INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 3, MARCH 2026 | ISSN: 2456-4184 | IJNRD.ORG

An Integrated AI Approach for DNA Sequence-Based Genetic Disease Prediction

[1]K. Sudha Pavani, [2]Samala Sindhu, [3]Gunti Ashwini, [4]Eillandula Devi Jaahnavi, [5]Chinnapatlori Yogasri

[1]Assistant Professor,[2 3 4 5]B. Tech Students

[1 2 3 4 5]Department of Computer Science and Engineering (Data Science)

[1 2 3 4 5]CMR Technical Campus, Hyderabad, India

Abstract : This paper proposes an integrated Artificial Intelligence (AI) framework for DNA sequence-based genetic disease prediction. To enhance the accuracy without increasing complexity, our approach combines Machine Learning (ML) and Deep Learning (DL) techniques. We first collected DNA sequences and analyzed them to identify the key biological characteristics. These include k-Mer counts, GC proportions and repeat sequence patterns, which act as indicators of the genomic composition and the mutation likelihood. To understand the behavior of different genetic patterns, we experimented with several classical ML models such as, Logistic Regression, Support Vector Machine (SVM), Random Forest and XGBoost. Interestingly, XGBoost consistently achieved the highest precision, scalability and robustness. This performance shows its ability to capture complex feature relationships, making it the optimal ML baseline. We used a one-dimensional Convolutional Neural Network (1D-CNN) to capture spatial dependencies and hidden sequential patterns directly from DNA sequences for in-depth feature extraction. We observed that combining ML and DL approaches allows the hybrid framework to perform better than the individual models. This leads to higher reliability and better prediction accuracy across diverse genetic patterns. This hybrid model provides a reliable solution for early genetic related disease detection and prediction, which supports personalized and precision medicine.

IndexTerms **- Logistic Regression, Support Vector Machine (SVM), Random Forest, XGBoost, 1D-CNN, Hybrid framework**

I. INTRODUCTION

Small changes in our DNA or Deoxyribonucleic acid can disturb the functioning of the cell. Every cell in the body holds the DNA. This DNA works like a long instruction sheet that guides how cells grow, repair themselves and perform daily activities. The code inside DNA is written using four chemical bases: Adenine(A), Thymine(T), Cytosine(C) and Guanine(G). These bases attach to a sugar-phosphate backbone structure to form nucleotides. It acts as the basic units of the DNA strand. When these units join together, they twist into a double-stranded, ladder-like shape. Specific stretches within this long molecule are known as genes. It carries the information needed to produce proteins that support biological functions such as metabolism, immunity, hormone balance and tissue repair. All genes are packaged into chromosomes inside the cell's nucleus. It helps to maintain order and ensuring genetic information is safely passed from one generation to the next generation.

Even tiny alterations in the genetic code can change the behaviour of gene and may eventually lead to health issues. These variations can appear in different forms, including single-base substitutions, insertions, deletions or unusual expansions of repeated sequences. Some disorders, like Sickle Cell Anaemia caused by mutations in the HBB gene and Cystic Fibrosis linked to changes in the CFTR, appear at birth. Others disorders, like hereditary breast cancer associated with BRCA1 and BRCA2 mutations, may not show up until adulthood. Because even minor changes in DNA sequences can affect protein function, identifying these variations early is extremely important in life.

With recent advancements of sequence technologies such as Next-Generation Sequencing (NGS), Whole-Genome Sequencing and Whole-Exome Sequencing, researchers now generate enormous amounts of genetic data. Manually scanning this data for subtle, disease-related variations is nearly impossible. Most traditional diagnostic methods still rely on clinical symptoms or known mutations. It can lead to overlook the hidden patterns present in DNA sequences. Because of these limitations, there is a clear need for advanced computational techniques that can handle large genomic datasets and to identify and detect the complex sequence patterns.

In recent years, Artificial Intelligence (AI) has become a valuable tool for this work. ML techniques can help to identify meaningful insights such as shifts in GC content, frequent occurrences of specific base fragments or unusual repetition patterns, which shows the possible disease links. Deep-learning models, particularly 1D-CNNs, take a direct approach by learning from the raw nucleotides sequence itself. When ML and DL are combined, the strengths complement each other: ML provides interpretable patterns, while DL uncovers more complex patterns hidden in the sequence. Together, they often yield more reliable predictions than using either technique alone.

In this study, we use a combination of Machine Learning (ML) and Deep Learning (DL) methods to improve the prediction of genetic disorders. DNA sequences obtained from the NCBI repository will be cleaned and organized to extract meaningful

sequence-level information. These processed features will then be used to train a set of ML models such as Logistic Regression, SVM, Random Forest and XGBoost are mainly used to understand how each of them reacts to disease-related variations. At the same time, a custom 1D-CNN will analyse the raw sequences directly, enabling it to capture broader structural patterns that simple feature engineering may miss. After both modelling approaches generate their outputs, the results will be merged into a single hybrid system designed to enhance the accuracy and consistency. The overall aim is to supports the easier detection of genetic conditions and provide a practical, adaptable framework for future research in genomics and personalized healthcare.

II. RELATED WORK

Research on DNA-based disease prediction provides important biological insights and modern computational methods. Dasari and Bhukya [1] demonstrated that analysing pattern-frequency in exonic regions improves diagnostic precision. However, using manual features restricts the detection of deeper sequence-level patterns. The works of Sinden et al. [2], Massey and Jones [3], Depienne and Mandel [4], Cardoso and Marques [5], and Biscotti et al. [6] explained mechanisms, such as triplet repeat instability, repetitive DNA organization and anticipation in hereditary disorders. Yet, these biological findings have not led to creation of computational tools for disease prediction. Recent AI-driven studies aim to fill this gap. Mathur et al. [7] showed that convolutional neural networks can classify genomic sequences more efficiently than traditional machine learning models. At the same time, Bahado-Singh et al. [8] found that combining the epigenomic signatures with deep learning can reveal disease-specific methylation patterns. Broader surveys by Alharbi and Rashid [9] and Abass and Adeshina [10] highlighted the strengths and weaknesses of CNNs, RNNs, autoencoders and hybrid models for genomic analysis. Le [11], Lotfollahi and Theis [12] and Kourou et al. [13] pointed out that classical machine-learning methods are easier to interpret, but they struggle to capture high-order sequence dependencies. Pedregosa et al. [14] introduced Scikit-learn as a flexible framework for building machine-learning pipelines; however, its design does not fit the complex, high dimensional nature of biological sequences. More evidence of deep learning's potential comes from Montesinos López et al. [15] in genomic selection tasks and Alzoubi et al. [16] in SNP-based disease risk prediction, although both lack integrated interpretability. The HEC-ASD framework introduced by Ismail et al. [17] shows that combining multiple learning algorithms in an ensemble structure significantly enhances the robustness and consistency of gene-level predictive performance. Together, these findings reveal a persistent research gap. While biological studies clarify disease mechanisms and computational models display predictive strength, existing frameworks do not integrate interpretable machine-learning techniques with deep-learning driven feature extraction. This limitation emphasizes the need for a hybrid structure that combines the strengths of machine learning and deep learning for robust DNA-based disease prediction, as supported by recent computational genomics studies.

III. METHODOLOGY

This section outlines the full workflow of the hybrid AI framework developed for predicting genetic disorders from DNA sequences.

3.1 System Architecture

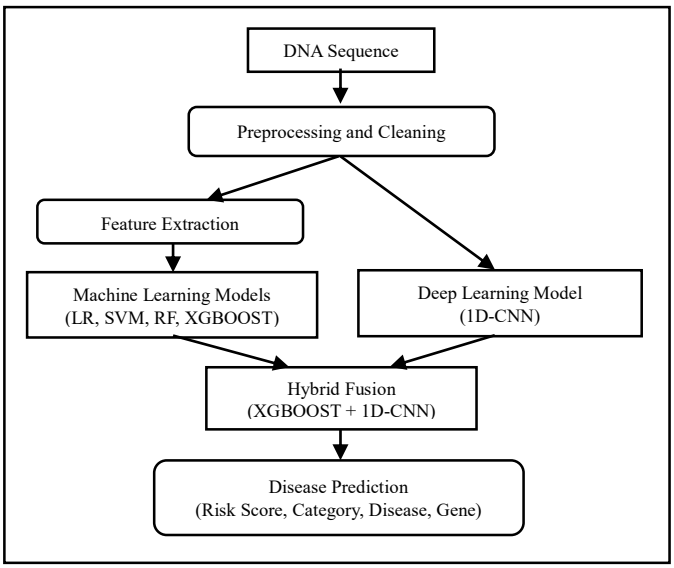

Figure 1. Proposed AI Framework for DNA Sequence-Based Disease Prediction

The proposed system follows a dual-path architecture for DNA sequence-based genetic disease prediction. Initially, raw DNA sequences undergo preprocessing and cleaning to ensure data quality. The processed data is then handled through two parallel paths. In the first path, handcrafted features such as k-mer frequencies, GC content, and repeat statistics are extracted and used by machine learning models. In the second path, a one-dimensional convolutional neural network (1D-CNN) learns deep sequence patterns directly from the DNA data. The features obtained from both paths are combined using a hybrid feature fusion approach and

processed within a hybrid model (XGBoost + 1D-CNN) to generate the final prediction. This integrated architecture improves prediction accuracy by leveraging both biological feature representation and deep learning-based pattern extraction.

3.2 Data Collection and Cleaning

DNA sequences representing both disease and healthy classes were obtained from the NCBI database in FASTA format. The sequences were retrieved and processed programmatically. Each entry was carefully inspected to confirm its validity, and any ambiguous nucleotides (e.g., N, R, Y, S, W) were removed, keeping only the standard bases {A, C, G, T}. All sequences were standardized to uppercase, and any corrupted or incomplete sequences were excluded from further analysis. After cleaning, class labels were assigned based on the metadata of the downloaded files. Finally, the processed dataset was divided into training and testing subsets to support model development and evaluation.

Algorithm 1: Preprocess Sequences ()

Input: Raw FASTA files containing DNA sequences
Output: Cleaned DNA sequences with assigned labels

1. Load all FASTA files from the input directory: files = loadFASTA(inputDirectory).
2. For each file in files:
   2.1. seq = readSequence(file)
   2.2. cleanSeq = removeNonACGT(seq)
   2.3. cleanSeq = toUpperCase(cleanSeq)
   2.4. label = extractLabel(fileMetadata)
   2.5. Store (cleanSeq, label)
3. Split the stored dataset into training and testing sets.
4. Return cleaned Sequences and labels.

Algorithm 1. Preprocess Sequences Algorithm

Algorithm 1 reads each FASTA file, removes invalid characters, standardizes sequences, and assigns class labels. The resulting cleaned DNA sequences form the input for feature extraction and model training, ensuring uniformity and reducing noise.

3.3 Feature Extraction

To numerically represent DNA sequences for machine-learning and deep-learning models, three categories of biologically relevant features were extracted: **GC content**, **k-Mer frequencies**, and **maximum repeat length**.

$$S = \{s_1, s_2, \dots, s_L\}$$

be of length L, where each $s_i \in \{A, C, G, T\}$.

3.3.1 GC Content

GC content measures the proportion of Guanine (G) and Cytosine (C) bases in a sequence. It is computed as:

$$GC\ Content = \frac{G + C}{A + T + G + C}$$

This feature reflects sequence stability and mutation likelihood.

3.3.2 K-Mer Frequencies

A k-Mer is a contiguous substring of length k:

$$k\text{-mer}_i = S[i{:}\, i + k - 1]$$

The normalized frequency of each k-Mer is computed as follows:

$$f(k\text{-mer}) = \frac{\text{count}(k\text{-Mer})}{(L - k + 1)}$$

All possible 2-mer and 3-mer patterns were considered, generating a complete local motif profile of the sequence.

3.3.3 Maximum Repeat Length

To quantify structural irregularities and repeated patterns, a run-length encoding (RLE) scan is applied:

$$R = \max\{\text{consecutive identical bases in } S\}$$

This value is strongly associated with expansion-based genetic disorders.

Algorithm 2: Extract Features ()

Input: DNA sequence S
Output: Feature vector F

1. Count occurrences of A, C, G, T in S.
2. Compute GC content: $GC = \frac{G+C}{A+T+G+C}$.
3. Compute normalized frequency for all 2-mer and 3-mer patterns.
4. Perform RLE scan to identify maximum repeat length.
5. Combine GC content, k-mer frequencies, and repeat length into vector F.
6. Return F.

Algorithm 2. Feature Extraction Algorithm

Algorithm 2 extracts composition-based, motif-based, and structure-based features, providing a biologically meaningful representation of genetic variation. These features capture both local sequence patterns and global compositional properties of DNA, which are essential for distinguishing between normal and disease-associated sequences. Additionally, the combination of statistical and structural features improves the robustness and generalization capability of the predictive models.

3.4 Development of Machine Learning & Deep Learning Models

The cleaned and feature-extracted dataset was used to train four supervised classification models: **Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost**.

3.4.1 Logistic Regression

Logistic Regression is used as a simple baseline model. It links the selected genomic features to disease probability through a linear rule and is quick to run and easy to interpret. Its main limitation is that it cannot capture the complex patterns often present in DNA sequences, so it mainly acts as a reference point for comparing stronger models.

Algorithm 3: Train Logistic Regression ()

Input: Feature matrix X, labels y
Output: Trained LR model

1. Standardize Xusing z-score normalization.
2. Initialize parameters $w$and b.
3. For each training iteration:
   3.1. Calculate predicted outputs using the logistic function: $\hat{y} = \sigma(w^T X + b)$
   3.2 Determine the gradients based on the prediction errors.
   3.3. Update the weights $w$and bias $b$using gradient descent.
4. After all iterations, return the optimized LR model

Algorithm 3. Train Logistic Regression Algorithm

Algorithm 3 updates model parameters through gradient descent on standardized features to ensure stable optimization. Training continues until the loss converges, after which the model outputs probability-based classifications. Although effective for baseline comparison, LR cannot represent nonlinear genomic patterns.

3.4.2 Support Vector Machine (SVM)

Support Vector Machine constructs a decision boundary that maximizes the margin between diseased and normal DNA sequences. Using the RBF kernel, it can model non-linear relationships in the genomic feature space, making it suitable for moderately complex patterns in the extracted features.

Algorithm 4: Train SVM ()

Input: Standardized features X, labels y
Output: Trained SVM classifier

1. Choose RBF kernel.
2. Set hyperparameters $C$and γ.
3. Compute kernel matrix for all samples.
4. Solve the maximum-margin optimization problem.
5. Return the trained SVM model.

Algorithm 4. Support Vector Machine Training Algorithm

Algorithm 4 determines the key support vectors that establish the optimal separating margin, while regularization helps control misclassification errors. The use of the RBF kernel allows the SVM to handle nonlinear separations, enhancing prediction accuracy for genomic sequences. Nonetheless, the computational cost of training increases significantly as the dataset size grows.

3.4.3 Random Forest

Random Forest builds a collection of decision trees; each trained on a different bootstrapped subset of the genomic features. Combining the outputs of multiple trees enhances the stability of predictions and allows the model to capture nonlinear interactions among features, including GC content, k-mer frequencies, and repeat sequences. This ensemble approach also helps to mitigate overfitting, which is common in single decision-tree models.

Algorithm 5: Train Random Forest ()

Input: Feature matrix X, labels y
Output: Trained RF model

1. Initialize number of trees T.
2. For each tree $t = 1$to T:
   2.1. Sample training data using bootstrapping.
   2.2. Grow decision tree on the sampled data.
3. Aggregate predictions from all Ttrees using majority voting.
4. Return the trained RF model.

Algorithm 5. Random Forest Training Algorithm

Algorithm 5 trains multiple decision trees on different bootstrapped subsets of the data and combines their predictions through majority voting. This ensemble approach captures complex non-linear patterns in genomic features and improves overall classification accuracy while maintaining generalization.

3.4.4 XGBoost

XGBoost is a gradient-boosted decision tree model optimized for both speed and accuracy. Each tree iteratively minimizes the errors of previous trees, while regularization, shrinkage, column sampling, and weighted boosting improve generalization. This makes XGBoost highly effective for modeling complex, non-linear interactions among genomic features.

Algorithm 6: Train XGBoost ()

Input: Feature matrix X, labels y
Output: Trained XGBoost model

1. Initialize model parameters (learning rate, max depth, subsample).
2. Compute initial predictions.
3. For each boosting round $r = 1$to R:
   3.1. Compute gradients and Hessians.
   3.2. Fit a new decision tree to the residual errors of the current model.
   3.3. Apply the shrinkage factor to update the ensemble's predictions.
4. Return the final trained XGBoost model.

Algorithm 6. XGBoost Training Algorithm

Algorithm 6 XGBoost demonstrates superior performance compared to other machine-learning models by effectively capturing complex, nonlinear relationships within genomic features. Its gradient boosting framework allows sequential learning from previous errors, which improves overall prediction accuracy and robustness. Additionally, its built-in regularization techniques help prevent overfitting, ensuring better generalization on unseen DNA sequence data.

3.4.5 1D Convolutional Neural Network (Deep Learning)

A one-dimensional convolutional neural network (1D-CNN) was employed to model sequential patterns in DNA sequences. The convolutional layers identify local sequence motifs, while max-pooling layers reduce feature dimensionality, and batch normalization is applied to enhance training stability and convergence.

Algorithm 7: Train 1D-CNN ()

Input: Raw DNA sequences and corresponding labels
Output: Trained 1D-CNN embedding model

1. Convert DNA sequences into numerical form (e.g., one-hot encoding).
2. Reshape sequences to match the CNN input format.
3. Apply a 1D convolutional layer followed by batch normalization and max-pooling.
4. Repeat convolution and pooling blocks to extract hierarchical features.

5. Flatten the final feature maps and connect to fully connected (dense) layers.
6. Train the dense layers using the Adam optimizer and cross-entropy loss.
7. Evaluate performance on a validation set to monitor learning.
8. Save the trained model for downstream prediction tasks.

Algorithm 7. 1D Convolutional Neural Network Training Algorithm

Algorithm 7 enables automatic learning of sequence motifs and long-range dependencies, improving accuracy for genetic disease prediction compared to classical ML models. The convolutional layers automatically detect important local patterns such as motifs, while deeper layers capture complex hierarchical representations of the DNA sequence. This deep learning approach reduces the need for manual feature engineering and enhances the model's ability to generalize across diverse genetic variations.

3.5 Model Evaluation

To measure the classification performance, the following metrics were computed:

Accuracy (Acc):

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Precision(Prec):

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

Recall(Rec):

$$Recall = \frac{TP}{(TP + FN)}$$

F1-Score(F1):

$$F1 - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

3.6 Combining Outputs into a Single Hybrid Prediction Model:

The proposed hybrid framework merges features derived from traditional machine-learning methods with embeddings learned by a 1D-CNN, enabling both interpretability of engineered features and the ability to capture complex patterns from raw DNA sequences.

Algorithm 8: Hybrid Prediction ()

Input: Raw DNA sequence, corresponding handcrafted features
Output: Predicted disease label

1. Compute handcrafted features from the DNA sequences (e.g., k-Mer counts, GC content, repeat).
2. Pass the raw sequences through the trained 1D-CNN to generate deep embeddings.
3. Combine the handcrafted features and CNN embeddings into a single hybrid feature vector.
4. Train or utilize an XGBoost classifier using the hybrid feature vector.
5. Predict the disease label for each DNA sequence.
6. Return the predicted label.

Algorithm 8. Hybrid Prediction Algorithm

Algorithm 8 combines manually extracted biological features with patterns learned through deep learning, resulting in improved prediction performance over using either machine-learning or deep-learning models alone.

IV. RESULTS AND DISCUSSION

Table 1 shows all models performance. Traditional ML models (Logistic, SVM) perform moderately, XGBoost improves results, 1D-CNN does even better, and the hybrid model achieves the best accuracy (94%).

Table 1. Comparative Analysis of Machine Learning, Deep Learning, and Hybrid Frameworks

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	75.00	61.00	58.00	58.00
SVM	77.38	39.00	50.00	44.00
Random Forest	77.38	64.00	54.00	52.00
XGBoost	88.10	89.00	76.00	80.00
1D-CNN	91.67	95.00	82.00	86.00
Hybrid XGBoost + 1D CNN Proposed Method	94.05	94.00	89.00	91.00

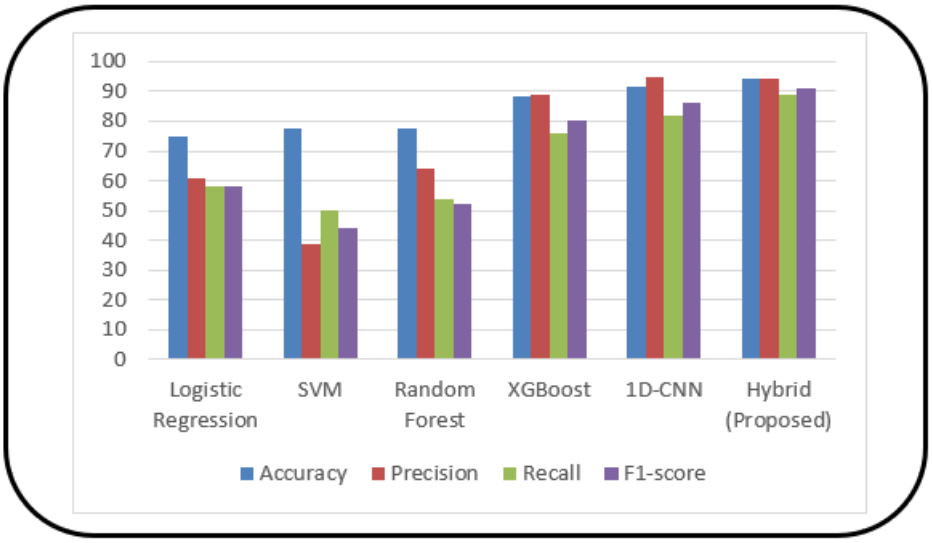

Figure 2. Evaluation of Predictive Performance Across Models

Figure 10 shows that traditional ML models perform moderately, XGBoost improves the results, and 1D-CNN performs even better by learning sequence patterns. The proposed hybrid system achieves the highest metrics overall, confirming that combining ML features with CNN embeddings gives the best performance.

V. CONCLUSION

The study explored a combined learning approach where simple, manually computed genetic features were used together with sequence patterns learned by a 1D-CNN. The handcrafted features give an overall picture of each DNA segment by capturing elements such as k-Mer counts, GC levels, and repeating regions. In contrast, the CNN focuses on the fine-grained patterns within the sequence that are difficult to design by hand. When these two sources of information were merged, the model delivered steady performance and achieved nearly 94% accuracy. The training and validation curves also remained smooth and consistent, indicating that the model was improving without showing signs of overfitting. Even so, the work is still constrained by the limited size and range of the genomic data used.

In the future, it may be helpful to study how specific variations influence the model's predictions and to test deeper or more targeted CNN structures. Adding other biological information—such as gene expression or epigenetic markers—could further enhance the model's ability to generalize. Most importantly, applying the approach to larger, clinically validated datasets will be essential to understand how well it holds up in real diagnostic environments.

ACKNOWLEDGEMENT

The authors sincerely express their gratitude to Mrs. K. Sudha Pavani, Assistant Professor, Department of Computer Science and Engineering (Data Science), CMR Technical Campus, for her valuable guidance and continuous support throughout this research work. The authors also thank the faculty and staff of the department for their encouragement and assistance. We are grateful to the National Center for Biotechnology Information (NCBI) for providing access to the genomic datasets used in this study. Additionally, we acknowledge the contributions of our peers and the open-source research community for their valuable resources and feedback, which helped improve the quality of this work.

REFERENCES

[1] C. M. Dasari and R. Bhukya, "Disease Diagnosis based on Various Pattern Frequency from Extracted Exons," *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, Oct. 7–9, 2022.

[2] R. R. Sinden, V. N. Potaman, E. A. Oussatcheva, C. E. Pearson, Y. L. Lyubchenko, and L. S. Shlyakhtenko, "Triplet repeat DNA structures and human genetic disease: dynamic mutations from dynamic DNA," *Journal of Biosciences*, vol. 27, no. 1 Suppl 1, pp. 53–65, Feb. 2002. doi: 10.1007/BF02703683.

[3] T. H. Massey and L. Jones, "The central role of DNA damage and repair in CAG repeat diseases," *Disease Models & Mechanisms*, vol. 11, no. 1, p. dmm031930, Jan. 2018. doi: 10.1242/dmm.031930.

[4] C. Depienne and J.-L. Mandel, "30 years of repeat expansion disorders: What have we learned and what are the remaining challenges?," *American Journal of Human Genetics*, vol. 108, no. 5, pp. 764–785, May 2021. doi: 10.1016/j.ajhg.2021.03.011.

[5] I. L. Cardoso and V. Marques, "Trinucleotide repeat diseases—anticipation diseases," *2018*. Fernando Pessoa University.

[6] M. A. Biscotti, E. Olmo, and J. S. P. Heslop-Harrison, "Repetitive DNA in eukaryotic genomes," *Chromosome Res.*, vol. 23, no. 3, pp. 415–420, Sep. 2015, doi: 10.1007/s10577-015-9499-z.

[7] G. Mathur, A. Pandey, and S. Goyal, "A comprehensive tool for rapid and accurate prediction of disease using DNA sequence classifier," *J. Ambient Intell. Humaniz. Comput.*, Jun. 25, 2022, pp. 1–17, doi: 10.1007/s12652-022-04099-y.

[8] S. Bahado-Singh et al., "Deep Learning/Artificial Intelligence and Blood-Based DNA Epigenomic Prediction of Cerebral Palsy," *MDPI*, 2019.

[9] F. Alharbi and M. Rashid, "A review of deep learning applications in human genomics using next-generation sequencing data," *Human Genomics*, vol. 16, no. 1, 2022, doi: 10.1186/s40246-022-00396-x.

[10] A. Abass and A. Adeshina, "Deep Learning Methodologies for Genomic Data Prediction: Review," *J. Artif. Intell. Med. Sci.*, 2021, SpringerLink.

[11] D.-H. Le, "Machine learning-based approaches for disease gene prediction," *Brief. Funct. Genomics*, vol. 19, no. 5–6, pp. 350–363, Dec. 4, 2020, doi: 10.1093/bfgp/elaa013.

[12] S. B. Lotfollahi and M. Theis, "Interpretable machine learning for genomics," *Hum. Genet.*, vol. 141, pp. 1499–1513, Oct. 2021, doi: 10.1007/s00439-021-02387-9.

[13] Y. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015, doi: 10.1016/j.csbj.2014.11.005.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[15] O. A. Montesinos-López, A. Montesinos-López, P. Pérez-Rodríguez, J. W. R. Martini, S. B. Fajardo-Flores, et al., "A review of deep learning applications for genomic selection," BMC Genomics, vol. 22, art. 19, Jan. 2021, doi: 10.1186/s12864-020-07319-x.

[16] H. Alzoubi, R. Alzubi, and N. Ramzan, "Deep Learning Framework for Complex Disease Risk Prediction Using Genomic Variations," Sensors, vol. 23, no. 9, Art. 4439, May 2023, doi: 10.3390/s23094439.

[17] E. Ismail, W. Gad, and M. H. Hashem, "HEC-ASD: a hybrid ensemble-based classification model for predicting autism spectrum disorder disease genes," BMC Bioinformatics, vol. 23, Art. 554, Dec. 2022, doi: 10.1186/s12859-022-05099-7.