Hindawi
Computational and Mathematical Methods in Medicine
Volume 2022, Article ID 9705275, 10 pages
https://doi.org/10.1155/2022/9705275

Research Article

Research on DNA-Binding Protein Identification Method Based on LSTM-CNN Feature Fusion

Weizhong Lu,[1,2] Xiaoyi Chen,[1] Yu Zhang,[3] Hongjie Wu,[1] Yijie Ding,[4] Jiawei Shen,[1] Shixuan Guan,[1] and Haiou Li[1]

[1] *School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China*
[2] *Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, China*
[3] *Suzhou Industrial Park Institute of Services Outsourcing, Suzhou 215123, China*
[4] *Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, 324000, P.R, China*

Correspondence should be addressed to Hongjie Wu; hongjiewu@usts.edu.cn

Received 25 November 2021; Revised 23 December 2021; Accepted 27 April 2022; Published 2 June 2022

Academic Editor: Prem Chapagain

Protein is closely related to life activities. As a kind of protein, DNA-binding protein plays an irreplaceable role in life activities. Therefore, it is very important to study DNA-binding protein, which is a subject worthy of study. Although traditional biotechnology has high precision, its cost and efficiency are increasingly unable to meet the needs of modern society. Machine learning methods can make up for the deficiencies of biological experimental techniques to a certain extent, but they are not as simple and fast as deep learning for data processing. In this paper, a deep learning framework based on parallel long and short-term memory(LSTM) and convolutional neural networks(CNN) was proposed to identify DNA-binding protein. This model can not only further extract the information and features of protein sequences, but also the features of evolutionary information. Finally, the two features are combined for training and testing. On the PDB2272 dataset, compared with PDBP_ Fusion model, Accuracy(ACC) and Matthew's Correlation Coefficient (MCC) increased by 3.82% and 7.98% respectively. The experimental results of this model have certain advantages.

1. Introduction

Protein is the most important component in the organism. It is closely related to life activities. As a special protein, DNA-binding protein can interact and bind with DNA to form different structures and functions [1]. The interaction between the protein and DNA is an important basis for cell life activities, which not only can achieve multiple functions such as DNA transcription and replication, but also it has a key role in the regulation of organisms. Therefore, the subject of DNA-binding protein [2] prediction is particularly significant. The prediction of DNA-binding protein is to judge whether a protein can combine with DNA. At present, in accordance with different feature information, the prediction of DNA-binding protein can be split into two categories, one is the method relied on protein structure feature information, and the other is the method relied on protein sequence feature information. Generally, methods relied on protein structure feature information have superior prediction results.

However, the traditional biometric technologies, such as filter combination analysis, X-ray diffraction and other methods, have gradually lagged behind the needs of modern society. Although the prediction accuracy rate is high, but they need strict experimental environment and accurate experimental equipment. These methods are high cost and low efficiency. In particular, the number of protein sequences has increased a lot, and these methods are becoming less and less applicable, which limits the research of proteins. Today, due to the cross-age development of computers, more time-saving and labor-saving machine learning methods [3] have come into the sight of researchers. Many researchers used machine learning methods to model based on the existing protein information, and predict the protein. Compared with the traditional biological experiment recognition technology, machine learning method is more efficient, accurate and simple. Cai et al. [4] first developed SVM-Prot based on SVM [5, 6] algorithm. Ma et al. [7] predicted correlations based on random forest algorithm. Gao et al. [8] proposed DBD-Hunter model to judge whether a protein can combine with DNA. Liu et al. [9] made prediction based on the sequence features of PseAAC (Pseudo Amino Acid Composition) combined with the random forest method. Zhao et al. [10] introduced a new volume fraction correction to extract new information from the complex structure of DNA-binding proteins, and further proposed the binding affinity between protein and DNA. Traditional machine learning methods have realized the recognition of DNA-binding protein to a certain extent. However, the effect of deep learning neural network model is better than that of traditional machine learning experiment, which can more effectively extract and train protein features and improve the accuracy of prediction. Alipanahi et al. [11] CNN model was constructed to identify DNA-binding proteins. Qu et al. [12] contributed a fused model of CNN and RNN for identifying DNA-binding proteins. Du et al. [13] proposed a new framework of MsDBPthat uses deep neural networks for learning and classification, which tested 67% accuracy on dataset PDB2272. Chen et al. [14] built a model based on graphical neural network and developed a protein classification predictor.It's accuracy on PDB2272 reached 64.17%. Li et al. [15] first used CNN to extract protein features, and input the features extracted by CNN into LSTM network for prediction, and the accuracy rate on PDB2272 reached 77.77%.

The model proposed in the paperconstructed a deep learning framework based on two neural network models: LSTM and CNN. The function of LSTM wasto extract protein sequence information, and the function of CNN was to extract useful features in evolutionary information. Finally, the extracted information was fused to train, and the result showed that the model improved the accuracy of prediction.

2. Materials and Methods

In this part, firstly, the datasets used in the model are introduced. Then, the framework and experimental process proposed in this paper are explained. Finally, the model algorithm in the experiment are displayed.

2.1. The Dataset. We acquired the internationally common dataset PDB14189 from Ma, Guo \& Sun (2016) [7] as train dataset, PDB2272 as test dataset. Both datasets are from the collection of DNA-binding protein in the UniProt database [10]. The PDB14189 dataset is divided into 7129 positive sequences and 7060 negative sequences. The PDB2272 is an independent test dataset. In the dataset, it contains 1153 positive sequences and 1119 negative sequences. This dataset is mainly used to test whether this method is improved compared with other methods [16–21]. The sequence similarity in PDB14189 is no more than 40%, and the sequence similarity in PDB2272 is no more than 25%. The number of positive and negative samples in PDB14189 and PDB2272 datasets are shown in Table 1.

2.2. Feature Extraction

2.2.1. The Position-Specific Scoring Matrix. The Position-Specific Scoring Matrix (PSSM) [22, 23] can construct the evolutionary information, whichis vital for biological analysis to do some prediction. Therefore, the PSSM has been worked in many relative researches. In this article, PSSM was obtained by searching the non-redundant (NR) database using PSI-BLAST [24]. The iteration and e-values were set to 3 and 0.001,respectively. The PSSM extracted from the protein was represented by an $L * 20$dimensional matrix, and the PSSM can be expressed as follow:

$$PSSM = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,20} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ P_{L,1} & P_{L,2} & \cdots & P_{L,20} \end{bmatrix} \quad (1)$$

Where L is the number of rows of the PSSM matrix and represents the length of the protein sequence. 20 is the column number of PSSM matrix, representing 20 different amino acid types. The $P_{i,j}$represents the conversion rate of amino acid$i$to amino acid j. $P_{i,j}$ is generally a positive integer or a negative integer. When $P_{i,j}$is a positive integer and $P_{i,j}$is larger, the probability is higher. Conversely, when $P_{i,j}$ is a negative integer, the smaller the $P_{i,j}$, the smaller the probability.

2.2.2. Sequence Encoding. Feature coding is an important work of deep learning. Different datasets have different features. Therefore, it is particularly important to choose an appropriate coding method. Common coding methods are divided into two categories: One-hot coding and Embedding coding. One-hot encoding can digitize any features. Embedding coding is a mapping, it was used to convert discrete variables into continuous variables.

In the dataset used in this paper, the protein sequence is composed of 20 different amino acids, which are represented by 20 different English letters. Several amino acids are arranged together in order to form a protein sequence. The protein sequence can be expressed as $S = S_1, S_2, \ldots, S_n$, where $S_i$stands for the i-th amino acid in sequence. S_i is shown below: Table 2 shows the Dictionary of 20 amino acids.

$$\mathrm{S_i} \in \{A, R, F, E, H, D, N, C, V, M, Q, G, I, L, K, P, W, Y, S, T\} \quad (2)$$

For different amino acids in protein sequence, One-hot encoding is used. When the wrong residue appears in the

Table 1: Introduction to the dataset.

Number \dataset	PDB14189	PDB2272
DBPs	7129	1153
Non-DBPs	7060	1119
Total	14189	2272

Table 2: Dictionary of 20 amino acids.

Amino acid	Sequence	Amino acid	Sequence
Alanine	A	Glutamine	Q
Arginine	R	Glycine	G
Phenylalanine	F	Isoleucine	I
Glutamicacid	E	Leucine	L
Histidine	H	Lysine	K
Asparticacid	D	Proline	P
Asparagine	N	Tryptophan	W
Cysteine	C	Tyrosine	Y
Valine	V	Serine	S
Methionine	M	Threonine	T

protein sequence, we use '*X*' instead. When a protein sequence of length L is used as input and encoded with one-hot, the output is an$20 * L$dimensional matrix. For example, for a protein sequence "$S = ANCKYVHIEN$", it is encoded in one-hot mode. As shown in Figure 1.

2.3. Framework of the Model. At present, deep learning neural network has been widely accepted and achieved good results in many industries. This section mainly describes two common deep learning models used to predict whether protein sequences are binding proteins: convolutional neural network (CNN) [25], long-short term memory networks (LSTM) [26] and their fusion models.

2.4. Long-Short Term Memory. Recurrent Neural Network (RNN) is a type of artificial neural network. Because the hidden state (h_t) of RNN has short-term memory function, RNN is often used in the classification task with sequence [27] as input, so RNN is used as the basic model for DNA-binding protein classification. However, for a long input sequence, if the derivative of the activation function is too large or too small, the training loss will become too large or too small in the reverse transfer process layer by layer. These two phenomena are called gradient explosion and gradient disappearance, respectively. In order to ensure that the previous input information can still affect the model prediction after a certain time and reduce the influence of gradient disappearance, we use a variant of RNN-LSTM neural network. LSTM neural network adds cell state (C_t) after hidden state to control the change of hidden state. Compared with the hidden state, the change of cell state is relatively slow, which can strengthen the memory function of LSTM to a certain extent. Figure 2 shows a classic LSTM cell structure.

In the LSTM cell structure shown in Figure 2.In the gate structure of the neuron, the input gate (i_t) receives all the inputs of the node, including the inputs of the upper neuron and the information of the last time point of the node. The forget gate (f_t) determines the information to be lost by this node, and determines the degree of information forgetting by controlling a value from 0 to 1. Neurons themselves need to determine the retention of information and save useful information. Finally, the experimental results are output by the output gate (o_t). Eachgate structure selects a different activation function. The three gate structures and hidden states are calculated as follows:

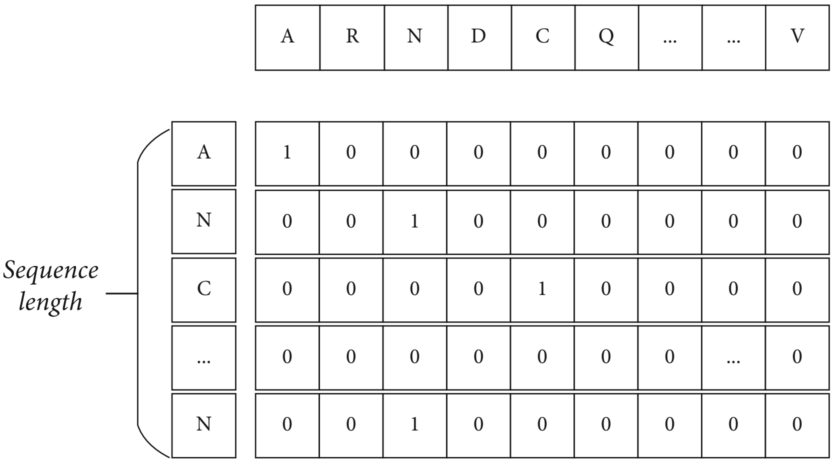
$$\begin{aligned} i_t &= \sigma(W_{xi}[h_{t-1}, x_t] + b_{xi}) \\ o_t &= \sigma(W_{xo}[h_{t-1}, x_t] + b_{xo}) \\ f_t &= \sigma(W_{xf}[h_{t-1}, x_t] + b_{xf}) \\ C_t &= f_t C_{t-1} + i_t \tanh(W_{xc}[h_{t-1}, x_t] + b_{xc}) \\ h_t &= o_t \tanh(C_t) \end{aligned} \tag{3}$$

The specific structure of LSTM can be explained by the above formula. Where the $\sigma$is the sigmoid function. i_t,f_t,o_t are three gate structures of LSTM, respectively. b is bias,and C is long-term memory in LSTM. W_{xi}, W_{xo},W_{xc} are three corresponding weight matrices of three different gate structures in the LSTM.

2.5. Convolutional Neural Network. Convolutional neural network has a very important position in deep learning. Compared with other classification algorithms, convolutional neural network needs much less data processing. In the early machine learning algorithms, the filter of the model was designed manually. However, CNN can learn the data features after enough training. In recent years, many network models have been developed based on convolutional neural networks, mainly including AlexNet [28], VGGNet [29] and ResNet [30].

As shown in Figure 3, Convolutional neural network is usually composed of multi-layer convolution layer and pooling layer. The input data to the model is usually a two-dimensional matrix. Multiple convolution cores are defined and applied to the whole data for convolution process. Finally, the feature mapping matrix corresponding to the convolution core is obtained. Each convolution core represents a feature detector and scans the corresponding features on the data. The pool stage is usually connected behind the convolution layer to reduce the dimension of the feature map by taking the maximum or average value. Deep learning models can usually contain multiple convolution layers to learn more complex abstract features.

2.6. Model Fusion. This section mainly displays the deep learning model proposed. As shown in Figure 4, the biggest difference between our proposed model and other existing models is that our model is a parallel structure. LSTM and CNN canextract different information features at the same time. Finally, the extracted information was fused as



Sequence length

	A	R	N	D	C	Q	...	...	V
A	1	0	0	0	0	0	0	0	0
N	0	0	1	0	0	0	0	0	0
C	0	0	0	0	1	0	0	0	0
...	0	0	0	0	0	0	0	...	0
N	0	0	1	0	0	0	0	0	0

Figure 1: One-hot Encoding.

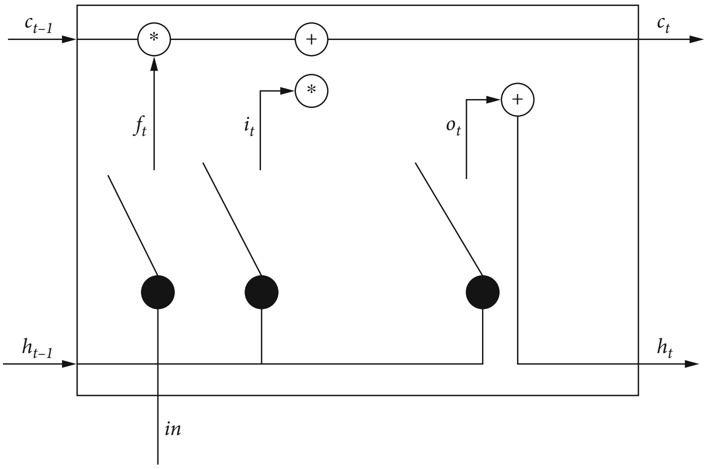


Figure 2: Classic LSTM Cell Structure.

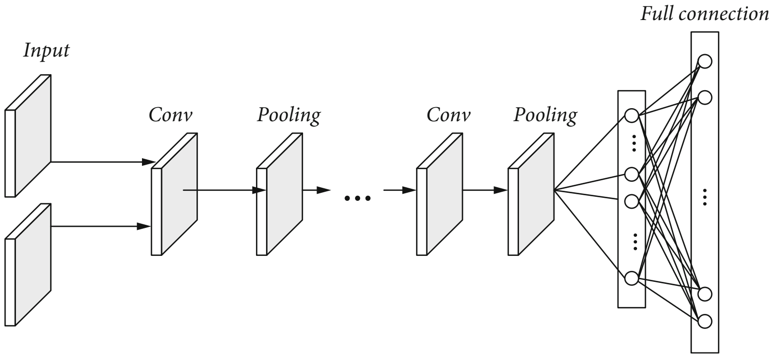


Figure 3: Convolutional Neural Network Structure.

the input of MultiLayer Perceptron (MLP) [31] for training and classification. Other existing models are series structure, which connects CNN and LSTM in series. In series structure, those two networks cannot extract features at the same time.

The output of the model was a probability value and 0.5 was set as the dividing point. When the output is greater than 0.5, it is predicted that this protein can bind to DNA. When the output is less than 0.5, the result is just the opposite.

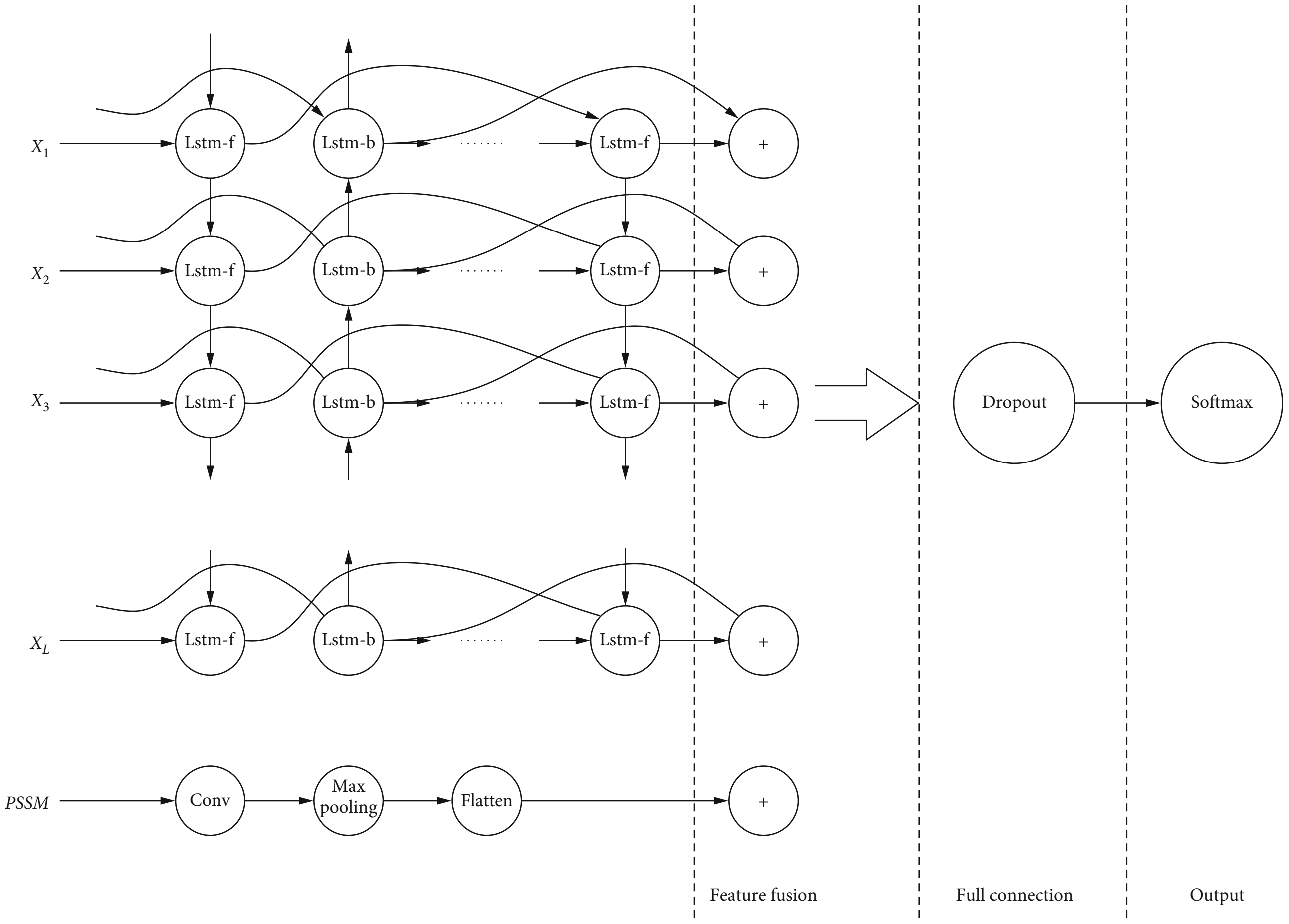


Figure 4: Network Model.

```
Algorithm: Pseudo algorithm for predicting DNA-binding protein
Input: Training dataset after data preprocessing
initialize weights
when iteration n =1:
for n < max epoch:
   for input_data $x_1$ to $x_L$:
   a. protein sequence features as input through LSTM
   b. PSSM matrix as input through CNN
   c.Fuse sequence feature information and PSSM information, and input into the fully connected neural network.
   d. calculate loss function
   e. find the optimal parameter gradient
    f. update network parameters through back propagation
Output: Trained network parameters
```

Algorithm 1: Pseudo code of the algorithm.

2.7. Model Algorithm. In this section, the specific algorithm of the model is described in detail. The protein sequence features and PSSM matrix wereput into two parallel neural networks. Finally, the extracted information was fused as the input of MLP full connection layer for training and classification. The specific pseudo algorithm is shown in Algorithm 1.

Table 3: Setting of hyperparameters.

Hyperparameter	Setting
Epoch	70
Learning rate	0.001
Batch size	64
Optimizer	Adam
Loss function	Binary cross entropy loss

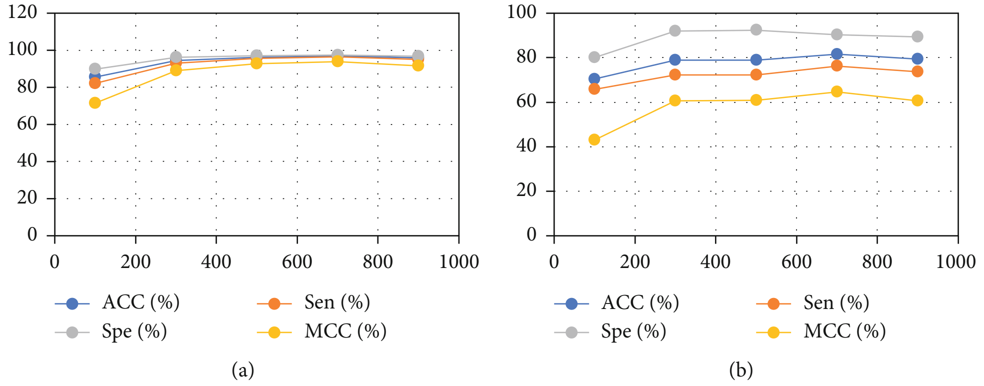


Figure 5: Comparative experiment of different sequence lengths. (a) is the result of different sequence lengths in dataset PDB14189. (b) is the result of different sequence lengths in dataset PDB2272.

3. Results and Discussion

3.1. Evaluation Index. In this experiment, four evaluation indicators were used. They are accuracy (*ACC*), sensitivity (*Sen*), specificity (*Spe*) [32] and Matthew's Correlation Coefficient (*MCC*). The formulas of these four evaluation indexes are as follows:

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$
$$Sen = \frac{TP}{(TP + FN)}$$
$$Spe = \frac{TN}{(TN + FP)}$$
$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\left(\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}\right)} \quad (4)$$

TP is the size of positive sequences correctly identified.

TN is the size of negative sequences correctly identified.

FP is the size of negative sequences incorrectly identified.

FN is the size of positive sequences incorrectly identified.

Sen is sensitivity, which is the percentage of correctly identified positive sequence.

*Spe*is specificity, which is the percentage of correctly identified negative sequence.

ACC is accuracy, which is the percentage of correctly identified sequence.

MCC is Matthew's Correlation Coefficient, which means the prediction quality of the binary classification model, with a range of [-1,1]. The smaller the *MCC*, the worse the prediction quality of the algorithm [32, 33].

3.2. Model Hyperparameter. The experimental code of model in the paper was implemented through the PyTorch framework. In addition, hyperparameters are very important to the model. Only by constantly adjusting the hyperparameters can we get the optimal training model. Hyperparameters are

Table 4: Result comparison of whether selecting Dropout.

	ACC(%)	Sen(%)	Spe(%)	MCC(%)
With dropout	*79.83*	*73.59*	91.35	62.06
Without dropout	*79.44*	*74.29*	87.89	60.32

Table 5: Result comparison of different weight_decay.

	ACC(%)	Sen(%)	Spe(%)	MCC(%)
weight_decay =0.1	*66.80*	*64.28*	68.22	32.04
weight_decay =0.01	*81.59*	*76.23*	90.23	64.63
weight_decay =0.001	*78.91*	*72.05*	93.00	61.05
weight_decay =0.0001	*76.75*	*69.62*	93.51	57.78

Table 6: Result comparison on PDB14189.

Methods	ACC(%)	Sen(%)	Spe(%)	MCC(%)
MsDBP	80.29	80.87	79.72	60.61
PSSM-PP + PHY	82.67	79.95	85.39	65.4
BP ± NBP + PHY	80.40	76.88	83.92	60.9
PSSM	79.62	76.02	83.21	59.4
64 optimal featuresa	86.90	83.76	90.03	72.2
PHY	77.65	73.54	81.76	55.5
PSSM-PP + BP_NBP	83.68	81.01	86.34	67.4
PSSM-PP	81.69	78.92	84.45	63.5
ALL features	84.64	82.23	87.06	70.6
StackDPPred(one-hot)	76.00	79.27	72.71	52.10
PDBP-CNN	82.02	87.49	76.50	64.69
PDBP-fusion	82.81	86.45	79.13	66.1
Our methods	96.93	96.46	97.41	93.86

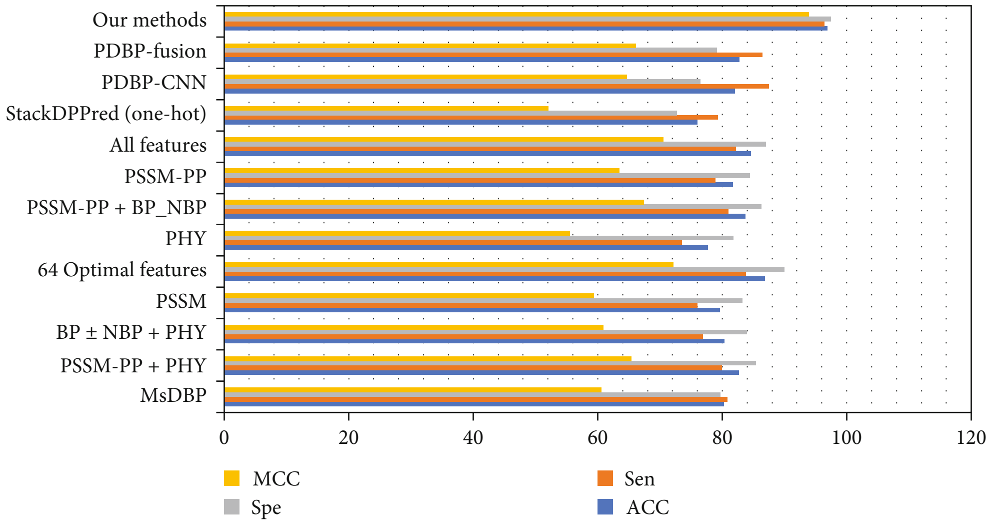


Figure 6: Result Comparison on PDB14189.

often initially set based on experience. The settings of parameters are shown in Table 3.

Table 7: Result comparison on PDB2272.

Methods	ACC(%)	Sen(%)	Spe(%)	MCC(%)
Qu et al. [12]	48.33	49.07	48.31	−3.34
DPP-PseAAC [17]	58.10	59.10	56.63	16.25
MsDBP [13]	66.99	66.42	70.69	33.97
Local-DPP [19]	50.57	58.72	8.76	4.564
PseDNA-Pro [37]	61.88	59.90	75.28	24.30
PDBP-Fusion [15]	77.77	73.31	66.85	56.65
Our methods	81.59	76.23	90.23	64.63

3.3. Result. In this section, we first compared the comparative experiments based on different length sequences. Next, we considered whether to add a Dropout [34] layer and Regularization, and conducted two sets of comparative experiments. Then, other parameters were selected to obtain the best training model. Finally, wetested on the train dataset PDB14189 and the test dataset PDB2272, and compared the performance with other existing models.

3.4. Result of Different Sequence Lengths. In the data processing phase, we select different maximum lengths (from 100 to 900) to encode DNA sequences to evaluate the overall performance. Figure 5 shows that the result is the best when the protein sequence length is 700.

3.5. Model Performance whether Selecting Dropout. When the model was used to train the dataset, it was easy to form an over-fitting phenomenon. In order to prevent the problem, a dropout method was proposed. The direct function of dropout is to reduce the number of intermediate features, so as to reduce redundancy and increase the orthogonality between features in each layer. In each training batch, the interaction between hidden layer nodes is reduced by ignoring the general feature detector to improve experimental results. Comparative experiments are shown in Table 4.

3.6. Model Performance whether Selecting Regularization. Like dropout, regularization [35, 36] is also a method to prevent the training model from overfitting. We can understand regularization as "constraint", which is convenient to understand. The more complex the model, the easier it is to overfit. The role of regularization is to correct the problem. Some are in the model design stage, and some are in the model training stage. The purpose is to prevent overfitting. Therefore, we set the hyperparameter of weight_decay, which is a method of weight decay. Weight decay is to subtract a gradient from the gradient of each update. As shown in the formula (5).In this method, a penalty term is added to the model loss function to make the learned model parameters smaller, which is a common method of overfitting. The results were different when the value of weight_decay was different. The experimental results are shown in Table 5.

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha \nabla f_t(\theta_t) \tag{5}$$

θ is the model parameter vector, $\nabla f_t(\theta_t)$ is the gradient of loss function at t time, and α is the learning rate.

When other parameters are determined, we adjust weight_decay and find that the training result is best when weight_decay =0.01. So during training, we adjust weight_decay to 0.01 to optimize our final training result.

3.7. Result Comparison on the Benchmark Dataset. In order to demonstrate the effectiveness of our proposed model. On the benchmark dataset PDB14189, we compared DNABP [7], MsDBP [13], StackDPPred [22], PDBP-CNN [15] and

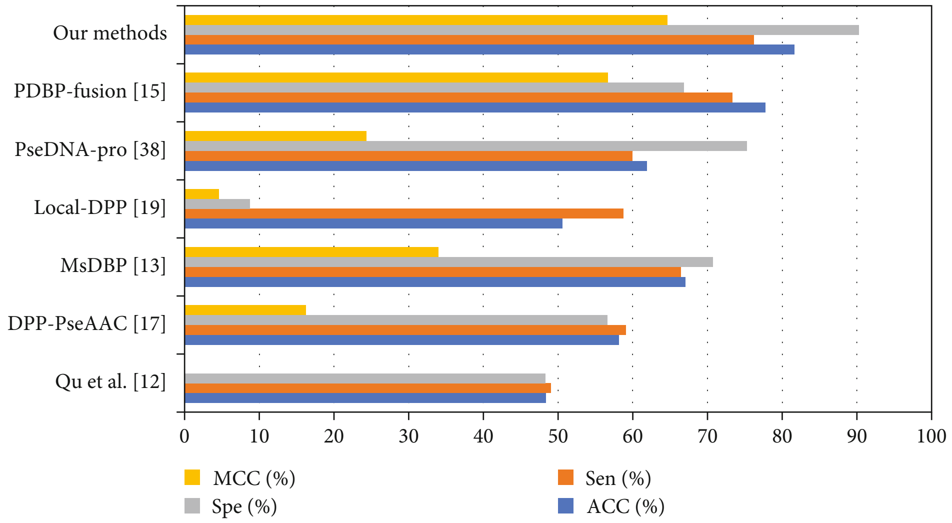


Figure 7: Result Comparison on PDB2272.

PDBP-Fusion [15]. The DNABP method adopts various sequence features. The specific experiment comparison is shown in Table 6.

For a visual comparison, we show this as a bar graph in Figure 6.

3.8. Result Comparison on the Test Dataset. On the PDB2272 dataset, different methods were compared. In Table 7,the ACC of our proposed model is 81.59%, which is 3.82% higher than the ACCof PDBP_Fusion model. From this indicator of MCC, the MCC of our model is 64.63%, which is 7.98% higher than the MCC of PDBP_Fusion. Therefore, the method has certain advantages.

As shown in Figure 7, the method plays a role in identifying DNA-binding protein. In conclusion, our model is effective. It is a reliable deep learning neural network algorithm.

4. Conclusions

DNA-binding proteins are essential for the regulation of life activities. And in pharmaceutical engineering, DNA-binding proteins are key components of steroids, antibiotics, and anticancer drugs. Therefore, the identification of DNA-binding proteins is of great significance. In this paper, good recognition performance is achieved by only extracting protein features and combining deep learning algorithm pairs to determine whether related proteins have a preference for interacting with DNA. The main work of this paper is as follows: a DNA-binding protein fusion recognition model based on LSTM and CNN is proposed. In view of the weak ability of traditional protein feature representation, we use LSTM and CNN to extract protein sequence information and local information, respectively, to improve the ability of protein feature representation. By effectively extracting protein sequence features and local features, the modeling ability of protein depth features and the recognition ability of DNA-binding proteins are significantly improved.

Compared with traditional methods at the forefront of this field, the experimental results verify the superiority and stability of the model. In the future, we plan to use different biological features and continue to improve overfitting to further improve the prediction speed and accuracy of the model.

Data Availability

The dataset is available in the references cited.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (61902272, 62073231, 61772357, 62176175, 61876217, 61902271), National Research Project (2020YFC2006602), Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS2166), Opening Topic Fund of Big Data Intelligent Engineering Laboratory of Jiangsu Province (SDGC2157). The authors acknowledge and thank the anonymous reviewers for their suggestions that allowed the improvement of our manuscript.

References

[1] M. S. Nogueira and O. Koch, "The development of target-specific machine learning models as scoring functions for docking-based target prediction," *Journal of Chemical Information and Modeling*, vol. 59, no. 3, pp. 1238–1252, 2019.

[2] Q. Kaiyang, "A review of DNA-binding proteins prediction methods," *Current Bioinformatics*, vol. 14, no. 3, pp. 246–254, 2019.

[3] L. Wei, M. Liao, X. Gao, and Q. Zou, "Enhanced protein fold prediction method through a novel feature extraction

technique," *IEEE Transactions on Nanobioscience*, vol. 14, no. 6, pp. 649–659, 2015.

[4] C. Z. Cai, L. Y. Han, and Z. L. Ji, "SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3692–3697, 2003.

[5] Y. Qian, H. Meng, W. Lu, Z. Liao, Y. Ding, and H. Wu, "Identification of DNA-binding proteins via Hypergraph based Laplacian Support Vector Machine," *Current Bioinformatics*, vol. 16, no. 1, 2022.

[6] Y. Zou, Y. Ding, L. Peng, and Q. Zou, "FTWSVM-SR: DNA-binding proteins identification via fuzzy twin support vector machines on self-representation," in *Interdisciplinary Sciences: Computational Life Sciences*, pp. 1–13, Springer, 2021.

[7] X. Ma, J. Guo, and X. Sun, "DNABP: Identification of DNA-binding proteins based on feature selection using a random forest and predicting binding residues," *PloS One*, vol. 11, no. 12, article e0167345, 2016.

[8] M. Gao and J. Skolnick, "DBD-hunter: a knowledge-based method for the prediction of DNA–protein interactions," *Nucleic Acids Research*, vol. 36, no. 12, pp. 3978–3992, 2008.

[9] B. Liu, S. Wang, and X. Wang, "DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation," *Scientific Reports*, vol. 5, no. 1, pp. 1–11, 2015.

[10] H. Zhao, Y. Yang, and Y. Zhou, "Structure-based prediction of DNA-binding proteins by structural alignment and a volume-fraction corrected DFIRE-based energy function," *Bioinformatics*, vol. 26, no. 15, pp. 1857–1863, 2010.

[11] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.

[12] Y. H. Qu, H. Yu, X. J. Gong, J. H. Xu, and H. S. Lee, "On the prediction of DNA-binding proteins only from primary sequences: a deep learning approach," *PLoS One*, vol. 12, no. 12, article e0188129, 2017.

[13] X. Du, Y. Diao, H. Liu, and S. Li, "MsDBP: exploring DNA-binding proteins by integrating multiscale sequence information via Chou's five-step rule," *Proteome Research*, vol. 18, no. 8, pp. 3119–3132, 2019.

[14] D. Chen and L. Wei, "A useful tool for the identification of DNA-binding proteins using graph convolutional network," *Current Proteomics*, vol. 18, no. 5, pp. 661–668, 2021.

[15] G. Li, X. Du, X. Li, L. Zou, G. Zhang, and Z. Wu, "Prediction of DNA binding proteins using local features and long-term dependencies with primary sequences based on deep learning," *Peer J*, vol. 9, article e11262, 2021.

[16] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting DNA-protein binding," *Bioinformatics*, vol. 32, no. 12, pp. i121–i127, 2016.

[17] M. S. Rahman, S. Shatabda, S. Saha, M. Kaykobad, and M. S. Rahman, "DPP-PseAAC: A DNA-binding protein prediction model using Chou's general PseAAC," *Theoretical Biology*, vol. 452, pp. 22–34, 2018.

[18] P. Zhang, L. Tao, X. Zeng et al., "A protein network descriptor server and its use in studying protein, disease, metabolic and drug targeted networks," *Briefings in Bioinformatics*, vol. 18, no. 6, pp. 1057–1070, 2017.

[19] L. Wei, J. Tang, and Q. Zou, "Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information," *Information Sciences*, vol. 384, pp. 135–144, 2017.

[20] Y. Wang, Y. Ding, F. Guo, L. Wei, and J. Tang, "Improved detection of DNA-binding proteins via compression technology on PSSM information," *PLoS One*, vol. 12, no. 9, article e0185587, 2017.

[21] Y. Zou, H. Wu, X. Guo et al., "MK-FSVM-SVDD: A Multiple Kernel-based Fuzzy SVM Model for Predicting DNA-binding Proteins via Support Vector Data Description," *Current Bioinformatics*, vol. 16, no. 2, pp. 274–283, 2020.

[22] W. Lu, Z. Song, Y. Ding et al., "Use Chou's 5-Step Rule to Predict DNA-Binding Proteins with Evolutionary Information," *BioMed Research International*, vol. 2020, 9 pages, 2020.

[23] L. Wei, M. Liao, X. Gao, and Q. Zou, "An improved protein structural classes prediction method by incorporating both sequence and structure information," *IEEE Transactions on Nanobioscience*, vol. 14, no. 4, pp. 339–349, 2015.

[24] A. A. Schaffer, L. Aravind, T. L. Madden et al., "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements," *Nucleic Acids Research*, vol. 29, no. 14, pp. 2994–3005, 2001.

[25] S. Chauhan and S. Ahmad, "Enabling full-length evolutionary profiles based deep convolutional neural network for predicting DNA-binding proteins from sequence," *Proteins: Structure, Function, and Bioinformatics*, vol. 88, no. 1, pp. 15–30, 2020.

[26] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016, https://arxiv.org/abs/1603.01360.

[27] G. Klaus, K. S. Rupesh, K. Jan, R. S. Bas, and S. Jürgen, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2015.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," 2014, https://arxiv.org/abs/1409.1556.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, USA, 2016.

[31] S. K. Pal and S. Mitra, "Multilayer Perceptron, Fuzzy Sets, Classifiaction," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 683–697, 1992.

[32] W. Lu, Z. Song, Y. Ding, H. Wu, and H. Huang, "A Prediction Method of DNA-Binding Proteins Based on Evolutionary Information," in *International Conference on Intelligent Computing*, pp. 418–429, Springer, Cham, 2019.

[33] X.-F. Wang, P. Gao, Y.-F. Liu, H.-F. Li, and L. Fan, "Predicting thermophilic proteins by machine learning," *Current Bioinformatics*, vol. 15, no. 5, pp. 493–502, 2020.

[34] Y. Ding, C. Yang, J. Tang, and F. Guo, "Identification of protein-nucleotide binding residues via graph regularized k-

local hyperplane distance nearest neighbor model," in *Applied Intelligence*, pp. 1–15, Springer, 2022.

[35] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, 1995.

[36] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *2017 international joint conference on neural networks (IJCNN)*, pp. 2684–2691, IEEE, Anchorage, AK, USA, 2017.

[37] B. Liu, J. Xu, S. Fan, R. Xu, J. Zhou, and X. Wang, "PseDNA-pro: DNA-binding protein identification by combining chou's PseAAC and physicochemical distance transformation," *Molecular Informatics*, vol. 34, no. 1, pp. 8–17, 2015.