Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

Robust Synthetic DNA Generation Using GAN

Richa Saxena
Computer Science & Engineering
Moradabad Institute of Technology
Moradabad, India
richasaxena2006@gmail.com

Himanshu Saini
Computer Science & Engineering
Moradabad Institute of Technology
Moradabad, India
himanshurahul9927@gmail.com

Krati Gupta
Computer Science & Engineering
Moradabad Institute of Technology
Moradabad, India
kratigupta064@gmail.com

Kushagra Bhatnagar
Computer Science & Engineering
Moradabad Institute of Technology
Moradabad, India
kushagrabhatnagar27@gmail.com

Bhumi Joshi
Computer Science & Engineering
Moradabad Institute of Technology
Moradabad, India
bhumijoshi2005@gmail.com

***Abstract*—The generation of synthetic data is also another active area of research in machine learning and data science, considering the continuous need for large quantities of data in these fields. However, it is challenging to obtain high-quality real data due to issues of privacy, availability, and ethics. The main contribution of this paper is to propose a hybrid generative model by integrating two different generative architectures, namely Deep Convolutional Generative Adversarial Networks and Wasserstein Generative Adversarial Networks with Gradient Penalty, for generating synthetic data. The use of DCGAN in this paper is motivated by its ability to extract high-level feature representations of data using deep convolutional layers. At the same time, the incorporation of WGAN-GP into our model is based on the robustness it offers in dealing with issues such as mode collapse and unstable gradients, as encountered in the traditional GAN architectures. In order to ascertain the reliability and quality of the generated synthetic data, the proposed framework has incorporated various stages of evaluation, verification, and validation. The quality and efficiency of the generated synthetic data can be ascertained through the application of various machine learning evaluation parameters such as precision, recall, and F1 score.These metrics assess the quality of the produced synthetic data in facilitating the classification or prediction task. Furthermore, the proposed framework has verification steps, such as using SeqKit to assess the characteristics and statistical properties of the produced synthetic data. An additional validation step is incorporated in the proposed framework to verify if the produced synthetic data still contains significant patterns and realistic nucleotide distribution. The experimental results demonstrate the effectiveness of the proposed model in producing high-quality synthetic tabular data with significant characteristics of the original data set.**

***Index Terms*—Synthetic DNA Generation, DCGAN, WGAN-GP, Bioinformatics, Evaluation Metrics**

ABBREVIATIONS

GAN	Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty
DNA	Deoxyribonucleic Acid

I. INTRODUCTION

The rapid increase in the generation of genomic data has had a considerable effect on the growth of contemporary biological research, leading to the emergence of breakthroughs in areas such as bioinformatics, disease prediction, and personalized medicine. However, the key problem with the growth of biological datasets is that there is limited access to large quantities of high-quality DNA sequence data due to the issues of privacy and ethics associated with the use of the data, as well as the high cost of acquiring it.

However, in recent times, the GANs have received significant interest because of their capability to learn complex data distributions and produce synthetic data that is very similar to the real data. The general framework for the GANs is composed of two main neural networks: the generator and the discriminator. These two networks are trained in an adversarial way, wherein the generator generates synthetic data while the discriminator checks the authenticity of the synthetic data. Despite the fact that promising results have been obtained through the application of the GANs-based approaches, the conventional GANs have some problems that are quite serious. These problems include instability in the training process, gradient vanishing, and mode collapse.

A. Motivation

The generation of synthetic DNA sequence has gained importance for various applications, including data augmentation of genomic data, data sharing for preserving privacy, and analysis of biological data on a large scale. The usage of statistical models might not be sufficient to manage complex structural relationships in the data. The usage of traditional GAN models for synthetic data generation might not be suitable for DNA sequence generation due to the instable nature of the training process and the quality of the data produced. Considering the limitations of traditional models for synthetic data generation, it has become essential to utilize a powerful model for synthetic data generation. Utilizing convolutional neural networks for one-hot encoded matrices

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

of DNA sequence can be utilized for determining structural relationships of nucleotide sequences. Utilizing powerful models for synthetic data generation has opened doors to newer opportunities.

B. Contributions

In the proposed work, we propose a hybrid architecture that is able to inherit the feature learning capability of DCGANs, along with the training stability of WGANs. This is achieved by proposing a synthetic DNA sequence generation framework that is able to inherit the feature learning capability of DCGANs. This is achieved by encoding the DNA sequence in a structured numerical form using the one-hot encoding mechanism. This enables convolutional neural networks to learn the feature dependencies in the DNA sequence. In addition, the proposed framework is able to inherit the training stability of WGANs by using the Wasserstein loss function. This ensures that the proposed architecture is able to avoid GAN-related issues. Thus, the proposed architecture is able to generate synthetic DNA sequences that can be used in genomic studies. From the experimental results, it is clear that the proposed architecture is able to generate high-quality synthetic DNA sequences.

II. Literature Survey

In this proposed study, GANs will be employed in generating biologically relevant synthetic DNA sequences that resemble real genomic data. Before the advent of deep learning-based generative models, several approaches were employed in generating synthetic DNA sequences. One of the earliest approaches was the Markov chain model [1]–[3] in which DNA sequences were generated based on the probability of transition from one nucleotide to another. Although Markov models were simple and computationally inexpensive, these approaches only considered short-range dependencies in DNA sequences. However, long-range dependencies [4] in DNA sequences, as seen in real genomic data, cannot be modeled by Markov chain approaches. One of the most popular approaches in generating synthetic DNA sequences was the Hidden Markov Model [5]–[7] in which hidden states were introduced to model biological phenomena, including gene regions or motifs.

The HMM was extensively employed in gene prediction and sequence alignments. However, these approaches heavily rely on transition probabilities, making these approaches limited in modeling biological phenomena. Other statistical simulation techniques, like Monte Carlo simulation [8], were also employed to generate synthetic sequences. The techniques include probabilistic sampling methods [9], among others. The techniques generate sequences by random sampling of statistical distributions [10]. Although these techniques could be used to approximate simple sequence features, they could not maintain higher-order dependencies, hence failing to generate synthetic sequences with biological realism. Because of these limitations, these methods are not suitable for modern large-scale genomic studies, where the data is highly complex and multidimensional. This is because these methods rely heavily on manual assumptions, do not learn complex patterns in the data, and cannot learn non-linear relationships within the DNA sequences. In contrast, the proposed model which is DCGAN architecture [11], was proposed with the aim of improving the stability and quality of generated data by utilizing CNNs. The DCGAN architecture enables the learning of hierarchical features and complex structures from the input data. Therefore, the DCGAN architecture is applicable to the modeling of genomic sequence data with complex structures. However, the GAN and DCGAN models are limited by their instability and mode collapse problems. The instability and mode collapse problems are tackled by the WGAN architecture [12]–[16]. The WGAN architecture is an advancement over the GAN and DCGAN models in that it uses the Wasserstein distance metric to measure the difference between the real and generated data distributions. The WGAN architecture provides more meaningful gradient updates and improves the stability of the models. The WGAN-GP architecture is an advancement over the WGAN architecture and is obtained by replacing the clipping function with the gradient penalty term. In the proposed work, the DCGAN and WGAN-GP models are utilized for generating synthetic DNA sequences in a realistic fashion. DCGAN is useful in helping the model learn the patterns in the sequence with the help of the deep convolutional structures. Through adversarial training, the generator network is able to learn the intricate dependencies between nucleotides in the DNA sequences. Furthermore, the model also includes a set of rigorous verification and validation processes. In the first place, the generated sequences are validated using machine learning performance measures such as precision, recall, F1 score, and accuracy. In the second place, the sequences are validated using a tool called SeqKit [17] that analyzes the distribution of nucleotides, sequence composition, and other statistical measures to determine whether the generated sequences of DNA follow realistic patterns. This framework guarantees that the synthetic sequences generated will not only be statistically valid but will also be valid from a biological perspective. Additionally, the proposed research also considers the 3D visualization of the DNA sequences to better understand the structural and compositional characteristics of the generated data. In the visualization component of the model, different nucleotides will be represented using different colors for better interpretation. For instance, adenine (A), thymine (T), guanine (G), and cytosine (C) will be represented using different color mappings [18]–[20]. This will enable the researchers to better understand the spatial arrangement of the nucleotides in the generated DNA sequences.

III. Proposed Methodology

The proposed framework for the generation of synthetic DNA sequences is based on the hybrid framework that utilizes the feature learning property of the DCGAN and the stable training property of the WGAN. The proposed framework is based on the two neural networks, which include the generator G and the critic D.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

The generator is responsible for the generation of the synthetic DNA sequences based on the random noise vectors, and the critic is responsible for the evaluation of the similarity between the generated sequences and the real DNA sequences. During the adversarial training, the generator improves its capability to generate the realistic and meaningful DNA sequences.

A. Data Representation

DNA sequences consist of four nucleotides:

$$\mathcal{N} = \{A, T, C, G\} \tag{1}$$

As neural networks require numerical input, each nucleotide is represented using **one-hot encoding** in the form of a vector. In the vector representation, each nucleotide is represented as a binary vector:

$$A = [1, 0, 0, 0] \tag{2}$$

$$T = [0, 1, 0, 0] \tag{3}$$

$$C = [0, 0, 1, 0] \tag{4}$$

$$G = [0, 0, 0, 1] \tag{5}$$

For a DNA sequence of length L, the encoded representation can be expressed as:

$$X \in \mathbb{R}^{L \times 4} \tag{6}$$

This matrix representation enables convolutional neural networks to learn spatial dependencies among nucleotides.

B. Latent Space Modeling

For generative models, the latent space is a compressed feature space from which the model generates synthetic data. The input to the generator is a randomly selected vector from the prior probability distribution:

$$z \sim P_z(z) \tag{7}$$

Usually, $P_z(z)$ is a Gaussian or a uniform distribution. The latent vector represents the structural patterns that the generator will later transform into meaningful DNA sequences.

The generator learns a nonlinear transformation from the latent space to the data space:

$$G : z \to X_{fake} \tag{8}$$

where X_{fake} is the generated DNA sequence matrix. During training, the generator gradually starts to learn how to map the various parts of the latent space to the most plausible DNA patterns.

The latent representation helps the model create diverse sequences by sampling different noise vectors, thereby ensuring variability and diversity in the generated synthetic dataset.

C. DCGAN Generator

The generator network receives a random noise vector sampled from a latent distribution:

$$z \sim P_z(z) \tag{9}$$

The generator learns a mapping function that transforms the latent vector into synthetic DNA sequences:

$$G(z; \theta_g) \to x_{fake} \tag{10}$$

where G is the generator network, θ_g is the parameters of the generator, and x_{fake} is the generated DNA sequence.

The generator of the DCGAN model consists of several transposed convolutional layers, which progressively transform the latent vector into a structured sequence. Each of these layers applies the transformation:

$$h_{l+1} = \sigma(W_l h_l + b_l) \tag{11}$$

where:

- W_l represents learnable weights
- b_l represents bias terms
- σ represents the activation function
- h_l represents intermediate feature maps

Batch normalization is applied after convolution operations to stabilize training and accelerate convergence.

D. WGAN Critic Network

Unlike the traditional discriminator that uses a probabilistic approach to distinguish between real and fake data, the novel framework utilizes a critic network that is inspired by the Wasserstein GAN. The critic network scores the real data as well as the generated data.

$$D(x; \theta_d) \tag{12}$$

where θ_d is a symbol for the critic parameters.

The critic is designed to reward real data with high scores and punish generated data with low scores, helping the generator improve in its creation of realistic DNA sequences.

E. Wasserstein Loss Function

The Wasserstein distance is used in WGAN to measure the difference between the real data distribution P_r and the generated data distribution P_g:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \tag{13}$$

In practice, this distance is approximated using the following loss function:

$$L = \mathbb{E}_{x \sim P_r}[D(x)] - \mathbb{E}_{z \sim P_z}[D(G(z))] \tag{14}$$

The generator minimizes the critic score of generated samples:

$$\min_G -\mathbb{E}_{z \sim P_z}[D(G(z))] \tag{15}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

The critic maximizes the difference between real and generated samples:

$$\max_D \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{z\sim P_z}[D(G(z))] \tag{16}$$

To enforce the Lipschitz constraint required for Wasserstein distance, weight clipping is applied:

$$w \leftarrow clip(w, -c, c) \tag{17}$$

F. Training Stability Techniques

Training GAN models is usually challenging because of problems like vanishing gradients, unstable convergence, and mode collapse. To overcome these challenges in training GAN models, various techniques that stabilize training are integrated into the proposed framework.

Firstly, the proposed framework uses the Wasserstein loss function instead of the common binary cross-entropy loss used in standard GANs.

Secondly, weight clipping is used in the proposed framework to satisfy the Lipschitz constraint of the Wasserstein distance:

$$w \leftarrow clip(w, -c, c) \tag{18}$$

where c is a predefined clipping threshold.

Third, the batch normalization technique is applied in the generator network. This stabilizes the gradient flow and speeds up the convergence of the generator.

Finally, the critic network is updated several times for every update of the generator network. This ensures that the critic network estimates the Wasserstein distance correctly before the generator makes any update.

G. Adversarial Training Process

Training is performed in an iterative fashion, where the critic is trained several times before training the generator. This is done for stable training. During each iteration of training, the critic network is trained to differentiate between the real DNA sequences from the dataset and the synthetic DNA sequences produced by the generator. This helps the critic approximate the Wasserstein distance between the two data distributions.

The generator then updates its parameters based on the feedback from the critic, with the aim of creating sequences that are more and more similar to real DNA samples. Updating the critic more often helps provide more accurate gradients, which in turn ensures the stability of the adversarial training and prevents problems such as mode collapse.The overall adversarial training procedure of the proposed DCGAN–WGAN model is summarized in Algorithm 1.

H. Synthetic DNA Sequence Generation

Once the training process converges, the generator is employed to generate new synthetic DNA sequences. A random noise vector z is sampled from the predefined noise distribution and is used as input to the generator network G.

Algorithm 1 Adversarial Training of DCGAN–WGAN

Require: Real DNA dataset D
Ensure: Trained generator model
1: Convert DNA sequences to one-hot encoded matrices
2: Initialize generator G and critic D
3: **for** each training epoch **do**
4: Sample mini-batch of real DNA sequences $x \sim P_r$
5: Sample random noise vector $z \sim P_z(z)$
6: Generate synthetic sequences $x_{fake} = G(z)$
7: Compute critic loss using Wasserstein objective
8: Update critic parameters
9: Sample new noise vector z
10: Generate sequences $x_{fake} = G(z)$
11: Compute generator loss
12: Update generator parameters
13: **end for**
=0

The generator takes the random noise vector and generates an output that is similar to the patterns observed in the real DNA sequences that the generator was exposed to during the training process. The generated output is then mapped to the nucleotide representation to obtain the final DNA sequences.

This process is repeated several times to obtain the final set of synthetic DNA sequences, which can be used for further analysis and experimentation.

I. Architecture Overview

The proposed architecture is a combination of the strengths of DCGAN and WGAN for the generation of realistic synthetic DNA sequences. The generation of synthetic DNA sequences starts with the generation of a random noise vector z from a latent distribution, which is given as input to the DCGAN-based generator.

The generated sequences are then passed through the discriminator (critic) of the WGAN, which comprises convolutional layers and LeakyReLU activation functions. This critic is used to evaluate the quality of the generated data. The critic compares the generated sequences with the real DNA sequences from the dataset, which have been represented using the one-hot representation.

The training objective is based on the Wasserstein loss function, defined as the distance between the real and generated data distributions. This improves the training stability and results in better synthetic DNA generation. This is achieved by maximizing the distance in the discriminator, while minimizing it in the generator. Through this adversarial training, the model learns to gradually generate biologically plausible DNA sequences that closely resemble real genomic data.

IV. Results and Evaluation

For the evaluation of the effectiveness of the proposed model based on the combination of the DC-GAN and WGAN architectures, experiments were performed on the encoded DNA sequence data. The generated sequences were compared

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

to the actual sequences using various evaluation criteria such as sequence similarity, precision, recall, and F1-score. As depicted in Table I, the highest similarity score of 0.91 is obtained using the proposed model compared to the standard GAN architectures.

TABLE I: Performance Comparison of GAN Models

Model	Similarity	Precision	Recall	F1 Score
Standard GAN	0.78	0.74	0.72	0.73
DCGAN	0.83	0.80	0.78	0.79
WGAN	0.86	0.84	0.81	0.82
Proposed Model	**0.91**	**0.88**	**0.86**	**0.87**

The results have shown that the proposed architecture offers better performance with respect to the traditional GAN models. The proposed architecture is a hybrid of DCGAN and WGAN models. The proposed architecture offers better stability and quality with respect to the generated DNA sequences.

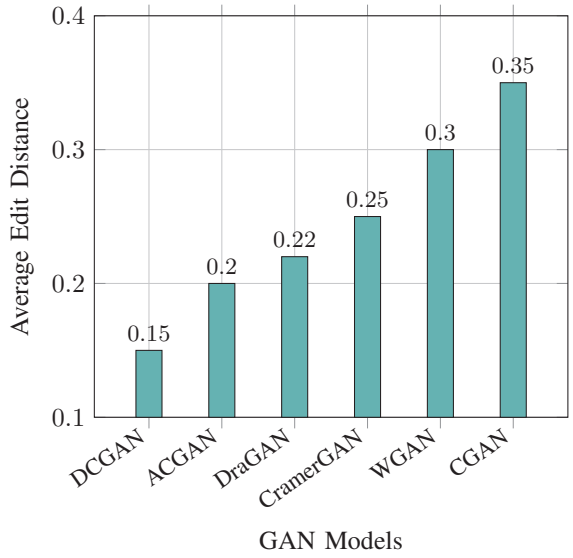

Fig. 1: Average edit distance comparison

Figure 1 illustrates the comparison of the average edit distance for various GAN architectures that are applied for synthetic DNA sequence generation. The edit distance is the measure of similarity between the generated synthetic DNA sequences and the real DNA sequences. The lower the edit distance, the higher the similarity between the generated synthetic DNA sequences and the real DNA sequences.

As seen from the figure, the DCGAN architecture has the least edit distance of 0.15, thereby showing that the synthetic DNA sequences generated are closer to the real DNA sequences. On the other hand, the WGAN and the CGAN architectures have higher edit distances, showing that the synthetic DNA sequences generated are quite different from the real DNA sequences. The gradual increase in the edit distance for the various GAN architectures shows the variations that exist in the performance of the various architectures.

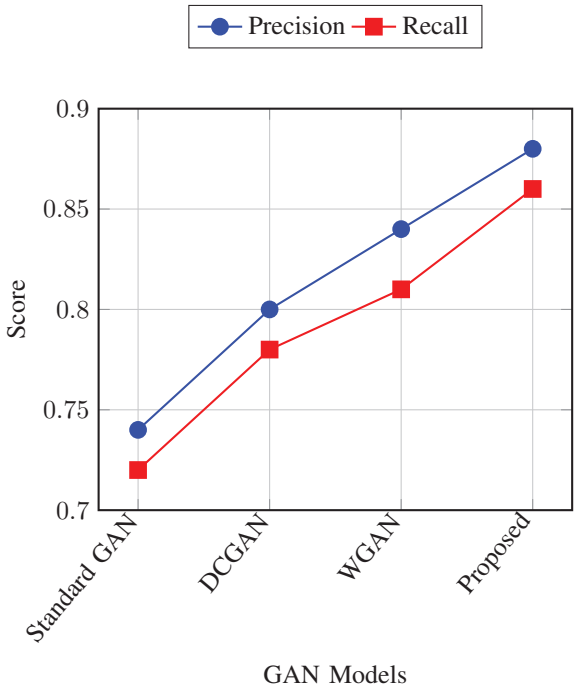

Fig. 2: Comparison of precision and recall scores for different GAN models.

$$ED(s_1, s_2) = \min(ins + del + sub) \tag{19}$$

where:

- ins : number of insertion operations
- del : number of deletion operations
- sub : number of substitution operations

As depicted in Eq. (19) the edit distance is a measure that indicates the minimum number of operations that must be performed to transform a given DNA sequence into another sequence. In this regard, the operations that can be performed include the insertion, deletion, and substitution of nucleotides. The lower the edit distance is, the more similar the sequences will be.

Figure 2 presents the comparison of precision and recall scores of various GAN models. It is observed that the proposed model has the highest precision and recall values, showing better performance in terms of generating accurate and reliable DNA sequences compared to Standard GAN, DCGAN, and WGAN.

V. Conclusion

This paper proposes a hybrid generative model that utilizes the DCGAN and WGAN architectures for the generation of synthetic DNA sequences. The proposed model utilizes the capability of the DCGAN model to learn features using the convolutional neural network and the stable optimization mechanism of the WGAN model.The proposed hybrid model has achieved better results compared to the traditional GAN model in terms of sequence similarity, precision, and recall,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

as observed from the experimental results. The generated synthetic DNA sequences show promising results with respect to the characteristics of the biological data.

The proposed approach can be utilized for various bioinformatics applications, including data augmentation, genomic analysis, and privacy-preserving sharing of biological data.

VI. Future Work

The proposed architecture of DCGAN-WGAN [21] shows promising results in terms of synthetic DNA sequence generation there are still many avenues that can be explored in the future. One such direction that can be explored is the incorporation of conditional generative models that can help in the generation of DNA sequences based on certain attributes. This would help in the controlled generation of data that can be used in genomic studies.

Another potential way to go is to leverage the power of advanced sequence-aware architectures, like those that use attention mechanisms or transformer architectures [22], [23]. The biological sequences tend to exhibit intricate patterns over long stretches, and it is expected that using these architectures could improve the biological relevance [24], [25] of the generated data.

Future studies could also investigate the application of the proposed generative framework on larger and more diverse genomic data sets in order to assess the scalability and generalization potential [26] of the proposed approach. Moreover, incorporating domain-specific constraints during training could potentially improve the biological validity of generated sequences.

Lastly, the model that is being proposed can be extended to facilitate the development of hybrid generative models that incorporate generative adversarial networks with other deep learning techniques [27]–[30] for the generation of synthetic biological data. This will likely result in the development of even more robust synthetic data that could be used for various applications such as genomic predictions, bioinformatics, and even the sharing of biological data.

References

[1] Z. Shi, F. Xie, B. Wang, and W. Yang, "The aep for hidden markov tree models," *IEEE Transactions on Information Theory*, 2026.

[2] J. Lian, W. Wang, and P. Jia, "Almost sure convergence of nonhomogeneous switching markov chains with absorbing states: A graph-based approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2026.

[3] S. Bouwhuis, C. Pauls, M. Garnier-Villarreal, and D. Pavlopoulos, "The interplay between job demands & resources and mental health: a novel approach using hidden markov models," *Journal of Behavioral Medicine*, pp. 1–19, 2026.

[4] Z. Xing, T. Ye, Y. Yang, D. Cai, B. Gai, X.-J. Wu, F. Gao, and L. Zhu, "Segmamba-v2: Long-range sequential modeling mamba for general 3d medical image segmentation," *IEEE Transactions on Medical Imaging*, 2025.

[5] V. Kalal and R. Patel, "A comprehensive feature analysis using hidden markov model for genomic sequence pattern identification," *Computer Methods and Programs in Biomedicine*, 2024.

[6] Y. Ma and H. Zhang, "The hidden markov model and its applications in bioinformatics," *Bioinformatics and Biology Insights*, 2025.

[7] K. H. Choo and J. Lee, "Exploring hidden markov models in the context of genetic and neurological disorder analysis," *Advances in Computational Medicine*, 2024.

[8] N. Alanazi, R. Aljeraiwi, M. Almutairi, and A. N. Alodhayb, "A survey on recent progress on monte carlo simulation methods in radiation detection," *Journal of Radiation Research and Applied Sciences*, vol. 19, no. 1, p. 102146, 2026.

[9] C. Costa, É. Pereira, S. Costa, P. M. Ferreira, A. Amorim, L. Prieto, and N. Pinto, "Stutter modeling in probabilistic genotyping for forensic dna analysis: A casework-driven assessment," *Genes*, vol. 16, no. 9, p. 1053, 2025.

[10] Z. Linzhou, L. Xuewen, X. Wenwu, M. Zhangfeng, X. Baiping, and J. Hao, "Uav-to-ground channel modeling:(quasi-) closed-form channel statistics and manual parameter estimation," *China Communications*, vol. 23, no. 1, pp. 47–66, 2026.

[11] M. I. Y, M. Y. A, P. Privietha, and P. P. Jemima, "Deep convolutional gan for realistic image generation using cifar-10," in *2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2025.

[12] X. Li, Y. Zhang, and H. Chen, "Evaluating the impact of input noise and erp-based penalties for eeg generation using wgan-gp," *Computers in Biology and Medicine*, 2025.

[13] A. I. Sehsah, M. Hassan, and M. Elhoseny, "Tp-vwgan: Hybrid variational autoencoder and wasserstein gan for protein structure generation," *Scientific Reports*, 2025.

[14] J. T. Mtetwa and E. Chikodza, "Adaptive gradient penalty for wasserstein generative adversarial networks," *Mathematics*, vol. 13, no. 16, p. 2651, 2025.

[15] Y. Zhang and X. Liu, "Empirical study of wgan and wgan-gp for enhanced data generation," in *Proceedings of the International Conference on Artificial Intelligence and Computational Engineering*. IEEE, 2024.

[16] L. Tirel and P. Fernandez, "Hybrid pix2pix and wasserstein gan with gradient penalty for image denoising," *IEEE Access*, vol. 12, pp. 120 345–120 356, 2024.

[17] S. Gao, Y. Xia, X. Li, F. Cui, Q. Zhang, Q. Zou, and Z. Zhang, "Acp-esm2: enhancing anticancer peptide prediction with pre-trained protein language models," *IEEE Transactions on Computational Biology and Bioinformatics*, 2025.

[18] S. Deb, A. Das, B. Biswas, J. L. Sarkar, S. B. Khan, S. Alzahrani, and S. Rani, "Enhancing image security via block cyclic construction and dna-based lfsr," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5516–5523, 2024.

[19] T. Fukuba, S. Goto, M. K.-S. Wong, Y. Minegishi, S. Hyodo, Y. Makabe-Kobayashi, Y. Sugai, and K. Hamasaki, "Development and evaluation of automated gene collector–atgc-12s for environmental dna sample archive at aquatic environments," in *OCEANS 2022, Hampton Roads*. IEEE, 2022, pp. 1–5.

[20] E. Garzón, R. Golman, Z. Jahshan, R. Hanhan, N. Vinshtok-Melnik, M. Lanuzza, A. Teman, and L. Yavits, "Hamming distance tolerant content-addressable memory (hd-cam) for dna classification," *IEEE Access*, vol. 10, pp. 28 080–28 093, 2022.

[21] H. Mehwish, H. Shakir, M. Rashid, A. Aamir, and R. Q. Khan, "Benchmarking vanilla gan, dcgan, and wgan architectures for mri reconstruction: A quantitative analysis," *arXiv preprint arXiv:2602.00221*, 2026.

[22] M. M. Haque, S. K. Paul, R. R. Paul, N. Islam, M. A. Rashidul Hasan, and M. E. Hamid, "Improving performance of a brain tumor detection on mri images using dcgan-based data augmentation and vision transformer (vit) approach," in *GANs for Data Augmentation in Healthcare*. Springer, 2023, pp. 157–186.

[23] U. Jameel and N. Belcari, "High-fidelity ct image denoising with de-transgan: A transformer-augmented gan framework with attention mechanisms," *Bioengineering*, vol. 12, no. 12, p. 1350, 2025.

[24] N. J. Wu, M. Aquilina, B.-Z. Qian, R. Loos, I. Gonzalez-Garcia, C. C. Santini, and K. E. Dunn, "The application of nanotechnology for quantification of circulating tumour dna in liquid biopsies: a systematic review," *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 499–513, 2022.

[25] L. Y. Venkataramana, S. Kamath, and S. L. Narayanan, "Diverse breast-omics: Ethnic-specific genomic framework for risk prediction in breast cancer," in *2026 9th International Conference on Computational Intelligence in Data Science (ICCIDS)*. IEEE, 2026, pp. 1–6.

[26] J. Wang, B. Wang, S. Zhou, B. Cao, W. Li, and P. Zheng, "Dnacse: Enhancing genomic llms with contrastive learning for dna barcode identification," *Journal of Chemical Information and Modeling*, vol. 66, no. 2, pp. 976–993, 2026.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 04, ICTEM 2.0 (2026)

[27] R. Mahmood, J. Lucas, D. Acuna, D. Li, J. Philion, J. M. Alvarez, Z. Yu, S. Fidler, and M. T. Law, "How much more data do i need? estimating requirements for downstream tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 275–284.

[28] J. van Rhijn and T. Schied, "Monte carlo simulation of stochastic differential equations using generative adversarial networks," *Computational Statistics*, vol. 38, pp. 1423–1440, 2023.

[29] L. Obaid, K. Hamad, and S. Barakat, "Incident duration reliability assessment using monte carlo simulation and kernel density estimation," *International Journal of Transportation Science and Technology*, 2024.

[30] J. Güldenstein and M. Schubert, "Deep learning based measurement model for monte carlo self-localization," in *Proceedings of the International Conference on Intelligent Robotics*. Springer, 2024.