Tandem repeats detection in dna sequences using p-spectrum based algorithm. In *2017 Conference on Information and Communication Technology (CICT)*, pages 1–5. IEEE, 2017.

[162] Oxana Lundström. *Intrinsic disorder and tandem repeats-match made in evolution: Computational studies of molecular evolution*. PhD thesis, Department of Biochemistry and Biophysics, Stockholm University, 2023.

[163] Hossein Savari, Hassan Shafiey, Abdorreza Savadi, Nayyereh Saadati, and Mahmoud Naghibzadeh. Statistics and patterns of occurrence of simple tandem repeats in sars-cov-1 and sars-cov-2 genomic data. *Data in Brief*, 36:107057, 2021.

[164] Tetsuya Akaishi, Kei Fujiwara, and Tadashi Ishii. Variable number tandem repeats of a 9-base insertion in the n-terminal domain of severe acute respiratory syndrome coronavirus 2 spike gene. *Frontiers in Microbiology*, 13:1089399, 2023.

[165] Simson Tarigan, NLP Indi Dharmayanti, Dianita Sugiartanti, Ryandini Putri, Andriani, Harimurti Nuradji, Marthino Robinson, Niniek Wiendayanthi, and Fadjry Djufri. Characterization of two linear epitopes sars cov-2 spike protein formulated in tandem repeat. *PLoS One*, 18(1):e0280627, 2023.

[166] Gavin Hanson and Jeff Coller. Codon optimality, bias and usage in translation and mrna decay. *Nature reviews Molecular cell biology*, 19(1):20–30, 2018.

[167] Shuai Xia, Lijue Wang, Yun Zhu, Lu Lu, and Shibo Jiang. Origin, virological features, immune evasion and intervention of sars-cov-2 omicron sublineages. *Signal Transduction and Targeted Therapy*, 7(1):241, 2022.

[168] Rita Gemayel, Marcelo D Vinces, Matthieu Legendre, and Kevin J Verstrepen. Variable tandem repeats accelerate evolution of coding and regulatory sequences. *Annual review of genetics*, 44:445–477, 2010.

[169] Susan T Lovett. Encoded errors: mutations and rearrangements mediated by misalignment at repetitive dna sequences. *Molecular microbiology*, 52(5):1243–1253, 2004.

[170] John K Blackwood, Ewa A Okely, Rabaab Zahra, John K Eykelenboom, and David RF Leach. Dna tandem repeat instability in the escherichia coli chromosome is stimulated by mismatch repair at an adjacent cag· ctg trinucleotide repeat. *Proceedings of the National Academy of Sciences*, 107(52):22582–22586, 2010.

[171] Hoang Dang Khoa Do and Joo-Hwan Kim. A dynamic tandem repeat in monocotyledons inferred from a comparative analysis of chloroplast genomes in melanthiaceae. *Frontiers in Plant Science*, 8:246368, 2017.

[172] Hoang Dang Khoa Do Hoang Dang Khoa Do and Kim JooHwan Kim JooHwan. A dynamic tandem repeat in monocotyledons inferred from a comparative analysis of chloroplast genomes in melanthiaceae. 2017.

[173] Susan T Lovett and Vladimir V Feschenko. Stabilization of diverged tandem repeats by mismatch repair: evidence for deletion formation via a misaligned replication intermediate. *Proceedings of the National Academy of Sciences*, 93(14):7120–7124, 1996.

[174] Marie C Schoelmerich, Rohan Sachdeva, Lucas Waldburger, Jacob West-Roberts, and Jillian F Banfield. Borg tandem repeats undergo rapid evolution and are under strong selection to create new intrinsically disordered regions in proteins. *bioRxiv*, pages 2022–05, 2022.

[175] DL Gonzalez, S Giannerini, and R Rosa. On the origin of degeneracy in the genetic code. *Interface focus*, 9(6):20190038, 2019.

[176] Deendayal Dinakarpandian, Venetia Raheja, Saumil Mehta, Erin G Schuetz, and Peter K Rogan. Tandem machine learning for the identification of genes regulated by transcription factors. *BMC bioinformatics*, 6:1–10, 2005.

[177] Maren Maanja, Peter A Noseworthy, Jeffrey B Geske, Michael J Ackerman, Adelaide M Arruda-Olson, Steve R Ommen, Zachi I Attia, Paul A Friedman, and Konstantinos C Siontis. Tandem deep learning and logistic regression models to optimize hypertrophic cardiomyopathy detection in routine clinical practice. *Cardiovascular Digital Health Journal*, 3(6):289–296, 2022.

[178] Aggraj Gupta, Chandan Bhat, Emir Karahan, Kaushik Sengupta, and Uday K Khankhoje. Machine learning based tandem network approach for antenna design. In *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, pages 1–2. IEEE, 2022.

[179] Omar Shouman, Wassim Gabriel, Victor-George Giurcoiu, Vitor Sternlicht, and Mathias Wilhelm. Prospect: Labeled tandem mass spectrometry dataset for machine learning in proteomics. *Advances in Neural Information Processing Systems*, 35:32882–32896, 2022.

[180] Joshua E Elias, Francis D Gibbons, Oliver D King, Frederick P Roth, and Steven P Gygi. Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature biotechnology*, 22(2):214–219, 2004.

[181] S Codrean, B Kruit, N Meekel, D Vughs, and F Béen. Predicting the diagnostic information of tandem mass spectra of environmentally relevant compounds using machine learning. *Analytical chemistry*, 95(42):15810–15817, 2023.

[182] David Edwards, Jason Stajich, and David Hansen. *Bioinformatics: tools and applications*. Springer Science & Business Media, 2009.

[183] Sebastian Böcker. Searching molecular structure databases using tandem ms data: are we there yet? *Current Opinion in Chemical Biology*, 36:1–6, 2017.

[184] Anne MP Canuto, Fernando Pintro, and João C Xavier-Junior. Investigating fusion approaches in multi-biometric cancellable recognition. *Expert Systems with applications*, 40(6):1971–1980, 2013.

[185] Chandra Mohan Dasari and Raju Bhukya. Comparative analysis of protein synthesis rate in covid-19 with other human coronaviruses. *Infection, Genetics and Evolution*, 85:104432, 2020.

[186] Qianfan Wu, Adel Boueiz, Alican Bozkurt, Arya Masoomi, Allan Wang, Dawn L DeMeo, Scott T Weiss, and Weiliang Qiu. Deep learning for predicting disease status using genomic data. Technical report, PeerJ Preprints, 2018.

[187] Bo Ram Beck, Bonggun Shin, Yoonjung Choi, Sungsoo Park, and Keunsoo Kang. Predicting commercially available antiviral drugs that may act on the novel coronavirus (sars-cov-2) through a drug-target interaction deep learning model. *Computational and structural biotechnology journal*, 18:784–790, 2020.

[188] Gciniwe S Dlamini, Stephanie J Müller, Rebone L Meraba, Richard A Young, James Mashiyane, Tapiwa Chiwewe, and Darlington S Mapiye. Classification of covid-19 and other pathogenic sequences: A dinucleotide frequency and machine learning approach. *Ieee Access*, 8:195263–195273, 2020.

[189] Ru-Fang Yeh, Lee P Lim, and Christopher B Burge. Computational inference of homologous gene structures in the human genome. *Genome research*, 11(5):803–816, 2001.

[190] Chris Burge and Samuel Karlin. Prediction of complete gene structures in human genomic dna. *Journal of molecular biology*, 268(1):78–94, 1997.

[191] Mario Stanke and Burkhard Morgenstern. Augustus: a web server for gene prediction in eukaryotes that allows user-defined constraints. *Nucleic acids research*, 33(suppl_2):W465–W467, 2005.

[192] Daniel R Zerbino, Premanand Achuthan, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Jyothish Bhai, Konstantinos Billis, Carla Cummins, Astrid Gall, Carlos García Girón, et al. Ensembl 2018. *Nucleic acids research*, 46(D1):D754–D761, 2017.

[193] National Center for Biotechnology Information Search database. `https://www.ncbi.nlm.nih.gov/`. Accessed: 2019-5-15.

[194] Sarah J Wheelan, Deanna M Church, and James M Ostell. Spidey: a tool for mrna-to-genomic alignments. *Genome research*, 11(11):1952–1957, 2001.

[195] NIH:U.S. National Library of Medicine. `https://ghr.nlm.nih.gov/gene/`. Accessed: 2019-5-16.

[196] Simon C Warby, Alexandre Montpetit, Anna R Hayden, Jeffrey B Carroll, Stefanie L Butland, Henk Visscher, Jennifer A Collins, Alicia Semaka, Thomas J Hudson, and Michael R Hayden. Cag expansion in the huntington disease gene is associated with a specific and targetable predisposing haplogroup. *The American Journal of Human Genetics*, 84(3):351–366, 2009.

[197] Maciej Figiel, Wlodzimierz J Krzyzosiak, Pawel M Switonski, and Wojciech J Szlachcic. Mouse models of sca3 and other polyglutamine repeat ataxias. In *Movement Disorders*, pages 991–1016. Elsevier, 2015.

[198] Suran Nethisinghe, Maria Lucia Pigazzini, Sally Pemble, Mary G Sweeney, Robyn Labrum, Katarina Manso, David Moore, Jon Warner, Mary B Davis, and Paola Giunti. Polyq tract toxicity in sca1 is length dependent in the absence of cag repeat interruption. *Frontiers in cellular neuroscience*, 12:200, 2018.

[199] Hussein Daoud, Véronique Belzil, Sandra Martins, Mike Sabbagh, Pierre Provencher, Lucette Lacomblez, Vincent Meininger, William Camu, Nicolas Dupré, Patrick A Dion, et al. Association of long atxn2 cag repeat sizes with increased risk of amyotrophic lateral sclerosis. *Archives of neurology*, 68(6):739–742, 2011.

[200] X Zhou, C Wang, D Ding, Z Chen, Y Peng, H Peng, X Hou, P Wang, W Ye, T Li, et al. Analysis of (cag) n expansion in atxn1, atxn2 and atxn3 in chinese patients with multiple system atrophy. *Scientific reports*, 8(1):3889, 2018.

[201] National Center for Biotechnology Information Search database. `https://www.ncbi.nlm.nih.gov/gene/`. Accessed: 2019-5-15.

List of Publications

Journal Publications:

1. **Praveen Gugulothu** , and Raju Bhukya.“Exploring Coronavirus Sequence Motifs through Convolutional Neural Network for Accurate Identification of Covid-19" *Computer Methods in Biomechanics and Biomedical Engineering*.(**Accepted, October 2024**, Indexing: SCIE, IF: 1.9, Publisher: Taylor & Francis).

2. **Praveen Gugulothu** , and Rahu Bhukya. "Coot-Lion Optimized Deep learning Algorithm for COVID-19 Point Mutation Rate Prediction using Genome Sequences." *Computer Methods in Biomechanics and Biomedical Engineering* (**Published 2023**, Indexing: SCIE, IF: 1.9, Publisher: Taylor & Francis)
   DOI: `https:doi.org10.1080/10255842.2023.2244109`

3. **Praveen Gugulothu**, and Raju Bhukya. "DNA Sequence Clustering and ERSIT-GRU for Repeat Detection in COVID -19 Prediction." *Current Neuropharma- cology*:(**Accepted**, Indexing: SCIE, IF: 5.69, Publisher: Bentham Science).

4. **Praveen Gugulothu** , and Raju Bhukya. "Genome -Wide Analysis for Covid-19 detection for Tandem Repeat Error and substitution Error using Harris Hawks Optimization" *Computers and Electrical Engineering* .(**Under Revision**, Indexing: SCIE, IF: 4.3, Publisher: Elsevier)

Other Publications:

1. Kishore Babu, **Praveen Gugulothu** , Raju Bhukya . ”Heart Abnormality Classification using ECG and PCG Recodrings with Novel PJM-DJRNN”, *Expert Systems with Applications* (**Under Revision**, Indexing: SCIE, IF: 8.1, Publisher: Elsevier).