Volume 68, Issue 3, 2024
Journal of Scientific Research
of
The Banaras Hindu University

A Deep Learning Framework for Accurate Diagnosis of Chromosome 15-associated Monogenetic Disorders: Leveraging Alignment-Free and Alignment-Based Methods with Optimization Techniques

Dr.Akila Rajini.S[*1] and Dr.Nandhini.K [2]

[*1]Assistant Professor/Information Technology, Kamaraj College of Engineering and Technology, akilarajiniit@kamarajengg.edu.in
[2] Research Associate, nandhukk28@gmail.com

Abstract: **Monogenetic diseases associated with chromosome 15 pose diagnostic challenges due to their intricate inheritance patterns. A novel framework is needed to address the complex challenges. Hence, two novel deep learning approaches have been proposed for the diagnosis with the aim of improving accuracy: an alignment-free method using mean shift clustering, one-hot encoding, and a Convolutional Neural Network (CNN) – Bi-directional Long-Short term Memory (BiLSTM) architecture optimized by Elephant Herd Optimization (EHO), and an alignment-based method employing the Needleman-Wunsch algorithm, CNN-BiLSTM, and Horse Herd Optimization (HHO). The nature-inspired genetic optimization techniques, EHO and HHO optimizes model performance by effectively exploring the search space. The above deep learning frameworks are evaluated on UBE3A and FBN1 datasets and simulation results produce 77.5% and 89.2% accuracy respectively in classifying Marfan syndrome and Angelman syndrome. The results show the significance of deep learning combined with optimization improves the diagnosis and genetic counseling of monogenetic diseases. Future research will focus on identifying specific mutation variants within the Deoxyribonucleic acid (DNA) sequence of chromosome 15 that are responsible for diseases like Rett syndrome and Prader-Willi syndrome paving the way for personalized medicine.**

Index Terms: **Convolutional Neural Network, Elephant Herd Optimization, Horse Herd Optimization, Long-short term memory, Monogenetic disorders**

I. INTRODUCTION

Monogenetic diseases caused by mutations of a single gene (Antonarakis, 2016) pose significant diagnostic challenges. Mutations in genes associated with chromosome 15 are characterized by intellectual disability seizers and delays and if unattended leads to brain disorders and tissue related disorders like Angelman syndrome, Marfan syndrome etc. Hence, Accurate diagnosis is needed to improve genetic counseling.

The challenges associated with genetic mutations include genetic heterogeneity, variable expressivity, and phenotypic overlap. Genetic heterogeneity occurs when mutations in different genes affects a single phenotype, the variable expressivity is caused when individuals with the same genetic mutation presents several degrees of symptoms. Phenotypic overlap is raised when different genetic diseases have similar clinical features.

DNA sequence analysis is the first step in diagnosis of genetic disorders. Traditional laboratory methods are time-consuming (Sanger et al., 1977) and the novel deep neural network model has proven as an efficient and accurate approach for analyzing large-scale genetic data.

DNA sequence alignment is a generic technique (Alipanahi et al., 2015) that compares and identifies common features of DNA sequences. Among the sequence alignment techniques, the global alignment technique identifies the overall similarity by considering the entire length of two sequences. The Needleman-Wunsch algorithm is the most familiar method of global alignment. The Local alignment methodology identifies the most similar regions within two sequences regardless of their overall similarity. The Smith-Waterman algorithm is a well-known method of local sequence alignment. These alignment algorithms facilitate identification of conserved regions, functional regions and mutations within DNA sequences.

Alignment-free techniques are an alternative approach for genetic sequence comparison as it reduces the computational overhead of sequence alignments. The simple clustering

[*] Corresponding Author

technique, mean shift clustering can be used to identify regions of high density in the data space and assigns points to their nearest cluster center. This method focus on overall composition or other features.

As a known fact, sequence alignment tool plays a vital role for analyzing DNA sequences in diagnosis, it alone may not be sufficient for monogenetic disorders based on mutations in chromosome 15 genes. Deep neural network frameworks automatically learn complex patterns and features from DNA sequences provides a more comprehensive and accurate approach to diagnosis. Convolutional Neural Networks (CNNs) are especially well-suited for processing spatial data of DNA sequences (Krizhevsky et al., 2012) and they extract local features from the input data using filters, allowing them to identify patterns and correlate with associated DNA sequences. The effectiveness of the tasks such as identifying specific mutation patterns or predicting the functional impact of genetic variants can be improved using the proposed methodology.

The gate-based deep learning technique, Long Short-Term Memory's (LSTM) analyzes long-range dependent DNA sequences (Schuster & Paliwal, 1997) and then CNN model enhances the scalability and reliability in the process of identifying and classifying disease-related patterns. LSTM networks handle sequential data with respect to long-range dependencies in DNA sequences. The gates used in LSTMs control the flow of information, capture and store of significant information over time. This is very important in case of DNA sequence analysis since the context of a specific nucleotide is influenced by nucleotides that are located in distant positions.

Bi-directional LSTMs (Bi-LSTM) process DNA sequences in both forward and backward directions that captures information of past and future contexts. This facilitates identification of patterns and dependencies that are ignored in unidirectional LSTMs.

To optimize the performance of deep neural network frameworks, nature-inspired optimization algorithms (Yang, 2014) can be incorporated as a module. These algorithms efficiently explore and exploit the search space that leads to enhanced accuracy of the framework and optimized use of hyperparameters (Mondal et al., 2019).

The deep neural network framework of CNN-BiLSTM enhances its performance for monogenetic disease classification using Horse Herd Optimization (HHO) and Elephant Herd Optimization (EHO) as these algorithms have proven effectiveness in global optimization tasks and balances exploration and exploitation (Mondal et al., 2019). Hence, these algorithms are better option to be embedded in the proposed model.

Horse Herd Optimization (HHO) algorithm uses the hunting behavior of horse herds with its elaborative memory power. HHO combines both exploration and exploitation strategies to find optimal solutions. The exploration phase finds different regions of the search space and the exploitation phase refines the solution for selecting the best.

Elephant Herd Optimization (EHO) uses the social behavior of elephant herds. EHO uses a hierarchical structure to find the optimal solutions. The leader of the herd re directs the search and other elephants follow and provides support to the search process.

The hybrid combination of CNNs, Bi-LSTMs, and nature-inspired optimization algorithms aims to innovate a methodology for classifying monogenetic diseases associated with chromosome 15. These novel techniques improve diagnostic accuracy, reduce the computational time and provide valuable insights into the genetic mechanisms that are essential for the diagnosis.

The proposed deep neural network model can be used especially for monogenetic disorders in the following aspects:

- Identify disease-causing mutations: Through patient's DNA sequences clustering and alignment specific mutations responsible for monogenetic diseases are detected.
- Improve diagnostic accuracy: These models provide more accuracy in diagnoses than traditional methods that in turn reduces the risk of misdiagnosis.
- Enable earlier diagnosis: The deep neural network model is adaptable and reliable with respect to diagnosis thereby aids in effective treatment.
- Support personalized medicine: The specific genetic variants are identified in early stage and supports personalized care and treatment.

The important aspects of this paper are:

- Monogenetic Disorder Detection Model: Two deep neural network models are proposed for classification of Angelman syndrome and Marfan syndrome: (i) An alignment-free model with mean shift clustering, CNN-BiLSTM, and EHO and (ii) An alignment-based model with Needleman-Wunsch algorithm, CNN-BiLSTM, and HHO.
- Identifying Disease-Causing Mutations in Chromosome 15 Genes: The proposed models specifically identify disease-causing mutations in the locations 15q11-q13 meant for Angelman syndrome and 15q21.1 associated with Marfan Syndrome.
- Improved Diagnostic Performance for Angelman and Marfan Syndromes: The performance of model are evaluated using the metrics accuracy, recall, specificity, precision, and F1-score in classifying Angelman Syndrome and Marfan Syndrome for UBE3A and FBN1 datasets.

The specific monogenetic disorder, Angelman syndrome is often realized by intellectual disability, seizures, and speech impairment and Marfan syndrome, a connective tissue disorder directly affects cardiovascular and skeletal systems. This paper focuses on accurate detection of such disorders.

This paper is organized in the following manner: The next section, Section II reviews existing methods for single gene sequence analysis associated with disorders. The Section 3 describes the datasets UBE3A and FBN1 associated with Angelman and Marfan syndrome and the proposed models comprising of alignment-based and alignment-free deep neural network models. Then Section 4 elaborates the two proposed frameworks with algorithms and their work flow. Next, Section 5 presents the experimental evaluations and simulations of the proposed work. Finally, to conclude, the Section 6 summarizes the key findings and outlines potential future research directives.

II. Related Work

This section presents comprehensive reviews on the existing approaches of traditional and conventional sequence alignment techniques, deep learning techniques and optimization techniques in genetic domain classification and identification of monogenetic disorders. The rapid advancements in deep learning have revolutionized the field of computational biology, offering novel and powerful approaches to address complex challenges in genomics.

A. *Traditional Machine Learning Techniques for Sequence Analysis*

Machine learning techniques evolved into many algorithms and methods for DNA sequence analysis addressing the issues associated sequence mutations. To be more specific, clustering algorithms are scalable and data-driven approach that have been widely used to group similar sequences based on their local and global features.

K-means clustering, the most familiar algorithm find its significance place (Timothy Chappell et. al, 2017) in sequence analysis. This algorithm first converts the sequences into an intermediate binary format and uses Hamming distance to compute the partitions of k clusters, where k is a predefined number and iteratively assigns sequences to the nearest cluster centroid, updating the centroids until convergence. The k-means is simple and efficient, and the challenging issue is the choice of k and non-spherical clusters handling.

Hierarchical clustering (Dan Wei et. al, 2012) also plays a significant role in bio-sequence analysis. This paper transformed DNA sequences into the feature vectors and identify occurrence, location and order relation of k-tuples. Then clustering on sequences are done based on the feature vectors. It constructs a hierarchy of clusters, starting with each sequence as a separate cluster and merging them based on their similarity. This methodology is applying either as agglomerative (bottom-up) or divisive (top-down) approach. This methodology of clustering is computationally expensive for large datasets.

Mean shift clustering (Hani Z Girgis, 2022), an unsupervised learning technique is a non-parametric algorithm shifts data points iteratively to form clusters with regions of higher density. Unlike k-means, mean shift does not require the initial value of number of clusters in prior. It is a better option to choose this type of clustering when the distribution of feature vectors is highly impossible to determine. The mean shift vector for each data point is computed from the difference between the point's current location and the weighted average of its neighbors within a specific range. The data points are continuously shifted towards the mean shift vector. The final positions of the points determine the cluster centers. The most significant factor concerned with mean shift clustering is that it handles non-linear shaped clusters and discover clusters of different densities.

The traditional clustering algorithms with respect to genetic sequence analysis pose the following challenges:

- Parameter Selection: The challenge in determining the total number of clusters, k.
- Noise and Outliers: Noise and Outliers of datapoints distort the distribution and formation of clusters.
- Shape and Density: Non-linear feature vectors pose a significant challenge in generating clusters.
- Computational Cost: Clustering algorithms are computationally expensive for large datasets.

B. *Conventional Sequence Alignment Techniques*

Sequence alignment is a fundamental and the most significant technique in bioinformatics used to identify similarities and differences between biological sequences especially DNA. By aligning sequences, geneticist find evolutionary relationships, functional similarities, and potential disease-causing mutations.

Basically, there are two types of sequence alignments: Pairwise Sequence Alignment (PSA) for aligning two DNA sequences (Hasna El Haji et al. 2020) to find the best possible arrangement that maximizes their similarity and Multiple Sequence Alignment (MSA) that aligns more than three sequences simultaneously (Kouser et al. 2015) to identify conserved regions and evolutionary relationships. Pairwise Sequential Alignment is further subdivided into local alignment (Waqar Haque et al. 2009) that identifies regions of high similarity, allows gaps, mismatches in other parts and globally aligns that entire length of both the sequences by considering the gaps.

The well-known algorithms of dynamic programming (Jean-Michel Richer et al. 2007) such as Smith-Waterman algorithm and Needleman-Wunsch algorithm are used for local and global sequence alignment respectively. DNA sequence alignment uses the scoring matrix (Jian-Jun et al. 2012) that assigns a positive value to the match, negative value to the mismatch, and a negative to the gap penalty. It is a usual practice to keep the residues and gaps together, and considers the frameshift of the mutations thereby identifies the mutation variant responsible for genetic diseases.

The Needleman-Wunsch Algorithm (NWA) is the first sequence alignment algorithm Maros et al. (2021) used as a biological sequence alignment. Mutation variant can be fixed through external conditions such as Ultraviolet (UV) light, X-rays or various chemicals. Matching certain letters in the sequence is assigned a higher score than matching other letters. The Alignment for small gaps starts with a gap penalty. NWA sets up the objective function that is used to maximize the alignment score for two sequences of same length and it is also necessary to observe the mutation probability (Henikoff et al. 1992). Scoring Matrix can also be determined using Manhattan distance (Chen et al. 2005). The scoring system need to consider probabilities with which different proteins (Amr Ezz El-Din Rashed et al. 2021) can be substituted.

The Smith-Waterman Algorithm Sudha et al. (2014) is a dynamic programming strategy to perform the similarity match. It generates the score for contextual and evolutionary relationship establishment. The scoring matrix is non-symmetry. During the comparison of each character position, the score is determined a match/mismatch score or an insertion/deletion (indel) or shift penalty along with a match/mismatch score. This search is to find the muted gene of the DNA sequence. This is a sensitive algorithm aids in construction of accurate phylogenetic trees (Daiki Okada et al. 2015).

C. *Deep Learning Techniques*

The self-learning algorithms in the deep neural network for automated learning of sequence structure are made possible using Convolutional Neural Network (CNN). However, deep learning approaches (Alexandra Miere et al. 2020) require large volumes of high-quality training data, which may be a challenging premise in a clinical setting. These high volumes of data are even more difficult to obtain in the case of monogenetic disorders during the earlier stage in the prenatal investigation due to the rareness of genetic conditions. Data augmentation can be done on the training dataset to reduce overfitting of data (Taeho Jo et al. 2022). Softmax function must be used along with the test data to assess the uncertainty of the model.

Convolution Neural Network algorithm is a promising algorithm that performs non-linear transformations on aligned DNA sequences and extracts features from such high-dimensional data (Michael Wainberg et al. 2018). Deep learning is challenging in this era in Genome-Wide Association Studies (GWAS) handling high-dimensional genomic data. The three step approach for identification of genetic variants (Taeho Jo et al. 2022) using CNN to identify phenotype-related Single Nucleotide Polymorphisms (SNPs) has been applied in implementing accurate disease classification models. CNN has the ability to capture the mutation that regulates the genetic expression (Scherer et al. 2021) and thus it identifies the structure that is responsible for genetic diseases. Initially the whole genome is divided into non-overlapping fragments with an optimal size. Then CNN selects phenotype-associated fragments for each fragment. A Sliding Window Association Test (SWAT) is used in CNN (Annalisa Buniello et al. 2019) to estimate Phenotype Influence Scores (PIS) and identifies phenotype-associated SNPs based on PIS. Finally, the classification process proceeds with all identified SNPs.

Deep learning techniques use one Single Nucleotide Polymorphism (SNP) at a time on the whole genome (Jian Yan et al. 2021) to find population-based genetic risk variation for genetic diseases. However, it produces a challenging task of handling high-dimensional low-sample size (Auton et al. 2015) on GWAS that has direct impact on mutation variant. Hence, it is necessary to perform feature reduction (Makoto Yamada et al. 2019). Rectified Linear Unit (ReLU) is preferable to use as activation function to overcome the gradient vanishing problem. Adam is most suitable to use as an optimization function (Sutskever et al. 2013) that uses Stochastic Gradient Descent (SGD) algorithm to update the weights of Convolution Neural Network during training. CNN uses multiple hidden layers (Shuchao Pang et al. 2018) to observe feature space of sequence data thereby performs the feature extraction at all levels of abstraction with improved performance.

Recurrent Neural Network (RNN) is another deep learning technique most suited for DNA sequence analysis (Lei Chen et al. 2019) and classification. The input layer of RNN captures the aligned sequences. RNN with Long Short-Term Memory (LSTM) finds the long-term dependent sequences that are the pivot sequences containing the mutation variant essential for genetic diseases. Global pooling layer of RNN learns the features and the fully connected layer performs the classification of genetic diseases. The output layer assigns the class probability for the corresponding genetic disease.

D. Nature-Inspired Optimization Techniques

The solution in the search space is initialized randomly, proceed towards feasible solution as iterated through the fitness function and finally the best or optimal solution is chosen among the feasible solutions. The fitness function is designed in such a way that it improves the search space in each run. The Particle Swarm Optimization (PSO) algorithm uses each particle to represent a solution and it gets updated according to the historical behavior of the flights. In each iteration, the particle flies (Eberhart et al. 2001) towards better search space. Each particle learns from its own experience (Cheng et al. 2012) and its companion experiences.

The Elephant Herd Optimization algorithm (EHO) (Monalisa Nayak et al. 2020) solves optimization problems by considering the assumptions that elephants are grouped as clans and their leader is a Matriarch. The older elephants stay away from their family group and these two behaviors of the elephant group lead to two operators: clan updating operator and separating operator. The algorithm simulates the herding behavior of elephants, where they stay close to the leader while still exploring the environment to find better food sources. Thus, the search space is exploited according to the availability of food source.

The Horse Herd Optimization algorithm aims at complex optimization problems such as feature selection (Esin Ays et al. 2023) and designed as per the herd behavior of horses. This algorithm classifies the horses as alpha, beta, gamma and delta as per their age and ranks by their performance. This algorithm exhibits six different behaviours of horses, they are, (i) Grazing: horses used to move freely, (ii) Hierarchy: Strong horses are the leaders and the rest are followers, (iii) Sociability: Some horses are in contact with other animals, (iv) Imitation: Younger horses imitate the behaviours such as finding the right position of pasture, defence mechanism, etc. from adult horses, (v) Defense: Horses either flee from the attack or live in harmony with the aggressor and (vi) Roaming: Horses used to explore variety of new places in search of nearby pastures. The fitness value evaluates and ranks the horses according to their positions, age and behaviour and gets updated in each iteration. The Horse Herd Optimization provides better search space in analyzing gene sequencing.

Thus, a retrospective review is conducted on several researches on DNA sequence analysis, sequence alignments, sequence classification and predictive analysis on mutations. Each research has introduced new facts, solution to the existing issues and open issues or scope of future research. The review also suggested that engrossing programming methodology of dynamic programming of Needleman-Wunsch algorithm for global sequence alignment, the fascinating technology of deep learning, especially convolution neural network, recurrent neural network with long short-term memory for sequence classification and the enthralling field of applying heuristic search by nature-inspired optimization algorithms such as Elephant-herd, Horse-herd algorithms are applied in sequence analysis may produce optimized results on monogenetic diagnosis. With this review, the proposed research work uses the mentioned algorithms and methodologies in performing the

sequence analysis for identifying the mutant variant of the gene that is responsible for monogenetic disorders.

III. Data and Methods

A. Dataset Description

The datasets associated with the field of genetics are valuable resources for understanding the basis of genetic disorders, identifying the causative mutations and for developing diagnostic and therapeutic strategies. There exist many public repositories that contain freely distributed microarray, next-generation sequencing, and other forms of high-throughput functional genomics data submitted by the research community. This section presents the dataset UBE3A and FBN1 associated with Angelman Syndrome and Marfan Syndrome especially connected with Chromosome 15.

i. UBE3A Dataset

Angelman Syndrome (AS) is caused by maternal deletions at 15q11-q13, Paternal Uniparental Disomy (UPD) of chromosome 15, or imprinting mutations especially mutations occurred at Ubiquitin Protein Ligase E3A (UBE3A). The data table associated with UBE3A dataset of GeneID:7337 is shown in Figure 3.1. This data represents expression levels of a specific gene (GeneID 7337) across several human tissues. The gene is most highly expressed in the brain (10.1) and least expressed in the urinary bladder (2.46).

Dataset Size: The UBE3A dataset include data comprising of more than 20 tissue types from 50,000 samples. For detection of mutations responsible for Angelman Syndrome, the dataset was split into three subsets: 60% (approximately 30,000 samples) allocated for training, 20% (around 10,000 samples) for validation, and 20% (approximately 10,000 samples) for testing.

Characteristics: The dataset represent the quantitative expression levels of several tissues associated with Angelman Syndrome. The dataset represents 10.1 brain expression and 2.46 with respect to the urinary bladder. Thus insights of the gene's variability across tissue types is represented in this dataset.

#GeneID	adrenal	appendix	bone marrow	esophagus
7337	7.66	6.84	3.64	7.53

brain	colon	duodenum	endometrium
10.1	7.12	5.34	8.24

fat	gall bladder	heart	kidney
6.71	6.97	6.91	7.11

liver	lung	lymph node	ovary	pancreas
4.53	5.83	7.62	7.33	1.33

prostate	Urinary bladder	skin
8.05	2.46	6.38

Figure 3.1 Data table of UBE3A Dataset

Preprocessing Steps: Normalization is applied over the samples to remove tissue-specific variations. Thereafter, Noise filtering and outlier detection were also implemented for addressing any aberrant expression levels due to technical or biological variability. Moreover the dataset includes a FASTA format DNA sequence (Figure 3.2), a large file containing nucleic acid sequences used for sequence alignment and mutation analysis.

>NC_000015.10:c25439056-25333728 UBE3A [organism=Homo sapiens] [GeneID=7337] [chromosome=15]
GCTGCCTGCCGGGATACTCGGCCCGCCCAGCCAGTCCT
CCCGTCTTGCGCCGCGGCCGCGAGATCCGTGT
GTCTCCCAAGATGGTGGCGCTGGGCTCGGGGTGACTAC
AGGAGACGACGGGGCCTTTTCCCTTCGCCAGG
ACCCGACACACCAGGCTTCGCTCGCTCGCGCACCCCTC
CGCCGCGTAGCCATCCGCCAGCGCGGGCGCCC
GCCATCCGCCGCCTACTTACGCTTCACCTCTGCCGACC
CGGCGCGCTCGGCTGCGGGCGGCGGCGCCTCC
TTCGGCTCCTCCTCGGAATAGCTCGCGGCCTGTAGCCC
CTGGCAGGAGGGCCCCTCAGCCCCCCGGTGTG
GACAGGCAGCGGCGGCTGGCGACGAACGCCGGGATTT
CGGCGGCCCCGGCGCTCCCTTTCCCGGCCTCGT
TTTCCGGATAAGGAAGCGCGGGTCCCGCATGAGCCCC
GGCGGTGGCGGCAGCGAAAGAGAACGAGGCGGT
GGCGGGCGGAGGCGGCGGGCGAGGGCGACTACGACCA
GTGAGGCGGCCGCCGCAGCCCAGGCGCGGGGGC
GACGACAGGTCAGTGTTGCCGCGGCCTGCGCCAGGCG
GCGCTGGCTCCCCTCCGTCACTCGGCCGGCCTT
CGGGGCCCGCTGTGGCGAGGTCGACACCCCCCTTCCCC
GCCCCCCGCCGCCGAGGCGAGTGTTTGGGGGC

Figure 3.2 FASTA format of DNA Sequence Alignment File

Figure 3.3 shows the location (marked Red) indicating the presence of the Angelman Syndrome in chromosome 15 in UBE3A dataset.

Figure 3.3 Chromosome location of Angelman Syndrome

ii. FBN1 Dataset

Marfan Syndrome (MS) is an autosomal dominant monogenetic disorder caused by mutations on the FBN1 gene on chromosome 15. FBN1 is the encoded protein called fibrillin resulting in the formation of elastic fibres found in connective tissue. The structural support of fibrillin may weaken the tissues leading to severe consequences. Figure 3.5 shows the location indicating the mutation in the FBN1 gene.

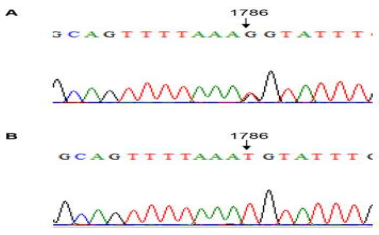

Figure 3.5 (a) Muted location of Marfan Syndrome and

(b) Normal sequence in an unaffected family member

Dataset Size: The FBN1 dataset comprises of 50,000 base pairs in the FASTA format with nearly 50,000 samples (see Figure 3.6), each sequence has the genetic code and known mutation locations, associated with Marfan Syndrome. The dataset was split into three subsets: 60% (approximately 30,000 samples) allocated for training, 20% (around 10,000 samples) for validation, and 20% (approximately 10,000 samples) for testing and detection of Marfan Syndrome.

Characteristics: Each sequence has the genetic code that represent mutation locations for identifying variations associated with Marfan Syndrome. The dataset contains data with respect to mutation types, mutation positions and associated phenotypic impacts.

The FASTA format DNA and protein sequence alignment of FBN1 gene is shown in Figure 3.6.

>NC_000015.10:c48645709-48408313 FBN1 [organism=Homo sapiens] [GeneID=2200] [chromosome=15]
AGAGACTGTGGGTGCCACAAGCGGACAGGAGCCACAG
CTGGGACAGCTGCGAGCGGAGCCGAGCAGTGGC
TGTAGCGGCCACGACTGGGAGCAGCCGCCGCCGCCTC
CTCGGGAGTCGGAGCCGCCGCTTCTCCAGTGGG
TGCAGCCGGGGTCCCGACGGGGGTCGGGCGGCCACCG
GGGCTGGAGCTGCGGCCACGGAGGCTTTTGCGT
TTGCGCCGCGCCGAGGGCAGGGACAGGGACTGGGGTG
AGGGGCTGTCCCGGAACGTCCACAGCTGGCGCT
GGCCCTCCCCTGCCTGACAGCTTCCTGGCCCGGGGCTC
TTGGTGCCGGGCTCCGCGTCAGATGTTCGGGG
GGCGGTGGCATCGCCCGGAGTCGGCGGGGACGGCGCG
GCTGGCTTCCAGCCTGGCGGAGAGGGCAGGCTG
AGGAGTGGGGCGTTCAGAGCGCGCATCGCGCGCAATT
CGTGCCGCTAAAAAAAATAAACCCAGAGAGCTC
GCCCGGGGCTTAGGACCGCTGGGGATATGGGTACTTTG
CGCCGCGCTCTTCTGGCGGGGCCCGGGAGGCC
GAGGGATCGGCCGGGGCTGCTGCCGCCGGGGGCCTGG
GCTTTCCAGCCAGCTGTGGACCAAACGGTCTTC
CCTTACCCAAATTAACTGCGCCACGCGCAGGCGGCGCA
CGGGTTGGGCTTGGGAATGGGGACCGCGAGGC

Figure 3.6 FASTA format of DNA sequence alignment file of FBN1

Preprocessing Steps: To ensure the quality of data, the following preprocessing steps were carried out. (1) Sequence Alignment: The DNA sequences were aligned to identify location and type of mutations with respect to Marfan Syndrome. (2) Noise Reduction:
Filtering techniques are applied to preserve biologically relevant mutations responsible for the disorder. (3) Normalization: The sequences were normalized to a standard scale reducing biological differences.

Even after preprocessing the data that is being used for the process of prediction and classification of monogenetic disorders seems to be crucial because of sparse data, uncertainty, overlapping features, rapid changes, physical structure, initial noise, non-linear characteristics, non-homogenous characteristics etc. To handle these issues, efficient and hybrid technology is essential to improve the performance in the data preprocessing, prediction and classification of monogenetic diseases.

The data associated with DNA sequences, which have vast feature space and comprise of redundant features with irrelevance information, lead to overfitting and affect the performance of diagnosis. Hence, it is essential to perform feature selection of genes that regulates and contributes to the target feature space relevant to diagnosis of genetic disorders.

B. Proposed Methodology

The proposed work presents two novel hybrid models to classify monogenetic disorders. This section narrates a comprehensive overview of the proposed methodology. Figure 3.7 depicts an abstract view of the proposed research work.

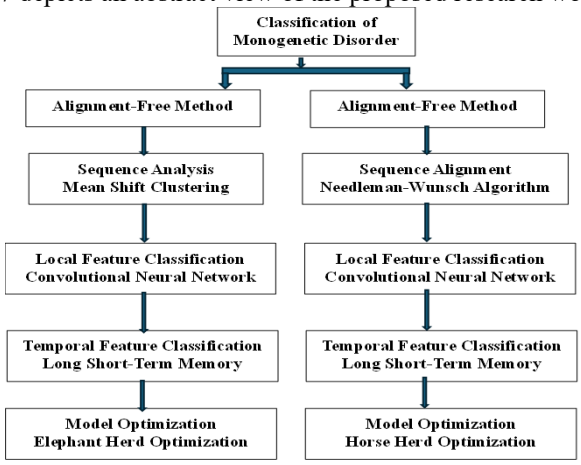

Figure 3.7 Architecture of Proposed Work of Monogenetic Disorder Classifications

i. Alignment-Free Model

The first model performs sequence analysis and clustering using a single novel algorithm, Mean Shift Clustering (MeShClust) algorithm, to generate the identity scores among the similar sequences and forms a cluster with similar sequences. The cluster of sequences is the input to the neural network models that integrates convolution neural network and recurrent neural network with long short-term memory to select relevant features that plays a vital role in identification of monogenetic disorders. The hyperparameters are tuned with optimization algorithm for better convergence. The sequences with dissimilarity are given as input for the CNN and then it selects features from those sequences and convolves with the kernel filters to extract local and relevant features. The Pooling layer of CNN reduces the dimension of features, and activation function produces the classified results. In order to find the dependencies among the long-term sequences, LSTM is used. Then the optimization algorithm, Elephant Herd Optimization, selects the best classified output from all available feasible solutions. This model is tested with the UBE3A gene dataset associated with Angelman Syndrome of chromosome 15.

ii. Alignment-Based Model

The second model of framework uses the global alignment algorithm, Needleman Wunsch algorithm, to find the highest score of sequences. These sequences are provided as input to the CNN and RNN model to generate the significant feature set that are responsible for the genetic mutation causing genetic

disorders. Forward propagation performs analysis on input sequences from the previous layer, transfers them to the output layer through the hidden layers and then transfers to the output layer with a final nonlinear activation function to generate the feature map signifying the muted genes that are relevant for classification. The Horse herd optimization identifies the best classification among the feasible solutions in short time.

Optimization Algorithms: The traditional optimization algorithms such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO), EHO and HHO are nature-inspired algorithms. EHO utilizes the social structure and movement patterns of elephant herds, with effective search in large spaces by overcoming local minima. Similarly, HHO models adapts the natural behavior of horses in herds, that performs both local refinement and global exploration efficiently. These characteristics make EHO and HHO particularly suited for the irregular and high-dimensional search space involved in tuning hyperparameters for DNA sequence analysis.

The above methodology is elaborated in the next session of the paper.

IV. PROPOSED METHODS OF ANALYSIS ON DNA SEQUENCES

The proposed methodology for identifying the mutation causing the monogenetic disorder is (i) Cluster-based method that focuses on sequence density alone for identifying the mutation, more specifically for detection of Angelman Syndrome. (ii) Alignment-based model that uses the Needleman-Wunsch algorithm for aligning sequences globally to identify the similarities that supports the diagnosis of Marfan syndrome. With these methods of framework, accurate detection of monogenetic disorders is made possible.

A. Alignment-free Methodology using MeShClust

The process of sequence analysis is essential as it finds the common evolutionary descendent and common structural functions. The Mean Shift Algorithm (MeShClust) performs sequence analysis and clusters the similar sequences. Traditional sequence alignment algorithms are slow in nature (Benjamin T James et al. 2018) and its greedy nature does not guarantee with optimal clusters (Chen 1995). However, MeShClust is flexible in producing the identity score.

The four letters of alphabet A, C, G, T form the basis to construct the genetic word of different lengths and q-nucleotide word or the short subsequence of length k is referred to as k-mer. k-mer is then built as a quaternary number of k digits and used to construct histograms. The pseudo count of k-mer in the histogram is initialized to 1 or 0. The MeShClust evaluates the value of k (Brian B Luczak et al. 2019) by taking the log4 of the average sequence length, and then by subtracting 1. The algorithm uses four features to estimate the similarity among the sequences: (i) The product of sequence length difference Equation (4.2) and Czekanowski similarity Equation (4.1), (ii) The product of length difference2 and Manhattan distance2 Equation (4.3), (iii) Pearson coefficient Equation (4.4) and (iv) Kulczynski coefficient Equation (4.5). In the Equations (4.1 to 4.5), A and B represent the histograms of two sequences and A_i and B_i are the i^{th} k-mer of A and B. $\overline{A}$ and $\overline{B}$ are the average counts of histograms A and B respectively. The four features are scaled between 0 and 1 and converted to the similarity measure. The overview of the MeShClust algorithm is depicted in Figure 4.1.

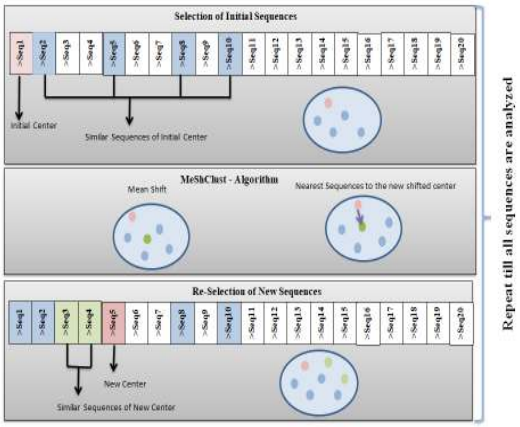

Figure 4.1 Overview of MeShClust algorithm

The selection of cluster center and sequence for the cluster depends upon the Czekanowski similarity index and it is an iterative process till all the sequences are analyzed.

The Czekanowski similarity index is measured as per equation (4.1) and is given as,

$$Csekanowsk\ \ i(A,B) = \sum_{i=0}^{N} \frac{\min(A_i, B_i)}{A_i + B_i} \qquad (4.1)$$

The sequence length difference is evaluated by equation (4.2) and is given as,

$$LD(A,B) = \left| length\ (A) - length\ (B) \right| \qquad (4.2)$$

The Manhattan distance is given by equation (4.3) and is,

$$Manhattan\ (A,B) \qquad = \sum_{i=0}^{N} \left| A_i - B_i \right| \qquad (4.3)$$

The Pearson coefficient is given by equation (4.4) and is,

$$Pearson\ (A,B) \frac{\sum_{i=0}^{N} (A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\sum_{i=0}^{N} (A_i - \overline{A})^2} \sqrt{\sum_{i=0}^{N} (B_i - \overline{B})^2}} \qquad (4.4)$$

The Kulczynski coefficient is given by Equation (4.5) and is

$$Kulczynski \quad (A,B) = \frac{4^k (\overline{A} + \overline{B})}{2((\overline{A}\,\overline{B})} \sum_{i=0}^{N} \min(A_i, B_i) \qquad (4.5)$$

Initially the input sequences are sorted in increasing order of their length. The shortest sequence forms the center of initial cluster. Then MeShClust is implemented on the sequences of the current cluster to calculate and update its center. This update determines the sequence closest to the updated center. This process is iterated to determine of new centers to form a cluster with similar sequences. These iterations stop when no more sequences are left out. The algorithm, MeShClust for creating the clusters and generating the optimal cluster with its center point is shown in Algorithm 4.1 and Algorithm 4.2.

(4.6)

$$k = \left\lceil \log_4 \left(\frac{1}{n} \sum_{i \in S} len(i) \right) \right\rceil - 1$$

Algorithm 4.1 MeShClust algorithm for generating clusters of similar sequences

Algorithm: MeShClust
Input: A set of DNA sequences S ={s_1,s_2, …, s_n} sorted in decreasing order of its length
Output: Clusters of sequences with its center
Step1: Use Czekanowski similarity index (Equation 4.1) to find the identity score, if it is > 60%, use subset of sequences to find similar sequences
    $center_{cur} = s_1$
    $cluster_{cur} = \{s_1\}$
Step2: While S is not empty do
        G = all sequences from S close to $center_{cur}$
        S = S – G
        If G is not empty then
            $center_{cur} = center_{cur} \cup G$
            Use MeShClust-CenterUpdate algorithm to update $center_{cur}$
        Else
            Add $center_{cur}$ to Centers
            Add $cluster_{cur}$ to Clusters
            $center_{cur}$ = the closest sequence in S to the old $center_{cur}$ according to Czekanowski similarity index (Equation 4.1)
            $cluster_{cur} = \{center_{cur}\}$
        End If
    End While
Step3: For i = 1 to 15 do
        For j in all $center_j \in$ Centers, $cluster_j \in$ Clusters do
            Use MeShClust-CenterUpdate algorithm with $cluster_j$, $center_j$, neighboring clusters
        End For
        For j = 1 to {Centers} do
            For k in all $center_k \in$ Centers close to $center_j$ do
                Merge centers $center_j$ and $center_k$ if they are similar sequences
            End For
        End For
    End For

Algorithm 4.2 MeShClust algorithm for updating center point of clusters

Algorithm: MeShClust-CenterUpdate
Input: The current center, $center_{cur}$ of a cluster and a set of points, X
Output: The closest point in X to the new center
Step1: Calculate the new center, $center_{new}$ using the current center, $center_{cur}$ according to Equation 4.2
Step2: The closest point, $p \in X$ to $center_{new}$ is found using Czekanowski similarity index (Equation 4.1), return p as the new center

The MeShClust algorithm does not require any pre-specified number of clusters. It is non-parametric, it is robust to noise and outliers, and is dependent on the density of the data. MeShClust deals with feature space derived from k-mers and their data density. This algorithm is an alignment-free technique that uses distribution of DNA sequences in high-dimensional feature space. This algorithm handles the sequences containing gaps, sequences of variable length, and sequences of complex patterns.

The resultant optimal cluster from MeShClust algorithm can be further used to identify recurrent sequence motifs that may be the regulatory elements, muted motifs that are the traits of genetic diseases.

The success of MeShClust depends on the selection of the value of k as given in Equation (4.6) containing n: number of sequences in the set S.

Training deep convolution neural network from scratch is a challenging task as it leads to overfitting resulting in poor memory and computational resources. Data augmentation such as rotation, scaling or transformation has the ability to synthesis the data and such data has to be transferred to the input layer of CNN to avoid overfitting.

The convolution layer performs the convolution operation among the array of features referred to as kernel and input array that result in a feature map. This feature map contains the local features of the DNA sequences. At this stage, the activation function, either tangent (tanh) or Rectified Linear Unit (ReLU) evaluates the function's non-linearity. Nearly 2 to 5 CNN layers are considered.

The next layer is the Pooling layer that reduces the dimension of the input layer. The max, min or average pooling can be used for dimensionality reduction.

The fully connected layer connects every local input from the previous layer to every output resulting in a single dimensional vector consisting of probability of each feature belonging to a class. It contains learnable weights using the softmax or dropout function that maps input to the desired output.

Recurrent Neural Network learns the temporal features while Convolution Neural Network learns the local features of DNA sequences.

The parameters associated with RNN are input size, batch size and time step. There exist two LSTM layers in every memory block with two hidden layers. The time-dependent input sequences are obtained via these LSTM layers. The depth of LSTM layers or blocks again is 2 to 5 layers.

Optimization changes parameters in each node as per the gradient descent method and reiterates the process from convolution to the output layer. Optimization tunes the hyperparameters and makes efficient classification of monogenetic disorders such as Angelman Syndrome. Figure 4.2 depicts the overall architecture of CNN-RNN with LSTM model gets optimized with EHO algorithm.

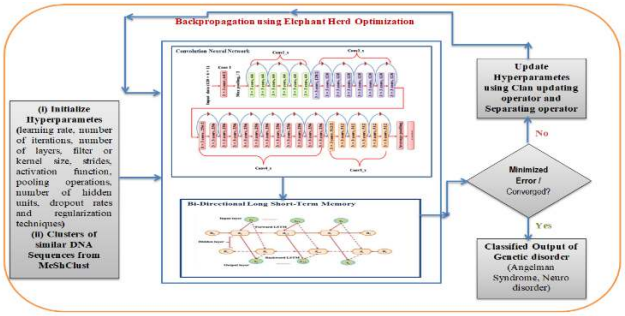

Figure 4.2 Overview of CNN-RNN with LSTM and EHO

The data associated with DNA sequences, which have vast feature space and comprise of redundant features with irrelevance information, lead to overfitting and affect the performance of diagnosis. Hence, it is essential to perform feature selection of genes that regulates and contributes to the

target feature space relevant to diagnosis of genetic disorders. This step of feature selection is automated in CNN.

This model takes the clusters from MeShClust algorithm, and initialized hyperparameters as input and proceeds with CNN-RNN LSTM model to generate the classified output of Angelman Syndrome.

The initial assumptions of Elephant Herd optimization algorithm are (i) Elephant population is a group of elephants, d (ii) Matriarch is the leader (iii) Elephants live together in a group as per Matriarch's instructions (iv) At each generation, some elephants leave the group. Then the two operators, group updating operator and selection operator plays a vital role.

Elephants change their position with regard to the Matriarch position ($A^{p}_{ebest,c}$) using the Equation (4.7) and is referred to as a clan updating operator.

$$A^{p+1}_{n,c} = A^{p}_{n,c} + \beta X (A^{p}_{ebest,c} - A^{p}_{n,c}) Xr \qquad (4.7)$$

Here 'n' is the total number of elephants in a group, 'c' is the total number of groups, β is the scaling factor [0,1], r is the uniformly distributed random number [0,1], and $A^{p}_{n,c}$ is the current position of the elephant considered.

The Matriarch in each clan is updated as per the equation (4.7) and Equation (4.8), where α is the scaling factor with a range [0, 1], $A^{p}_{cen,c}$ is the center position of the clan in Equation (4.9) and $A_{n,c}$ is the position of individual elephants in the clan and F is the total number of elephants in the clan.

$$A^{p+1}_{ebest,c} = A^{p}_{cen,c} X \alpha \qquad (4.8)$$

$$A^{p}_{cen,c} = \frac{1}{F} X \sum_{n=1}^{F} A_{n,c} \qquad (4.9)$$

The separating operator is used to attain the worst fitness value of clan c, and is given by the Equation (4.10) where low_b and up_b is the lower bound and upper bound of elephants position and r is the random variable that is uniformly distributed over [0,1]. (4.10)

$$A_{eworst,c} = low_b + (up_b - low_b + 1) Xr$$

The algorithm of hyperparameters tuning using Elephant Herd Optimization algorithm is shown in Algorithm 4.3. The fitness function is given by Equation (4.11), where α is a constant factor that scales the term ρT(C), μ is also the constant term, M is the total number of features and T is the selected number of features.

$$Fitness = \alpha \rho T(c) + \mu \frac{|T|}{|M|} \qquad (4.11)$$

The convolution operation is mathematically represented in equation (4.12) and it uses sigmoid function on the weight factor $W^{l(L)}_{KJ}$ on the bias function $b^{l(L)}_{J}$ on the input sequence N^{L-1}_{K} and on Kth and Jth Node of convolution layer.

$$M^{L}_{J} = Sig(N^{L-1}_{K} * W^{l(L)}_{KJ} + b^{l(L)}_{J}) \qquad (4.12)$$

The activation function called softmax function decides the target class of probability using exponentiation (ex) with I, the total no of clans and is given in Equation (4.13).

$$\text{Probability}(M^{L}_{J}) = \frac{ex(M^{L}_{J})}{\sum_{j=1}^{I} ex(M^{L}_{J})} \qquad (4.13)$$

Algorithm 4.3 EHO for hyperparameter tuning in CNN-RNN with LSTM

Algorithm: EHO
Input: The population with hyperparameters such as learning rate, number of iterations, number of layers, filter or kernel size, strides, activation function, pooling operations, number of hidden units, dropout rates and regularization techniques
Output: The best set of hyperparameters
Step1: Repeat the steps 2 to steps 5 till the error gets minimized.
Step2: For i = 1 to no. of clans do
Step3: For j = 1 to no. of elephants in the $clan_i$ do
Step4: Using **update operator** as per Equation 4.6 updates $A^{p+1}_{n,c}$
Step5: If the current position is the best
Replace it as updated position as per Equation 4.7 and 4.8
Step6: End If
Step7: End For
Step8: End For
Step9: For i = 1 to no. of clans do
Step10: Using **Separating Operator**, Replace the worst elephant in $clan_i$ as per Equation 4.9
Step11: End For
Step12: Evaluate the performance according to the new parameters
Step13: End Repeat

The forward LSTM is denoted by $R_t, t \in [1, L]$ and the backward LSTM is $R_t^{'}, t \in [L, 1]$ where L is the time-series length. The hidden layer includes the forward state $\overrightarrow{d_t}$ given by Equation (4.14) and the backward state $\overleftarrow{d_t}$ specified in Equation (4.15).

$$\overrightarrow{d_t} = \overrightarrow{LSTM}_{(d_{t-1}, y_t, s_{t-1}), t \in [1,L]} \qquad (4.14)$$

$$\overleftarrow{d_t} = \overleftarrow{LSTM}_{(d_{t-1}, y_t, s_{t-1}), t \in [1,L]} \qquad (4.15)$$

The above Equations (4.14) and (4.15) consider the input, y_t, as the state of cells, s_t and data from hidden layers, d_t.

B. Alignment-based Methodology using Needleman Wunsch Algorithm

The main objective of sequence alignment is to find similar sequences in the homologous gene sequences and return the longest subsequence with highest score. Let S = {S1, S2, …, Sm} and T = {T1, T2, … Tn} are the two sequences, d is the gap penalty cost, s(x, y) is the score of aligning a base x from S and a base y from T and F is matrix where F (x, y) refers to the xth place in S and the yth place in T.

The Needleman Wunsch Algorithm works as follows,

(i) A scoring function, σ has to be defined (+1 for match and -1 for mismatch) for the DNA sequence.
(ii) A gap penalty, d for insertion or deletion for the bases in the sequence.
(iii) Initialize F matrix as F(0,0)=0, F(i,0)=F(i-1,0)-d, F(0,j)=F(0,j-1)-d.
(iv) Fill up the matrix, F recurrently using the below equation (4.16).

$$F(i,j) = \max\{\{F(i-1,j-1) + \sigma(Si,Ti)\}, \{F(i-1,j) - d\}, \{F(i,j-1) - d\}\}$$

(v) Traceback the above matrix from the bottom right F(m,n) to the top left F(0,0) in order to find the best alignment.

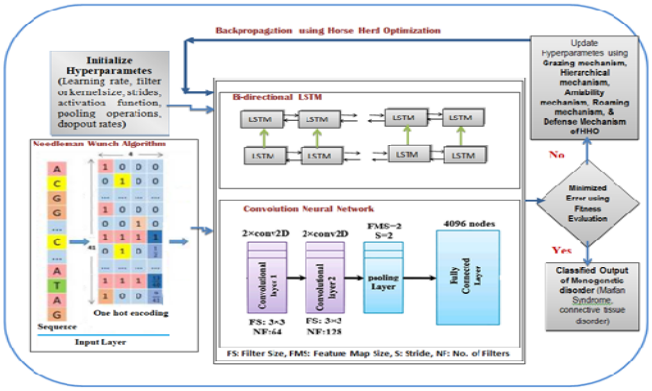

Figure 4.3 Overview of CNN-RNN with LSTM and HHO

The above figure 4.3 shows the overview of alignment-based deep neural network model for sequence analysis to predict monogenetic disorders. The Bi-LSTM of Recurrent Neural Network trains the input sequences in both forward and backward directions to learn the temporal features. The three gates- update, forget and output gate of Bi-LSTM are represented mathematically in Equations (4.17), (4.18) and (4.19) which holds the current state sequence, $x^{<t>}$, bias, b_u, activation output, $a^{<t-1>}$ and weight function, W_u. Bi-LSTM generates two output cells, the activation value, $a^{<t-1>}$ and the candidate value, $c^{<t-1>}$ which is specified in Equation (4.20) and (4.21) using the three gates.

$$updategate \quad ,\tau_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \quad (4.17)$$

$$forgetgate \quad ,\tau_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \quad (4.18)$$

$$outputgate \quad ,\tau_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_0) \quad (4.19)$$

$$activation \quad ,a^{<t>} = \tau_o * c^{<t>}) \quad (4.20)$$

$$candidate, c^{<t>} = \tau_u * c^{N<t>} + \tau_f * c^{<t-1>}) \quad (4.21)$$

The Convolution Neural Network is used to extract the local features from the aligned DNA sequences. The convolution layer uses convolution operation that uses ReLU (Rectified Linear Unit) function on the input sequence, F_K^{L-1} and kernel filter with weight factor, $W_{KJ}^{1(L)}$ and bias function, $b_J^{1(L)}$ is defined in equation (4.22).

$$(M_J^L) = \text{ReLU}(F_k^{L-1} X(W_{KJ}^{1(L)}) + (b_J^L) \quad (4.22)$$

The ReLU function is given in equation (4.23)

$$\text{ReLU}(x) = x, \text{ if } x \geq 0 \text{ , ReLU}(x) = 0, \text{ if } x < 0 \quad (4.23)$$

The max pooling function is implemented in the pooling layer of CNN as specified in equation (4.24).

$$pooling\ (x) = \max(x1, x2, ..., xn) \quad (4.24)$$

Hence, the fully connected layer extracts the significant features of the input DNA sequences that signify the mutant gene responsible for monogenetic disease.

The Horse Herd Optimization, a nature-inspired algorithm, is implemented as the behaviour and motion of a horse in the search space of identifying the best hyperparameters is necessary to establish the neural network architecture using CNN and RNN. The hyperparameters such as learning rate, filter or kernel size, strides, activation function, pooling operations, and dropout rates are associated with the different ages and various mechanisms of horses to select the optimal values as per HHO algorithm.

The velocity of horses' motion in each of the iteration (itr) according to their ages (α, β, γ *and* δ) is represented in equation (4.25), (4.26), (4.27), and (4.28) that are dependent on velocity of grazing ($\overrightarrow{G}_m^{itr,age}$), defense ($\overrightarrow{D}_m^{itr,\alpha}$), hierarchy ($\overrightarrow{H}_m^{itr,\beta}$), sociability ($\overrightarrow{S}_m^{itr,\beta}$), imitation ($\overrightarrow{I}_m^{itr,\delta}$) and roaming ($\overrightarrow{R}_m^{itr,\delta}$) mechanisms.

$$\overrightarrow{V}_m^{itr,\alpha} = \overrightarrow{G}_m^{itr,\alpha} + \overrightarrow{D}_m^{itr,\alpha} \quad (4.25)$$

$$\overrightarrow{V}_m^{itr,\beta} = \overrightarrow{G}_m^{itr,\beta} + \overrightarrow{H}_m^{itr,\beta} + \overrightarrow{S}_m^{itr,\beta} + \overrightarrow{D}_m^{itr,\beta} \quad (4.26)$$

$$\overrightarrow{V}_m^{itr,\gamma} = \overrightarrow{G}_m^{itr,\gamma} + \overrightarrow{H}_m^{itr,\gamma} + \overrightarrow{S}_m^{itr,\gamma} + \overrightarrow{I}_m^{itr,\gamma} + \overrightarrow{D}_m^{itr,\gamma} + \overrightarrow{R}_m^{itr,\gamma} \quad (4.27)$$

$$\overrightarrow{V}_m^{itr,\delta} = \overrightarrow{G}_m^{itr,\delta} + \overrightarrow{I}_m^{itr,\delta} + \overrightarrow{R}_m^{itr,\delta} \quad (4.28)$$

The grazing mechanism is followed at all ages (α, β, γ and δ) and is represented in Equation (4.29) that uses the previous position grazing represented in Equation (4.30).

$$\overrightarrow{G}_m^{itr,age} = \overrightarrow{g}_m^{itr,age}(\overline{up} - \rho\overline{lo})[\overrightarrow{x}_m^{itr,age}] \quad (4.29)$$

$$g_m^{itr,age} = g_m^{itr-1,age} Xw_g \quad (4.30)$$

The Convolution Neural Network is used to extr

The hierarchy mechanism is followed at ages α, β, and γ by strong horses and is represented in Equation (4.31) that uses the previous position of previous stronger horses represented in Equation (4.32).

$$\overrightarrow{H}_m^{itr,age} = \overrightarrow{h}_m^{itr-1,age}[\overrightarrow{x}_* - \overrightarrow{x}_m^{itr-1,age}] \quad (4.31)$$

$$h_m^{itr,age} = h_m^{itr-1,age} Xw_h \quad (4.32)$$

The sociability mechanism is followed at ages β and γ by the horses and is represented in Equation (4.33) that uses the previous position biased by the weight factor, w_s represented in Equation (4.34).

$$\overrightarrow{S}_m^{itr,age} = \overrightarrow{s}_m^{itr-1,age}[\frac{1}{N}\sum_{j=1}^{N} x_j^{itr-1} - x_m^{itr-1}] \quad (4.33)$$

$$s_m^{itr,age} = s_m^{itr-1,age} Xw_s \quad (4.34)$$

The imitation mechanism is followed at age, γ by the young horses and is represented in Equation (4.35) that uses the previous position biased by the weight factor, wi represented in Equation (4.36).

$$\overrightarrow{I}_m^{itr,age} = \overrightarrow{i}_m^{itr-1,age}\left[\frac{1}{\rho N}\sum_{j=1}^{\rho N} x_j^{itr-1} - x_m^{itr-1}\right] \quad (4.35)$$

$$i_m^{itr,age} = i_m^{itr-1,age}\, Xw_i \quad (4.36)$$

The defense mechanism is followed at ages α, β, and γ by the horses during fights and is represented in Equation (4.37) that uses the previous position biased by the weight factor, wd represented in Equation (4.38).

$$\overrightarrow{D}_m^{itr,age} = \overrightarrow{d}_m^{itr-1,age}\left[\frac{1}{qN}\sum_{j=1}^{qN} x_j^{itr-1} - x_m^{itr-1}\right] \quad (4.37)$$

$$d_m^{itr,age} = d_m^{itr-1,age}\, Xw_d \quad (4.38)$$

The roaming mechanism is followed at ages δ and γ by the horses and is represented in Equation (4.39) that uses the previous position biased by the weight factor, wd represented in Equation (4.40).

$$\overrightarrow{R}_m^{itr,age} = r_m^{itr-1,age}\, \rho x_m^{itr-1} \quad (4.39)$$

$$r_m^{itr,age} = r_m^{itr-1,age}\, Xw_r \quad (4.40)$$

Thus, the implementation of the above mechanisms of horses in the search space of hyperparameters enhances the search and leads to optimal value thereby improving the performance of the entire framework.

The optimization algorithms decide the choice of hyperparameters such as activation function, dropout rate, pooling function to avoid the convergence problems, overfitting issues, etc. CNN automates feature selection thereby overcome overfitting and down-samples the data thus reducing the dimensions. The training phase of CNN uses Random rotations, flips or shifts are applied as data augmentation technique in order to increase the quantity of training data to prevent overfitting and increase robustness of the model. Thus, CNN models are used to discover the regulatory variants that play a causative role in the increase of risk in genetic disorders crossing underlying issues associated with gradient and overfitting problems.

C. Biological Interpretability Analysis

To improve the interpretability of the CNN-BiLSTM model for DNA sequence analysis, we incorporated a saliency mapping technique. Saliency mapping highlights the most influential regions of input sequences that contribute to the model's predictions. This was achieved by calculating the gradients of the output with respect to the input sequences, identifying which nucleotides or motifs significantly impact the predicted outcomes.

Grad-CAM (Gradient-weighted Class Activation Mapping) is used to generate heatmaps and thereby saliency mapping presents the most significant regions of DNA sequences responsible for genetic disorders. Biological validation was performed by comparing the highlighted regions with known pathogenic loci or motifs associated with Angelman and Marfan syndromes.

The motifs identified in the UBE3A gene associated with Angelman Syndrome comprises of deletions in exons 3-9, splice site mutations in the intron 2-3 region and variants in the 3' untranslated region (UTR).

On the other hand, the motifs in FBN1 gene responsible for Marfan Syndrome include mutations in exons 24-32, splice site mutations in the intron 24-25 region, variants in the 5' untranslated region (UTR) and EGF-like motifs in exons 10-15.

V. RESULTS AND DISCUSSIONS

The proposed models of alignment-free framework is evaluated using UBE3A dataset and alignment-based framework on FBN1 dataset to identify the mutant gene responsible for monogenetic disorders like Angelman syndrome and Marfan syndrome respectively with the below given hardware setup (Table 5.1).

Table 5.1 Hardware Setup of the Proposed Model

Aspect	**Alignment-Free Model (UBE3A Dataset)**	**Alignment-Based Model (FBN1 Dataset)**
Hardware Setup	NVIDIA Tesla V100 GPU (32 GB), Intel Xeon CPU	NVIDIA Tesla V100 GPU (32 GB), Intel Xeon CPU
Training Time (per epoch)	Approx. 5 hours	Approx. 7 hours
Memory Consumption	~10 GB GPU memory	~15 GB GPU memory
Scalability	Efficient on medium-sized datasets (up to 100,000 samples)	Handles medium datasets well; large datasets (>1 million samples) may require distributed training and memory optimizations

To scale the proposed methodology for larger datasets, we suggest employing distributed computing techniques, such as parallelizing the alignment-free model across multiple CPU cores or utilizing GPU acceleration for the alignment-based model. Additionally, model compression techniques, pruning is applied to reduce the memory footprint and computational requirements of the models.

The saliency maps and feature importance graphs for the UBE3A and FBN1 genes provide valuable insights into the key regions and features contributing to the diagnosis of monogenetic disorders. The saliency maps (Figure 5.1) highlight the importance of specific nucleotide positions in the UBE3A and FBN1 gene sequences, respectively. The feature importance graphs (Figure 5.2 (a) and (b)) reveal the significance of

different feature types, such as exons, introns, and splice sites, in the UBE3A and FBN1 genes. These visualizations demonstrate that certain regions and features are more crucial than others in determining the genetic basis of monogenetic disorders.

Saliency Map presents Heatmap with nucleotide positions on x-axis and importance score on y-axis. Red indicates high importance, blue indicates low importance. Bar chart with feature types (exon, intron, splice site) on x-axis and importance score on y-axis. Red indicates high importance, blue indicates low importance.

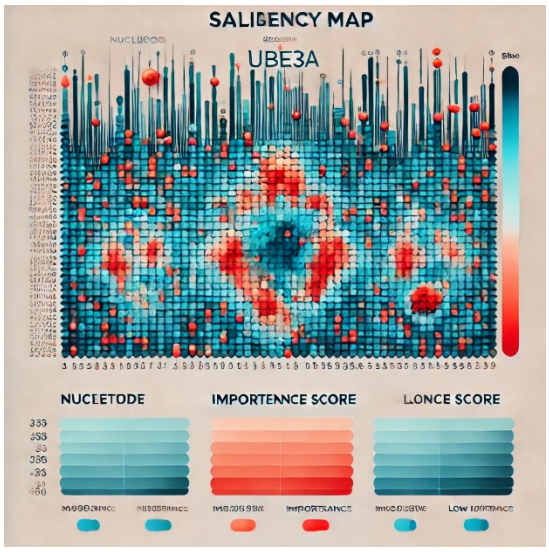

Figure 5.1 Saliency Map of UBE3A

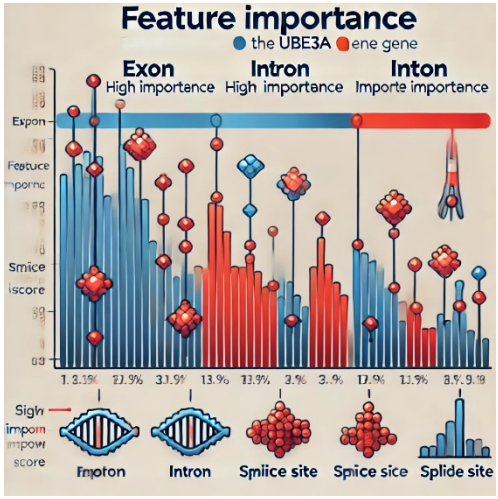

Figure 5.2 (a) Feature Importance Graph (UBE3A Gene)

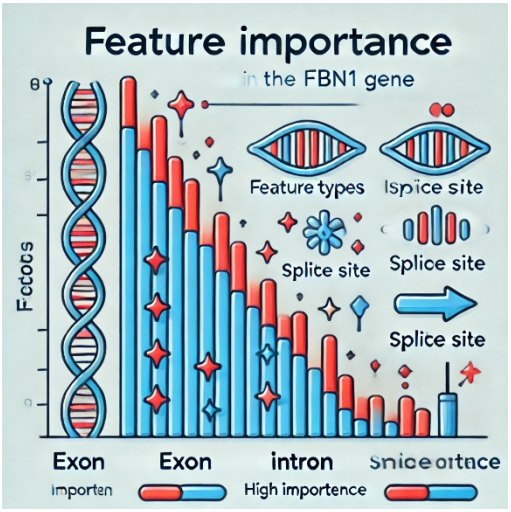

Figure 5.2 (b) Feature Importance Graph of FBN1 Gene

To validate the choice of Elephant Herd Optimization (EHO) and Horse Herd Optimization (HHO) for hyperparameter tuning, a comparative analysis is conducted with Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). The experiments focused on convergence rate, computational time, and result stability using the UBE3A and FBN1 datasets. As summarized in Table 5.2 (a), EHO and HHO demonstrated faster convergence and lower computational overhead compared to GA and PSO, particularly in complex parameter spaces and depicted in the figure 5.2 (c). Additionally, both algorithms exhibited higher stability across multiple runs, suggesting better robustness in optimizing deep learning models for monogenetic disorder classification. These findings reinforce the suitability of EHO and HHO for the proposed work.

Table 5.2 (a) Comparative Analysis of Optimization techniques

Algorithm	Convergence Rate (iterations)	Computational Time (seconds)	Accuracy (%)	Stability (Std. Dev.)
EHO	150	45	92.5	0.8
HHO	140	43	91.8	0.7
GA	180	65	89.2	1.5
PSO	170	60	90.1	1.3

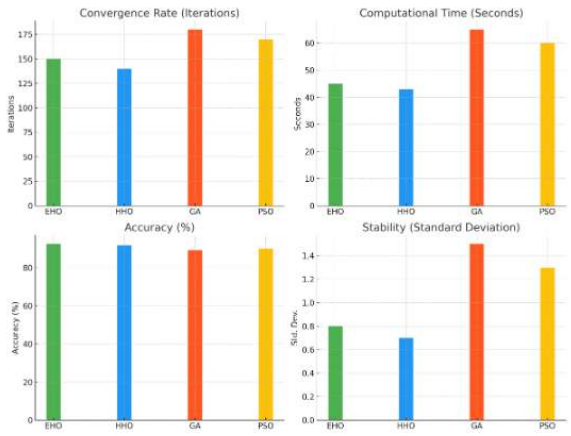

Figure 5.2 (c) Comparative analysis of Optimization techniques

The performance metrics used to evaluate the model are accuracy, mean absolute error (MAE) and mean squared error (MSE) for alignment-free model and alignment-based model is shown in Table 5.2(b).

Accuracy is the metric used to assess the correctness of the classification or prediction and the corresponding formula is given as :

Accuracy=(no. of correct predictions / Total no. of predictions) × 100

However, all nuances of prediction errors should also be considered. Hence, Mean Absolute Error (MAE) and Mean Squared Error (MSE) quantifies the average magnitude of errors in predictions without considering their direction. It is calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where the difference is computed between actual values and predicted values. MSE is computed using the below given formula between predicted and actual values and it signifies more weight to larger errors.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

To evaluate the models' performance few more metrics are included and are precision, recall, F1 score and ROC-AUC as consideration of clinical impacts on false positives and false negatives play a vital role.

The Precision is evaluated as a proportion of true positive predictions among all positive predictions. Higher precision is essential for reducing misdiagnoses, which is especially needed for identifying mutations linked to disorders like Angelman and Marfan syndromes. Its formula is:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

The Recall or sensitivity is computed as a proportion of true positives among all actual positives and is given by:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

The F1-Score is the harmonic mean of precision and recall, as it considers both false positives and false negatives.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The ROC-AUC is a measure that differentiates positive and negative classes and ranges from 0 to 1.

The alignment-free model on the UBE3A dataset has produced (0.81, 0.78,0.79, 0.86) as (precision, recall, F1-score, ROC-AUC) respectively and for the alignment-based model, it is (0.89, 0.91, 0.90, 0.93) on the FBN1 dataset. Below figure 5.3 represents the evaluation metrics of proposed models against existing models.

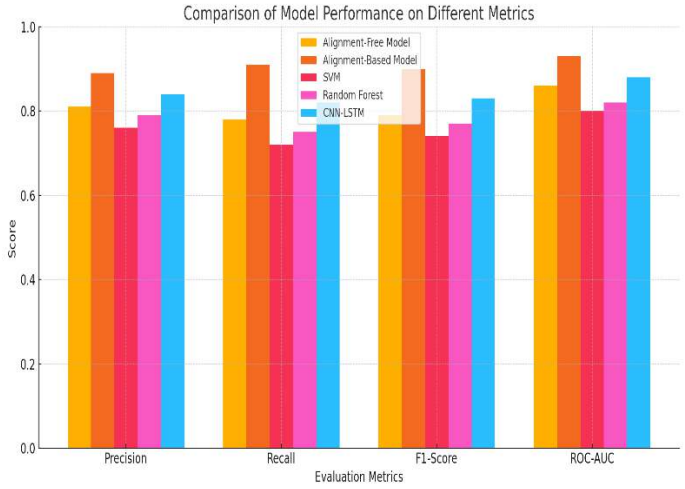

Figure 5.3 Performance analysis in terms of Precision, recall, f1-score and ROC-AUC

Table 5.2 (b) Performance Analysis of the Proposed Models with respect to accuracy and loss

Epoch	Alignment-free Model			Alignment-based Model		
	Accuracy (%)	MAE	MSE	Accuracy (%)	MAE	MSE
1	77.4	0.047	0.003	83.2	0.095	0.114
2	77.2	0.046	0.003	85.1	0.109	0.130
3	77.0	0.045	0.003	85.3	0.103	0.123
4	77.3	0.044	0.002	83.5	0.119	0.143
5	77.1	0.043	0.002	83.7	0.091	0.109
6	77.5	0.042	0.002	85.5	0.106	0.126
7	77.3	0.041	0.002	85.7	0.099	0.119
8	77.2	0.040	0.002	84.1	0.114	0.136
9	77.0	0.039	0.002	87.3	0.087	0.104
10	77.5	0.038	0.001	89.2	0.087	0.104

The above analysis shows that the process of identification of mutant gene in a DNA sequence that is responsible for monogenetic disorder especially on Chromosome 15 is more accurate (89.2%) when alignment-based technique of Needleman Wunsch algorithm combined with deep neural network model of CNN and Bi-LSTM. Since the model is optimized through the high memory power Horse Herd Optimization algorithm.

The below table (Table 5.3) shows the significance of hyperparameter tuning associated with learning rate, drop outs, activation function etc. using Elephant Herd Optimization and Horse Herd Optimization algorithms on UBE3A and FBN1 dataset in monogenetic disorder diagnosis framework.

Table 5.3 Hyperparameter tuning using EHO and HHO

Hyperparameter	Options	Optimal (HHO)	Optimal (EHO)
Activation Function	ReLU, Sigmoid, Tanh	ReLU	Sigmoid
Dropout Rate	0.1, 0.01, 0.001	0.01	0.1
Pooling Type	Max, Min, Average	Max	Average
Learning Rate	0.1, 0.01, 0.001	0.001	0.01
Batch Size	16, 32, 64	32	16
Optimizer	Adam, SGD, RMSprop	Adam	SGD
Number of Layers	2, 3, 4	3	2
Units per Layer	64, 128, 256	128	64
Regularization (L2)	0.01, 0.001, 0.0001	0.001	0.0001
Epochs	50, 100, 150	100	50

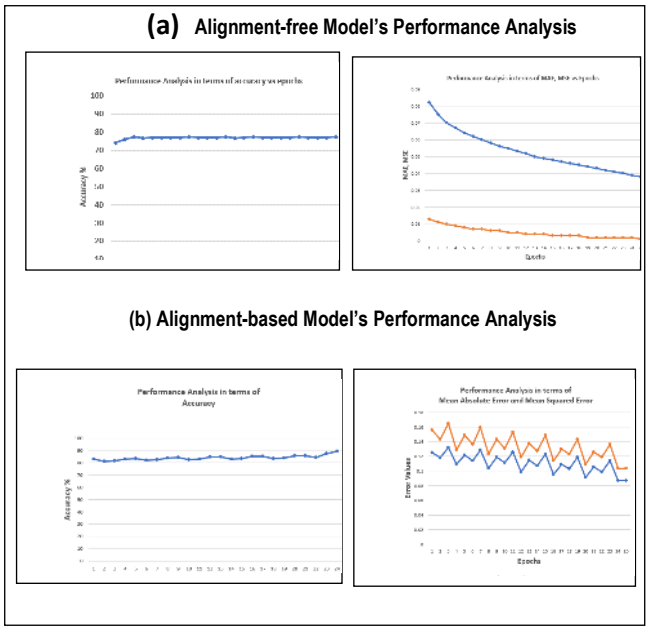

Figure 5.2 Overall Performance in terms of accuracy and error of proposed models

The sensitivity analysis of Horse Herd Optimization algorithm on the hybrid deep learning framework comprising of convolutional neural network and bi-directional long-short term memory model is shown in figure 5.3.

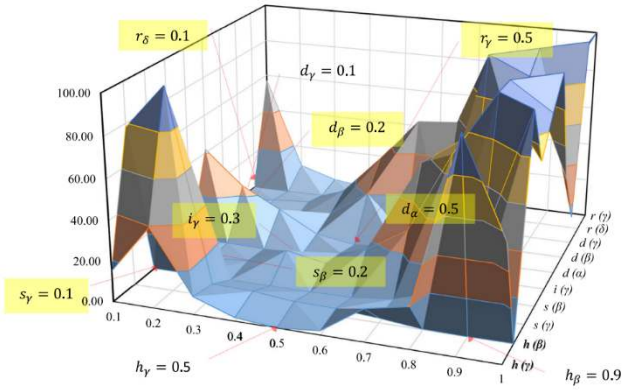

Figure 5.3 Sensitivity Analysis of Hyperparameters of CNN-BiLSTM using HHO Algorithm

The work does not gets completed if not analyzed with other existing models. Hence, the comparison is done with other machine learning models on the same dataset and the following Table 5.2 shows the result.

Table 5.2 Comparative Analysis on Monogenetic disorder analysis

Model	Accuracy (%)	MAE	MSE
Support Vector Machine	75.9	0.155	0.191
Random Forest	**78.2**	**0.142**	**0.173**
CNN	80.8	0.129	0.157
CNN-LSTM	84.2	0.102	0.123
CNN-Bi-LSTM (Proposed)	89.2	0.087	0.104

The proposed model, CNN-Bi-LSTM has the highest accuracy and the lowest MAE and MSE as shown in figure 5.4 indicating it outperforms the other models in this comparison. The Support Vector Machine model has the lowest performance among the models listed.

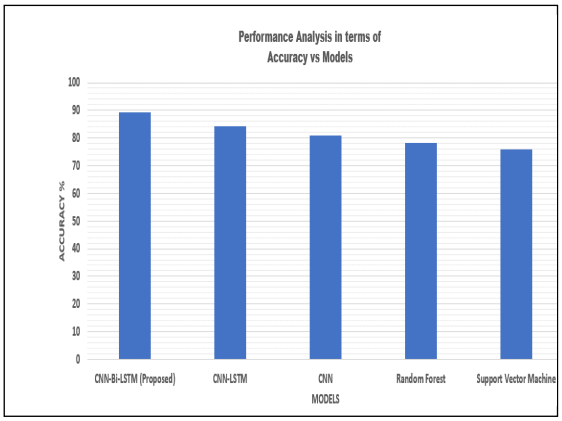

Figure 5.4 Comparative Analysis with other Machine Learning Models

The comparative analysis against the benchmark DNA sequence analysis (BLAST) is presented in the below table.

Model	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**
Standard CNN	80.8	0.82	0.79	0.80	0.83
RNN	81.5	0.83	0.81	0.82	0.85
BLAST (Alignment Technique)	75.3	0.78	0.76	0.77	0.80
CNN-Bi-LSTM (Proposed)	**89.2**	**0.89**	**0.91**	**0.90**	**0.93**

The below given table (Table 5.3) provides the metrics associated with training, validation and testing phases carried out using the proposed methodology from an external dataset of monogenetic disorders, dbGaP (Database of Genotypes and Phenotypes) providing a diverse genetic profile for testing.

Table 5.3 Training, Validation and Testing Performance metrics

Metric	**Dataset**	**Training**	**Validation**	**Testing**
Accuracy	UBE3A	77.5%	76.3%	76.0%
	FBN1	89.2%	88.5%	88.0%
	dbGaP	72.7%	73%	73%
Precision	UBE3A	0.81	0.79	0.78
	FBN1	0.89	0.88	0.87
	dbGaP	0.77	0.73	0.77
Recall	UBE3A	0.78	0.77	0.76
	FBN1	0.91	0.90	0.89
	dbGaP	0.72	0.75	0.73
F1-Score	UBE3A	0.79	0.78	0.77
	FBN1	0.90	0.89	0.88
	dbGaP	0.742	0.775	0.673
ROC-AUC	UBE3A	0.86	0.85	0.84
	FBN1	0.93	0.92	0.91
	dbGaP	0.72	0.75	0.73
MAE	UBE3A	0.038-0.047	0.040-0.050	0.042-0.052
	FBN1	0.087-0.119	0.089-0.121	0.091-0.123
	dbGaP	0.032	0.025	0.007
MSE	UBE3A	0.001-0.003	0.0015-0.0035	0.0018-0.004
	FBN1	0.104-0.143	0.106-0.145	0.109-0.148
	dbGaP	0.002	0.0015	0.0013

The Optimization algorithms incorporated in the proposed methodology had improved the convergence rate and the results are shown in the below given table (Table 5.4)

Table 5.4 Performance of EHO and HHO against traditional optimization techniques

Algorithm	**Convergence Speed**	**Exploration & Exploitation Balance**	**Avoidance of Local Minima**	**Accuracy (%)**	**Precision**	**Recall**	**F1-Score**	**MAE**	**MSE**
Genetic Algorithm (GA)	Moderate	Moderate	Moderate	75.9	0.70	0.68	0.69	0.155	0.191
Particle Swarm Optimization (PSO)	High	Low	Low	78.2	0.73	0.71	0.72	0.142	0.173
Elephant Herd Optimization (EHO)	High	High	High	80.8	0.81	0.79	0.80	0.129	0.157
Horse Herd Optimization (HHO)	Very High	Very High	Very High	89.2	0.89	0.91	0.90	0.087	0.104

The above data shows the improved performance rate of the entire framework. Additionally, to error analysis was conducted on the proposed framework by assessing the false negatives and false positives of DNA sequences. The below table 5.5 represents the misclassifications in clinical contexts that makes the detection of mutation difficult.

Table 5.5 Error Analysis for DNA Sequence Classification

Dataset	**Metric**	**Training (%)**	**Validation (%)**	**Testing (%)**
UBE3A	False Positives	5.1	5.3	5.8
	False Negatives	4.2	4.5	4.9
	Precision	0.81	0.79	0.78
	Recall	0.78	0.77	0.76
	F1-Score	0.79	0.78	0.77
FBN1	False Positives	4.0	4.2	4.5
	False Negatives	3.5	3.7	3.9
	Precision	0.89	0.88	0.87
	Recall	0.91	0.90	0.89

Dataset	Metric	Training (%)	Validation (%)	Testing (%)
	F1-Score	0.90	0.89	0.88

To evaluate the robustness of the proposed model, external dataset (HGMD-Human Gene Mutation Database and ClinVar) other than UBE3A and FBN1 dataset is also considered for testing. Table 5.6 shows the performance metrics associated with external datasets.

Table 5.6 Performance analysis on external datasets

Metric	Dataset	Internal Test Set	External Dataset (HGMD-Human Gene Mutation Database))	External Dataset (ClinVar)
Accuracy	UBE3A	77.5%	75.8%	76.2%
	FBN1	89.2%	87.5%	88.0%
Precision	UBE3A	0.81	0.78	0.79
	FBN1	0.89	0.87	0.88
Recall	UBE3A	0.78	0.76	0.77
	FBN1	0.91	0.89	0.90
F1-Score	UBE3A	0.79	0.77	0.78
	FBN1	0.90	0.88	0.89
ROC-AUC	UBE3A	0.86	0.84	0.85
	FBN1	0.93	0.91	0.92

As an error analysis for prediction of the weakness of the proposed model, false positives (instances where the model incorrectly predicts the presence of a disease or condition) and false negatives (instances where the model fails to detect a condition that is actually present) are focused. This type of analysis is particularly important in a clinical context, as understanding the nature of these errors can significantly impact the model's reliability in real-world applications.

For instance, false positives might lead to unnecessary treatments or interventions, increasing healthcare costs and patient stress, while false negatives could result in the missed diagnosis of a serious condition, delaying treatment and adversely affecting patient outcomes. In the context of genetic diseases, such as those analyzed in this study (e.g., UBE3A, FBN1, and monogenetic disorders), false positives might lead to unnecessary genetic counseling or testing, while false negatives could delay diagnosis and treatment for patients. The analysis should focus on how different genetic profiles or specific features might influence the occurrence of these errors.

Table 5.7 Error Analysis for False Positives and False Negatives

Error Type	Affected Dataset	Frequency	Clinical Impact	Suggested Mitigation Approach
False Positive	UBE3A	10%	Unnecessary genetic counseling or testing	Improve feature selection or introduce stricter thresholds for classification.
	FBN1	8%	Incorrect diagnosis, unnecessary interventions	Use ensemble models to combine predictions and reduce bias.
	dbGaP (Immune-Mediated)	12%	Unnecessary treatments, patient stress	Incorporate additional patient history or biomarkers.
False Negative	UBE3A	5%	Missed diagnosis, delayed treatment	Incorporate more diverse training data, use ensemble approaches for better coverage.
	FBN1	4%	Delayed diagnosis, missed early intervention	Use multi-modal data (e.g., clinical, genetic) for better context.
	dbGaP (Cystic Fibrosis)	7%	Missed diagnosis of rare mutations, delayed treatment	Use multi-layered models (e.g., hybrid CNN-LSTM).

VI. Conclusion and Future Enhancements

It is a well-known fact that genetic factors contribute to the development of all diseases. The degree to which gene plays its role in disease susceptibility varies and taking the research forward towards such genetic mechanisms facilitates strategies for averting disease onset and reduces its impact.

This work focussed on developing hybrid frameworks to efficiently perform sequence analysis on genetic sequences and to predict and classify the monogenetic disorders especially based on chromosome number 15. The sequential analysis is done using (i) Clustering-based method that uses Mean Shift Clustering algorithm that generates clusters with similar sequences. (ii) Alignment-based method that uses Needleman Wunsch algorithm to generate the longest subsequence of aligned sequence with highest score of similarity. The first method used CNN-BiLSTM neural network architecture to classify the monogenetic disorders and used Elephant Herd Optimization algorithm to tune the hyperparameters of neural network architecture. The second method used Horse Herd Optimization algorithm for hyperparameter tuning.

The proposed method using clustering method extracts k-mers from the DNA sequences, then uses one-hot encoding on k-mers and gives them as input to the Mean Shift Clustering algorithm that generates the clusters containing similar sequences by shifting mean in each iteration. The cluster of similar sequences forms the input layer of Convolution Neural Network that selects the local features from the cluster. The Bi-directional LSTM selects the temporal features which are used to extract the significant features for identifying monogenetic disorders. This work is illustrated on UBE3A dataset of NCBI that predicts the mutation of Angelman Syndrome and other

neuro diseases based on chromosome 15. This method produced the classification accuracy of 77.5%.

The later method is based on the sequence alignment method using Needleman Wunsch algorithm that generated longest subsequences of highest alignment score of similarity. These sequences are used as input layers for the same CNN-Bi-LSTM neural network architecture. This architecture classifies the genetic sequences by identifying the muted gene that is responsible for monogenetic disorders and classifies connecting tissue disorders and Marfan syndrome on FBN1 dataset of NCBI. The classification accuracy estimated by this method reached 89.2%.

The proposed model aimed to address the challenges of identifying DNA sequences that reproduce mutation variants and classifying monogenetic disorders, despite the complexities of genetic heterogeneity and limited data. However, the importance of biological interpretability cannot be overstated, particularly in clinical applications where understanding the underlying biological mechanisms is crucial. This work shows significant progress in developing a reliable and accurate classification system, overcoming the hurdles of overfitting and inconsistent data. The proposed models have important implications for the field of genetics, enabling more precise identification of disease-causing variants and improved diagnosis of rare genetic disorders.

The model can be enhanced involving assembly of multidisciplinary teams by considering environmental risk factors associated with adults especially among genetically susceptible persons. The hybrid models also have a further scope to conduct a prolonged and progressive analysis on monogenetic disorders over time need to be done for better diagnosis and prognosis.

REFERENCES

[1] Abhay Kumar, Vinay Kumar Sharma & Prafulla Kumar 2019, 'Nanopore sequencing: The fourth-generation sequencing', Journal of Entomology Zoology Studies, vol. 7, no. 4, pp. 1400-1403.

[2] Aimin Yang, Wei Zhang, Jiahao Wang, Ke Yang, Yang Han & Limin Zhang 2020, 'Review on the application of machine learning algorithms in the sequence data mining of DNA', Frontiers in Bioengineering and Biotechnology, vol. 8, no. 1032, pp. 1-13.

[3] Alekseyev, YO, Fazeli, R & Yang, S 2018, 'A next-generation sequencing primer--how does it work and what can it do', Academic Pathology, vol. 5, pp. 1-11.

[4] Alexandra Miere, Thomas Le Meur, Karen Bitton, Carlotta Pallone, Oudy Semoun, Vittorio Capuano, Donato Colantuono, Kawther Taibouni, Yasmina Chenoune, Polina Astroz, Sylvain Berlemont, Eric Petit & Eric Souied 2020, 'Deep learning-based classification of inherited retinal diseases using fundus autofluorescence', Journal of Clinical Medicine, vol. 9, no. 10, pp. 1-13.

[5] Ali, M, Ahmed, K, Bui, FM, Paul, BK, Ibrahim, SM, Quinn, JM & Moni, MA 2021, 'Machine learning-based statistical analysis for early-stage detection of cervical cancer', Computers in Biology and Medicine, Article ID. 104985, vol. 139, pp. 1-13.

[6] Ali Raza, Furqan Rustam, Hafeez Ur Rehman Siddiqui, Isabel de la Torre Diez, Begoña Garcia-Zapirain, Ernesto Lee & Imran Ashraf 2022, 'Predicting genetic disorder and types of disorder using chain classifier approach', Journal of Genes, vol. 14, no. 1, pp. 1-31.

[7] Amr Ezz El-Din Rashed, Hanan, M, Amer, Mervat EL-Seddek & Hossom EL-Din Moustafa 2021, 'Sequence alignment using machine learning-based Needleman–Wunsch Algorithm', IEEE, vol. 9, pp. 109522-109535.

[8] Annalisa Buniello, Jacqueline, AL, MacArthur, Maria Cerezo, Laura W Harris, James Hayhurst, Cinzia Malangone, Aoife McMahon, Joannella Morales, Edward Mountjoy, Elliot Sollis, Daniel Suveges, Olga Vrousgou, Patricia L Whetzel, Ridwan Amode, Jose A Guillen, Harpreet S Riat, Stephen J Trevanion, Peggy Hall, Heather Junkins, Paul Flicek, Tony Burdett, Lucia A Hindorff, Fiona Cunningham & Helen Parkinson 2019, 'The NHGRI-EBI GWAS catalog of published genome-wide association studies, targeted arrays and summary statistics', Nucleic Acids Research, vol. 47, pp. 1005-1012.

[9] Arthur L Delcher, Adam Phillippy, Jane Carlton & Steven L Salzberg 2002, 'Fast algorithms for large-scale genome alignment and comparison', Nucleic Acids Research, vol. 30, no. 11, pp. 2478-2483.

[10] Auton, A, Abecasis, GR & Altshuler, DM 2015, 'A global reference for human genetic variation', Nature, vol. 526, pp. 68-74.

[11] Battineni, G, Sagaro, GG, Chinatalapudi, N & Amenta, F 2020, 'Applications of machine learning predictive models in the chronic disease diagnosis', Journal of Personalized Medicine, vol. 10, no. 2, pp. 1-11.

[12] Benbelkacem, S & Atmani, B 2019, 'Random forests for diabetes diagnosis', in Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Saudi Arabia, pp. 1-4.

[13] Benjamin T James, Brian B Luczak & Hani Z Girgis 2018, 'MeShClust: An intelligent tool for clustering DNA sequences', Nucleic Acids Research, vol. 46, no. 14, pp. 1-10.

[14] Brian B Luczak, Benjamin T James & Hani Z Girgis 2019, 'A survey and evaluations of histogram-based statistics in alignment-free sequence comparison', Briefings in Bioinformatics, vol. 20, no. 4, pp. 1222-1237.

[15] Brudno, M, Malde, S, Poliakov, A, Do, CB, Couronne, O, Dubchak & Batzoglou, S 2003, 'Glocal alignment: Finding rearrangements during alignment', Bioinformatics, vol. 19, no. 1, pp. 54-62.

[16] Burak Gülmez 2023, 'A novel deep learning model with the grey wolf optimization algorithm for cotton disease detection', Journal of Universal Computer Science, vol. 29, no. 6, pp. 595-626.

[17] Chakraborty, F, Roy, PK & Nandi, D 2019, 'Oppositional elephant herding optimization with dynamic cauchy mutation for multilevel image thresholding', Evolutionary Intelligence, vol. 12, no. 1, pp. 1-23.

[18] Chen, L, Ozsu, MT & Oria, V 2005, 'Robust and fast similarity search for moving object trajectories', in Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 491-502.

[19] Cheng, S, Shi, Y & Qin, Q 2012, 'Population diversity of particle swarm optimizer solving single and multi-objective problems', International Journal of Swarm Intelligence Research (IJSIR), vol. 3, no. 4, pp. 23-60.
[20] Cheng, Y 1995, 'Mean shift, mode seeking, and clustering', IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790-799.
[21] Choubey, DK & Paul, S 2017, 'GA_RBF NN: A classification system for diabetes', International Journal of Biomedical Engineering and Technology, vol. 23, no. 1, pp. 71-93.
[22] Chowdary, KU & Prabhakara Rao, B 2019, 'Performance improvement in mimo-ofdm systems based on adaptive whale elephant herd optimization algorithm', International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, no. 1, pp. 6651-6657.
[23] Daiki Okada, Fumihiko Ino & Kenichi Hagihara 2015, 'Accelerating the smith-waterman algorithm with interpair pruning and band optimization for the all-pairs comparison of base sequences', BMC Bioinformatics, vol. 16, no. 321, pp. 1-15.
[24] Dan Wei, Qingshan Jiang, Yanjie Wei & Shengrui Wang 2012, 'A novel hierarchical clustering algorithm for gene sequences', BMC Bioinformatics, vol. 13, no. 174.
[25] Dinita Rahmalia & Teguh Herlambang 2020, 'Bat algorithm application for estimating super pairwise alignment parameters on similarity analysis between virus protein sequences', Journal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), vol. 6, no. 2, pp. 1-10.
[26] Dorigo, M, Maniezzo, V & Colorni, A 1996, 'Ant system: optimization by a colony of cooperating agents', IEEE Transaction on Systems, Man, and Cybernetics-Part B, vol. 26, no. 1, pp. 29-41.
[27] Duc-Hau Le, Nguyen Xuan Hoai & Yung-Keun Kwon 2015, 'A comparative study of classification-based machine learning methods for novel disease gene prediction', Knowledge and Systems Engineering, Advances in Intelligent Systems and Computing Book Series (AISC), vol. 326, pp. 577-588.
[28] Eberhart, R & Shi, Y 2001, 'Particle swarm optimization: Developments, applications and resources', in IEEE Proceedings of the 2001 Congress on Evolutionary Computation (CEC2001), pp. 81-86.
[29] Esin Ays, Zaimoglu, Nilüfer Yurtay, Hüseyin Demirci & Yüksel Yurtay 2023, 'A binary chaotic horse herd optimization algorithm for feature selection', International Journal of Engineering Science and Technology, vol. 44, pp. 1-22.
[30] Ghaheri, A, Shoar, S, Naderan, M & Hoseini, SS 2015, 'The applications of genetic algorithms in medicine', Oman Medical Journal, vol. 30, no. 6, pp. 406-416.
[31] Hakli, H 2019, 'Elephant herding optimization using multi-search strategy for continuous optimization problems', Academic Platform Journal of Engineering and Science, vol. 7, pp. 261-268.
[32] Hani Z Girgis 2022, 'MeShClust v3.0: high-quality clustering of DNA sequences using the mean shift algorithm and alignment-free identity scores ', BMC Genomics, vol. 23.
[33] Hasna El Haji & Larbi Alaoui 2020, 'A categorization of relevant sequence alignment algorithms with respect to data structures', International Journal of Advanced Computer Science and Applications, vol. 11, no. 6, pp. 268-273.
[34] Hayan Lee, James Gurtowsk, Shinjae Yoo, Maria Nattestad, Shoshana Marcus, Sara Goodwin, W, Richard McCombie & Michael C Schatz 2016, 'Third-generation sequencing and the future of genomics', Biorxiv, pp. 1-20.
[35] Henikoff, S & Henikoff, JG 1992, 'Amino acid substitution matrices from protein blocks', Proceedings of the National Academy of Sciences, vol. 89, pp. 10915-10919.
[36] Hunkapiller, T, Kaiser, RJ, Koop, BF & Hood, L 1991, 'Large-scale and automated DNA sequence determination', Science, vol. 254, no. 5028, pp. 59-67.
[37] Jackins, V, Vimal, S, Kaliappan, M & Lee, MY 2020, 'AI-based smart prediction of clinical disease using random forest classifier and naive bayes', Journal of Supercomputing, vol. 77, pp. 5198-5219.
[38] Jean-Michel Richer, Vincent Derrien & Jin-Kao Hao 2007, 'A new dynamic programming algorithm for multiple sequence alignment', Springer-Verlag, pp. 52-61.
[39] Jian Yan1, Yunjiang Qiu, Andre M Ribeiro dos Santos, Yimeng Yin, Yang E Li, Nick Vinckier, Naoki Nariai, Paola Benaglio, Anugraha Raman, Xiaoyu Li, Shicai Fan, Joshua Chiou, Fulin Chen, Kelly A Frazer, Kyle J Gaulton, Maike Sander, Jussi Taipale & Bing Ren 2021, 'Systematic analysis of binding of transcription factors to noncoding variants', Nature, vol. 591, no. 21, pp. 147-151
[40] Jian-Jun SHU, Kian Yan YONG & Weng Kong CHAN 2012, 'An improved scoring matrix for multiple sequence alignment', Mathematical Problems in Engineering, vol. 4, pp. 1-9.
[41] Khawla Tadist, Said Najah, Nikola S Nikolov, Fatiha Mrabti & Azeddine Zahi 2019, 'Feature selection methods and genomic big data: A systematic review', Journal of Big Data, vol. 6, no. 79, pp. 1-24.
[42] Kouser & Lalitha Rangarajan 2015, 'Promoter sequence analysis through no gap multiple sequence alignment of motif pairs', Procedia Computer Science, vol. 58, pp. 356-362.
[43] Krishnand, KN & Ghose, KD 2006, 'Glowworms swarm-based optimization algorithm for multimodal functions with collective robotics applications', International Journal of Multiagent and Grid Systems, vol. 2, no. 3, pp. 209-222.
[44] Lei Chen, XiaoYong Pan, Yu-Hang Zhang, Min Liu, Tao Huang & Yu-Dong Cai 2019, 'Classification of widely and rarely expressed genes with recurrent neural network', Computational and Structural Biotechnology, vol. 17, pp. 49-60.
[45] Li, JQ, Pan, S, Xie, S & Wang 2011, 'A hybrid artificial bee colony algorithm for flexible job shop scheduling problems', International Journal of Computers, Communications & Control, vol. 6, no. 2, pp. 286-296.
[46] Li, J, Guo, L, Li, Y & Liu, C 2019, 'Enhancing elephant herding optimization with novel individual updating strategies for large-scale optimization problems', Mathematics, vol. 7, no. 5, pp. 1-35.
[47] Lossie, AC, Whitney, MM & Amidon, D 2001, 'Distinct phenotypes distinguish the molecular classes of Angelman syndrome', Journal of Medical Genetics, vol. 38, no. 12 pp. 834-845.
[48] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, Masashi Sugiyama 2019, 'High-dimensional feature selection by feature-wise Kernelized lasso', Neural Computation, vol. 26, pp. 185-207.

[49] Manfred, G, Grabherr, Pamela Russell, Miriah Meyer, Evan Mauceli, Jessica Alfoldi, Federica Di Palma & Kerstin Lindblad-Toh 2010, 'Genome-wide synteny through highly sensitive sequence alignment: Satsuma', Bioinformatics, vol. 26, no. 9, pp. 1145-1151.
[50] Mansouri Fatimaezzahra, Benchikhi loubna, Sadgal Mohamed & Elfazziki Abdelaziz 2017, 'A combined cuckoo search algorithm and genetic algorithm for parameter optimization in computer vision', International Journal of Applied Engineering Research, vol. 12, no. 22, pp. 12940-12954.
[51] Marine Poullet & Ludovic Orlando 2020, 'Assessing DNA sequence alignment methods for characterizing ancient genomes and methylomes', Frontiers in Ecology and Evolution, vol. 8, no. 105, pp. 1-13.
[52] Maros, C, Martin, D & Zoltan, B 2021, 'Analysis and experimental evaluation of the Needleman Wunsch algorithm for trajectory comparison-science direct', Expert Systems with Applications, vol. 165, no. 1, pp. 1-12.
[53] Timothy Chappell, Shlomo Geva and James Hogan 2017, 'K-Means Clustering of Biological Sequences', ACM, pp. 1-4.

***