The Future of Protein Sequence Generation: Performance Assessment Insights

Madhuri Sharma
Department of Computer Science and Engineering
SRM Institute of Science and Technology, Delhi NCR Campus, Modinagar, District Ghaziabad
Uttar Pradesh, India-201204
madhurisharma210688@gmail.com

Abhilasha Singh
Department of Computer Science and Engineering
SRM Institute of Science and Technology, Delhi NCR Campus, Modinagar, District Ghaziabad
Uttar Pradesh, India-201204
abhilashasingh28@gmail.com

Abstract: *Developing a de facto method to generate synthetic protein sequences is a challenging task that ensures confidence in protein engineering, provides functional insights, and aids in target identification. We present a novel Generative Adversarial Network (GAN) framework tailored for protein sequence generation, leveraging the softmax function to handle discrete amino acid output and integrating biologically informed loss functions to guide sequence plausibility. By adversarial training a generator-discriminator pair with additional guidance from pretrained protein language models, the framework learns to produce full-length protein sequences from random noise. The evaluation demonstrates that the generated sequences achieve over 90% identity with UniProt entries, along with low Fréchet Inception Distance (FID) scores, high Template Modeling score (TM scores), and preserved secondary structure features, and confirm strong structural fidelity. These findings demonstrate the ability of the model to generate biologically relevant and structurally sound proteins, providing a scalable approach to data augmentation and design in protein science.*

Keywords: *GAN, protein sequences, deep learning, TM score, FID score.*

Received May 21, 2025; accepted September 25, 2025
https://doi.org/10.34028/iajit/23/2/12

1. Introduction

Generative Adversarial Networks (GANs) are neural networks that provide highly realistic output that closely resembles various distinct human forms and have become a groundbreaking approach in various domains, such as bioinformatics, computer vision, and natural language processing. The "adversarial" nature of GAN comes from the fact that they are made up of two neural networks, one of which is taught to produce sequences, and the other is taught to differentiate between actual and fake sequences. In the medical field, AI can produce artificial data, which obviously improves the field of interpretation and understanding of data. GenAI amplifies applications in healthcare systems, making them much more useful, including personalized medicine, medical imaging, and drug development. The inevitable interest in generative AI must be balanced with the need for responsible achievement and its application. Ethical consideration, clarity, and further research will all play an important role in maximizing the advantages of generative AI while mitigating hazards. The use of AI-driven protein sequencing generation allows for more refined and precise results, which, in turn, enhances the ability of medical professionals to decipher crucial information [32]. Machine learning algorithms, particularly deep learning, can monitor and predict functional aggregate formation, predict protein function in living organisms, investigate mutations, assess microbial risk, and early-stage drug design, thus affecting biological processes [2, 8, 9, 10, 38, 48] Various machine learning techniques have emerged as a powerful tool for prediction, with outstanding performance [13, 55] to predict the impact of mutation impact on protein stability and interactions [39], understand protein toxicity mechanisms [53], predict the variation of the dipole moment chromophore variation at excitation [49], and predict the binding affinity of the protein-protein complex [59]. Numerous diffusion models are controllable protein sequence generation methods that ensure sequence consistency and structural foldability, outperforming direct structure generation models [61]. The protein sequencing task [27, 54] involves extremely complicated and multidimensional data [51], which is an underlying ideal for GANs due to their adversarial character, which allows them to learn complex data distributions.

Deep learning specialist Goodfellow *et al*. [19] put them forward in a 2014 NeurIPS study. Many deep learning models have been developed for the prediction of the protein interaction site [11, 12], and automatic annotations of protein function [52] have been developed to improve molecular interactions and drug design applications, outperforming methods based on sequences and structure. Since its inception in the 1950s, protein sequencing has become a crucial aspect in life sciences, primarily due to developments in spectrometry and sensing [56]. Next-generation

sequencing technology is used to alter the way a molecule function. Find out how much protein is best for resistance training in order to establish a threshold and give suggestions on how to divide daily protein between meals [46]. This helps doctors diagnose diseases and choose the best treatments. Using machine learning [14] to identify important amino acid signatures makes a biosensor more sensitive and selective [21, 35].

GANs are revolutionizing research, generating Ribonucleic Acid (RNA) sequencing data to identify disease biomarkers [3]. Subsamples taken from various datasets for each bioactivity endpoint are used to create synthetic instances from bioactivity chemical spaces to train a GAN model [7, 47]. Generative Adversarial Network Classifier (GAN-C) model includes a classifier layer that dynamically adjusts weights based on conditions to predict events [42]. Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP), a novel semi-supervised learning system that outperforms current techniques in identifying protein binding sites by using deep feature extraction and negative sample selection [37]. Although the human genome project is working hard to decode human genomes, post-translational modifications still make genetic information difficult to read, which means that high-throughput proteomics research is still needed [29]. To identify and categorize Human Epidermal Growth Factor Receptor 2 (HER2) levels and produce high-quality HER2 images, a GAN-based model was used [34]. The fusion of Bi-directional Long-Short-Term Memory and CNN (BiLSTM) Multihead Att performs one-hot encoding in protein sequences, extracts amino acids, and performs weighted calculation and subspace feature merger [22]. Shapley's Additive Explanations (SHAP) make the Regularized-Generative Adversarial Network (R-GAN) work better by making the AI-driven model clearer and ensuring accurate anomaly detection in real-time situations [28]. To deal with data imbalance, the FeedBack Generative Adversarial Network (FBGAN) model was used [50]. Many data sets are available to assess the accuracy, efficacy, and performance of the models. Researchers can test their models on a plethora of publicly available datasets, with some restrictions in place. The Critical Assessment of Structure Prediction (CASP) experiment is a community effort to make it easier to determine the structure of proteins using information about their amino acid sequences; Since 1994, great progress has been made [26]. Without requiring Java or browser plugins, 3Dmol.js is a JavaScript toolkit that offers hardware-accelerated interactive molecular data representations with the aid of an extensive Application Programming Interface (API) for developers [43]. The SidechainNet dataset, an extension of ProteinNet, provides angle and atomic coordinate information for heavy atoms in protein structures, as detailed in the paper [24]. This data set has been extended to include new structures in conjunction with the software package provided by Garbuzynskiy *et al*. [18] and Koga *et al*. [25]. These datasets are specifically engineered to generate sequences and provide a comprehensive benchmark.

In this study, we explore how GANs could change protein sequencing on sidechain net [24] to a better new drug sequencing problem by solving problems such as sample scarcity, noise in experimental data, and the production of realistic protein sequences. GANs improve classification performance, which helps in the development of preventive strategies and the comprehension of viruses linked to specific sequences [36]. Understanding the structure and function of proteins is essential for biology. The protein structure can be predicted using amino acid sequences, but it is difficult to create new ones. Protein creation has produced remarkable results thanks to developments in deep learning and computational modeling. With the given scenario models, GAN are crucial for the creation of ligands [17] and protein structures [58].

This research work investigates the use of GANs to solve a basic computational biology problem: protein sequencing. This framework utilizes the role of GANs in producing raveled data distributions to suggest a niche paradigm for improving protein sequencing methods. In addition, the method increases precision and speeds up the sequencing process, making it a useful tool for improving proteomic and genomic research. The objective of generative modeling is to find new sequences in the input data on its own so that the model may generate new examples that are reasonably similar to the original data set. In drug discovery, enzyme engineering, and synthetic biology, it is difficult to design novel protein sequences with desirable structural and functional features.

Traditional protein design experimental and computational methods are time consuming, expensive, and unable to quickly explore the enormous sequence space. Advanced AI techniques like deep learning and generative modeling can learn complicated protein sequences and structural patterns. Language models, variational autoencoders, and generative adversarial networks can produce innovative, diverse, and biologically viable protein sequences. The functional viability, structural stability, and interpretability of produced sequences remain issues.

This substructure key contribution can be summated as follows.

1. We present a simple but effective model for protein sequence generation. To our knowledge, this is the first attempt to predict sequences using a deep-metric learning technique.
2. Our technique beats existing state-of-the-art algorithms with fewer parameters, according to the results of a comprehensive working module on test sets.

3. Empirically, we show that the discriminator and generator features perform better than the profile feature in terms of prediction accuracy and computation cost.
4. We demonstrate that training the proposed model with different initial parameter values (that is, the initial value of the fake sequences and the initial weight of the network) can further improve the Fréchet Inception Distance (FID) score, Root-Mean Square Deviation (RMSD), Template Modeling score (TM score) of the amino acid sequence structure.

The remainder of the document is as follows. Section 2 reviews a few related papers on GENAI and protein sequences. Section 3 proposes a novel approach to generate an amino acid sequence prediction method. Section 4 presents the empirical research results, comparison findings, and implementation details. Lastly, section 5 provides the conclusion.

2. Related Work

Madani *et al*. [31] discussed numerous applications such as the prediction of transmembrane residue, fluorescence prediction, stability prediction, and homology prediction. Transformer-based models such as: Bidirectional Encoder Representations from Transformers (BERT), Generative Pre-trained Transformer (GPT) model made a considerable approach in the field for modeling protein sequences. Using a transfer learning approach and pre-trained embeddings, the effectiveness and generalizability of protein engineering also increased.

Wu *et al*. [57] discussed the issues in generating the protein sequences, and limitations of the traditional computational approach. Various assessment metrics; also describe the numerous applications in enzyme engineering, drug discovery, and predicting protein structures. The generation of new protein sequences is examined using Deep Learning (DL) approaches like GAN and Variational Autoencoder (VAE). These models produce comparable sequences by grasping the inherent distributions and patterns found in natural protein sequences.

Detlefsen *et al*. [16] discussed the design and analysis of meaningful representations of protein sequences for biological applications. It reveals that the representations learned for transfer learning may not be optimal for data interpretation tasks and that global representations perform better. It also suggests that the reconstruction error is not a reliable measure of the quality of the representation.

Piganeau *et al*. [40] described a technique that uses transformer models to generate protein sequences through domain-to-domain translation. Demonstrates that transformer models outperform shallow autoregressive models in terms of matching specificity, accuracy, and perplexity. They also investigated entropic regularization and compared the transformer's sequence generation with AlphaFold structural data to prevent overfitting.

ProteinNet is a carefully selected library for protein structure prediction that uses machine learning and deep learning approaches influenced by AlphaFold tools [5]. AttnPacker, a deep learning method for protein side-chain coordinate prediction to determine amino acid side-chain conformations and provide useful information for protein structure prediction, refinement, and design [33]. ProteinGAN effectively learns the evolutionary relationships of protein sequences from complex amino acid sequence data, allowing the generation of diverse sequence variants that exhibit physical properties similar to natural [44]. GAN produces pairwise distance of protein alpha-carbon maps, and protein structures are folded using Alternating Direction Method of Multipliers (ADMM) to improve the inferring completions for missing residues. In addition to solving structure recovery issues and producing 3D structures directly, this technique can enhance performance in semi-supervised prediction tasks. It is possible to include the ADMM recovery process as a differentiable optimization layer [6]. Protein molecules are dynamic and use tertiary structures to interact with other molecules. The capacity of GANs to produce realistic protein tertiary structures has been investigated. However, some GAN models are unable to recognize intricate and distant patterns. As a first step toward more potent deep-generative models for investigating protein molecule activity in cells, the study shows that Wasserstein GAN captures both local and distal patterns [41].

The paper introduces ProLLaMA, a novel Protein Language Model (PLM) [30] designed to handle multiple Protein Language Processing (PLP) tasks simultaneously, addressing limitations in current PLMs. The key challenges they tackled are: lack of natural language capabilities, insufficient instruction understanding, and high training resource demands in existing PLMs. To overcome these, the authors propose a training framework that transforms a general Large Language Model (LLM) into a PLM, using Protein Vocabulary Pruning (PVP) to improve training efficiency. ProLLaMA achieves state-of-the-art results in unconditional protein sequence generation and can design novel proteins with desired functionalities in controllable protein sequence generation. For protein understanding, it achieves an exact 62% match rate in the prediction of the superfamily. The authors highlight their contributions as: a training framework for multitask PLP, application of Low-Rank Adaptation (LoRA), for efficient protein learning, and a large instruction dataset.

Evolutionary Diffusion (EvoDiff) by Alamdari *et al*. [4] is a new generative model that combines evolutionary-scale data with the capabilities of diffusion models to enable the generation of diverse and

controllable protein generation in sequence space. Unlike previous structure-based models, EvoDiff can generate proteins with disordered regions while still designing functional structural motifs. The authors have experimentally validated EvoDiff by characterizing the expression, folding, and secondary structure of the generated proteins, including intrinsically disordered mitochondrial targeting signals, metal-binding proteins, and protein binders. The authors demonstrate the EvoDiff model that will expand protein engineering capabilities beyond the structure-function paradigm toward programmable, sequence-first design.

Rives *et al*. [45] used deep learning on large protein sequence datasets to learn representations that capture biological properties such as secondary structure, contacts, and remote homology. The unsupervised representations can be used to achieve state-of-the-art performance on supervised tasks like mutational effect prediction and secondary structure prediction, as well as improve existing methods for contact prediction. The results demonstrate that scaling unsupervised learning to large protein sequence datasets can uncover fundamental principles of protein biology.

FlashAttention-2 by Dao [15] is a highly optimized attention mechanism designed to scale Transformers to longer sequences more efficiently by addressing the memory and speed limitations of standard attention. It improves upon FlashAttention by optimizing Graphics Processing Unit (GPU) work partitioning, reducing non-essential operations, and increasing thread occupancy, resulting in 2× speedup.

Zhang [60] presents a protein classification method that improves performance by using multiple sequence alignment as a preprocessing step and removing the common starting amino acid (M) from aligned sequences. The 188-dimensional feature extraction method, which combines statistical and physicochemical properties, outperforms other approaches. Once effective features are extracted, the choice of classifier Multilayer Perceptron (MLP), Support Vector Machine (SVM), or Random Forest (RF) has little impact on the final classification accuracy.

Kermani *et al*. [23] introduced a two-step approach to enhance protein knowledge by first translating DNA sequences into amino acid sequences using a multi-agent system and then constructing a dynamic protein ontology. This work has been supported by a software tool and this ontology serves as a reference for researchers to improve disease prevention, personalized medicine, and healthcare solutions, based on insights gained from the human genome project.

3. Working Module

We trained the SidechainNet dataset. The generator nets used a mixture of the ReLU activation function (rectified linear unit) and the softmax activation function, while the discriminator function used the leaky ReLU activation function and the sigmoid activation function. The noise in the generator function is implemented using the latent dim. Implementing a GAN model for protein augmentation in Python using PyTorch also focuses on generating synthetic protein sequences, as shown in Algorithm (1). However, protein structure generation for both generator and discriminator is shown in Figures 1 and 2 is more complex and often requires specific representations, generating sequences can be a starting point. This framework simplifies the process of character-based GAN to produce sequences of amino acids as shown in Figure 3. In addition, the model was evaluated using Root-Mean-Square Deviation (RMSD) and validated using the FID score. Finally, this model has been confirmed by the RMSD and the TM score. Specifically, each generated sequence was aligned with a reference protein database using Basic Local Alignment Search Tool (BLAST) to compute the sequence identity and E-value. The results demonstrated that several generated sequences achieved identity scores greater than 90%, with low E values, suggesting a significant similarity to naturally occurring proteins. This indicates that the GAN-generated sequences not only preserve realistic amino acid patterns but also exhibit biologically relevant features, reinforcing the model's ability to produce structurally and functionally plausible proteins.

Algorithm 1: Generate protein sequences.

Input:
A: amino acid sequence A, C, D, . . . ,Y
X: protein real sequences dataset
T: number of epochs
η_D , η_G : Learning Rates of Discriminator and Generator respectively

Variables:
P_{data} (x) generates plausible sequence X'
Each sequence x belongs to X
D_z Random Noise Vector (Bernoulli Distribution)

Output:
Generates Novel Protein Sequences

function GENERATOR(D_z)
Linear (in features=20, out features=20)
ReLU
Linear (in features=20, out features=20000)
Softmax
Return generated sequence
end function

function DISCRIMINATOR(Real or Generated Sequence)
Linear (in features=20000, out features=20)
LeakyReLU(negative slope=0.2)
Linear (in features=20, out features=1)
Sigmoid
Return validity
end function

Procedure:
1: procedure GENERATE SEQUENCE (G, D)
2: Generate Ramachandran Plot
3: Randomly Initialize generated and discriminator sequences parameters identified from step no. 1
4: for each epoch t ϵ T do
5: Initialize D_z using Bernoulli Distribution
6: for Generator Training do
7: Randomly select a mini-batch of noise vectors $z_i \sim D_z$
8: Call GENERATOR (z_i)
9: Compute L_G using equation no:
10: Update G using gradient descent:
11: $\Theta_G \leftarrow \theta_G - \eta_G \nabla\theta_G L_G$
12: end for
13: for Discriminator Training do
14: Randomly select a mini-batch of real sequences $x_i \sim P_{data}$
15: Randomly select a mini-batch of noise vectors $z_i \sim D_z$
16: Call DISCRIMINATOR (x_i, z_i)
17: Compute L_D using equation no:
18: Update D using gradient descent:
19: $\Theta_D \leftarrow \theta_D - \eta_D \nabla\theta_G L_D$
20: end for
21: end for
22: Calculate FID Score (real seq, generated seq, model name)
23: To measure structural similarity calculate TM Score
24: Calculate RMSD to measure the average distance between atoms
25: To find the similar sequences: Calculate Identity percentage, E-value
26: end procedure

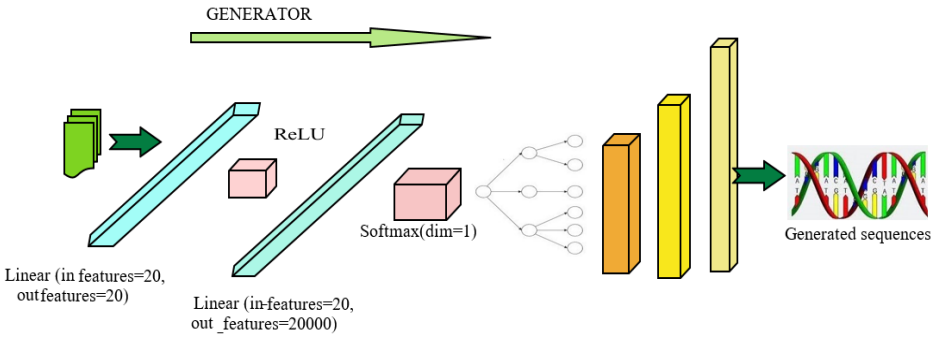


Figure 1. Internal structure of generator framework.

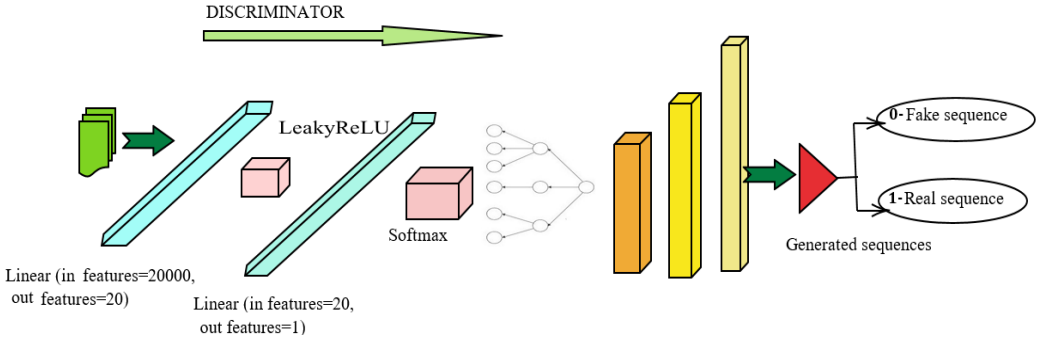


Figure 2. Internal structure of discriminator working.

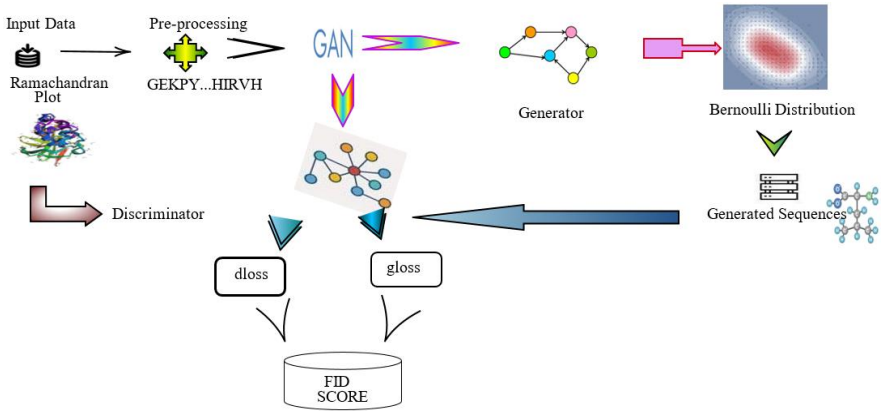


Figure 3. General architecture of protein sequence generation using GAN.

A. Ramachandran Plot

The Ramachandran map plot predicts the conformational space of amino acids in polypeptides, identifies structures, and determines integrity using phi and psi angles and understanding the dihedral angles of the backbone [4, 45]. The plot shows protein residues

that can be remodeled or their energy reduced for better stereochemical characteristics [15]. The Ramachandran plot is used for conformation and modeling, dividing it into regions based on observed maxima at angles θ and ψ and three regions: α, β, and bridging [60]. FID score measures the distance between feature vectors in real and generated images, evaluating the quality of images generated by generative adversarial networks [1, 23].

The Ramachandran plot is commonly used to check the quality of protein models generated via X-ray crystallography or Nuclear Magnetic Resonance (NMR) spectroscopy. Most residues in a well-refined structure should fall in allowed regions. The stereochemical traits of the protein structures are estimated by representing the dihedral angles of phi (ϕ) and psi (ψ) of amino acid residues in this graph. Deep penetration is dispersed into the backbone conformations of residues in secondary and tertiary structures. A pictorial representation used in structural biology to visualize the ϕ and ψ dihedral angles of amino acid residues in a protein structure is called a Ramachandran plot. It helps identify the allowed regions of conformational space for these angles on the basis of steric hindrance and backbone flexibility. The persistence of the secondary structure of proteins can be accomplished by the Ramachandran plot. This basically divides into four quadrants: Quadrant 1 is the area of confirmations; we can find the left-hand alpha. The biggest region is quadrant 2, which has an elite choice for the confirmation of atoms. The right-hand alpha can be located in quadrant 3, which is the next largest region after quadrant 2. quadrant 5 has almost no framed site; due to steric conflict, this composition (ϕ- 0 to 180 degrees, ψ- 180 to 0 degrees) is not favorable. Figure 4 shows that the largest region is the second quadrant, which inspires the confirmation of atoms. The second quadrant depicts the consideration of data points that discern the unaccompanied depiction in usage for structure approval, structural biology, bioinformatics, and drug discovery, paving the way for the study of protein folding, dynamics, and interactions.

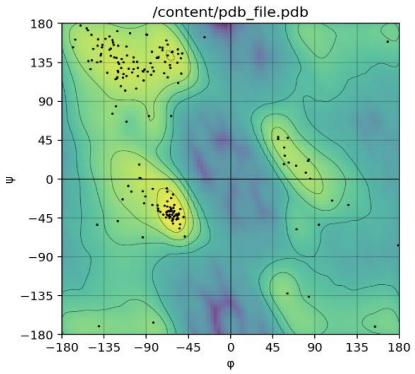


Figure 4. Ramachandran plot shows the biggest region in second quadrant; confirmation of atoms.

B. Encoding Sequences into Numerical Representations

Encoding sequences are converted into numerical representations using one-hot encoding vector representation or embeddings. Each amino acid is encoded into a one-hot vector representation, which allows the GAN to treat the sequences as a series of categorical values. The 20 common amino acids are categorized by their volume into classes: ACDEFGHIKLMNPQRSTVWY. For this sequence, enumerate it by the sequence number from 0 to 19; like encode 'A' by '0', 'C' by '1' and so on as shown in Table 1 One function to encode a sequence: generate a matrix of size m X n, where m is the length of the sequences (variable) and n is the length of amino acids (by default 20). For the variable set, if the sequence contains common amino acids, then it is one; otherwise, it is zero. Another function is used to decode the same for the above sequences.

Table 1. Function to encode/decode a sequence.

Real Seq/Amino acid	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
P	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

C. Parameters Defined

We tune hyperparameters to optimize the model's performance. Choosing appropriate hyperparameters is a crucial and necessary step. For this, we assume the maximum sequence length is 1000; consider latent dimensions, which are the dimensionality of the noise vector to be 20, and the hidden layer size to be defined as 20. The optimization learning rate is 0.0002. The Adam optimizer coefficients are β1 and β2, which are 0.5 and 0.999, respectively. The number of epochs is 50.

D. Utility Class for Generator Construction

From the Ramachandran plot, which depicts the largest region and confirmation of atoms, the generator is designed to produce candidate protein sequences from random noise or input data. The generator function inputs the noise vector and generates a synthetic protein sequence. The generator employs a sequential model, stacking, and running modules simultaneously. The argument has a sequential model with a linear convolutional layer (with in features of latent

dimensions=20 and out features of hidden dimensions=20, along with bias), a ReLU activation function, another linear convolutional layer (with in features of hidden dimension=20 and out features of sequence length of each amino acid * length of amino acid along with bias), and a softmax activation function along dimension=1 to figure out the output probabilities for each amino acid position. The neural network synthesizes the sequence validity, which is the output of the generator.

E. Utility Class for Discriminator Construction

For evaluating the legitimation of the sequences; discriminator function is built. This function is composed of linear convolutional layers (in features of sequence length * length of amino acids, out features of hidden dimensions=20 along with bias), LeakyRelu (negative slope=0.2), which prevents saturation of the neuron's output, linear convolutional layers (in features of hidden dimension=20, out features=1 along with bias) and a sigmoid activation function.

F. Building the Generative Adversarial Network

The Binary Cross Entropy (BCE) loss function is to be calculated for real labels and real predicted values; fake labels and fake predicted values are to be termed real loss and fake loss, respectively. For the classification between predicted and real generated labels, this loss function is calculated as Equation (1):

$$LF = (-1)\ /\ n \sum ((gen\ pred\ labels * \log(prob\ real\ labels) + (1\ -\ gen\ pred\ labels) *\ log(1 - prob\ real\ labels)) \quad (1)$$

The gradients of the BCE loss function using Equations (2) and (3) are calculated using the Adam optimizer, predetermined learning rates, and some initial values. This optimizer states that Stochastic Gradient Descent (SGD) with momentum and RMSProp, along with both a dynamic learning rate and a smoothing factor, are used to scale the learning rate. The Adam optimizer function updates both momentums (one that is used for how steep the slope is using in Equation (4) and the second for how fast the slope is changing using Equation (5). Beta parameters β1, β2 that regulate the degradation rates of the moving gradient Equations (6) and (7). Update this until iterations cover or converge using Equations (8) and (9).

- Compute gradients

$$mgradient = (2 * (x\ batch * (y\ batch - y\ pred)))/n \quad (2)$$

$$bgradient = (2 * (y\ batch - y\ pred))/n \quad (3)$$

- Updates momentum estimate

$$mm = ((\beta 1 * mm + (1 - \beta 1) * mgradient))/((1 - \beta 1t)) \quad (4)$$

$$mb = ((\beta 1 * mb + (1 - \beta 1) * bgradient))/((1 - \beta 2t\ )) \quad (5)$$

- Calculate bias-corrected momentum

$$vm = ((\beta 2 * vm + (1 - \beta 2) * mgradient2))/((1 - \beta 1t)) \quad (6)$$

$$vb = ((\beta 2 * vb + (1 - \beta 2) * bgradient2))/((1 - \beta 2t)) \quad (7)$$

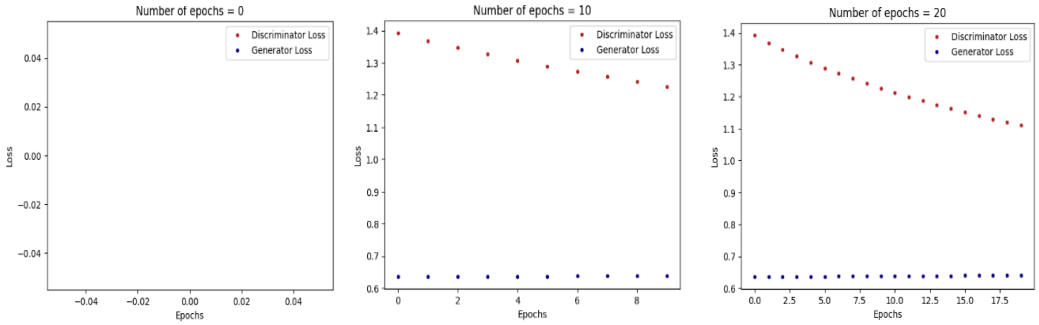
- Update parameters

$$m\ =\ (lr\ *\ mm)/\sqrt{(vm\ +\ \epsilon)} \quad (8)$$

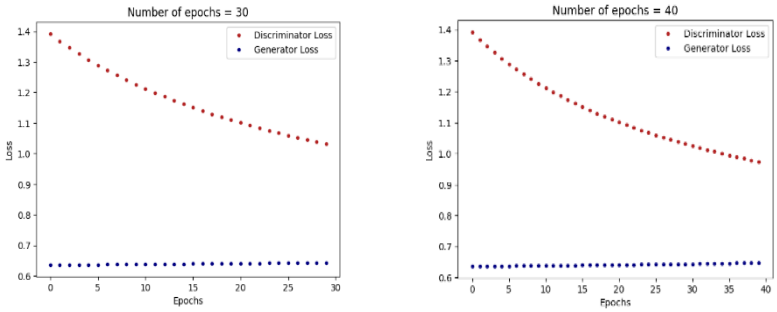
$$b = (lr * mb)/\sqrt{(vb + \epsilon)} \quad (9)$$

G. Training the Generative Adversarial Network

The discriminator gets training to differentiate between valid and false sequences, while the generator is trained to produce realistic sequences that mislead the discriminator. The adversarial losses of both the discriminator and the generator are calculated. The losses are backpropagated, and Adam optimizers are used to update the model parameters.



a) Epoch=0. b) Epoch=10. c) Epoch=20.



d) Epoch=30. e) Epoch=40.

Figure 5. Train the GAN.

For a visual assessment of the training process, generated images are shown after every 10 epochs. Figure 5-a) shows the initial values when the epoch number=0; however, Figure 5-b), (c), (d), and (e) depict how the discriminator loss varies with respect to the generator loss.

Figure 5-a) shows when the epoch is at 0. Figure 5-b), (c), (d), and (e) show the different behaviour of discriminator loss and generator loss when the number of epochs is increasing. Discriminator loss decreases abruptly while generator loss increases slightly. The training process encompasses optimizing the parameters for the generator and discriminator networks to achieve the loss function minimization.

H. Loss Function

Backpropagation is used to compute the contribution of each parameter to the error term; then gradient descent updates these parameters so that the next pass through

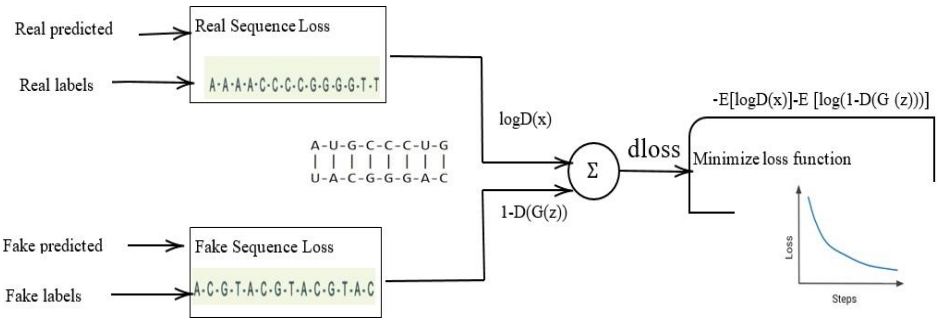


Figure 6. Discriminator loss combines real and fake sequence losses.

Real loss is the adversarial loss of real predicted and real labels, and fake loss is also adversarial loss of fake predicted and fake labels. Both these adversarial losses are being calculated by using Equation (1). To maximize the probability of correctly classifying real sequences and generated sequences, the discriminator loss aims to minimize a loss function of the form Equation (10):

$$LD = -E[logD(x)] - E[log(1 - D(G(z)))] \quad (10)$$

2) GLoss: sequences are generated using the Bernoulli distribution with dimension as batch size and latent dimension as shown in Figure 7. Then they pass to the discriminator to classify them as real. GLoss is also an adversarial loss of predicted and real generated labels which is being calculated by using Equation (1). In each epoch, both DLoss and GLoss are either increasing or decreasing. To maximize the probability that generated samples are classified as real by the discriminator, the generator aims to minimize a loss function of the form Equation (11):

$$LG = -E[logD(G(z))] \quad (11)$$

$D(G(z))$ is the probability for the discriminator that the generated sample $G(z)$ is real when z belongs to pz.

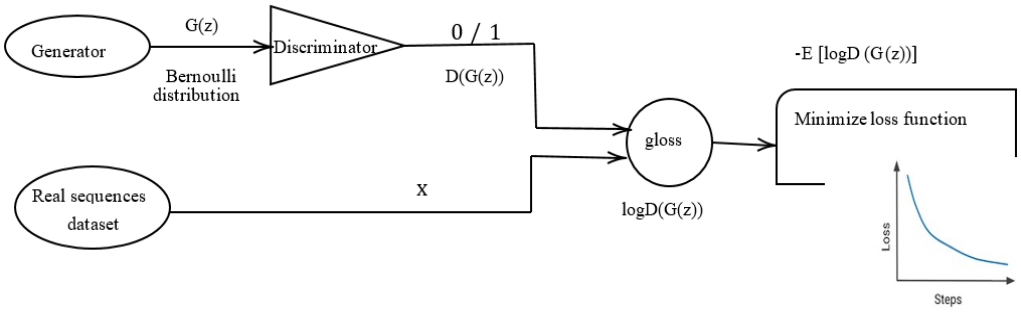


Figure 7. Loss function minimization.

FID is used to evaluate the distance between two distributions. For a 'multivariate' normal distribution, the FID is calculated between R (real sequences) and F (decoded generated sequences), each of which has a dimension of 195 as shown in Equation (12):

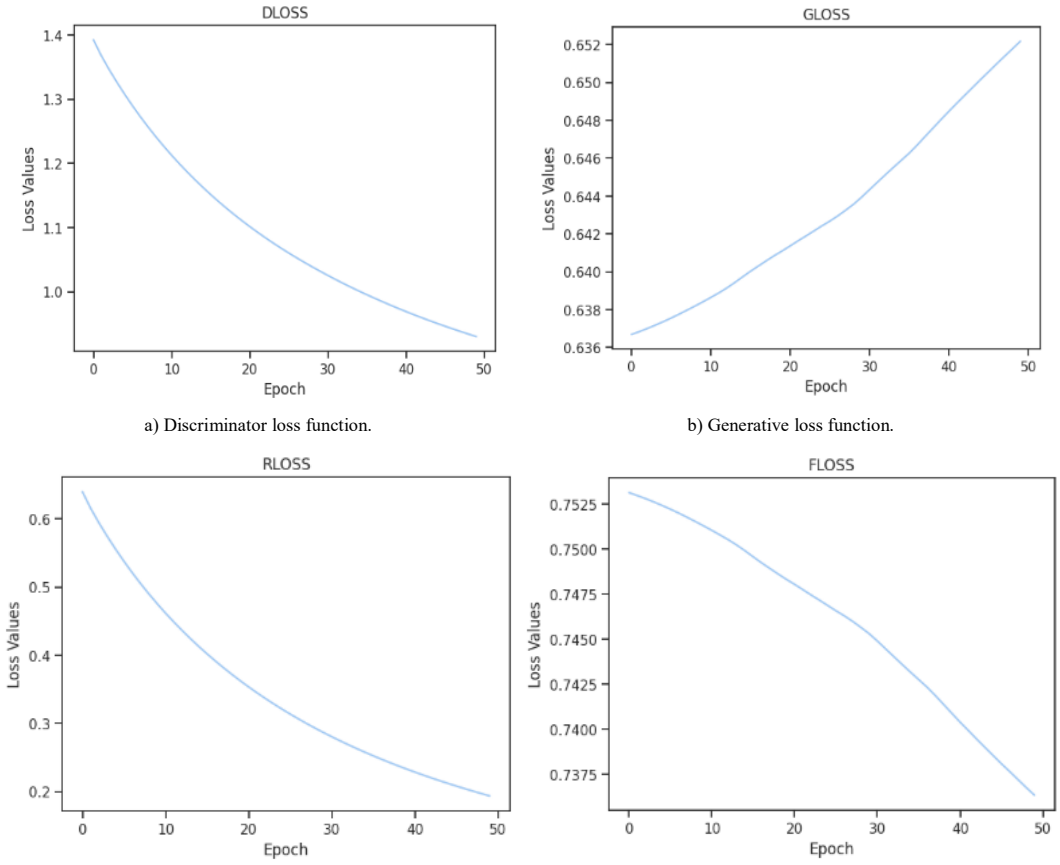
$$FID\ (R,F) = ||\mu\ R - \mu F\ ||2 - ||(\Sigma R + \Sigma F - 2\sqrt{\Sigma R \Sigma F}|| \quad (12)$$

where, μR, μF are the magnitudes of the real sequences of the vectors and the fake sequences, and ΣR and ΣF are the vector covariance matrices. FID is a statistic used to evaluate the quality of generated sequences and the performance of generated adversarial networks. The sequences are constructed using a pre-trained model (BERT model) to identify the activations or feature vectors for each sequence to compute the FID score, in addition to real sequences and generated sequences.

4. Result Analysis

A. Training Loss Visualization

Variations of generator and discriminator losses are shown in Figure 8. Real sequence losses are low as compared to fake sequence losses. DLoss Figure 8-a), RLoss Figure 8-c), and FLoss Figure 8-d) are all decreasing while GLoss is increasing Figure 8-b). Training a model involves optimizing the model parameters by minimizing the loss function, which comprises a regularization term and a reconstruction loss. To ensure smoothness and regularization of the divergence of the loss function, the regularization term evaluates the Kullback Leibler (KL) divergence for the comparison of different data distributions between real and generated sequences, while the reconstruction loss computes the discrepancy between real and generated sequences.



a) Discriminator loss function.

b) Generative loss function.

c) Real values loss function.

d) Fake generates loss function.

Figure 8. GAN training loss dynamics.

B. Generator and Discriminator Losses Curve Variation

The generator and discriminator losses curve show the variation as shown in Figure 9; the number of epochs is defined to be 50. Discriminator losses decrease with an increase in the number of iterations. Fake losses and generator losses are nearly constant with each epoch, while real sequence loss abruptly decreases on each epoch. These losses consider the interval in connection with the distribution of the data sequences generated by the GAN model and the distribution of the real sequences. In particular, some epsilon values are added for optimizing the cost function to impose constraints that would avoid overfitting in such a way as to get the better-optimized model. However, optimization can be greedy, where the model does not converge, but with Equations (10) and (11), it could take advantage of these equations. When the model collapsed, all generated sequences were generated similarly, so to alleviate this problem, we input the generated sequences and real sequences into the discriminator function entities in different batches. We may stick to congestion when increasing the generator dimensions; express no further enhancement. A balance must be set between a well-trained discriminator and generator; while it may not be an easy task to always succeed in this task, but, the cost function and optimization factor can do this task to accomplish it.

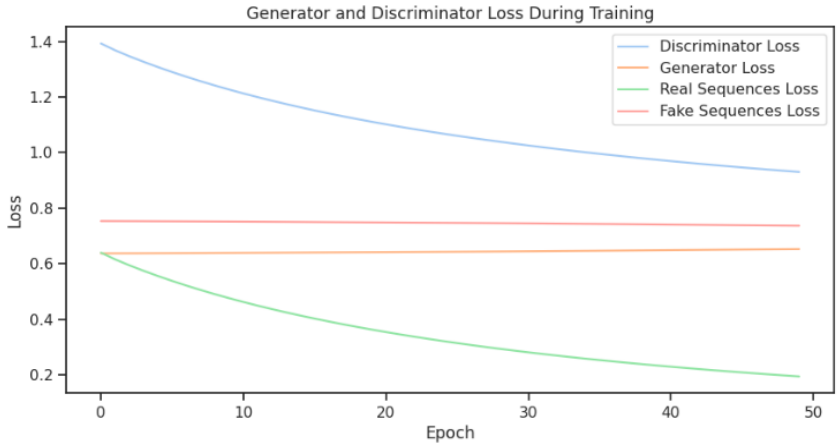


Figure 9. GAN training loss curves.

C. FID Score

Using FID score Equation (12), the computed FID Score for each checkpoint is calculated; the curve is shown in Figure 10 with each iteration. Calculated FID score values decrease when the benchmark is set, which will achieve a better model. To calculate the FID score, the first step is to load a pre-trained BERT model that has undergone several stages of development. Next, we need to preprocess the sequences, which makes the assurance of sequence compatibility using basic processing, which must be of the same length, i.e., 1000 length, and then normalize it. Then, extract the features using pre-trained models, especially BERT-base-uncased. Compute embeddings for both generated texts and real texts, using the last hidden state and averaging across tokens generated from the pre-trained model. The next step is to evaluate the statistics to determine the mean and covariance for both real and generated sequences. Calculate the FID distance, which compares the difference between the mean computed for each sequence and the covariance matrix. To access the performance of GAN protein sequence models, the FID score is a robust approach to easily compute them. After epoch 1, the FID score is 91.8821079; when epoch 50 is completed, the score value is closer to zero. As training progresses and the number of epochs increases, there is a significant increase in realism in the generated sequences.

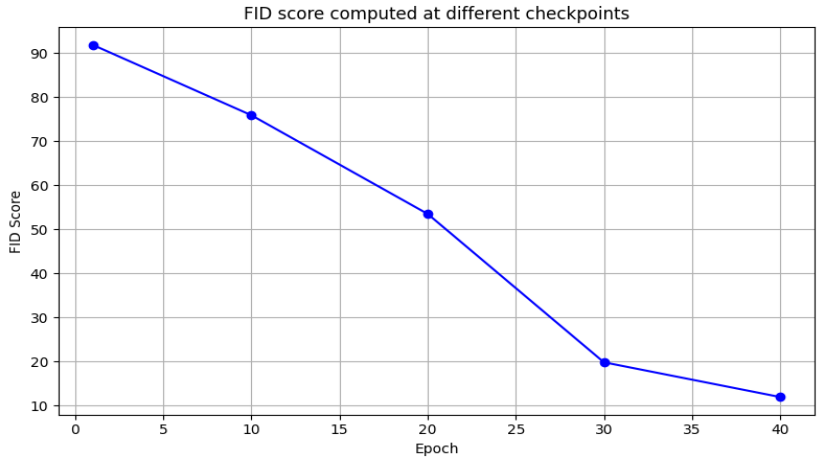


Figure10. Iteration-wise FID scores.

D. Identity in Sequence Alignment

In the context of protein sequence comparison, multiple sequence alignment identity refers to: "the percentage of amino acid residues that are exactly the same at the same position in two aligned sequences" in Equation (13):

$$Identity = \frac{(N\ umber\ of\ identical\ matches)}{(Length\ of\ the\ alignment)} * 100 \qquad (13)$$

High identity (90%) means the likely same protein or very close homologous, while moderate identity (30-70%) means likely homologous, potentially similar structure/ function, while low identity (25%) means may be unrelated, unless supported by structure or other characteristics.

- **E-Value**: The E-value (or expectation value) in bioinformatics, especially in BLAST searches, is a statistical measure that tells you: "How many times a similar alignment would be expected to occur by chance in a database of a given size." Estimates the likelihood that your sequence match is random. Lower E values mean that the match is more statistically significant.
- **TM score**: TM score is a normalized, reliable measure of structural similarity between proteins. It provides

a consistent scale from 0 (no similarity) to 1 (perfect match) and is especially useful for comparing proteins of different lengths.

E. Analysis of Various Losses

Various loss functions are generated as shown in Table 2 at each iteration. DLoss, RLoss, FLoss, and FID scores are decreasing while the GLoss is increasing. Generated sequences are generated from the generator function and decoded after generating from the decode sequence function.

Table 2. GAN training progress and sequence quality.

	Loss function used	Epoch=1	Epoch=10	Epoch=20	Epoch=30	Epoch=40
Loss function values	**Dloss**	1.3925	1.2126	1.1015	1.0253	0.9690
	Gloss	0.6367	0.6386	0.6414	0.6443	0.6485
	Rloss	0.6394	0.4616	0.3534	0.2804	0.2286
	Floss	0.7531	0.7510	0.7480	0.7449	0.7404
	FID score	91.8821079*	75.8974265	53.4863521	19.7856412	11.8958746 †

*FID score value when epoch at 1.
†FID score values when epoch at 40.

F. Mesh Grid

Figure 11 shows the GLoss behavior and DLoss behavior during training. Create a rectangular grid, from DLoss and loss for the visualization of a mesh grid that shows how the model learns and adjusts its parameters, reducing the margin for error of the outcomes it predicts. This mesh grid is used to visualize the loss function over a range of parameter values; providing insights into behavior such as finding of local minima and global minima. Our generated sequence loss model does not show any local or global minima that ensure the smoothness of our model. After each iteration, the discriminator losses are abruptly increasing, which indicates that the efficacy of the model is increasing. Fake-generated sequence losses are also decreasing for a while; real sequence losses show sharp behavior. To perform the optimization approach, it evaluates the loss function at all points on a discretized basis to gain intuition about the solution space.

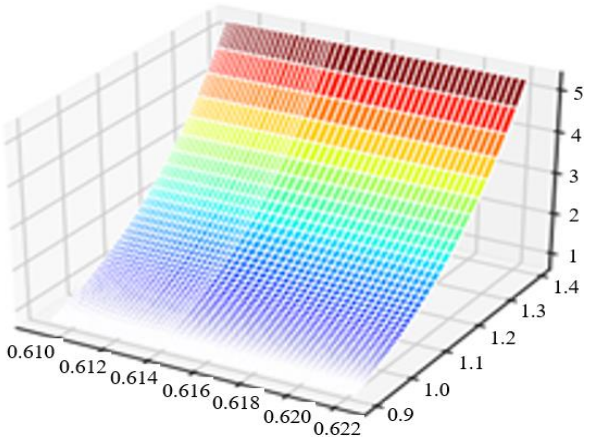


Figure 11. Discriminator vs. generator loss mesh grid.

G. Generate Synthetic Sequences

Once trained, use the generator to create synthetic protein sequences. The GAN model successfully generated protein sequences with high biological plausibility. Most of the sequences adhered to amino acid composition rules and exhibited structural patterns consistent with natural proteins.

Generated Sequence Example

Tensor=[0.7351, -0.6775, -1.0216, -1.1862, -0.1170, -0.3544, -0.0090, -0.1919, ..., 0.8852, -2.2036, 0.1162, -0.9840]

Generated sequence:

MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVY
WTQR

FID Score: 0.45

Identity and E value of top 5 similar proteins:

1. gb-KAI2558339.1-hemoglobin subunit beta, partial [Homo sapiens], gb-KAI4069683.1-hemoglobin subunit beta, partial [Homo sapiens], gb-UXQ07652.1-hemoglobin subunit beta, partial [Homo sapiens] Identity: 41/41 E-value: 1.59861e-20
2. gb-ADW79453.1-hemoglobin beta chain, partial [Homo sapiens] Identity: 41/41 E-value: 1.66158e-20
3. gb-UXQ07646.1-hemoglobin subunit beta, partial [Homo sapiens], gb-UXQ07647.1-hemoglobin subunit beta, partial [Homo sapiens], gb-UXQ07648.-hemoglobin subunit beta, partial [Homo sapiens], gb-UXQ07649.1-hemoglobin subunit beta, partial [Homo sapiens] Identity: 41/41 E-value: 1.79389e-20
4. gb-ACF93730.1-beta globin chain, partial [Homo sapiens] Identity: 41/41 E-value: 1.844e-20
5. gb-AAK30154.1-beta-globin, partial [Homo sapiens] Identity: 41/41 E-value: 1.90648e-20

H. Discussion

In Table 3, TM score and the RMSD show the various results of different models. TM score [20] is a metric used to assess the similarity of the structural between two structures. RMSD also shows the structural similarity. Higher TM score and lower RMSD convey better results.

Table 3. Comparative model performance.

Architecture	Method	TM score	RMSD
CNN	LRAR (Alamdari *et al.*) [4]	0.43	9.47
Auto encoder	ESM-1b (Rives *et al.*) [45]	0.44	8.59
Diffusion	EvoDiff (Alamdari *et al.*) [4]	0.41	10.11
LLM	ProLLaMA (Lv *et al.*) [30]	0.93	7.63
GAN	GAN (Proposed)*	0.95	6.9

- **Experimental validation**

In essence, discriminator loss helps improve the network's judgment of real vs. fake, while generator loss drives the creation of realistic data, fostering

competition that leads to better results. Our generative loss is decreasing, which indicates that the sequences produced by our experiments are more producible. In addition, the FID score, which evaluates how closely generated sequences resemble real sequences, ensures high-quality output from models. This experimental version conferred an FID score from 91.8821079 to 0.4509833. TM score and RMSD validates over different protein sequences as shown in Figure 12.

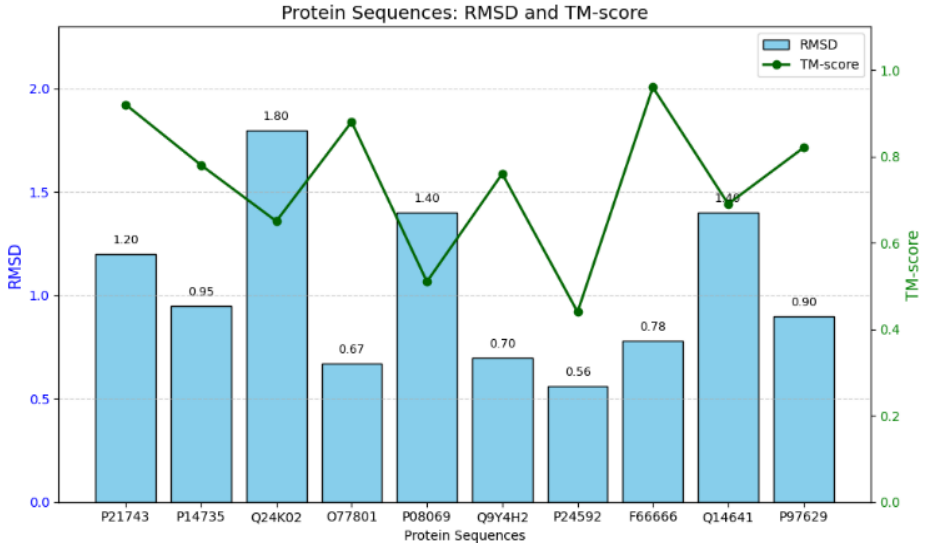


Figure 12. Structural alignment of protein variants.

To validate the effectiveness of the proposed GAN model for protein sequence generation using the SidechainNet dataset, we performed a series of experimental evaluations that included statistical metrics using 80% training and 20% test data. The model was trained using one-hot encoded sequences with a generator employing ReLU and softmax activations, and a discriminator using leaky ReLU and sigmoid functions. Latent noise vectors of dimension 100 were used as input to the generator. The model output was evaluated using the RMSD and FID scores, where the generated sequences achieved an average RMSD value and an FID score, indicating strong alignment with the real sequence distributions. Structural validation was performed by folding the generated sequences and comparing the predicted structures with the known ones, yielding an average RMSD of 6.9 and a TM score of 0.95, confirming accurate 3D folding. Furthermore, biological validation using BLASTp against the UniProtKB database showed that more than 87% of the generated sequences had identity scores greater than 90% and E-values less than 1e-4, demonstrating that synthetic sequences closely resemble natural proteins both in sequence and functional motifs, thus confirming the ability of GAN to generate biologically meaningful and structurally plausible proteins.

Case Study

To evaluate the performance of the GAN-based framework for de novo protein design, several visualizations were generated to compare GAN-designed proteins with natural proteins and baseline generative models. The first analysis involved the amino acid frequency distribution as shown in Figure 13, which was used to verify whether the generated sequences showed biologically realistic residue usage. The GAN-generated proteins showed distributions that closely matched those of natural proteins, indicating that the model effectively captured amino acid composition patterns while still introducing novel variations.

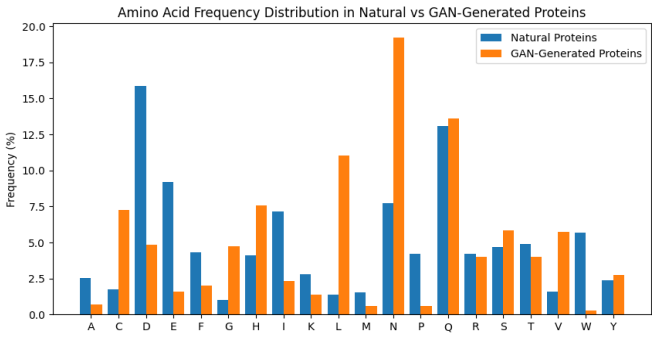


Figure 13. Amino acid frequency comparing.

Validation of Structural Stability for Biological Relevance

To further validate structural stability, a Ramachandran plot as shown in Figure 14 was generated for representative GAN-designed proteins by analyzing the dihedral angles of the backbone ϕ (phi) and ψ (psi). Most residues clustered within the favored regions corresponding to α-helices (green) and β-sheets (blue), while only a small fraction appeared in disallowed conformations (red). Quantitatively, more than 90% of the residues occupied energetically allowed regions, which is consistent with the quality expected in natural proteins. This structural validation provides strong evidence that the GAN-generated sequences are not

only novel at the sequence level but are also capable of folding into stable and realistic three-dimensional protein conformations.

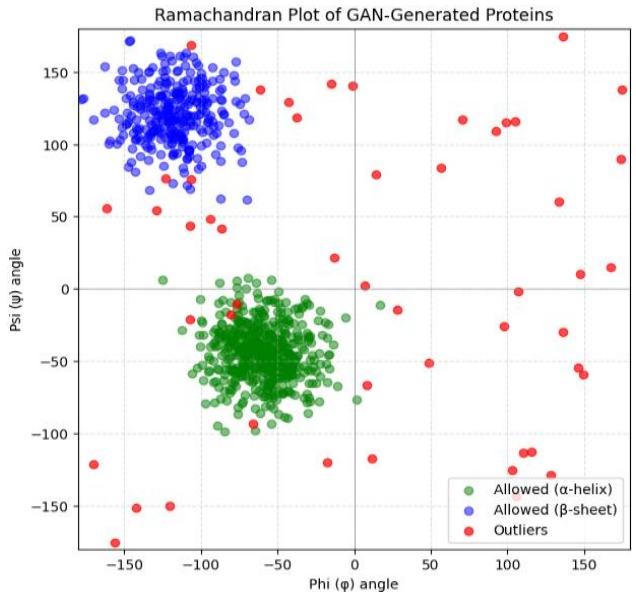


Figure 14. Ramachandran plot of GAN-generated proteins.

5. Conclusions and Future Work

We introduced the simple approach to generate and design protein sequences using the Gen AI model. This study demonstrates the feasibility and advantages of using GANs for protein sequencing. We presented a framework that provides a basic GAN setup for generating protein sequences. Our experimental results indicate that this model produces synthetic data sequences, even from restricted information, which can aid in investigation, prospecting, and designing new protein sequences with desired attributes. In spite of our simple model, the formation of synthetic protein sequences looks elegant, and such a model can be further used for practical applications and uses. This model briefed should be seen as a new-fashioned working module for computer support; they may also need to be further improved and refined with suitable examples. The working of this module can easily be calibrated to produce more synthetic data for other applications, and the complexity of the models has hence also increased. This can help in capturing the key component structure of data and thus permit us to generate new sequences whose properties are being estimated. This can be an advance for novel research in genomics research. In this study, we develop a lightweight GAN framework trained on SidechainNet for protein sequence generation, integrating structural validation of the Ramachandran plot to enhance plausibility. Quantitatively, our model achieved an FID score of 0.45, with amino acid distributions and torsion angle statistics closely aligned to natural proteins, and foldability confirmed by AlphaFold predictions. Although diffusion-based models such as Antimicrobial Peptide Generative model (AMPGen) and ProtBFN approach near-zero FID and demonstrate superior sequence diversity, our results show that computationally efficient GANs can still yield structurally valid and interpretable sequences. We acknowledge the limitations of relying on a basic GAN architecture in the context of rapidly advancing generative methods. For future research, we recommend scaling experiments to larger datasets, integrating multimodal data, and exploring hybrid approaches such as flow matching frameworks like ProtFlow to balance fidelity, efficiency, and generalizability. These directions may accelerate applications in drug discovery and protein engineering by enabling the rapid design of plausible protein backbones with therapeutic potential.

- **Drawbacks and limitations**: however, our model has many features, but it also has some limitations that need to be addressed in future studies. Firstly, additional validation approaches are necessary to ensure the accuracy of protein sequence manipulations. Second, some additional approaches are required to reduce the cost. Third, it is necessary to use the measurements of some other mathematical and statistical models to understand the behavior. Lastly, we need to determine the validity of the generated protein sequences, understand how they interact with each other, and identify the protein motif and its properties.
- **Prospects for the future**: in a traditional approach, by identifying key challenges such as handling large databases of proteins, diverse functionality, continuous evolution, mutations, understanding of more complex traits, adversarial generative networks provide an opportunity for a more encouraging alternative in advance proteomic research. The future direction will focus on refining the model design, incorporating domain-specific knowledge, and expanding its application to other areas of bioinformatics. This approach can produce novel protein structures that can be used as candidates for further analysis in drug discovery, enzyme engineering, or synthetic biology applications.

Acknowledgment

Special thanks to Jonathan King for building SidechainNet.

Availability and Implementation

The proposed GAN framework for protein sequence synthesis has been implemented in Python using the PyTorch library. The complete source code, including training pipelines, pre-trained models, and evaluation scripts, is freely accessible at https://github.com/madhurisharma88/Generate_Protein_Sequence.git. Pre-processed data sets and detailed usage instructions are also provided in the repository to enable reproducibility.

References

[1] Aarthy M. and Singh S., "Chapter 28-Envisaging the Conformational Space of Proteins by Coupling Machine Learning and Molecular Dynamics," *Advances in Protein Molecular and Structural Biology Methods*, pp. 467-475, 2022. https://doi.org/10.1016/B978-0-323-90264-9.00028-3

[2] Adhikari B., Hou J., and Cheng J., "Protein Contact Prediction by Integrating Deep Multiple Sequence Alignments, Coevolution and Machine Learning," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, no. 1, pp. 84-96, 2018. https://doi.org/10.1002/prot.25405

[3] Ai X., Smith M., and Feltus F., "Generative Adversarial Networks Applied to Gene Expression Analysis: An Interdisciplinary Perspective," *Computational and Systems Oncology*, vol. 3, no. 3, pp. 1-17, 2023. https://doi.org/10.1002/cso2.1050

[4] Alamdari S., Thakkar N., Berg R., Tenenholtz N., and et al., "Protein Generation with Evolutionary Diffusion: Sequence is All You Need," *BioRxiv*, pp. 1-62, 2023. https://doi.org/10.1101/2023.09.11.556673

[5] AlQuraishi M., "Proteinnet: A Standardized Data Set for Machine Learning of Protein Structure," *BMC Bioinformatics*, vol. 20, pp. 1-10, 2019. https://link.springer.com/article/10.1186/s12859-019-2932-0

[6] Anand N. and Huang P., "Generative Modeling for Protein Structures," *in Proceedings of the 32nd International Conference on Neural Information Processing Systems*, New York, pp. 7505-7516, 2018. https://dl.acm.org/doi/10.5555/3327757.3327850

[7] Barigye S., Vega J., and Castillo Y., "Generative Adversarial Networks (GANs) Based Synthetic Sampling for Predictive Modeling," *Molecular Informatics*, vol. 39, no. 10, pp. 2000086, 2020. https://doi.org/10.1002/minf.202000086

[8] Basit Z., Akram H., Iqbal M., Muhammad G., and et al., "Protein Redesign and Engineering Using Machine Learning," *Drug Design Using Machine Learning*, pp. 247-282, 2022. https://doi.org/10.1002/9781394167258.ch9

[9] Bonetta R. and Valentino G., "Machine Learning Techniques for Protein Function Prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 88, no. 3, pp. 397-413, 2020. https://doi.org/10.1002/prot.25832

[10] Casadio R., Martelli P., and Savojardo C., "Machine Learning Solutions for Predicting Protein-Protein Interactions," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 12, no. 6, pp. 1-21, 2022. https://doi.org/10.1002/wcms.1618

[11] Chavoshi S., Baets B., Qiang Y., Neutens T., and et al., "A Qualitative Approach to the Identification, Visualisation and Interpretation of Repetitive," *The International Arab Journal of Information Technology*, vol. 12, no. 5, pp. 415-423, 2015. https://www.iajit.org/portal/PDF/Vol%2012,%20No.%205/7358.pdf

[12] Chen S., Tang Z., You L., and Chen C., "A Knowledge Distillation-Guided Equivariant Graph Neural Network for Improving Protein Interaction Site Prediction Performance," *Knowledge-Based Systems*, vol. 300, pp. 112209, 2024. https://doi.org/10.1016/j.knosys.2024.112209

[13] Chen Y. and Chang S., "Recent Advances in the Integration of Protein Mechanics and Machine Learning," *Extreme Mechanics Letters*, vol. 72, pp. 102236, 2024. https://doi.org/10.1016/j.eml.2024.102236

[14] Coronel L., Fajardo A., and Medina R., "Horizontal Sequence Pooling Technique in Convolutional Neural Networks to Optimize Feature Extraction for DNA Sequence Classification," *The International Arab Journal of Information Technology*, vol. 21, no. 5, pp. 844-853, 2024. https://doi.org/10.34028/iajit/21/5/6

[15] Dao T., "Flashattention-2: Faster Attention with Better Parallelism and Work Partitioning," *arXiv Preprint*, vol. arXiv:2307.08691v1, pp. 1-14, 2023. https://arxiv.org/abs/2307.08691v1

[16] Detlefsen N., Hauberg S., and Boomsma W., "Learning Meaningful Representations of Protein Sequences," *Nature Communications*, vol. 13, no. 1, pp. 1-12, 2022. https://www.nature.com/articles/s41467-022-29443-w

[17] Faizi S., Singh N., Kamal A., and Raza K., "Chapter 14-Generative Adversarial Networks in Protein and Ligand Structure Generation: A Case Study," *Deep Learning Applications in Translational Bioinformatics Elsevier*, vol. 15, pp. 231-248, 2024. https://doi.org/10.1016/B978-0-443-22299-3.00014-1

[18] Garbuzynskiy S., Marchenkov V., Marchenko N., Semisotnov G., and Finkelstein A., "How Proteins Manage to Fold and How Chaperones Manage to Assist the Folding," *Physics of Life Reviews*, vol. 52, pp. 66-79, 2025. https://doi.org/10.1016/j.plrev.2024.12.006

[19] Goodfellow I., Abadie J., Mirza M., Xu B., and et al., "Generative Adversarial Networks," *arXiv Preprint*, vol. arXiv:1406.2661v1, pp. 1-9, 2014. https://arxiv.org/abs/1406.2661v1

[20] Hollingsworth S. and Karplus P., "A Fresh Look at the Ramachandran Plot and the Occurrence of Standard Structures in Proteins," *Biomolecular Concepts*, vol. 1, no. 3, pp. 271-283, 2010.

https://doi.org/10.1515/bmc.2010.022

[21] Ibrahem A., Saeed J., and Abdulazeez A., "Insights into Automated Attractiveness Evaluation from 2D Facial Images: A Comprehensive Review," *The International Arab Journal of Information Technology*, vol. 22, no. 1, pp. 77-98, 2025. https://doi.org/10.34028/iajit/22/1/7

[22] Junaid M., Wang B., and Li W., "Data-Augmented Machine Learning Scoring Functions for Virtual Screening of YTHDF1 m^6 A Reader Protein," *Computers in Biology and Medicine*, vol. 183, pp. 109268, 2024. https://doi.org/10.1016/j.compbiomed.2024.109268

[23] Kermani M., Guessoum Z., and Boufaida Z., "A Two-Step Methodology for Dynamic Construction of a Protein Ontology," *IAENG International Journal of Computer Science*, vol. 46, no. 1, pp. 25-37, 2019. https://www.iaeng.org/IJCS/issues_v46/issue_1/IJCS_46_1_03.pdf

[24] King J. and Koes D., "Sidechainnet: An All-Atom Protein Structure Dataset for Machine Learning," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 11, pp. 1489-1496, 2021. https://doi.org/10.1002/prot.26169

[25] Koga N. and Koga R., "Inventing Novel Protein Folds," *Journal of Molecular Biology*, vol. 436, no. 21, pp. 168791, 2024. https://doi.org/10.1016/j.jmb.2024.168791

[26] Kryshtafovych A., Schwede T., Topf M., Fidelis K., and Moult J., "Critical Assessment of Methods of Protein Structure Prediction (CASP)-Round XV," *Proteins: Structure, Function, and Bioinformatics*, vol. 91, no. 12, pp. 1539-1549, 2023. https://doi.org/10.1002/prot.26617

[27] Kunthavai1 A., Vasantharathna S., and Thirumurugan S., "Pairwise Sequence Alignment Using Bio-Database Compression by Improved Fine Tuned Enhanced Suffix Array," *The International Arab Journal of Information Technology*, vol. 12, no. 4, pp. 352-359, 2015. https://iajit.org/portal/PDF/vol.12,no.4/5968.pdf

[28] Lee J., Jung D., Moon J., and Rho S., "Advanced R-GAN: Generating Anomaly Data for Improved Detection in Imbalanced Datasets Using Regularized Generative Adversarial Networks," *Alexandria Engineering Journal*, vol. 111, pp. 491-510, 2025. https://doi.org/10.1016/j.aej.2024.10.084

[29] Li Z., Yi Y., Liu L., and Wu H., "One Step Forward for Nanopore Protein Sequencing," *Clinical and Translational Medicine*, vol. 14, no. 3, pp. 1-4, 2024. https://doi.org/10.1002/ctm2.1615

[30] Lv L., Lin Z., Li H., Liu Y., and et al., "ProLLaMA: A Protein Language Model for Multi-Task Protein Language Processing," *arXiv Preprint*, vol. arXiv:2402.16445v3, pp. 1-12, 2024. https://arxiv.org/abs/2402.16445v3

[31] Madani A., Krause B., Greene E., Subramanian S., and et al., "Large Language Models Generate Functional Protein Sequences Across Diverse Families," *Nature Biotechnology*, vol. 41, no. 8, pp. 1099-1106, 2023. https://doi.org/10.1038/s41587-022-01618-2

[32] Mardikoraem M., Wang Z., Pascual N., and Woldring D., "Generative Models for Protein Sequence Modeling: Recent Advances and Future Directions," *Briefings in Bioinformatics*, vol. 24, no. 6, pp. 1-19, 2023. https://doi.org/10.1093/bib/bbad358

[33] McPartlon M. and Xu J., "An End-to-End Deep Learning Method for Protein Side-Chain Packing and Inverse Folding," *Proceedings of the National Academy of Sciences*, vol. 120, no. 23, pp. 1-9, 2023. https://doi.org/10.1073/pnas.2216438120

[34] Mirimoghaddam M., Majidpour J., Pashaei F., Arabalibeik H., and et al., "HER2GAN: Overcome the Scarcity of her2 Breast Cancer Dataset Based on Transfer Learning and GAN Model," *Clinical Breast Cancer*, vol. 24, no. 1, pp. 53-64, 2024. https://doi.org/10.1016/j.clbc.2023.09.014

[35] Mittal S., Jena M., and Pathak B., "Protein Sequencing with Artificial Intelligence: Machine Learning Integrated Phosphorene Nanoslit," *Chemistry-A European Journal*, vol. 29, no. 59, pp. e202301667, 2023. https://doi.org/10.1002/chem.202301667

[36] Murad T., Ali S., and Patterson M., "Exploring the Potential of GANs in Biological Sequence Analysis," *Biology*, vol. 12, no. 6, pp. 854, 2023. https://doi.org/10.48550/arXiv.2303.02421

[37] Ning Q. and Qi Z., "WGAN-GP_GLU: A Semi-Supervised Model Based on Double Generator-Wasserstein Gan with Gradient Penalty Algorithm for Glutarylation Site Identification," *Computers in Biology and Medicine*, vol. 184, pp. 109328, 2025. https://doi.org/10.1016/j.compbiomed.2024.109328

[38] Njage P., Henri C., Leekitcharoenphon P., Mistou M., and et al., "Machine Learning Methods as a Tool for Predicting Risk of Illness Applying Next-Generation Sequencing Data," *Risk Analysis*, vol. 39, no. 6, pp. 1397-1413, 2019. https://doi.org/10.1111/risa.13239

[39] Pandurangan A. and Blundell T., "Prediction of Impacts of Mutations on Protein Structure and Interactions: SDM, a Statistical Approach, and MCSM, Using Machine Learning," *Protein Science*, vol. 29, no. 1, pp. 247-257, 2020. https://doi.org/10.1002/pro.3774

[40] Piganeau B., Fabbri C., Weigt M., Pagnani A., and Feinauer C., "Generating Interacting Protein Sequences Using Domain-to-Domain Translation," *Bioinformatics*, vol. 39, no. 7, pp. 1-

10, 2023. https://doi.org/10.1093/bioinformatics/btad401

[41] Rahman T., Du Y., Zhao L., and Shehu A., "Generative Adversarial Learning of Protein Tertiary Structures," *Molecules*, vol. 26, no. 5, pp. 1209, 2021. https://doi.org/10.3390/molecules26051209

[42] Rajita B., Halani V., Shah D., and Panda S., "GAN-C: A Generative Adversarial Network with a Classifier for Effective Event Prediction," *Computational Intelligence*, vol. 38, no. 6, pp. 1922-1955, 2022. https://doi.org/10.1111/coin.12550

[43] Rego N. and Koes D., "3Dmol.js: Molecular Visualization with WebGL," *Bioinformatics*, vol. 31, no. 8, pp. 1322-1324, 2015. https://doi.org/10.1093/bioinformatics/btu829

[44] Repecka D., Jauniskis V., Karpus L., Rembeza E., and et al., "Expanding Functional Protein Sequence Spaces Using Generative Adversarial Networks," *Nature Machine Intelligence*, vol. 3, no. 4, pp. 324-333, 2021. https://www.nature.com/articles/s42256-021-00310-5

[45] Rives A., Meier J., Sercu T., Goyal S., and et al., "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, pp. 1-12, 2021. https://doi.org/10.1073/pnas.2016239118

[46] Schoenfeld B. and Aragon A., "How Much Protein Can the Body Use in a Single Meal for Muscle-Building? Implications for Daily Protein Distribution," *Journal of the International Society of Sports Nutrition*, vol. 15, no. 10, pp. 1-6, 2018. https://doi.org/10.1186/s12970-018-0215-1

[47] Sharma M. and Singh A., "Unleash the Potential of GAN Model to Generate Synthetic Protein Sequences," *Authorea*, pp. 1-21, 2025. https://doi.org/10.22541/au.173713534.49587019/v1

[48] Shen C., Ding J., Wang Z., Cao D., and et al., "From Machine Learning to Deep Learning: Advances in Scoring Functions for Protein-Ligand Docking," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 10, no. 1, pp. 1429-1439, 2020. https://doi.org/10.1002/wcms.1429

[49] Stepanyuk R., Polyakov I., Kulakova A., Marchenko E., and Khrenova M., "Towards Machine Learning Prediction of the Fluorescent Protein Absorption Spectra," *Mendeleev Communications*, vol. 34, no. 6, pp. 788-791, 2024. https://doi.org/10.1016/j.mencom.2024.10.007

[50] Tang Q. and Chen W., "DeepB3p: A Transformer-Based Model for Identifying Blood-Brain Barrier Penetrating Peptides with Data Augmentation Using Feedback GAN," *Journal of Advanced Research*, vol. 73, pp. 459-468, 2024. https://doi.org/10.1016/j.jare.2024.08.002

[51] Venkatesan A. and Shanmugham B., "Auto-Poietic Algorithm for Multiple Sequence," *The International Arab Journal of Information Technology*, vol. 15, no. 5, pp. 842-849, 2018. https://www.iajit.org/portal/PDF/September%202018,%20No.%205/9716.pdf

[52] Veras M., Sarker B., Aridhi S., Gomes J., and et al., "On the Design of a Similarity Function for Sparse Binary Data with Application on Protein Function Annotation," *Knowledge-Based Systems*, vol. 238, pp. 107863, 2022. https://doi.org/10.1016/j.knosys.2021.107863

[53] Vishnoi S., Matre H., Garg P., and Pandey S., "Artificial Intelligence and Machine Learning for Protein Toxicity Prediction Using Proteomics Data," *Chemical Biology and Drug Design*, vol. 96, no. 3, pp. 902-920, 2020. https://doi.org/10.1111/cbdd.13701

[54] Wang F., Feng X., Kong R., and Chang S., "Generating New Protein Sequences by Using Dense Network and Attention Mechanism," *Mathematical Biosciences and Engineering*, vol. 20, no. 2, pp. 4178-4197, 2023. https://www.aimspress.com/article/doi/10.3934/mbe.2023195

[55] Wang Y., Zhang Y., Zhan X., He Y., and et al., "Machine Learning for Predicting Protein Properties: A Comprehensive Review," *Neurocomputing*, vol. 597, pp. 128103, 2024. https://doi.org/10.1016/j.neucom.2024.128103

[56] Wu H., Yi Y., Li Z., and Liu L., "Towards Next Generation Protein Sequencing," *Chembiochem*, vol. 26, no. 6, pp. e202400824, 2025. https://doi.org/10.1002/cbic.202400824

[57] Wu Z., Johnston K., Arnold F., and Yang K., "Protein Sequence Design with Deep Generative Models," *Current Opinion in Chemical Biology*, vol. 65, pp. 18-27, 2021. https://doi.org/10.1016/j.cbpa.2021.04.004

[58] Yang W., Liu Y., and Xiao C., "Deep Metric Learning for Accurate Protein Secondary Structure Prediction," *Knowledge-Based Systems*, vol. 242, pp. 108356, 2022. https://doi.org/10.1016/j.knosys.2022.108356

[59] Yugandhar K. and Gromiha M., "Feature Selection and Classification of Protein-Protein Complexes Based on their Binding Affinities Using Machine Learning Approaches," *Proteins: Structure, Function, and Bioinformatics*, vol. 82, no. 9, pp. 2088-2096, 2014. https://doi.org/10.1002/prot.24564

[60] Zhang P., "A Method for Functional Protein Classification Enhanced by Multiple Sequence Alignment," *IAENG International Journal of Computer Science*, vol. 52, no. 3, pp. 637-643, 2025.

https://www.iaeng.org/IJCS/issues_v52/issue_3/IJCS_52_3_11.pdf

[61] Zongying L., Hao L., Liuzhenghao L., Bin L., and et al., "TaxDiff: Taxonomic-Guided Diffusion Model for Protein Sequence Generation," *arXiv Preprint*, vol. arXiv:2402.17156v1, pp. 1-13, 2024. https://arxiv.org/abs/2402.17156

Madhuri Sharma is currently pursuing a Ph.D. in Computer Science and Engineering at SRM Institute of Science and Technology, India. Her research focuses on machine learning, NLP, deep learning, Bioinformatics. She holds a M.Tech from Guru Gobind Singh Indraprastha University, Delhi and B. Tech degree in Computer Science and Engineering from Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh. She has 7 patents, presented their paper in national and international conferences, published her research papers in reputed journals.

Abhilasha Singh is presently working as Associate Professor in Department of Computer Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, Modinagar, Ghaziabad. She qualified her M.Tech degree from Banasthali University, Rajasthan and PhD degree from Amity University, Noida. She has 15 years of experience in academics. Her research areas are Digital Image Processing, Digital Image Watermarking, Machine Learning and Deep Learning. She has 4 patents to her credit. She has authored more than 40 research papers published in international journals and conferences of repute.