Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 13, No. 3, 2025, 261-273.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Improving the Hierarchical Classification of Protein Families and Model Interpretation with the Grad-CAM Method and Transformers

Naeimeh M. Karimi and Mehdi Rezaeian*

Computer Engineering Department, Yazd University, Yazd, Iran.

Article Info

Article History:
Received 31 December, 2024
Revised 27 January 2025
Accepted 24 February 2025

DOI:10.22044/jadm.2025.15493.2667

Keywords:
Protein Classification, Deep Learning, CNN Neural Network, Interpretability, Transformer Models.

**Corresponding author: mrezaeian@yazd.ac.ir (M. Rezaeian).*

Abstract

In the era of massive data, analyzing bioinformatics fields and discovering its functions are very important. The rate of sequence generation using sequence generation techniques is increasing rapidly, and researchers are faced with many unknown functions. One of the essential operations in bioinformatics is the classification of sequences to discover unknown proteins. There are two methods to classify sequences: the traditional method and the modern method. The conventional methods use sequence alignment, which has a high computational cost. In the contemporary method, feature extraction is used to classify proteins. In this regard, methods such as DeepFam have been presented. This research is an improvement of the DeepFam model, and the special focus is on extracting the appropriate features to differentiate the sequences of different categories. As the model improved, the features tended to be more generic. The grad-CAM method has been used to analyze the extracted features and interpret improved network layers. Then, we used the fitting vector from the transformer model to check the performance of Grad-CAM. The COG database, a massive database of protein sequences, was used to check the accuracy of the presented method. We have shown that by extracting more efficient features, the conserved regions in the sequences can be discovered more accurately, which helps to classify the proteins better. One of the critical advantages of the presented method is that by increasing the number of categories, the necessary flexibility is maintained, and the classification accuracy in three tests is higher than that of other methods.

1. Introduction

Proteins are encoded in the genome of living organisms [1]. Interpreting these codes is critical because proteins perform many cellular functions and play an essential role in biological processes [2]. With the increasing progress of sequencing technologies, there is a large number of unknown sequences. To discover the function of an unknown protein in the alignment method, that protein must be compared with a massive database of known proteins to extract some characteristics of the unknown protein. Considering that the number of amino acids in the protein and the number of sequences in a database are large, this process is very time-consuming.

For this reason, different machine learning algorithms were used to extract knowledge from bioinformatics voluminous data [3]. Some of the commonly used algorithms in genomics and biological systems are support vector machines [4, 5], random forests [6], Bayesian networks [7], and the hidden Markov model [8]. The efficiency of machine learning algorithms is highly dependent on the selection of appropriate features [3]. These features should be selected by experienced engineers, which is quite a difficult task.

Considering that bioinformatic data is big data, traditional methods to discover the characteristics of unknown proteins are time-consuming and need the necessary accuracy, so researchers should process the data with new algorithms and computational models [9, 10, 11]. Deep learning is a suitable method in many fields, including bioinformatics. Various research studies have been done on protein sequences. One important research is the classification of protein sequences. In the classification of sequences, many factors must be considered in extracting features from the sequences, including the number and order of amino acids, the relationship of adjacent or distant amino acids, and other factors. For this reason, it is difficult to extract features from sequences with a large number of amino acids. Different deep-learning architectures have been used in protein classification. One architecture that extracts features from sequences well is the CNN architecture [12, 13].

The DeepFam method [12] uses CNN architecture and features extracted from each layer to classify proteins. In short, in the DeepFam model, first, the input data enters the pre-processing stage, and each sequence is converted into an encoded matrix. Then, the coded matrix is entered into eight paths in order; each path consists of a pair of convolution and max pooling layers. The difference of each path is only in the kernel size in the convolution layer. The output of all paths merges and is then transferred to the FC layer to continue the process. The final output is obtained using the soft-max mechanism. This research improves on the DeepFam method and aims to transform local features into global features. In processing sequences, several convolution layers, one after the other, can help transform the features from local to global to some extent, obtaining long-range relationships.

In part 2, we discuss the methods of sequence matching in alignment algorithms, deep learning models in proteins, the Grad-CAM interpretability method, and the characteristics of transformer models in general. In section 3, we explain the technique presented in this research. Section 4 analyzes the method presented and the database used in this research, and section 5 discusses the performance of the convolution layer. In this section, we have examined and interpreted the essential features of this layer.

2. Related studies

2.1. Alignment algorithms

One method for categorizing protein sequences is alignment-based. This method uses a two-by-two comparison of the sequences to determine the degree of sequence similarity. Generally, proteins are compared in three ways: local, global, and multiple alignments. Local and general methods can be analyzed in optimal and heuristic categories. Dynamic methods provide the optimal solution in aligning two sequences, but they have a lot of time and space complexity [14]. The time complexity of dynamic methods for aligning m sequences with length n is equal to $O(n^m 2^m)$ [15] and is considered NP-hard problems. Time and space complexity is more noticeable in multiple sequence methods because many sequences must be compared, and more time and space are needed to implement optimal algorithms. For this reason, heuristic algorithms are used to solve these two problems. Table 1 shows the general classification of alignment methods and several famous algorithms for each category.

The heuristic algorithms for multiple sequences can be classified as progressive and iterative [14]. Progressive methods have good speed and accuracy, but they may stop at the local optimum, and if an error occurs, they will propagate it to the end of the work [14]. From advanced algorithms, CLUSTALW [16], MAFFT [17], MUSCLE [18], and T-COFFEE [19] algorithms can be mentioned. CLUSTALW algorithm is an intelligent method for MSA that uses sequence scores. The MAFFT method uses FFT and is suitable for aligning large sequences [17]. In the MUSCLE method, a more accurate criterion calculates the distance between the sequences and reduces time and space complexity. This method is considered a fast method for alignment and consists of three stages [18]. The first stage is the quick estimation of the distance, the second stage is the progressive alignment, and the third stage is the correction of the second stage [18]. In the T-COFFEE method, the sequences are compared two by two, which is suitable for aligning short sequences.

Iterative methods can be combined with progressive methods to get better results. In general, iterative methods create an initial alignment and repeat this process by modifying the alignment of the previous step to converge to a good result [14]. Various algorithms have been presented for iterative methods, including the VDGA method [20], MOMSA method [21], PHMM [22], and GRPAM method [23]. VDGA, MOMSA, and GRPAM methods use genetic algorithms for multiple sequence alignment.

Hidden Markov models are probabilistic models and assign probability to possible states (gap, match, non-match) to check all possible states [22].

Table 1. Methods based on alignment.

		Algorithms
Local Alignment	Optimal methods	FASTA [24] Smith Waterman [25] Grapped BLAST [26] BLAT [27] BLASTZ [28]
	Heuristic methods	BLAST [26] PatternHunter [29]
Global Alignment	Optimal methods	FOGSAA [30] Needleman-Wunsch [31] GLASS [32]
	Heuristic methods	LAGAN [33] BLASTZ [28] NUMmer [34] AVID [35] ACANA [36]
Multiple Alignment	Progressive methods	CLUSTALW [16] T-COFFE [19] MAFFT [17] VDGA [20] MOMSA-W [21] GAPAM [23]
	Iterative methods	MUSCLE [18]

2. 2. Deep learning architectures in bioinformatics

Deep learning is a dedicated subset of machine learning methods that have entered the field of learning based on massive data with parallel computing power [37]. Deep learning has made significant progress in various fields, such as image processing [38], sound processing [39], and natural language processing [40]. In this research, we have benefited from deep learning to process bioinformatics data. Feature extraction is the most important step in data processing methods [41]. Sequence analysis in bioinformatics aims to discover the relationships and functions in the cell, which requires the discovery of these functions and the extraction of important features hidden in the sequences. One of the valuable methods for extracting important and key features from sequences is deep learning. There are various architectures of deep learning in the field of bioinformatics, including CNN architecture [12, 13, 42], RNN [43, 44], Deep RL [45, 46], Deep SVM [47], DST-NN [48], CVAE [49], Ensemble deep learning [50], Diffusion Models [51] and transformer models [52] [53].

2.3. Interpretability methods

Due to the expansion of the use of deep learning models in various fields, interpreting and understanding the output of these models has become necessary. In general, neural networks comprise several layers with nonlinear activation functions. This problem makes it difficult to interpret the network. For this reason, the interpretation of deep networks has become an important research topic. These methods can be divided into local and global or model-dependent and model-independent. In local methods, the model is interpreted for a specific instance. Various tools such as LIME [54] and SHAP [55] [56] have been proposed for local interpretation of models. The goal of global methods is to discover the general behavior of the model. In model-dependent methods, the interpretability method depends on the model architecture.

In contrast, model architecture is not considered in model-independent methods, and the interpretability method is effective for many models and algorithms. In this research, we use the Grad-CAM method [57]. This method is local and model-dependent and plays an important role in various fields, such as image processing, medicine, text processing, and protein interpretation.

For each specific class, we can calculate the cost distribution gradient based on the output of the convolution layer (Equation 1). $A_{i,j}^{k}$ is the value of each unit (i,j) in channel k of the convolution layer. $\frac{\partial y^{c}}{\partial A_{ij}^{k}}$, indicates the effect of small changes of ∂A_{ij}^{k} on the output value [57]. The value of Z is all the units in the convolution layer, which, in the case of the problem raised in this article, is the sequence size, and the value of α_{k}^{c} shows the weight of the channels of the convolution layer [57].

$$\alpha_{k}^{c} = \frac{1}{z}\sum_{i}\sum_{j}\frac{\partial y^{c}}{\partial A_{ij}^{k}} \quad (1)$$

The following relationship helps combine the weight of the channels and filter the less important parts with ReLU [57]. This action affects the areas with positive significance in the thermal map.

$$L^{c}_{Grad-CAM} = \mathrm{Re}\,LU(\Sigma_k \alpha^{c}_{k} A^{k}) \quad (2)$$

2.4. Transformer model

Transformer architecture is a valuable architecture based on an attention mechanism. First, it started working in natural language processing and then expanded to data processing in bioinformatics. This model can find long dependencies without sequential processing. The processing of protein sequences by transformers is increasing rapidly, and the accuracy of discovering the features hidden in the sequences is high. In this study, we will not discuss the architecture of this model because we have not used this architecture in the presented model, and only the transformer has been used to interpret the model.

The article [58] has investigated the characteristics of embedding vectors in three different models: Bert, XLNet, and ALBERT. These models show that the insertion vectors have important features of proteins, such as folding structures in proteins and binding sites. Another research is the ESM model [59] developed by the Meta-AI research team. This model is trained on 86 billion amino acids from 250 million protein sequences. This amount of protein applies much information to the model and the fitting vectors. ESM model has applications such as predicting secondary and tertiary structure, identifying long-range relationships, and containing physicochemical information of sequences. This research used the ESM model's embedding vectors to confirm the extracted features.

3. The method presented

To classify proteins, we need to distinguish their features from each other. CNN architecture is one of the deep learning architectures that extracts features in sequences well. Convolution layers play the role of MERS in alignment methods. Due to differences in alignment methods, mers that are far apart lead to less important relationships than mers that overlap. In contrast, the difference between two proteins may originate from mers that are far apart.

The convolution layers consider these points extensively and process the sequences efficiently. We have used convolution and max pooling layers to extract features from proteins. First, the features are extracted with a convolution and max pooling layer, but other features are extracted hierarchically with a deeper network. In other words, some features are hidden in other features, and their extraction requires a deeper network. Therefore, it is necessary to apply several convolution layers to the sequence in order (Figure 1). Also, with a fixed value of k in k-mer, an effective feature that can correctly classify the desired sequence may not be obtained. This operation indirectly uses mers of different sizes in the sequence. For this reason, we use the results of all the layers to get more suitable features according to Figure 1, which we have chosen up to level two due to processing limitations in this research. Figure 3 shows the proposed method's general process. This process has three general stages: pre-processing, feature extraction, and protein classification, each of which will be explained in detail.

3.1. Coding protein sequences

Protein sequences consist of 21 amino acid permutations, and processing this sequence requires a pre-processing step to convert it into a numerical matrix. A deep network can process it. In coding the sequences, we have to do several steps. In the first step, the sequences' length (the network's input size) must be equal in the deep convolutional network. The length of sequences is set to 1000. If there is a sequence whose length is less than 1000, the '-' character is used to pad the sequence. In the second step, the amino acids should be coded into values. We have used IUPAC [60] for coding and coded the input data according to Equation 3 [12] and the labels according to Equation 4 [12]. In the following, the charset variable means the names of amino acids ({A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y, X}), and N_{label} is the number of tags and label set is the names of the tags.

$$X_{i,j} = \begin{cases} 1 & if\ s_i = jth\ base\ in\ charest \\ 0.5 & if\ s_i = B\ and\ jth\ base\ in\ charest \in \{D, N\} \\ & or\ s_i = Z\ and\ jth\ base\ in\ charest \in \{E, Q\} \\ & or\ s_i = J\ and\ jth\ base\ in\ charest \in \{I, L\} \\ 0 & otherwise \end{cases} \quad (3)$$

$$Y_q = \begin{cases} 1 & if\ y = ith\ in\ labelset \\ 0 & otherwise \end{cases} \quad (4)$$

$$i \in \{1, ..., L\},\ q \in \{1, ..., N_{label}\}$$

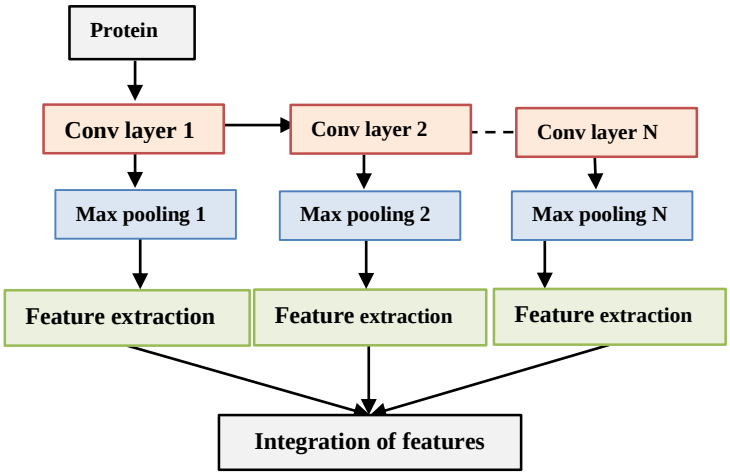


Figure 1. Integration of abstract features from each layer.

3.1. The Developed Model

Conserved regions in proteins are sensitive and important areas for classifying proteins. The length of these regions depends on the size of the mers. Achieving the most optimal mer length is tricky, so different lengths are used for mers. Some of these areas are found in the first convolution, but others must be discovered hierarchically in different network layers by merging different mers. The output of this convolution layer is injected into two other layers: convolution and max pooling. Then, the output of the second convolution enters another Max Pooling, and finally, the first Max Pooling and the second Max Pooling are merged. We combine the features extracted from different layers with different k-mers and filters and inject them into the FC layer in a flattened form. We have also used the soft-max function that expresses the final result between the classes as a probability distribution. In the deep neural network, the training must be repeated during different stages so that the network reaches the necessary convergence. For evaluation in each step, we used the cross-entropy loss function with an L2 regularizer. The described model is shown in detail in Figure 3.

We used Xavier [61] to initialize the network's weights, which converged to the desired solution faster than without weights. Deep neural networks need a suitable optimizer. For this purpose, we used the Adam optimizer, which works well for sparse gradients [62]. In the following, we will examine the two-layer max pooling algorithm.

3.3. Max Pooling layers

The action of the max-pooling layer in the deep network is to select the maximum value from the defined range. In this research, the protected areas are obtained with the help of the convolution layer. The protected area with the maximum value is determined by the max-pooling layer.

3.3.1. Max pooling 1

Suppose for a specific k in k-mer, the score for all the substrings of a sequence is calculated, and these values are placed in a row of the matrix (
Figure 2). We repeat this operation for the desired number of filters (N_flt), and finally, by using max polling, we calculate the maximum score for each line.

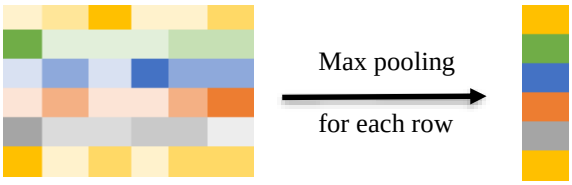


Figure 2. The first layer of Max Pooling.

3.3.2. Max pooling 2-N

The output of convolution layers 2 to N is a matrix, each cell of which is the result of combining several mers from the previous layer. In max-pooling layers 2 to N, some protected areas are selected with more certainty because we use overlap. The maximum value that falls in the overlapping area is a choice with a higher degree of importance. This operation is performed according to Figure 4 in each row with the number p of the Max Pooling operation, which is overlapping.

4. Analysis of the presented method

4.1. Dataset

The validity of the proposed method is checked using the COG database, a phylogenetic classification of proteins encoded in 21 complete genomes of eukaryotes, bacteria, and archaea [63].

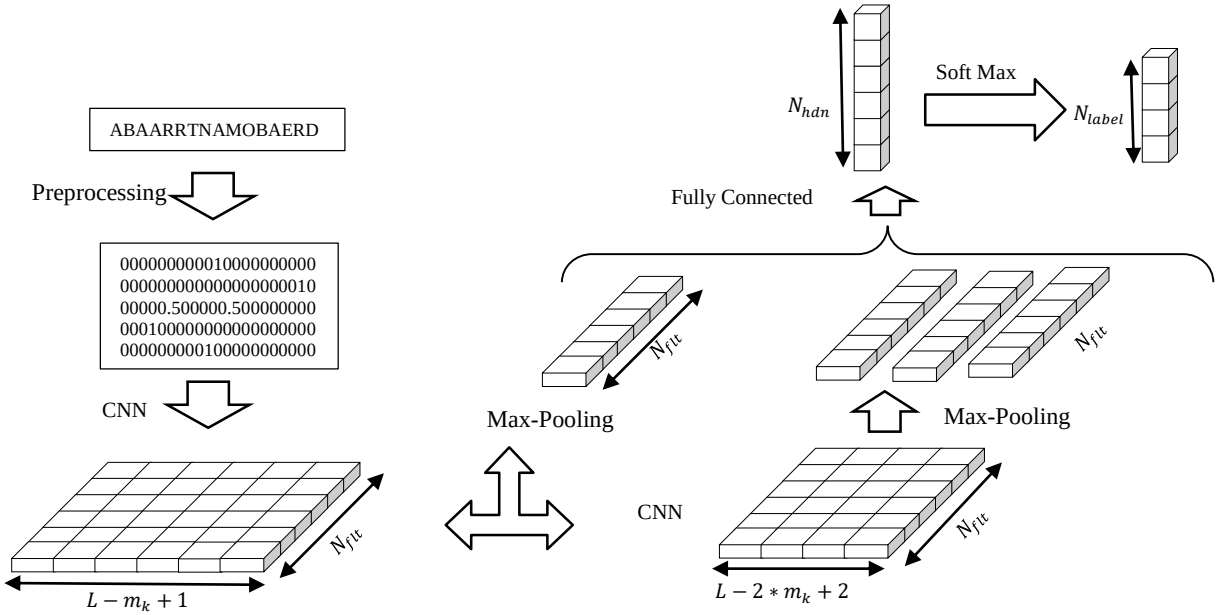


Figure 3. Details of the proposed method (variable N_{hdn} shows the number of nodes in the FC layer).

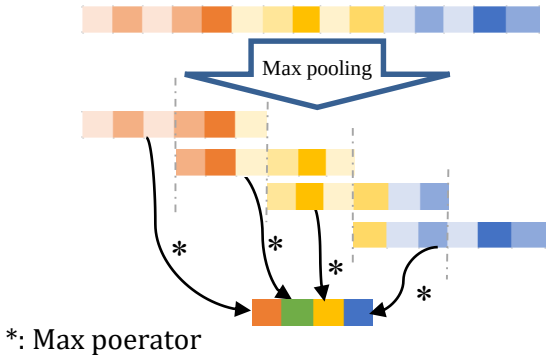


Figure 4. Max Pooling action for one row of the output matrix of convolution layers 2 to N.

Tatuso et al. [64] published the first version of this database in 1997, and Galperin et al. [65] made the latest version available to researchers on the NCBI website in 2014. Interpreting the functions of a cell's proteins is necessary to understand its functions. The COG database is one of the most widely used databases for interpreting the functions hidden in a cell's protein. For this reason, to evaluate the proposed model, we have used the COG database to classify proteins. The following discusses the applied filters [12] on the COG database. Proteins that belong to more than one class have been removed. Sequences with a length greater than 1000 have also been removed because the length of most sequences is less than 1000 [12]. By applying filters, the number of proteins is 1652408, and the number of groups is 4655. The threshold values are 100, 250, and 500; first, the classes with less than 100 samples are removed. The name of this database is COG-100-2829, and the number of proteins and their groups is shown in

Table 2. In the same way, the filter is also done for the threshold limit of 250 and 500, and their names are COG-250-1796 and COG-500-1074, respectively [12].

Table 2. Database specifications.

	Threshold		
	100	250	500
Dataset	COG-500-1074	COG-250-1796	COG-100-2829
#Group	2892	1796	1074
#Protein	1565976	1389595	1129428

A view of the database specifications after applying the filters can be seen in Figure 5. The average number of samples in classes is below two thousand in all three databases. In other words, many classes have less than two thousand samples, and a few contain more samples, although their number is less than others. We have performed several hypothesis tests with different criteria on three COG databases. It helps to recognize the nature of databases and understand their differences. Databases with more differences lead to better and more comprehensive model designs. To test the assumption, we first used different methods related to the normality of the data, the Shapiro-Wilk test [66], to check whether the data dispersion between different classes has a normal distribution. The answer to this test is negative for all three databases. Another test is the parametric

hypothesis test, in which we have used the t-test method for pairs of variables [67] to check whether there is a significant difference in data distribution in different classes between these three databases? The answer to this test was positive. Therefore, these three databases have differences from each other and are suitable for testing the designed model.

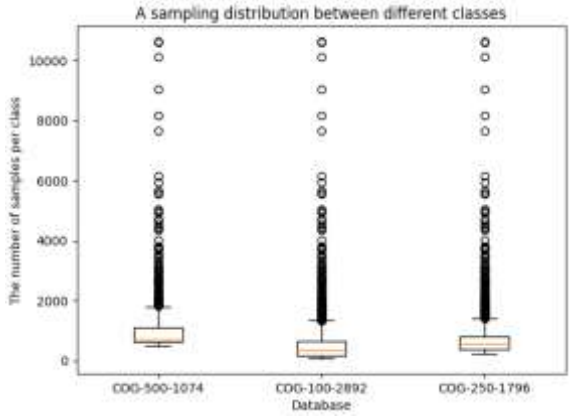


Figure 5. Distribution of samples in different databases.

4.2. Model parameters

A deep neural network has many parameters, and adjusting them is very time-consuming. To reduce the time needed to set the parameters in the presented model, we used the parameters presented in the DeepFam article [12]. In addition to these parameters, two more parameters have been added in this research: the network's depth and the number of steps, and the step length in Max Pooling for the Max Pooling layer is 2 to N (Table 3).

Table 3. The values of the parameters used in the model.

Parameters	values
N_{flt}: #Filters	250 [12]
N_{hdn}: #Nodes in FC layers	2000 [12]
m_k: size of kernel	8,12,16,20,24,28,32,36 [12]
λ: Regularizing factor	0.0005 [12]
Size of batch	100 [12]
Learning rate	0.001 [12]
Length of max pooling 2:N	$\frac{L}{2} - 2 * m_k + 1$
Length of max pooling step/p	$\frac{L}{4}$
# epoch	20

4.3. Evaluation of the presented model

We have used the COG database to evaluate the presented model. We have considered accuracy as the evaluation criterion and applied 3-fold cross-validation. The result of this evaluation can be seen in Table 4.

The PHMM method is one of the suitable methods for modeling protein functions with high accuracy, and the presented method has performed better than the PHMM method in all three mentioned experiments. With the number of categories increasing, our method is flexible and has performed better in the COG-100-2892 database, which has the most significant number of categories, with a relatively large difference from the other four models. The features extracted from the provided deep model help correctly assign the sequences to the respective categories.

Table 4. Evaluation of the presented method with other methods.

	COG-500-1074	COG-250-1796	COG-100-2892
The present method	95.93	95.15	93.91
DeepFam*	95.40	94.08	91.40
PHMM*	91.75	91.78	91.67
3-mer LR*	85.59	81.15	75.44
Protvec LR*	37.05	41.76	47.34

The number of changes in the PHMM method with three COG databases is less than that of the rest of the models, and the method presented in this article has better stability than other models after PHMM (Figure 6). The COG-100-2892 database is more challenging than other databases because it has a broader range of classes, and classes with sizes between 100 and 250 have less homology. Our method and the PHMM method have dealt well with the challenges of this class compared to the other two databases.

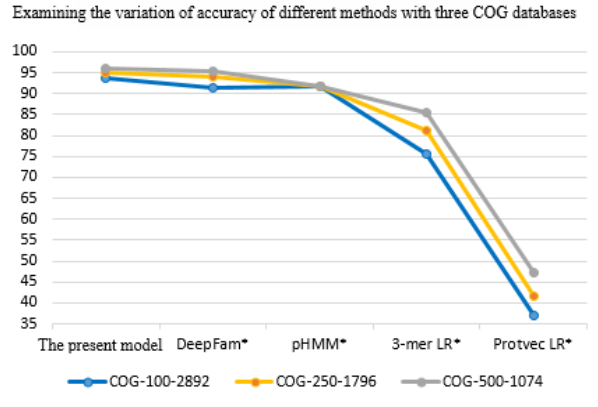


Figure 6. Examining the variation of accuracy of different methods with three COG databases.

In addition to the findings outlined earlier, we evaluated the model using several performance metrics, including Accuracy, Precision, Recall, F1-score, MCC, and AUC. The data were processed using 3-fold cross-validation for the training and testing phases. The table below provides details for each fold and the final micro average of each metric.

Table 5. Evaluation of model performance using various metrics.

		Accuracy	Precision	Recall	F1-score	MCC	AUC
COG-500-1074	0	95.953	95.953	95.953	95.953	95.946	0.99999
	1	95.945	95.945	95.945	95.945	95.938	0.99994
	2	95.892	95.892	95.892	95.892	95.886	0.99998
Average		**95.93**	**95.93**	**95.93**	**95.93**	**95.92**	**0.99997**
COG-250-1796	0	95.047	95.047	95.047	95.047	95.041	0.99999
	1	95.160	95.160	95.160	95.160	95.154	0.99990
	2	95.234	95.234	95.234	95.234	95.229	1.00000
Average		**95.15**	**95.15**	**95.15**	**95.15**	**95.14**	**0.99996**
COG-100-2892	0	93.844	93.844	93.844	93.844	93.839	0.99982
	1	93.952	93.952	93.952	93.952	93.947	1.00000
	2	93.948	93.948	93.948	93.948	93.943	0.99992
Average		**93.91**	**93.91**	**93.91**	**93.91**	**93.91**	**0.99991**

5. Interpretation of the presented method

a) Interpretation of performance in the convolution layer

Protein data consists of several amino acids, and the number and order of amino acids and the relationship of the sequences with each other are important. The large number of amino acids in a sequence complicates the extraction of important features from the sequences. Deep networks are a suitable solution to overcome the problems of this type of data. Protein sequences may belong to different categories with slight differences. For this reason, it is challenging to design a model that can detect slight differences between multiple sequences. In the presented model, as shown in Figure 7, squares 1 to 4 are calculated in the first convolution layer to extract local features. The result of these four squares is square 5 in the second convolution layer to extract global features. Suppose k in k-mer is equal to four, in the first convolution, the first character is only related to the next three characters, but in the second convolution, the first character is indirectly related to the next nine characters. Therefore, with the value of k in k-mer, more than $2 \times k - 1$ information is extracted, which helps to improve the prediction of protein categories.

b) Interpretation of features in the convolution layer

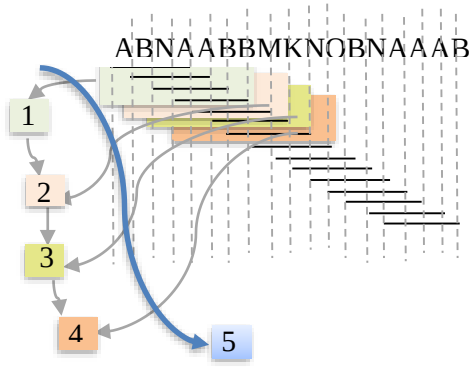
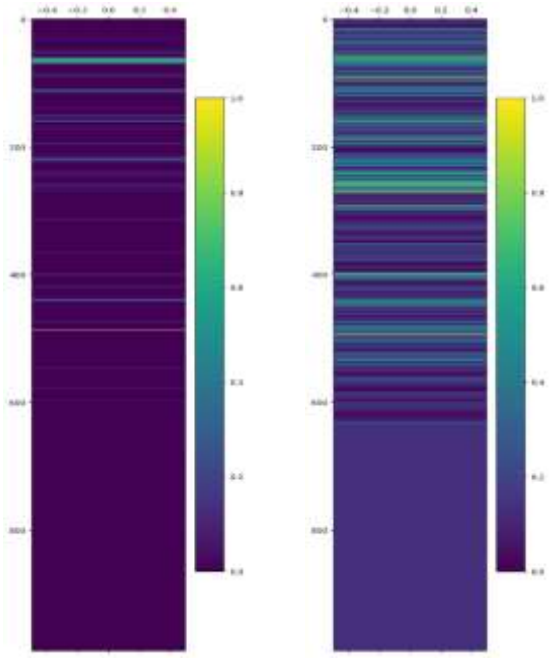


Figure 7. Operation of the first and second layer of convolution.

As explained, different methods exist to interpret the convolution network; we used the Grad-CAM method in this article. We compared two models (presented and DeepFam methods) to check the characteristics. In the
Figure 8, the significant areas are identified. This image is drawn for the layer with k-mer equal to 12. As you can see, the shape related to DeepFam has extracted the local features, and the decision is based on the local feature of the convolution layer. However, the figure related to the presented method is inclined to global and local features.



a) DeepFam method b) The method presented

Figure 8. Interpretation of convolution layer with k-mer=12.

To check the work's correctness, we have used transformer models. As explained in the previous sections, these models extract key features from the sequences. We have applied the desired sequence as input to the ESM model and obtained the

embedding vectors from this model. An embedding vector was extracted for each amino acid in a protein sequence and subsequently reduced in dimensionality using the t-SNE technique. Then, the norm of the embedding vector was used as a criterion for coloring each amino acid, providing a more intuitive visual representation of the relationships between amino acid features. The color dispersion of amino acids in Figures 8 (a) and 9 are closer. More details are in the attachment.

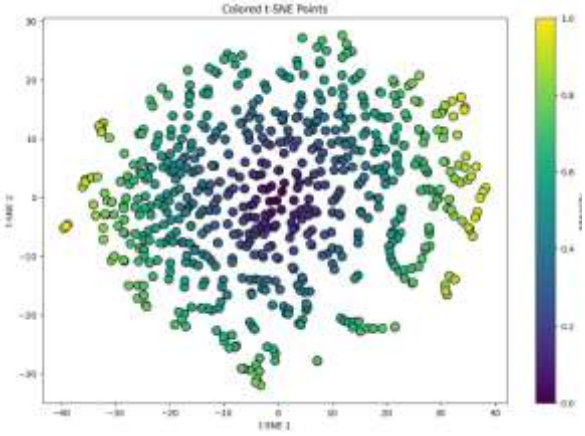


Figure 9. Display the output of the ESM model for sequence number one.

6. Conclusion

Convolutional networks effectively extract local features and identify homology in protein sequences. For protein sequence analysis, both local and global features are essential. We employed successive convolutional layers to model long-range dependencies between amino acids and utilized global feature extraction. The Grad-CAM method and transformer-based models were used to interpret the proposed model. Interpretability results indicate that the model can identify not only local features but also higher-level global features. The COG-100-2892 database exhibits the lowest homology between sequences compared to the other two databases. Analysis confirms that the proposed method performs better than DeepFam, PHMM, and other previous methods. These results suggest that our method has successfully uncovered long-range dependencies in protein sequences and does not classify proteins based solely on local homology. Therefore, employing consecutive convolutional layers provides a practical approach for extracting global features, improving protein sequences' classification accuracy.

References

[1] C. Yu, S.-Y. Cheng, R. L. He and S. S.-T. Yau, "Protein map: an alignment-free sequence comparison method based on various properties of amino acids," *Gene,* vol. 486, no. 1-2, pp. 110-118, 2011.

[2] F. Zhang, H. Song, M. Zeng, Y. Li, L. Kurgan and M. Li, "DeepFunc: a deep learning framework for accurate prediction of protein functions from protein sequences and interactions," *Proteomics,* vol. 19, no. 12, p. 1900019, 2019.

[3] P. Larranaga, B. Calvo, R. . Santana, C. Bielza, J. Galdiano, I. Inza, J. Lozano, R. Armananzas, G. . Santafe, A. Perez and V. Robles, "Machine learning in bioinformatics," *Briefings in bioinformatics,* vol. 7, no. 1, pp. 86-112, 2006.

[4] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li and H. Jiang, "Predicting protein--protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences,* vol. 104, no. 11, pp. 4337-4341, 2007.

[5] Y. Ge, S. Zhao and X. Zhao, "A step-by-step classification algorithm of protein secondary structures based on double-layer SVM model," *Genomics,* vol. 112, no. 2, pp. 1941-1946, 2020.

[6] Z. Lv, S. Jin, H. Ding and Q. Zou, "A random forest sub-Golgi protein classifier optimized via dipeptide and amino acid composition features," *Frontiers in bioengineering and biotechnology,* vol. 7, p. 215, 2019.

[7] C. L. P. Gupta, A. Bihari and S. Tripathi, "Protein Classification using Machine Learning and Statistical Techniques: A Comparative Analysis," *arXiv preprint arXiv:1901.06152,* 2019.

[8] O. Yakhnenko, A. Silvescu and V. Honavar, "Discriminatively trained markov model for sequence classification," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, IEEE, 2005, pp. 8--pp.

[9] W. Zheng, L. Yang, . R. J. Genco, J. Wactawski-Wende, M. Buck and Y. Sun, "SENSE: Siamese neural network for sequence embedding and alignment-free comparison," *Bioinformatics,* vol. 35, no. 11, pp. 1820-1828, 2019.

[10] B. Dogan, "An alignment-free method for bulk comparison of protein sequences from different species," *Balkan Journal of Electrical and Computer Engineering,* vol. 7, no. 4, pp. 405-416, 2019.

[11] S. Biđin, I. Vujaklija, T. Paradžik, A. Bielen and D. Vujaklija, "Leitmotif: protein motif scanning 2.0," *Bioinformatics,* vol. 36, no. 11, pp. 3566-3567, 2020.

[12] S. Seo, M. Oh, Y. Park and S. Kim, "DeepFam: deep learning based alignment-free method for protein family modeling and prediction," *Bioinformatics,* vol. 34, no. 13, pp. i254-i262, 2018.

[13] D. Zhang and M. Kabuka, "Protein Family Classification from Scratch: A CNN based Deep Learning Approach," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 2020.

[14] A. Dabba, A. Tari and D. Zouache, "Multiobjective artificial fish swarm algorithm for

multiple sequence alignment," *INFOR: Information Systems and Operational Research*, vol. 58, no. 1, pp. 38-59, 2020.

[15] M. S. Waterman, T. F. Smith and W. A. Beyer, "Some biological sequence metrics," *Advances in Mathematics*, vol. 20, no. 3, pp. 367-387, 1976.

[16] J. D. Thompson, D. G. Higgins and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic acids research*, vol. 22, no. 22, pp. 4673-4680, 1994.

[17] K. Katoh, K. Misawa, K.-i. Kuma and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," *Nucleic acids research*, vol. 30, no. 14, pp. 3059-3066, 2002.

[18] R. C. Edgar, "MUSCLE: a multiple sequence alignment method with reduced time and space complexity," *BMC bioinformatics*, vol. 5, no. 1, p. 113, 2004.

[19] C. Notredame, D. G. Higgins and J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment," *Journal of molecular biology*, vol. 302, no. 1, pp. 205-217, 2000.

[20] F. Naznin, R. Sarker and D. Essam, "Vertical decomposition with genetic algorithm for multiple sequence alignment," *BMC bioinformatics*, vol. 12, no. 1, p. 353, 2011.

[21] H. Zhu, Z. He and Y. Jia, "A novel approach to multiple sequence alignment using multiobjective evolutionary algorithm based on decomposition," *IEEE journal of biomedical and health informatics*, vol. 20, no. 2, pp. 717-727, 2015.

[22] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics (Oxford, England)*, vol. 14, no. 9, pp. 755-763, 1998.

[23] F. Naznin, R. Sarker and D. Essam, "Progressive alignment method using genetic algorithm for multiple sequence alignment," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 615-631, 2012.

[24] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proceedings of the National Academy of Sciences*, vol. 85, no. 8, pp. 2444-2448, 1988.

[25] W. R. Pearson, "Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms," *Genomics*, vol. 11, no. 3, pp. 635-650, 1991.

[26] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic acids research* M. Bhagwat, L. Young and . R. R. Robison, "Using BLAT to find sequence similarity in closely related genomes," *Current protocols in bioinformatics*, vol. 37, no. 1, pp. 1-41, 2012., vol. 25, no. 17, pp. 3389-3402, 1997.

[27] S. Schwartz, W. J. Kent, A. Smit, Z. Zhang, R. Baertsch, . R. C. Hardison, D. Haussler and W. Miller, "Human--mouse alignments with BLASTZ," *Genome research*, vol. 13, no. 1, pp. 103-107, 2003.

[28] B. Ma, J. Tromp and M. Li, "PatternHunter: faster and more sensitive homology search," *Bioinformatics*, vol. 18, no. 3, pp. 440-445, 2002.

[29] A. Chakraborty and S. Bandyopadhyay, "FOGSAA: Fast optimal global sequence alignment algorithm," *Scientific reports*, vol. 3, p. 1746, 2013.

[30] A. Wong, T. Reichert, D. Cohen and B. Aygun, "A generalized method for matching informational macromolecular code sequences," *Computers in biology and medicine*, vol. 4, no. 1, pp. 43-57, 1974.

[31] S. Batzoglou, L. Pachter, J. P. Mesirov, B. Berger and E. S. Lander, "Human and mouse gene structure: comparative analysis and application to exon prediction," *Genome research*, vol. 10, no. 7, pp. 950-958, 2000.

[32] M. Brudno, . C. B. Do, G. M. Cooper, M. F. Kim, E. Davydov, E. D. Green, A. Sidow and S. Batzoglou, "LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA," *Genome research*, vol. 13, no. 4, pp. 721-731, 2003.

[33] A. L. Delcher, A. Phillippy, J. Carlton and S. L. Salzberg, "Fast algorithms for large-scale genome alignment and comparison," *Nucleic acids research*, vol. 30, no. 11, pp. 2478-2483, 2002.

[34] N. Bray, I. Dubchak and L. Pachter, "AVID: A global alignment program," *Genome research*, vol. 13, no. 1, pp. 97-102, 2003.

[35] W. Huang, D. M. Umbach and L. Li, "Accurate anchoring alignment of divergent sequences," *Bioinformatics*, vol. 22, no. 1, pp. 29-34, 2006.

[36] S. Min, B. Lee and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851-869, 2017.

[37] N. Liu, J. Han, D. Zhang, S. Wen and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 362-370.

[38] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577-585.

[39] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294-3302.

[40] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, p. e0141287, 2015.

[41] M. Zeng, F. Zhang, F.-X. Wu, Y. Li, J. Wang and M. Li, "Protein--protein interaction site prediction through combining local and global features with deep neural networks," *Bioinformatics*, vol. 36, no. 4, pp. 1114-1120, 2020.

[42] W. Zhong and F. Gu, "Predicting Local Protein 3D Structures Using Clustering Deep Recurrent Neural Network," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

[43] B. Panda and B. Majhi, "A novel improved prediction of protein structural class using deep recurrent neural network," *Evolutionary Intelligence*, pp. 1-8, 2018.

[44] R. Jafari and . M. M. Javidi, "Solving the protein folding problem in hydrophobic-polar model using deep reinforcement learning," *SN Applied Sciences*, vol. 2, no. 2, p. 259, 2020.

[45] H. Hou, T. Gan, Y. Yang, X. Zhu, S. Liu, W. Guo and J. Hao, "Using deep reinforcement learning to speed up collective cell migration," *BMC bioinformatics*, vol. 20, no. 18, pp. 1-10, 2019.

[46] B. Liu, C.-C. Li and K. Yan, "DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks," *Briefings in Bioinformatics*, 2019.

[47] P. Baldi and G. Pollastri, "The principled design of large-scale recursive neural network architectures--dag-rnns and the protein structure prediction problem," *Journal of Machine Learning Research*, vol. 4, no. Sep, pp. 575-602, 2003.

[48] D. Bhowmik, S. Gao, M. T. Young and A. Ramanathan, "Deep clustering of protein folding simulations," *BMC bioinformatics*, vol. 19, no. 18, pp. 47-58, 2018.

[49] Y. Cao, T. A. Geddes, J. Y. H. Yang and P. Yang, "Ensemble deep learning in bioinformatics," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 500-508, 2020.

[50] Z. Guo, J. Liu, Y. Wang, M. Chen, D. Wang, D. Xu and J. Cheng, "Diffusion models in bioinformatics: A new wave of deep learning revolution in action," *arXiv preprint arXiv:2302.10907*, 2023.

[51] S. Zhang, R. Fan, Y. Liu, S. Chen, Q. Liu and W. Zeng, "Applications of transformer-based language models in bioinformatics: a survey," *Bioinformatics Advances*, vol. 3, no. 1, 2023.

[52] T. N. Kinyanjui, K. Mugoye and R. Kibuku, "Multi-Head Self-Attention Fusion Network for Enhanced Multi-Class Crop Disease Classification," *Journal of AI and Data Mining*, vol. 13, no. 2, pp. 227-240, 2025.

[53] V. Vimbi, N. Shaffi and M. Mahmud, "Interpreting artificial intelligence models: a systematic review on the application of LIME and SHAP in Alzheimer's disease detection," *Brain Informatics*, vol. 11, no. 1, p. 10, 2024.

[54] C. Molnar, "Interpretable machine learning," 2020.

[55] P. H. "Game theory: A Multi-leveled approach," 2015.

[56] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618-626.

[57] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher and N. F. Rajani, "Bertology meets biology: Interpreting attention in protein language models," *arXiv preprint arXiv:2006.15222*, 2020.

[58] "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, p. e2016239118, 2021.

[59] I.-I. Comm, "Abbreviations and symbols for nucleic acids, polynucleotides, and their constituents," *Biochemistry*, vol. 9, no. 20, pp. 4022-4027, 1970.

[60] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249-256.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[62] R. L. Tatusov, M. Y. Galperin, D. A. Natale and E. V. Koonin, "The COG database: a tool for genome-scale analysis of protein functions and evolution," *Nucleic acids research*, vol. 28, no. 1, pp. 33-36, 2000.

[63] R. L. Tatusov, E. V. Koonin and D. J. Lipman, "A genomic perspective on protein families," *Science*, vol. 278, no. 5338, pp. 631-637, 1997.

[64] M. Y. Galperin, K. S. Makarova, Y. I. Wolf and E. V. Koonin, "Expanded microbial genome coverage and improved protein family annotation in the COG database," *Nucleic acids research*, vol. 43, no. D1, pp. D261-D269, 2015.

[65] N. M. Razali, . Y. B. Wah and others, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests," *Journal of statistical modeling and analytics*, vol. 2, no. 1, pp. 21-33, 2011.

[66] R. C. Blair and J. J. Higgins, "Comparison of the power of the paired samples t test to that of Wilcoxon's signed-ranks test under various population shapes," *Psychological Bulletin*, vol. 97, no. 1, p. 119, 1985.

Appendix

We considered the following two sequences for the interpretation part of the model. To ensure greater clarity, we examined the effect of varying sequence scales by considering two sequences with different lengths (636 and 144).

Sequence 1:

MQKALFNLVLRGLEKQVPATGLGLFRLAFGLVAFQEICFLYYFRQLIFDPVPYLDIASPSVHLFLVLWAIAALCLALGLYTRLAAIANYLFWLVFTVFTPM
WKDFDGGFDQLMLGSSLLLIFLPSERAWSLDRLRLAWRHSTVDRCYALPRTVPVLCYFLPLAVSLGFIYFDSVIHKLFAEFWRNGLGPWLPSSLPYYMSPL
DMGWLLEIEPLQRAIGYTIIAFQFAFLFLLYFRRFRVPLMLVGLSLHAGIIVSLNIYPFGFGMLVHYFLMVPFRWWRTLGRTLRPAEPALQVFYDERCPLCL
KTVLAIEHFDVFRAVEFRGLQTHAATAPALEDIPERDLLGDLYAVDREGRRYSGVATYARILVAMRYPALAGLAMRLPGLATIADRVYRRIADNRVRLGC
DASCAPAPGRTEPDLAQRIGRWVGGSLQQRANRISRMLVVVLILQLNCTLHYAILYRLGVDTKANEAGQVLTMLSNALISASHTFLGITPHPLYLHDHFQG
YEHILGIVHLDADGKERWLPFVDEEGRIVSPNWGRVHSMWANVAVTRHMDPRRLDKFVRKVTAFWGTRLGLDLNRTTFVLKLKTVKAPMDWEPGLRR
YNLAQPWEDVGRAVWRDGEMRLELDRDLEALSAD

Sequence 2:

MKKRWALLGIVAAIIIIGVAGINYKMYKDKQAREVSVNSIFPKAKETIANMDGDIAVINNPNSMLVLVNKSRRLPDGYRPPDLVIPKVRYSSEGDQEKKKM
RKEAARALEDMFQQADNERIFLFAVSGFRSFDRQKALNTM

In the figure below, we have plotted the behavior of the convolutional layer for sequence number one with different kernel values. As shown, the convolutional layer behaves differently with kernels of varying sizes. However, on average, the method presented in this article considers more features, highlights the most important ones among these features, and uses local and global features in decision-making.

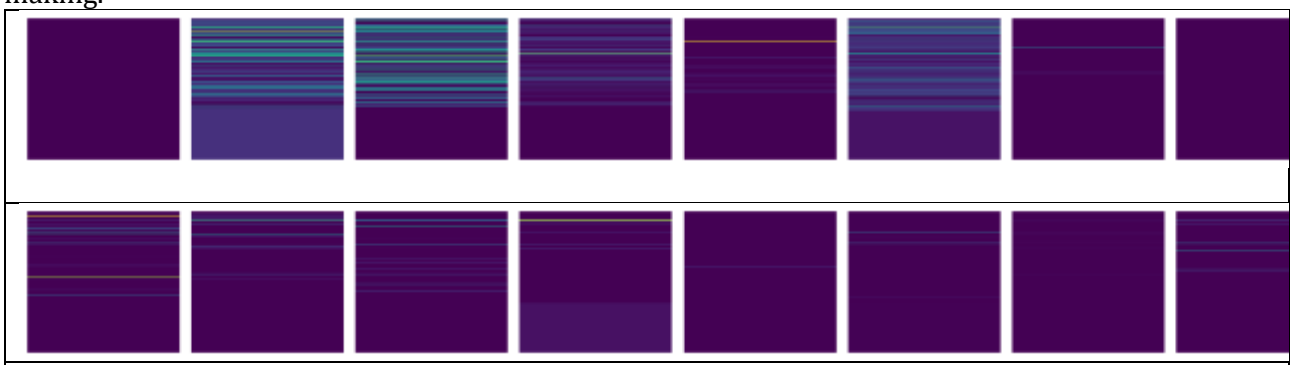


Figure 1. Comparison between the two presented models and the DeepFam model for the first sequence, respectively from the left, the convolution layer with different k-mers (from 8 to 36).

Figure 2. Comparison between the two presented models and the DeepFam model for the second sequence, respectively from the left, the convolution layer with different k-mers (from 8 to 36).

In the figure below, you can see the output of the ESM model for sequence number 2. First, we reduced the embedding vectors using the t-SNE method, and then a score was assigned to each amino acid. To give a score to each amino acid, we have calculated the distance of all points to the center, and with the help of this score,

the figure has been drawn. According to Figure 2, the behavior of convolution layers with k values (16, 20, 36) is closer to that of amino acids in Figure 3.

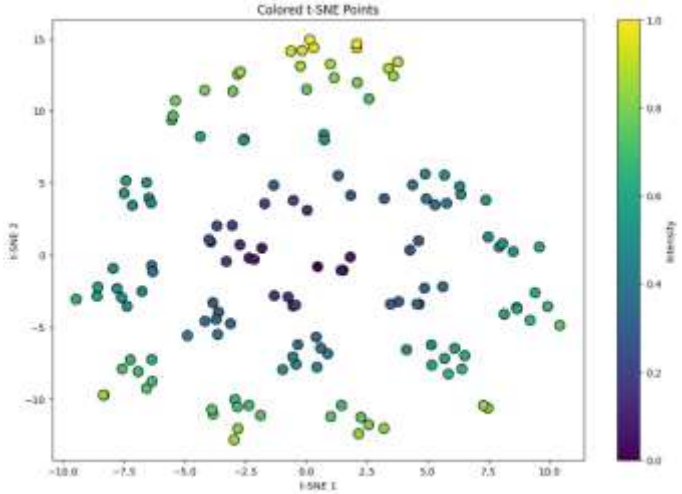


Figure 3. Plotting the output of the ESM model for sequence number two.

بهبود دسته‌بندی سلسله مراتبی خانواده های پروتئین و تفسیر مدل با روش Grad-CAM و ترنسفورمرها

نعیمه محمدکریمی و مهدی رضائیان*

دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران.

ارسال ۲۰۲۴/۱۲/۳۱؛ بازنگری ۲۰۲۵/۰۱/۲۷؛ پذیرش ۲۰۲۵/۰۲/۲۴

چکیده:

در عصر داده‌های حجیم، تحلیل رشته‌های بیوانفورماتیک و کشف توابع و عملکردهای آن از اهمیت بسزایی برخوردار است. نرخ تولید توالی‌ها با استفاده از تکنیک‌های تولید توالی به سرعت در حال افزایش است و محققان با توابع ناشناخته زیادی مواجه هستند. یکی از عملیات مهم در زمینه بیوانفورماتیک، دسته‌بندی توالی‌ها، به منظور کشف پروتئین های ناشناخته است. برای دسته‌بندی توالی‌ها، دو روش، روش سنتی و روش مدرن، وجود دارد. روش سنتی، از تطابق جفت توالی ها استفاده می‌کند که هزینه محاسباتی زیادی دارد. در روش مدرن، از استخراج ویژگی‌ها برای دسته‌بندی پروتئین‌ها استفاده می شود. در این راستا، روش‌هایی مانند DeepFam ارائه شده است. این پژوهش، بهبود مدل DeepFam است و تمرکز ویژه، روی استخراج ویژگی‌های مناسب جهت متمایز کردنِ توالی‌های دسته‌های مختلف می‌باشد. با بهبود مدل، ویژگی‌ها بیشتر متمایل به ویژگی‌های عمومی شدند. برای بررسی ویژگی‌های استخراج شده از روش Grad-CAM به منظورِ تفسیر لایه‌های شبکه بهبود یافته استفاده شده است. سپس بردار جاساز از مدل ترنسفورمر به منظور بررسی عملکرد Grad-CAM به کار بردیم. برای بررسی صحت عملکردِ روش ارائه شده از پایگاه داده COG که یک پایگاه داده حجیم از توالی‌های پروتئینی محسوب می‌شود، استفاده شده است. نشان داده‌ایم با استخراج ویژگی‌های کاراتر، نواحی حفاظت شده در توالی‌ها با دقت بیشتر کشف می شوند و به دسته‌بندی مطلوب‌تر پروتئین‌ها کمک می‌کند. یکی از مزیت‌های مهم روش ارائه شده این است که با افزایش تعداد دسته‌ها، انعطاف پذیری لازم، حفظ می‌شود و دقت دسته‌بندی در سه آزمایش، از دیگر روش‌ها، بالاتر است.

کلمات کلیدی: دسته‌بندی پروتئین، یادگیری عمیق، شبکه عصبی CNN، استخراج ویژگی، تفسیر پذیری، مدل‌های ترنسفورمر.