Graph Autoencoder-Driven Discovery of Structural Patterns in the AlphaFold E. coli Proteome

Parth Hasolkar, Farhaan Kammar, Shivtej Ghorpade, Yashvanth Kate, and Dr.Sharada K Shiragudikar

Department of Computer Science Engineering,
KLE Technological University, Hubballi, India
01fe23bcs201@kletech.ac.in, 01fe23bcs244@kletech.ac.in,
01fe24bcs422@kletech.ac.in, 01fe24bcs418@kletech.ac.in,
sharada.shiragudikar@kletech.ac.in

Abstract. Predictions of protein structures of high quality. from AlphaFold has made it possible to analyze the structure of proteins on a proteome scale, but au scalable prediction of thousands of structures tomated and scalable organizing structures tomated and scalable optimization of thousands of predicted structures remains challenging. This paper introduces a non-alignment-based framework of. use of unsupervised discovery of structural patterns in the E. coli proteome us ing graph based deep learning. All AlphaFold-predicted pro were 4,371. tein structures are reduced to contact graphs on a level of residues. ing pLDDT-based features of confidence. The training of a Graph Autoencoder is done. to train slim, structure-sensitive protein representations, which are subse. then clustered with the k-means. Coherence of the result of this structure. Root Mean Square Deviation and Contact are the measures used to evaluate clusters. Map Similarity. The findings indicate the existence of meaningful intra-cluster structctural consistency, meaning that graph based embeddings work well. detect protein formal interactions on a scale. This study highlights graph autoencoders as a scalable and interpretable method of large. scale protein structure analysis.

Keywords: Graph Autoencoder, Protein Structure, AlphaFold, Structural Clustering, Protein Graphs, RMSD, Contact Map Similarity

1 Introduction

Use of protein structures is one of the key problems in bioinformatics, and it remains challenging at the proteome scale. Traditional alignment-based methods such as TM-align [8] and DALI [10, 11] provide accurate structural comparisons but are computationally expensive. The availability of large repositories of protein structures predicted by AlphaFold [1, 2] highlights the need for scalable and automated methods for structural organization. In this context, deep learning has emerged as a promising non-alignment-based alternative, enabling the learning of compact, structure-aware representations that scale to large-scale protein structure comparison, building upon its demonstrated effectiveness in complex data-driven applications [18–22].

1.1 Background

Graph-based representations have been developed as viable alternatives to coordinate-based protein analysis, modeling proteins as graphs of interacting residues to capture both local and global structural information while remaining invariant to rigid-body transformations [5]. Compact structure-aware Graph Neural Networks (GNNs) and Graph Autoencoders (GAEs) do not require explicit structural alignment [4], making them well suited for large-scale unsupervised protein structure analysis [4, 5].

1.2 Motivation

Scalable automated structural organization of proteomes [2] will be required by the availability of AlphaFold-predicted proteomes [1]. Deep learning models based on graph representations allow graphs to be presented at the level of residues, reacting to the input enables superior performance by integrating pLDDT confidence scores [1, 13], encouraging graph autoencoders [4] to be confidence-aware and support unsupervised proteome-scale clustering [4, 6, 7].

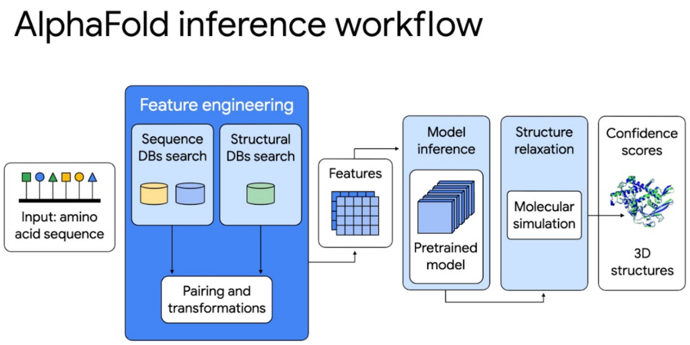

Fig. 1: Growth of the AlphaFold Protein Structure Database, demonstrating the rapid expansion of predicted proteomes and highlighting the need for scalable computational analysis.

1.3 Research Gaps

The vast majority of current protein clustering algorithms do not take into account AlphaFold confidence or explicit structural validation [1, 12, 13], motivating the development of scalable, confidence-sensitive, and structure-consistent clustering systems capable of operating at the proteome scale [2, 4, 6, 7].

Table 1: Research gaps in existing protein structure clustering methods and corresponding solutions proposed in this work [1, 2, 4, 8, 10, 11, 13].

Research Gap	**Limitations in Existing Methods**	**Proposed Solution**
Lack of confidence-aware modeling	AlphaFold-predicted structures are considered to be uniformly good, adding noise to the low-confidence regions [1, 13].	Assess scores of pLDDTs at the residue-level and convert them into node features in graphs as confidence-aware features.
Lack of structure-aware evaluation	Clustering quality is often measured by costs of statistical measures that are not indicative of structural similarity in three-dimension space [6, 7].	Post-clustering structural validation with the use of RMSD and Contact Map Similarity [8, 10, 11].

1.4 Objectives

The primary objective of this work is to develop a scalable and alignment-free framework for unsupervised clustering of AlphaFold-predicted protein structures using graph-based deep learning [1, 2, 4, 5]. Specifically, this study aims to:

- Construct residue-level protein graphs from AlphaFold-predicted structures using spatial contact information and pLDDT confidence scores [1, 2, 13].
- Learn compact, structure-aware embeddings of protein graphs using a Graph Autoencoder [4, 5].
- Perform unsupervised clustering of proteins in the learned embedding space at proteome scale [6, 7].
- Validate the resulting clusters using structure-aware metrics, including RMSD and Contact Map Similarity, as post-hoc evaluation measures [8, 10, 11].

This framework enables systematic exploration of large proteomic structural datasets while avoiding explicit structural alignment.

2 Related Work

The alignment-based methods, along with newer ones based on graphs, focus on alignment [8, 10, 11]. In research on protein structure, these methods are adopted [3]. You will find these methods in this section. This involves unsupervised analysis of protein structures based on reviewed models and graph-based approaches [4–7]. The prediction by AlphaFold was a motivating factor [1, 2].

2.1 Protein Structure Comparison

TM-align and DALI together with FATCAT alignment-based methods offer an accurate means of comparing protein structure at a very low cost; however, they are highly sensitive to structural variation, reducing their utility in large-scale clustering of AlphaFold-predicted protein set [8, 10, 11, 1, 2].

2.2 Graph-Based Protein Representations

Graph-based representations offer a useful paradigm of protein structure modeling that is claimed to be the most appropriate because of its ability to capture local and global structural information, and being insensitive to rigid-body transformations because the residues are the nodes and the spatial associations are their edges [5]. Despite GNNs applications to protein-related problems, the majority of these methods are based on supervised learning, which restricts their application to unsupervised organization of structures and massive clustering of AlphaFold-predicted proteomes [4, 6, 7, 1, 2].

2.3 Graph Autoencoders for Unsupervised Structural Analysis

Graph Autoencoders (GAEs) are unsupervised learners of low-dimensional models of graph structured data that combine both node features and topology encodings [4]. Although autoencoder-based models have been used to solve structural biology problems including fold classification [4, 5], they have so far been used to perform unsupervised, proteome-scale clustering of AlphaFold-predicted structures, which is the point of the current work [1, 2].

3 Proposed Methodology

This section presents an unsupervised, graph-based deep learning framework for clustering protein structures [1, 2]. AlphaFold-predicted structures are represented as graphs with integrated residue-level confidence information, enabling robust and structure-aware representations. The proposed methodology is scalable, alignment-free, and suitable for proteome-scale analysis.

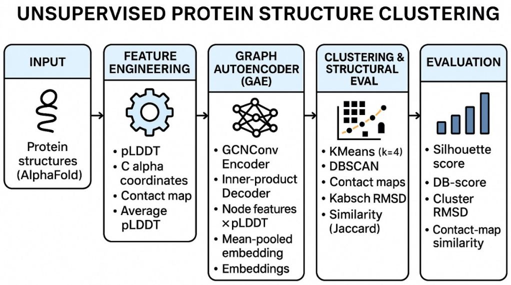

Fig. 2: Overall architecture of the proposed GAE-based protein structure clustering framework, illustrating data preprocessing, graph construction, representation learning, clustering, and evaluation stages.

Figure 2 summarizes the five main stages of the pipeline: data preprocessing, graph construction, representation learning using Graph Autoencoders [4, 5], unsupervised clustering, and structure-aware evaluation. Together, these components enable systematic exploration of structural patterns in large protein datasets.

3.1 Dataset Collection

We analyzed 4,371 AlphaFold-predicted protein structures from the *Escherichia coli* proteome (UniProt ID: UP000000625), obtained from the AlphaFold Protein Structure Database [1, 2]. Structures were parsed using BioPython [14], retaining only Cα atoms, and residue-level pLDDT scores were normalized and used as confidence indicators during graph construction.

Contact Map Definition Protein contact maps were constructed using a distance-based criterion between residue pairs:

$$C_{ij} = \begin{cases} 1, & \text{if } \|r_i - r_j\| \leq 8\,\text{Å}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where r_i and r_j denote the Cα coordinates of residues i and j, respectively.

Algorithm 1 Protein Structure Parsing and Contact Map Generation

```
Require: AlphaFold-predicted protein structures $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$
Require: Distance threshold $d = 8\,\text{Å}$
Ensure: Protein contact maps $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$
for each protein structure $P_i \in \mathcal{P}$ do
    Parse the .cif.gz file and extract C$\alpha$ coordinates $X_i$
    Extract and normalize residue-wise pLDDT scores $p_i$
    Initialize contact map $C_i$
    for each residue pair $(j, k)$ do
        if $\|x_j - x_k\| \leq d$ then
            $C_i(j, k) \leftarrow 1$
        else
            $C_i(j, k) \leftarrow 0$
        end if
    end for
end for
return $\mathcal{C}$
```

3.2 Graph Representation

Each protein structure is represented as an undirected weighted graph $G = (V, E)$, where nodes correspond to amino acid residues and edges denote residue–residue contacts. An edge $(i, j) \in E$ is formed if the Euclidean distance between the Cα atoms of residues i and j is at most 8 Å [10, 9].

Each node v_i is associated with a feature vector comprising the amino acid identity (one-hot encoded) and the normalized pLDDT confidence score provided by AlphaFold [1]. These features encode both biochemical properties and residue-level structural reliability.

Edge weights are defined as the inverse of the inter-residue distance:

$$w_{ij} = \frac{1}{\|r_i - r_j\| + \epsilon}, \tag{2}$$

where r_i and r_j denote Cα coordinates and ϵ ensures numerical stability [9].

This model describes local interactions and global fold topology without being sensitive to rigid-body motions and allows the passage of messages along the suggested Graph Autoencoder model [4, 5]. Algorithm 2 summarizes the graph construction process.

Algorithm 2 Graph Construction with Confidence-Aware Features

```
Require: Contact maps $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$
Require: Residue-wise pLDDT scores $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$
Ensure: Protein graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$
for each contact map $C_i \in \mathcal{C}$ do
    Initialize graph $G_i = (V_i, E_i)$
    Add one node to $V_i$ for each residue in $C_i$
    Assign normalized pLDDT values as node features
    for each residue pair $(j, k)$ such that $C_i(j, k) = 1$ do
        Add an undirected edge $(j, k)$ to $E_i$
    end for
end for
return $\mathcal{G}$
```

3.3 Graph Autoencoder (GAE) Architecture

A Graph Autoencoder (GAE) [4, 5] learns unsupervised complex representations of protein structures using the contact graph as input. The encoder is based on Graph Convolutional Networks and learns to combine local and long-range interactions between residues, while its inner product decoder recovers graph connectivity [4]. Node embeddings produced by the encoder are composed into compact representations at the level of the protein and preserve the structural organization of proteins to facilitate unsupervised clustering without alignment[6, 7].

Graph Autoencoder Formulation

(a) Graph Convolution A GCN layer updates node representations as [4, 5]:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right), \tag{3}$$

where $\tilde{A} = A + I$ includes self-loops, $\tilde{D}$ is the degree matrix, $W^{(l)}$ denotes learnable weights, and $\sigma(\cdot)$ is the ReLU activation [4, 5].

(b) Decoder The adjacency matrix is reconstructed using an inner-product decoder [4]:

$$\hat{A}_{ij} = \sigma\left(z_i^\top z_j\right), \tag{4}$$

where z_i and z_j denote latent residue embeddings [4].

(c) Confidence-Weighted Loss The reconstruction loss is weighted using AlphaFold pLDDT confidence scores [1, 13]:

$$\mathcal{L}_{\text{GAE}} = -\sum_{i,j} w_{ij}\left[A_{ij}\log\hat{A}_{ij} + (1 - A_{ij})\log(1 - \hat{A}_{ij})\right], \tag{5}$$

where $w_{ij} = p_i \cdot p_j$ emphasizes interactions between high-confidence residue pairs while suppressing unreliable contacts [1, 13, 4].

Algorithm 3 outlines the training procedure and protein-level embedding extraction [4, 6, 7].

Algorithm 3 Graph Autoencoder Training and Embedding Extraction

```
Require: Protein contact graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$
Ensure: Protein-level embeddings $\mathcal{Z} = \{z_1, z_2, \ldots, z_N\}$
Initialize GCN-based encoder and inner-product decoder parameters
for each training epoch do
    for each protein graph $G_i \in \mathcal{G}$ do
        Encode node features using the GCN encoder to obtain latent node embeddings $H_i$
        Reconstruct adjacency matrix $\hat{A}_i$ using the inner-product decoder
        Compute pLDDT-weighted reconstruction loss $\mathcal{L}_{\text{GAE}}$
        Update encoder and decoder parameters via backpropagation
    end for
end for
for each protein graph $G_i \in \mathcal{G}$ do
    Apply mean pooling over node embeddings in $H_i$ to obtain protein-level embedding $z_i$
end for
return $\mathcal{Z}$
```

3.4 Dimensionality Reduction and Clustering

The Graph Autoencoder will be trained to find protein embeddings and group them in an unsupervised clustering fashion [4, 6, 7]. UMAP was used to create a 2D map upon which visual inspection was conducted after denoising the protein embeddings using PCA [15, 16]. Each original embedding space was clustered using K-Means [15]. Only the two-dimensional UMAP projection allowed assessing the distance between clusters [15, 16].

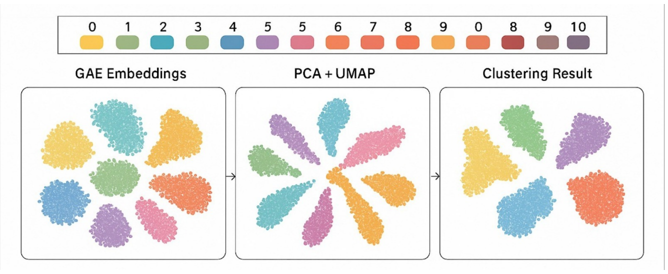

Fig. 3: Overview of dimensionality reduction and clustering applied to GAE embeddings. High-dimensional embeddings are projected using PCA and UMAP for visualization, while clustering is performed in the original embedding space using K-Means and DBSCAN [15, 16].

K-Means Objective K-Means minimizes the within-cluster variance [15]:

$$\mathcal{L}_{\text{cluster}} = \sum_{i=1}^{N} \|z_i - \mu_{c_i}\|^2, \tag{6}$$

where z_i denotes the embedding of protein i and μ_{c_i} is the centroid of cluster c_i [15].

3.5 Evaluation and Validation

The evaluation of clustering quality is measured by statistical indices such as Silhouette score (SS), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI) [6, 7] and structure-sensitive validation scores such as Root Mean Square Deviation (RMSD) [9], TM-score [8], and Contact Map Similarity (CMS) [10, 11] to measure cluster compactness, fold similarity, and consistency of interactions on a per-residue level.

Structural Validation Metrics

(a) Root Mean Square Deviation (RMSD)

$$\text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|r_i^{(1)} - r_i^{(2)}\|^2}, \tag{7}$$

where $r_i^{(1)}$ and $r_i^{(2)}$ denote coordinates of aligned residues in two protein structures [9]. Mean intra-cluster RMSD quantifies geometric coherence [8, 10].

(b) TM-Score

$$\text{TM} = \frac{1}{L_{\text{target}}} \sum_{i=1}^{L_{\text{aligned}}} \frac{1}{1 + \left(\frac{d_i}{d_0}\right)^2}, \tag{8}$$

where d_i denotes distances between aligned residues [8]. TM-score provides a length-invariant measure of global structural similarity [8, 11].

(c) Contact Map Similarity (CMS)

$$\text{CMS} = \frac{|C^{(1)} \cap C^{(2)}|}{|C^{(1)} \cup C^{(2)}|}, \tag{9}$$

where $C^{(1)}$ and $C^{(2)}$ denote binary contact maps [10, 12]. CMS evaluates preservation of residue interaction patterns within clusters [10, 12].

Algorithm 4 Unsupervised Clustering and Structural Validation

```
Require: Protein embeddings $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$
Require: Number of clusters $k$
Ensure: Cluster assignments and structural validation metrics
Apply K-Means clustering on $\mathcal{Z}$ to obtain cluster labels
for each cluster $c$ do
    Compute average pairwise RMSD among proteins in $c$
    Compute average Contact Map Similarity (CMS) within $c$
    Assess structural coherence based on RMSD and CMS values
end for
return Cluster labels and structural validation results
```

4 Final Results

This section presents the final clustering outcomes obtained from the proposed Graph Autoencoder (GAE)-based protein representation framework [4]. Protein-level embeddings learned by the GAE were clustered using K-Means with $k = 4$ [15], and cluster-wise structural statistics were computed to evaluate internal structural coherence and biological relevance [8, 10, 12].

4.1 Cluster-Level Structural Statistics

Table 2 summarizes cluster size and mean structural similarity metrics.

Table 2: Cluster-level structural statistics from GAE-based clustering of AlphaFold-predicted *E. coli* proteins.

ID	**Proteins**	**RMSD (Å)**	**CMS**
0	615	34.52	0.56
1	3702	32.78	0.42
2	18	51.19	0.31
3	37	57.81	0.38

4.2 Structural Coherence Analysis

Cluster 0 and 1 have smaller RMSD (34.5 Å and 32.8 Å) and larger CMS (0.56 and 0.42), which is evidence of high structural consistency [8–10]. The increased size of these clusters allows indicating that the learned embeddings encode protecting structural patterns dominating the *E. coli* proteome [4, 12].

Clusters 2 and 3, on the other hand, have higher RMSD and lower values of CMS indicating more structural variety that is expected in smaller clusters with less prevalent protein subfolds [10, 12]. The identified inverse correlation between RMSD and CMS across clusters proves the usefulness of the designed graph-based embedding structure and encourages future discussion of the relationship between them in the next subsection [4, 8, 12].

4.3 Relationship Between RMSD and Contact-Map Similarity

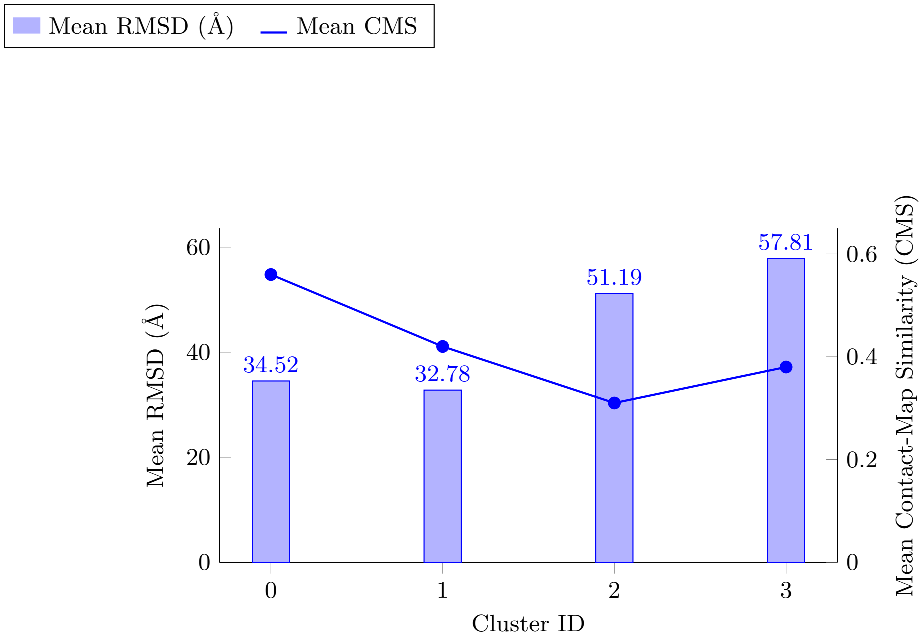

Fig. 4: Cluster-wise comparison of mean RMSD and mean contact-map similarity (CMS) for protein clusters obtained using K-Means clustering ($k = 4$) on GAE-derived embeddings. Lower RMSD values correspond to higher CMS, indicating stronger internal structural coherence.

4.4 Structural Reliability of the Dataset

Normalized pLDDT scores across clusters ranged from 0.73 to 0.92, indicating generally high confidence in the AlphaFold-predicted structures used for clustering [1, 2].

5 Future Work

Future research will be on more expressive graph architecture [4, 5]. This includes considering ways of capturing non-local interactions between residues and structural uncertainty [4, 13]. It will involve attention and variational autoencoders that will be used to capture such kinds of information in a better way [4, 5]. The representation of the quality of the representation will be enhanced by the addition of sequence-derived and evolutionary or functional features of biological data [17]. The validation will be based on the known protein domain families and definitive experiments to determine the actual functional importance [3, 12]. The framework will also be modified in order to aid comparative studies of homologous proteins on the basis of structural conservation and evolution [10, 11].

Acknowledgment

We would like to thank the faculty of KLE Technological University, Hubballi, who came to our aid when doing this project.

References

1. J. Jumper *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021, doi: 10.1038/s41586-021-03819-2.
2. M. Varadi *et al.*, "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models," *Nucleic Acids Research*, vol. 50, no. D1, pp. D439–D444, 2022, doi: 10.1093/nar/gkab1061.
3. H. M. Berman *et al.*, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.
4. T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders," *arXiv preprint arXiv:1611.07308*, 2016.
5. W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
6. J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis (DEC)," in *Proc. 33rd Int. Conf. on Machine Learning (ICML)*, 2016.
7. Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering," in *Proc. 26th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1965–1972, 2017.
8. Y. Zhang and J. Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," *Nucleic Acids Research*, vol. 33, no. 7, pp. 2302–2309, 2005, doi: 10.1093/nar/gki524.

9. W. Kabsch, “A solution for the best rotation to relate two sets of vectors,” *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923, 1976.
10. L. Holm and C. Sander, “Protein structure comparison by alignment of distance matrices,” *Journal of Molecular Biology*, vol. 233, no. 1, pp. 123–138, 1993.
11. L. Holm, “DALI and the persistence of protein shape,” *Protein Science*, vol. 31, no. 1, pp. 72–92, 2022, doi: 10.1002/pro.4204.
12. S. Bittrich *et al.*, “AlphaFold structure clustering and classification using Foldseek,” *bioRxiv preprint*, 2023, doi: 10.1101/2023.01.15.524127.
13. N. Hiranuma *et al.*, “Improved protein structure refinement guided by deep learning-based accuracy estimation,” *Nature Communications*, vol. 12, no. 1, p. 1340, 2021, doi: 10.1038/s41467-021-21640-8.
14. J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
15. F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
16. C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020.
17. C. Zhang, W. Zheng, S. M. Mortuza, Y. Li, and Y. Zhang, “DeepMSA2: improving protein multiple sequence alignment by building sequence profiles from predicted structural constraints,” *Nucleic Acids Research*, vol. 49, no. 19, pp. 11369–11381, 2021.
18. S. K. Shiragudikar, G. Bharamagoudar, K. K. Manohara, et al., “Insight Analysis of Deep Learning and a Conventional Standardized Evaluation System for Assessing Rice Crop’s Susceptibility to Salt Stress during the Seedling Stage,”*S*N Computer Science, vol. 4, p. 262, 2023.
19. S. Y. Malathi, G. R. Bharamagoudar, and S. K. Shiragudikar, “Diagnosing and Grading Knee Osteoarthritis from X-ray Images Using Deep Neural Angular Extreme Learning Machine,”*P*roc. Indian Natl. Sci. Acad., vol. 91, pp. 95–108, 2025.
20. S. K. Shiragudikar, G. Bharamagoudar, M. K. K., M. S. Y., and G. S. Totad, “Predicting Salinity Resistance of Rice at the Seedling Stage: An Evaluation of Transfer Learning Methods,”*i*n Intelligent Systems in Computing and Communication (ISCComm 2023), CCIS, vol. 2231, Springer, Cham, 2025.
21. S. Malathi, G. Bharamagoudar, S. K. Shiragudikar, and G. S. Totad, “Predictive Models for the Early Diagnosis and Prognosis of Knee Osteoarthritis Using Deep Learning Techniques,”*i*n Intelligent Systems in Computing and Communication (ISCComm 2023), CCIS, vol. 2231, Springer, Cham, 2025.
22. Shiragudikar, S. K., & Bharamagoudar, G. (2024). Enhancing rice crop resilience: Leveraging image processing techniques in deep learning models to predict salinity stress of rice during the seedling stage. *International Journal of Intelligent Systems and Applications in Engineering*, 12(14s), 116–124.