International Journal of Biology and Life Sciences
ISSN: 2957-9511 | Vol. 2, No. 3, 2023

Prediction of Protein Secondary Structure based on Multi-scale Convolutional Neural Network

Yu Xiao [a], Xiaozhou Chen [b, *]

School of Yunnan Minzu University, Mathematics and Computer Science, Kunming, China

[a] 20213037560002@ymu.edu.cn, [b, *] chxiaozhou@163.com

[*] **Corresponding author**: Xiaozhou Chen (Email: chxiaozhou@163.com)

Abstract: In the field of bioinformatics, the prediction of secondary structure of proteins is very important. It can be obtained from the prediction of primary structure (amino acid sequence) and can provide reference for the prediction of tertiary structure of proteins. Amino acid sequences of proteins are encoded with several features and then combined into the prediction network. Convolutional neural network has excellent performance in text and sequence information extraction. The amino acid sequence of protein is also a special sequence, so the convolutional neural network can be used to extract the information in the sequence. Moreover, the influence of amino acids on the formation of secondary structure varies with different distances, so in the experiment, convolutional neural networks with convolution nuclei of different sizes were used to form multi-scale convolution blocks to extract amino acid sequence information. At the same time, the sliding window technique is also used to show the interaction between the sequences, and a long amino acid sequence is divided into some amino acid fragments and input into the model. Finally, the accuracy of Q8 on the dataset CB6133_filtered reaches 71%.

Keywords: Protein Secondary Structure Prediction; Convolutional Neural Network; Sliding Window Technology; Feature Fusion.

1. Introduction

In the field of proteomics, an effective predictor of protein secondary structure with high accuracy has been very important [1], especially when the structure of an amino acid sequence fragment cannot be solved by high resolution experiments, such as X-ray crystallography, cryo-electron microscopy and NMR spectroscopy [2]. These experiments usually cost a lot of time and money, the experimental results will be interfered by various factors, cannot get ideal results. Due to various inconvenient factors in the physical experiment and the rapid development of computer network technology, we combined with the previous experience method, proposed to use the computer network to build a learning machine, using this machine to learn the protein with known secondary structure and amino acid sequence, and get the best correlation between secondary structure and amino acid sequence. The unknown secondary structure of a protein is thus predicted from its known amino acid sequence.

In the 21st century, various machine learning methods, especially artificial neural networks, have been used to improve performance, such as support vector machines [3], cyclic neural networks [4] (RNN), conditional neural field combination [5] and other probabilistic graphical models have been widely used. In recent years, big data, deep learning methods and other technologies have been widely applied, and it has become a research trend to predict the secondary structure of proteins combined with deep learning models. Among various deep learning methods, convolutional neural network (CNN) is well known for its excellent performance in processing images and sequences.

To predict protein secondary structure by amino sequence, each amino acid residue in amino sequence similar to natural language needs to be encoded, that is, transformed into corresponding numerical expression. After processing the feature encoded data and then input into the learning machine, the secondary structure classification can be carried out effectively. Moreover, the accuracy of the classification model is closely related to the encoding method of data characteristics. PSSM [6] spectral encoding and orthogonal encoding are often used in the secondary structure of proteins. In order to improve the classification effect, this paper adds the encoding method, including the physical and chemical properties of 7 amino acids.

Convolutional neural network is a specific type of deep neural network using translation invariant convolutional kernel, which can be used to extract local context features and has been shown to be effective in many natural language processing tasks [7]. Inspired by the success of convolutional neural network in natural language text, and the amino acid sequence of protein is also a special language sequence, it is a very feasible method to use convolutional neural network to extract amino acid sequence information. Experiments show that convolutional neural network does perform well in the prediction of protein secondary structure. Analysis of the secondary structure of the subject amino acid and the target protein shows that the formation of each secondary structure depends not only on the corresponding amino acid, but may also depend on the amino acid interacting with it. This interaction may result from the co-determination of the short or long range of different distances between the amino acid residues. Since the specific distance length between amino acids that interact with each other is uncertain, this paper uses CNN of convolution kernels of different sizes to extract short-range information from amino acid sequences.

2. Materials and Methods

2.1. Data Sets

In this paper, the data set CB6133 generated by PISCES CullPDB [8] and the public benchmark data set CB513 are used. There were 6128 protein sequences in CB6133 dataset and 514 protein sequences in CB513 dataset. Because dataset CB6133 contains protein sequences with homology greater than 25% from CB513, the deleted dataset is named CB6133_filtered, with a total of 5534 protein sequences. There are two ways to input the data for the learning model. First, 4427 amino acid sequences in 80% of the dataset CB6133_filtered are regarded as the training set, 10% have 553 amino acid sequences as the verification set, and 10% have 554 sequences as the test set. In addition, 90% of the data in the CB6133_filtered dataset is regarded as the training set, with 4980 amino acid sequences in total, 10% as the test set with 554 sequences in total, and finally the entire CB513 dataset is regarded as the test set. Then the sliding window technique was used to segment the sequence, and finally there were 1183318 amino acid fragments in dataset CB6133 and 84765 amino acid fragments in dataset CB513.

2.2. Data Preprocessing

2.2.1. PSSM

Location specific score matrix is a feature extraction method based on protein evolution information. PSSMS are generated by using the iterative BLAST method to discover more protein sequences that are evolutionarily related to the search sequence. The first BLAST search is performed on the query sequence, and the result of that search (higher than the selected E value) is used as the query sequence for the second BLAST search. After a second search, more sequences with evolutionary information can be found, and the process is repeated until no meaningful similar sequences are found. Given a protein sequence, the PSSM matrix is obtained:

$$PSSM = \begin{bmatrix} p_{1\to 1} & \cdots & p_{1\to 20} \\ \vdots & \ddots & \vdots \\ p_{L\to 1} & \cdots & p_{L\to 20} \end{bmatrix} \tag{1}$$

Where $p_{i\to j}$ represents the score of the i-th amino acid replaced by the j-th amino acid in the protein, and serial number 1-20 represents one of the 20 basic amino acids in alphabetical order. sigmoid function is used to normalize the elements of pssm matrix. The calculation formula of sigmoid function is as follows:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Through normalization processing, the value of pssm matrix is in the interval of [0,1], and the size of N×20 PSSM matrix is obtained.

2.2.2. Orthogonal Coding

The 20 basic amino acids of proteins are represented by the letters A, C, E, D, G, F, I, H, K, M, L, N, Q, P, S, R, T, W, V, and Y. Orthogonal coding is to represent each amino acid in the amino acid sequence of a protein as a 21-dimensional vector with the values of element 0 and 1, line up the basic amino acid of a protein in the above order, and add the letter X after the basic amino acid letter to represent some specific types of unknown amino acids, because the specific type cannot be determined by experimental methods sometimes. The amino acids of the target protein are arranged in a sequence, and each amino acid is compared with the basic amino acids in a row. The positions consistent with the basic amino acids are represented by 1, and different positions are represented by 0. Finally, the orthogonal matrix is obtained, which is the number of amino acids of the target protein.

2.2.3. Physical and Chemical Properties of Amino Acids

Considering that the feature coding used to predict the secondary structure of proteins was too simple, most of them adopted orthogonal coding and PSSM spectrum coding. In order to more comprehensively explore the relationship between the secondary structure and the amino acid sequence of proteins, the physical and chemical properties of amino acids were introduced in the experiment for coding.

2.3. Experimental Design

The deep learning model see Figure 1 was used in the experiment to predict the secondary structure of proteins, and a good performance was obtained. The model consists of three parts: feature embedding layer, multi-scale convolutional neural network layer and full connection layer. The feature embedding layer is used to map sparse data to dense data, and the multi-scale CNN layer is used to extract sequence information in various nuclear size ranges. The specific setting of nuclear size is based on the influence of various amino acid interactions on secondary structure, as shown in the table. This design can fully extract the sequence information, and then integrate the information through the full connection layer. Finally, the SoftMax activation function is used to classify the secondary structure.

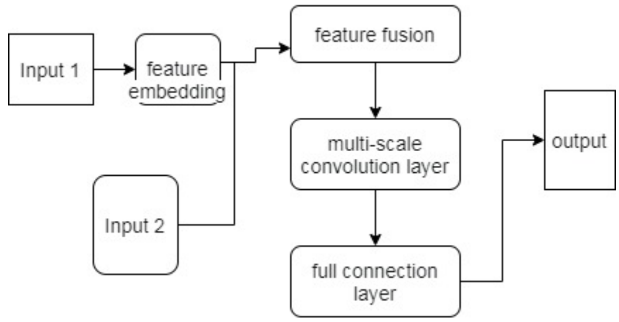

Figure 1. Multi-scale convolution model structure for predicting protein secondary structure

Feature embedding layer was used to map sparse data to dense data, and CNN was used to extract the feature information of amino acid sequence. In recent years, convolutional neural networks have been very effective in extracting information and have been applied in many fields. In particular, one-dimensional convolutional neural networks have performed well in extracting text and sequence information, and two-dimensional convolutional neural networks have performed well in extracting picture information. By comparing the process of image convolution, it is found that the essence of the image is a two-dimensional sequence of gray level, and there is a corresponding value on each pixel point. The features within the size range of kernel are extracted by using convolution kernel (filter). After the amino acid sequence data is processed into a numerical matrix with a size of (700, d), the values in this numerical matrix can be viewed as the values of each pixel in the image. The prediction of protein secondary structure is carried out by using the amino acid sequence of primary structure, because the sequence can use one-dimensional convolutional neural network to extract the hidden information in the sequence to predict its secondary structure, and the sequence can be processed as a matrix with features, so the information can be extracted by two-dimensional convolution. The influence of amino acids on the secondary structure varies with different distances, so in this paper, convolutional neural networks with different nuclear sizes are used to extract amino acid sequence information of proteins within a specified range.

2.3.1. Sliding Window Technology

The sliding window technique is often used to predict the secondary structure of proteins. We know that the primary structure of protein is the amino acid sequence, and the amino acid sequence is arranged in order. Therefore, sliding window technique is used here to present the interaction between these sequences. Generally, the size of the selection window is odd. The middle amino acid in the amino acid fragment selected in the sliding window is the secondary structure of the amino acid to be predicted in the experiment (target amino acid), and the adjacent amino acids before and after are input into the network as the influence factors of the secondary structure [9]. As for the selection of the size of the sliding window, according to the study of Yan Fu-ming, it is known that the long-range influence between amino acids is very important for the prediction of secondary structure. Through the experimental analysis, it is found that the experimental performance is better when the size of the sliding window is selected as 17. After the size of the sliding window is selected, the amino acid sequence of the protein is supplemented with 8 zeros at the beginning and end. The sliding window technique not only expresses the interaction between amino acids, but also increases the amount of data input to the network, which will improve the performance of the network prediction.

2.3.2. Feature Embedding

For convenience of memory, the protein sequence is expressed as: $P = \mathrm{a}_1, a_2, \cdots, a_n$, a_i representing the i-th amino acid residue, n is the total number of amino acids in the protein sequence. Through the sliding window technique introduced above, the protein amino acid sequence is slid by the window size of 17 to obtain an amino acid fragment composed of 17 amino acids, which can be expressed as formula $\tilde{a}_i = a_{i-8}, a_{i-7}, \cdots, a_i, \cdots, a_{i+7}, a_{i+8}$. The secondary structure of the protein sequence is expressed as formula: $S = s_1, s_2, \cdots, s_n$, s_i represents the secondary structure corresponding to the i-th amino acid.

Each amino acid residue was characterized by orthogonal coding and PSSM spectral coding. When the feature of amino acid residue is represented by orthogonal coding, it is a sparse vector $x_i^1 = (0, \cdots, 1, 0, \cdots, 0)$, so the input data feature is a sparse matrix of 17×21 represented by X_i^1, and the amino acid feature is represented by a 20-dimensional dense vector $x_i^2 = (p_{i1}, p_{i2}, \cdots, p_{i20})$ by PSSM spectral coding. p_{i1} is the standardized value of the fraction of the i-th amino acid residue replaced by the first amino acid. This gives you a spectral coded dense matrix of 17×20 in terms of X_i^2. Since the input features of amino acid sequences are divided into two parts of different types, in order to make the feature representation types consistent when they are input into the convolutional neural network layer, the 21-dimensional orthogonal encoded feature vectors are mapped into dense vectors using the embedding layer. The specific operation is to input the first type of data features into the feature embedding layer composed of the fully connected neural network layer. In the experiment, the mapping dimension will affect the performance of prediction. The larger the mapping dimension is, the better the performance will be, but it is impossible to enlarge indefinitely. Finally, we choose to set the output dimension of the embedded layer to 128, the weight matrix to $W \in \mathbb{R}^{21\times128}$, the Glorot uniform distribution initialization method to initialize the weight, and the bias item to be initialized to zero. The linear transformation $f = X_1 \cdot W + b$ yields a 128-dimensional dense matrix represented by $\tilde{X}_i^1$.

2.3.3. Multi-scale Convolution Design

The feature of the input amino acid fragment sequence ($\tilde{a}_i$) after fusion is expressed as $X_i = \left(\tilde{X}_i^1, X_i^2\right)$, where the dimension of each feature vector X is 163. In order to simulate the local dependence of adjacent amino acids, CNN with sliding window and linear correction function is used to extract short-range information. We input the eigenmatrix X_i of amino acid fragment ($\tilde{a}_i$) into the two-dimensional convolutional neural network, where the convolution kernel is F_f and the size is $f \times m$ (f is the size of the convolution kernel sliding along the amino acid sequence at one time, m is the dimension 149 of the amino acid feature), $\mathrm{Re}LU$ is the activation function W is the weight, and b is the bias vector. After the following convolution operation:

$$\hat{x}_j = F_j \otimes x_{j:j+f-1} = LeakyR\mathrm{e}LU\left(W * x_{j:j+f-1} + b\right)$$
$$j = i-8, \cdots, i, \cdots, i+8 \tag{3}$$

The feature map $\hat{X}_{if} = \left[\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{17}\right]$ of amino acid sequence ($\tilde{a}_i$) with respect to the convolution kernel F_f is obtained. Here, the convolution kernel varies in size. In order to fully extract the interaction between amino acids, convolution layers of multiple convolution kernels are used in the convolution part of the experiment through parallel connection or downward stacking. see Figure 2. each convolution layer and batch normalization layer and Droppout layer added in order to prevent gradient explosion.

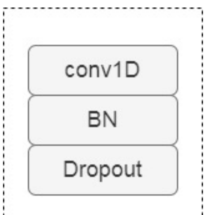

Figure 2. The specific setting of the convolution layer

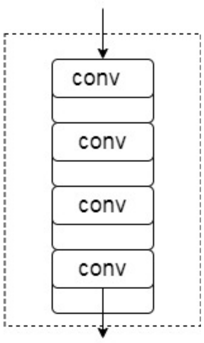

Figure 3. Structure of multi-scale convolution block

see Figure 3. the convolution layers of each convolution kernel of different sizes are directly stacked downward. Similarly, two layers are set for each convolution layer of the same convolution kernel, and the size of the convolution kernel is set to.

2.4. Experimental Optimization

2.4.1. Loss Function

The protein secondary structure label is in the orthogonal coding form, and the corresponding predicted label form is a probability array, where each probability is the probability that each amino acid in the target amino acid sequence corresponds to 8 secondary structures respectively, and the total probability sum is 1. According to the categorical_crossentropy loss function, we adopt the categorical_crossentropy loss function, the formula is as follows:

$$\text{Loss} = -\sum_{i=1}^{\substack{output\\size}} y_i \times \log \hat{y}_i \tag{4}$$

2.4.2. Optimization Function

SGD (stochastic gradient descent algorithm) is updated iteratively once according to each sample data. It only updates once at a time, so there is no redundancy, and it is faster, and new samples can be added. Although the iteration speed of SGD is fast, there is a cost associated with decreased accuracy. It is not the global optimal solution, but the final result is near the global optimal solution. Stochastic gradient descent algorithms can also cause severe shocks due to frequent updates. Its formula is expressed as follows:

$$\theta = \theta - \eta . \nabla_\theta J\left(\theta; x^{(i)}; y^{(i)}\right) \tag{5}$$

Based on SGD algorithm, Adagrad algorithm is proposed later, which improves the robustness of SGD algorithm. The low frequency parameters can be updated to a large extent, and the high frequency parameters can be updated to a small extent, which also has a good performance on sparse data. The most important thing is to reduce the manual adjustment of learning rate. RMSprop algorithm is an adaptive learning rate method proposed by Geoff Hinton, which solves the problem of sharp decline in learning rate of Adagrad algorithm. The method of exponential weighted average is used to eliminate the oscillations in gradient descent. The specific update is expressed as:

$$E\left[g^2\right]_t = 0.9E\left[g^2\right]_{t-1} + 0.1g_t^2 \tag{6}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E\left[g^2\right]_t + \varepsilon}} g_t \tag{7}$$

2.4.3. Activating Functions

Common activation functions mainly include Sigmoid, tanh, ReLU and ReLU6. The most commonly used activation function in convolutional neural networks is ReLU [10], whose formula expression is as follows:

$$f(x) = \max(0, x) \tag{8}$$

As can be seen from the formula, the function will compare the input data with 0 and select the data greater than 0, which will greatly speed up the network calculation speed. But, for better or worse, in speeding things up, this function could condemn some neurons to a death sentence and never fire. Therefore, some parameters of the network will not update the data and change their distribution, making the network unable to continue learning. The usual way to deal with the problem of changing data distribution is to add batch normalization to convolution. LeakyReLU as an activation function is a variant of LeakyReLU, which does a good job of avoiding problems with the ReLU function and alleviating model death. While LeakyReLU can also update the parameters of the network at $x \le 0$ to prevent the death of the model. The LeakyReLU function is defined as:

$$f(x) = \max(0.01x, x) \tag{9}$$

2.4.4. Experimental Test Index

In order to evaluate the excellent performance of the experimental model, the accuracy of Q8 is usually used to evaluate the model in the experiment of protein secondary structure prediction using amino acid sequence. The higher the accuracy of Q8, the better the prediction performance of the model for protein secondary structure. The formula is expressed as follows:

$$Q_8 = \frac{\sum_{i=1}^{8} T_i}{N} \times 100 \tag{10}$$

Where N represents the number of all amino acids in the evaluated data set, and T_i is the number of class i secondary structures correctly predicted as Class i secondary structures. There are a total of 8 secondary structures.

3. Experimental Results and Analysis

3.1. Experimental Results

When two layers are set in the convolutional network, the number of convolutional nuclei is set to 300, and the size of convolutional nuclei is set to, the convolutional layers of convolutional nuclei of different sizes are connected in parallel to form multi-scale convolutional blocks. The learning rate is set to 0.00003, the batch size is set to 512, and the number of iterations is set to 100. The dropout of 0.5 and l2 regularization of 0.05 are added to prevent overfitting of the model. EarlyStopping is also set to stop training in advance, when the model does not improve for 10 consecutive epochs, and finally the model stops training for 77 times. Two fully connected layers were set, and the number of neurons was set to 256 and 128 respectively.

On the dataset CB6133_filtered, the training accuracy reaches 0.7157, and the test accuracy reaches 0.7221. It can be seen from see Figure 4. that the convergence speed of the model is very fast. see Table 1. the accuracy, precision, recall rate, and Q8 accuracy of each data set.

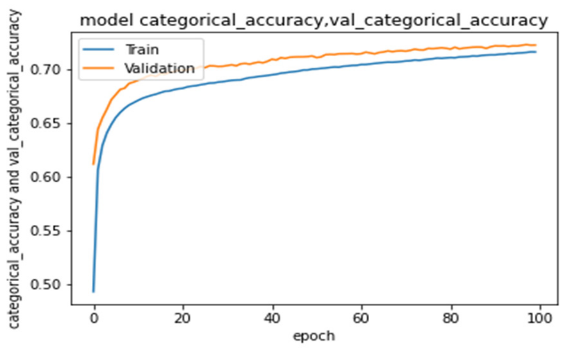

Figure 4. Prediction accuracy of the model on the dataset CB6133_filtered

Table 1. The accuracy, accuracy, recall rate and second-level structure Q8 accuracy of the model on the dataset CB6133_filtered

	accuracy	precision	recall	Q8
Training set	0.7157	0.8419	0.6033	
Verification set	0.7221	0.8323	0.6239	
Test set	0.7172	0.8155	0.6222	0.7172

3.2. Comparative Analysis

The influence of sliding window size on model performance was analyzed. In this regard, we conducted the following experiments, keeping all parameters of the experimental model unchanged, and setting sliding Windows of different sizes as 13, 15, 17 and 19. The influence of sliding window size on the experiment was compared by the accuracy of data set CB513. As shown in Table 4-4, when other parameters remain the same, the experimental model performs best when the window size is 17. see Table 2.

Table 2. The effect of sliding window size on model performance

Sliding window size	13	15	17	19
accuracy	0.6720	0.6754	0.7053	0.6832

The influence of convolution kernel size on experimental results was analyzed. As in the above analysis of the size of the sliding window, the size of the convolution kernel is changed by keeping all other parameters unchanged. Here, the convolution kernel size is set as 1, 3, 5, 7, 9, 11, 13, and the results are shown in Table 4-5. When the convolution kernel size is 1, the prediction performance of protein secondary structure on dataset CB513 is not very good. When the convolution kernel size is 11, the performance is the best; when the convolution kernel size is 7, 9, 11, the experimental results are not different. see Table 3.

Table 3. Effect of convolution kernel size on model performance

Convolution kernel size	1	3	5	7	9	11	13
Accuracy	0.4185	0.5363	0.5789	0.6754	0.5770	0.5881	0.5890

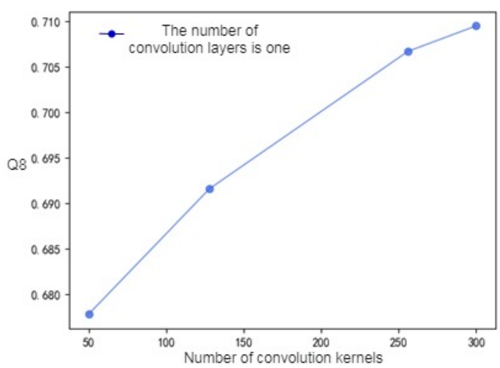

Figure 5. Prediction accuracy of the model on the dataset CB6133_filtered

The influence of the number of convolution nuclei on the experimental results was analyzed. Based on the above analysis, the sliding window size was set to 17 and multi-scale convolution was used to predict the protein secondary structure. Here we set the convolution kernel size to, which is experimentally shown to be better, and we stack each convolution layer of different sizes in order from smallest to largest. If other parameters of the model remain unchanged, the number of convolution nuclei in each layer will be changed, so as to analyze the influence of the number of convolution nuclei on the prediction of protein secondary structure. The analysis results are shown in the figure. It can be seen from Figure 4-8 that the more convolution nuclei, the better the convolution effect will be. Here, when the number of convolution nuclei is 300, the experiment performs best, and the accuracy rate reaches 0.7095 on the dataset CB6133_filtered. From there, for later experiments, you can choose to set the convolution kernel size to 300. see Figure 5.

4. Conclusion

In order to predict the secondary structure of eight types of proteins, the multi-scale convolutional neural network was used to predict the secondary structure of proteins, and the optimal model was obtained after several optimization experiments. The influence of each parameter on the experimental results is analyzed from various angles. The secondary structure of proteins can be predicted by amino acid sequence without spending a lot of time and money.

Acknowledgments

Thank professor of Xiaozhou Chen for guiding me in the direction of my research and the significance of the research results, and thank my senior brothers and sisters for answering my doubts about experimental methods. Special thanks the editor in chief and worthy reviewers for valuable suggestions for giving the final shape of the manuscript. This work was supported in part by a grant from the National Natural Science Foundation of China (Grant No.31460297).

References

[1] Yang Y, Gao J, Wang J, et al. Sixty-five years of the long march in protein secondary structure prediction: the final stretch[J]. Briefings in Bioinformatics, 2018, 19(3): 482–494.

[2] Noble M, Endicott J A , Johnson L N . Protein Kinase Inhibitors: Insights into Drug Design from Structure[J]. Science, 2004, 303(5665):1800-1805.

[3] Kandoi G, Leelananda S P, Jernigan R L, et al. Predicting protein secondary structure using consensus data mining (CDM) based on empirical statistics and evolutionary information[M]. Prediction of Protein Secondary Structure. Humana Press, New York, NY, 2017: 35-44.

[4] Spencer M, Eickholt J, Cheng J. A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction[J]. Computational Biology& Bioinformatics IEEE/ACM Transactions on, 2015, 12(1):103-112.

[5] Patel M S, Mazumdar H S. Knowledge base and neural network approach for protein secondary structure prediction[J]. Journal of Theoretical Biology, 2014, 361:182-189.

[6] WANG S, LI M, GUO L, et al. Efficient utilization on PSSM combining with Recurrent Neural Network for membrane protein types prediction[J]. Computational Biology and Chemistry, 2019, 81:9-15.

[7] Gao F, Yue Z , Wang J , et al. A Novel Active Semisupervised Convolutional Neural Network Algorithm for SAR Image Recognition[J]. Comput Intell Neurosci, 2017, 2017 (24): 3105 053.

[8] Zhao Y, Liu Y . OCLSTM: Optimized convolutional and long short-term memory neural network model for protein secondary structure prediction[J]. PLOS ONE, 2021, 16.

[9] Majid Vafaeipour, Omid Rahbari, Marc A. Rosen,Farivar Fazelpour,Pooyandeh Ansarirad. Application of sliding window technique for prediction of wind velocity time series[J]. International Journal of Energy and Environmental Engineering, 2014,5(2-3).

[10] Glorot X, Bordes A , Bengio Y . Deep Sparse Rectifier Neural Networks [J]. Journal of Machine Learning Research, 2011, 15:315-323.