

RESEARCH ARTICLE

Convolution-Bidirectional Temporal Convolutional Network for Protein Secondary Structure Prediction

YUNQING ZHANG¹, YUMING MA¹, AND YIHUI LIU¹

School of Computer Science and Technology, Shandong Academy of Sciences, Qilu University of Technology, Jinan 250353, China

Corresponding authors: Yuming Ma (mym@qlu.edu.cn) and Yihui Liu (yxl@qlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61375013, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2013FM020.

ABSTRACT As a basic feature extraction method, convolutional neural networks have some information loss problems when dealing with sequence problems, and a temporal convolutional network can compensate for this problem. However, ordinary temporal convolutional networks can not deal well protein secondary structure prediction because of their one-way analysis. Therefore, we propose an integrated deep learning model called Convolutional-Bidirectional Temporal Convolutional Network. for 3-state and 8-state protein secondary structure predictions based on a convolutional neural network and bidirectional temporal convolutional networks. Because the model combines the advantages of the convolutional neural network and bidirectional temporal convolution network, it can not only capture the local correlation in the amino acid sequence but also analyse the long-distance interaction in the amino acid sequence. Therefore, this model can effectively improve the accuracy of protein secondary structure predictions. The experimental results show that the combination of convolutional neural network and bidirectional temporal convolutional network is effective for predicting protein secondary structure.

INDEX TERMS Protein secondary structure prediction, temporal convolutional network, convolutional neural network, bidirectional prediction, deep learning.

I. INTRODUCTION

Proteins are essential for life activities, and their structures determine their functions. The structures of proteins include primary, secondary, tertiary, and quaternary structures. The primary structure is the basis of the protein structure and is known as the natural structure. The primary structure determines the stable spatial conformation, and these structures enable the biological function of proteins. The sequence space of protein is extremely complex. There are 20 possible residues at each position; therefore, the protein is highly diverse in sequence, structure, and function. The Protein secondary structure (PSS) is formed by folding of the protein primary structure, protein tertiary structure is in turn formed by

bending and folding of the secondary structure, which refers to a specific spatial structure resulting from the maintenance of multiple secondary bonds. The protein quaternary structure refers to a polymeric structure formed by multiple independent tertiary structure polypeptide chains linked by non-covalent bonds. Predicting the tertiary structure of proteins is a basic problem in computational biology, and predicting the secondary structure is a springboard for predicting the tertiary structure. Therefore, research on protein secondary structure prediction (PSSP) is particularly important.

Many methods have been used to predict PSSs, particularly, after machine learning, and statistical methods have been used to predict PSSs in recent years. The position-specific scoring matrix (PSSM) based on PSIBLAST reflects information regarding sequence evolution, amino acid conservation and mutations [1]. Combining

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu¹.

PSSM data with machine learning has led to major breakthroughs in PSSP. Support vector machines (SVMs) [2], [3], [4]. Neural networks (NNs) [5], [6], [7] and k-nearest neighbours [8] can improve the protein prediction accuracy to over 70%.

DeepCNF [9] used deep neural networks and conditional neural fields to predict 3-state and 8-state secondary structures. PSRM [10] uses big data to train support vector machines [11]. In contrast to machine learning methods, CNNs can automatically extract the local features of amino acid residues. Significant results have been achieved using CNN and RNN [12]. For example, Guo et al. fused an asymmetric convolution neural network with long-term and short-term memory neural network models and applied them to predict the secondary structure of eight types of proteins [13]. Heffernan used a long short-term memory (LSTM) bidirectional recurrent neural network (BRNN) that captured long-term interactions without using sliding windows [14]. Snyder et al. used bidirectional long- and short-term memory neural networks to capture long-range correlations between amino acid residues for secondary structure prediction [15]. Wang et al. Applied a CNN with conditional random fields for secondary structure prediction on the benchmark CB513 dataset.

In recent years, temporal convolutional network (TCN) are novel networks that have been proposed for sequence modelling [16]. Traditional convolution cannot capture the dependent information of long sequences owing to the limitation of the convolution kernel size. The TCN uses causal and expansion convolution based on traditional convolution and uses residual blocks instead of convolution. A residual block contains two layers of convolution and a non-linear mapping, with WeightNorm and Dropout incorporated in each layer to regularise the network. TCN has been shown to outperform long short-term memory networks in many sequence modelling problems.

These models either use a single information processing method to predict PSSs or are deficient in other aspects. In this paper, we propose Convolutional-Bidirectional Temporal Convolutional Network (C-BITCN), which takes advantage of CNN and bidirectional TCN in predicting PSS, where CNN can effectively extract short distance features near amino acids, and bidirectional TCN can effectively analyse the interactions between amino acids. The experiments showed that C-BITCN outperformed other advanced models.

II. C-BITCN

A. CONVOLUTIONAL NEURAL NETWORK

CNN can effectively extract local features of amino acid chains to improve the accuracy of the PSS. The input of the CNN was a sliding window intercepted and created by PSSM, and the size of each sliding window was 20×19 . The convolution layer extracts local features from the input data through local convolution and weight sharing. The neurones above the convolution layer transfer the features to the convolution

layer, which is realised by “sliding” the convolution kernel on the input matrix PSSM. In the experiment, after the PSSM matrix was input into the CNN, the dimensions of the fully connected layer were set to 20 and the output features of the fully connected layer were extracted. Fig 1 shows the process of extracting features after inputting the PSSM matrix into a CNN.

The convolution feature extraction process was completed using a feature extraction filter to filter the PSSM matrices one by one. The matrix of each region is multiplied by the corresponding weight plus an offset to obtain the feature map, which is defined as follows:

$$C^i = (c_1^i, c_2^i, \dots, c_k^i, \dots, c_{N-1}^i, c_N^i). \quad (1)$$

where C^i is the set of convolutional kernels in layer i , C_k^i is the k_{th} convolutional kernel in layer i , and N is the number of convolutional kernels in layer i . W_n^{i-1} is the region map generated by the input amino acid PSSM matrix with the previous layer of the convolution kernels. After the convolution of layer i , the feature map J_k^i is obtained and is defined as follows:

$$J_k^i = f\left(\sum_R W_n^{i-1} \times c_k^i + b_k\right). \quad (2)$$

b_k is the offset and f is the activation function. In this experiment, the activation function is ReLU, which was used to provide a nonlinear element for the output of the convolution layer.

The pooled layer in Fig 1 did not undergo any learning. Its function is to reduce feature dimensions and prevent overfitting. The neurones of the entire junction layer are connected to the previous layer, which is convenient for extracting the characteristics of amino acid residues. The input dimension of the softmax layer can be controlled by resizing the fully connected layer. The softmax layer is connected to the fully connected layer, and consists of three or eight neurones. The value of the fully connected layer satisfies the following equation:

$$\sum_{i=1}^3 P_i = 1 \quad i \in \{H, E, C\}. \quad (3)$$

$$\sum_{i=1}^8 P_i = 1 \quad i \in \{G, H, I, B, E, T, S, L\}. \quad (4)$$

B. TEMPORAL CONVOLUTIONAL NETWORK

TCN has been proposed to process sequences in recent years and has been proven to be superior to LSTM in language modeling and natural language processing. The TCN has both causal and expansive convolutional functions. As shown in Fig 2, causal convolution is represented as follows: the value of the next layer at time k only depends on the value of the previous layer at time k and before time k ; therefore, the causal convolution is unidirectional, and future data cannot be observed, unlike the traditional convolution. However, causal convolution alone cannot solve the problem of traditional

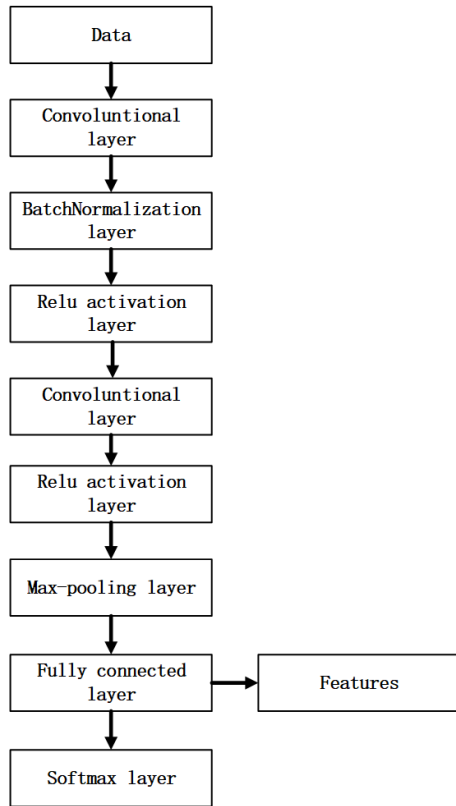


FIGURE 1. Convolutional neural network workflow.

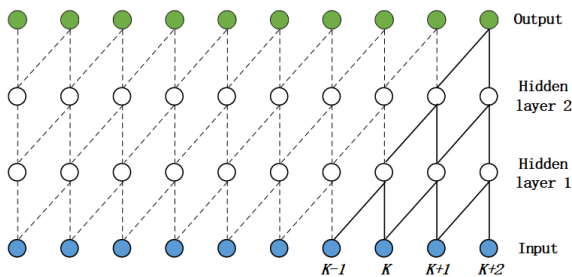


FIGURE 2. Conceptual diagram of causal convolution.

convolution neural network. Because the modeling length is limited by the size of the convolution kernel, it is necessary to build a sufficiently long dependency relationship by multi-layer linear stacking. Therefore, causal and dilated convolutions are required to address this problem. In dilation convolution, $d = 1$ in the first layer means sampling the input at each point, and $d = 2$ in the second layer means sampling at one sample every two points. Dilated convolution causes the receptive field to increase exponentially with the number of layers.

The properties of the sequential modelling task are as follows: assume that given a sequence of inputs x_0, \dots, x_T and wish to predict some corresponding outputs y_0, \dots, y_T . Typically, in order to predict the output y_t at moment t , we are restricted to using only those inputs that have already

been observed: x_0, \dots, x_t . Therefore, formally, a sequential modelling network is an arbitrary function f that generates a mapping: $X_{T+1} \rightarrow Y_{T+1}$. The mapping is as follows:

$$y_0, \dots, y_T = f(x_0, \dots, x_T). \quad (5)$$

Therefore, the TCN has two principles: the network cannot leak from the future to the past, and the output length of the network is consistent with the input length. The first principle requires the use of causal convolution, whereas the second principle achieves one-to-one correspondence by extracting information from the sequence in the expansion convolution and then outputting it, as illustrated in Fig 3.

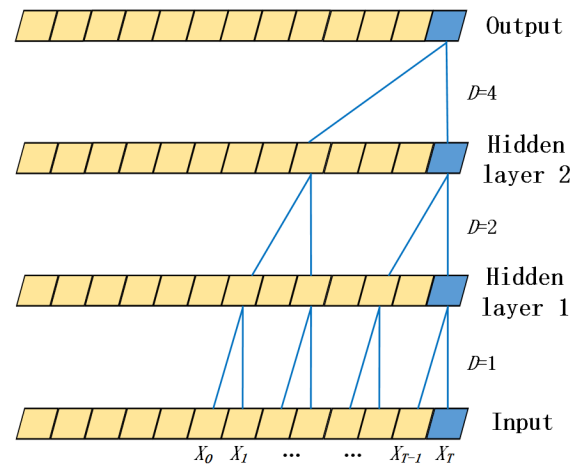


FIGURE 3. Schematic diagram of the expansion convolution.

Causal convolution can only extract the previous information linearly, but its time efficiency is low when processing long sequences. Therefore it is necessary to increase the dilated convolution to improve the processing efficiency. For a 1-D sequence input $x \in R^n$ and a filter $f : \{0, \dots, k - 1\} \rightarrow R$, the dilated convolution operation F on sequence elements s is defined as:

$$F(s) = (X * df)(s) = \sum_{i=0}^{k-1} f(i) \times X_{s-d \cdot i}. \quad (6)$$

In Equation (6) d is the expansion factor, k is the filter size, and $s - d \cdot i$ indicates the past direction. Therefore, dilation is equivalent to introducing a fixed step between every two neighbouring filters. When $d = 1$, dilation convolution can be transformed into regular convolution. Using a larger dilation allows the highest output level to represent a larger range of inputs, thereby effectively expanding the receptive domain of the CNN.

The residual block contains a branch that leads to a series of transformations F whose output is added to the input x of block:

$$O = Activation(x + F(x)). \quad (7)$$

Doing so can effectively change the mapping of layers, and yield better results for deeper networks. Although the

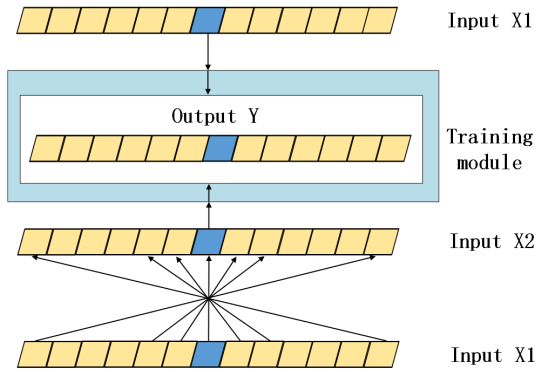


FIGURE 4. Bidirectional temporal convolutional network.

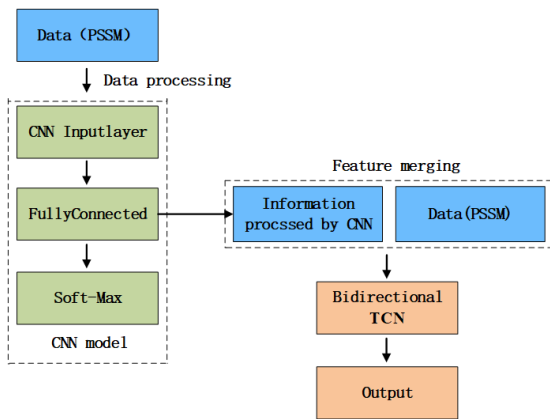


FIGURE 5. Structure of C-BITCN.

sensing domain of the TCN depends on the network depth n , filter size k , and expansion factor d , the network depth of the TCN may increase. At this time, the use of a residual module instead of a convolution layer makes the model more stable.

C. C-BITCN

1) BIDIRECTIONAL TEMPORAL CONVOLUTIONAL NETWORK Although the TCN has many advantages, its unidirectional prediction structure can not fully capture information in the protein sequence. Therefore, we modified the TCN and trained the model by taking forward and reverse protein sequences as the input information of the model simultaneously, and obtained the bidirectional TCN. The process for obtaining reverse input data is shown in Fig 4.

We modified the TCN into a bidirectional TCN so that the protein sequence X_1 and inverted sequence X_2 can be input into the training and prediction modules simultaneously for training and prediction, which can effectively improve the prediction accuracy.

2) INTEGRATE CNN AND BIDIRECTIONAL TCN

In this study, we propose an integrated deep learning model, C-BITCN, the structure of which is illustrated in Fig 5. First, the PSSM matrix is processed into 20×19 matrices, which are input into the 4-D CNN, and then the features processed by the CNN are extracted in the fully connected layer. After the extracted features were combined with the

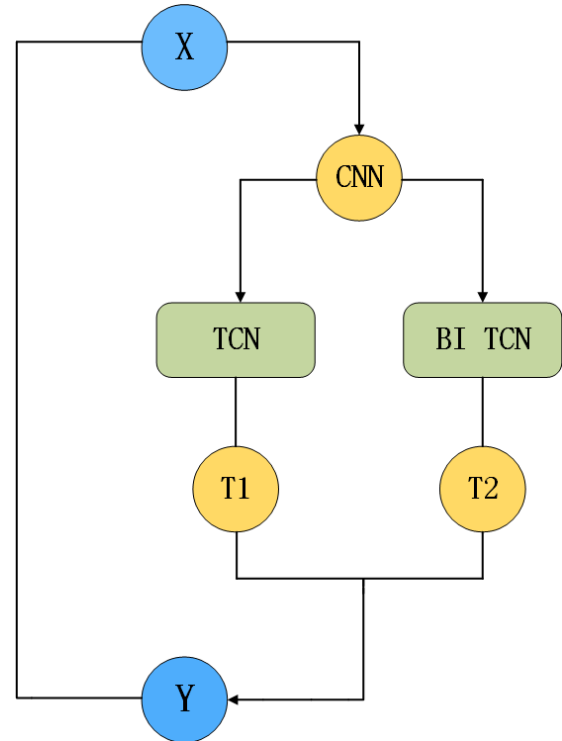


FIGURE 6. Integration of CNN and TCN.

TABLE 1. Number of proteins in the test datasets.

Test dataset	The amount of protein in the test set
CASP10	99
CASP11	81
CASP12	19
CASP13	22
CASP14	24
CB513	513

PSSM sequence, they were inputted into the sequential convolutional network, and the final classification results were obtained.

The details of the integration of CNN and TCN are shown in Fig 6. The amino acid sequence X to be predicted is processed and input into the CNN, and the features extracted by the CNN are simultaneously input into the TCN and reverse TCN at the same time, and the obtained T_1 , T_2 and original information X are input into the final classifier for classification.

$$T_1 = TCN(CNN(X)). \tag{8}$$

$$T_2 = BITCN(CNN(X)). \tag{9}$$

$$Y = FinalC(X, T_1, T_2). \tag{10}$$

Equations (8) - (10) describe this process. We record bidirectional TCN as $BITCN$, and the final classification module as $FinalC$. Input the output of TCN, the output of bidirectional TCN and X into $FinalC$ to get the classification result Y .

TABLE 2. Q3 accuracy of C-BITCN with different sliding window sizes.

Sliding Window	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
13	79.92	78.16	75.85	75.72	74.61	74.84
15	80.68	78.63	76.24	76.30	75.38	75.42
17	81.01	79.03	76.77	76.68	75.93	75.84
19	81.10	79.19	77.04	77.03	76.21	81.86

TABLE 3. Q3 results for C-BITCN with different feature input dimensions.

Features dimension	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
23	81.10	79.19	77.04	77.03	76.21	81.86
40	80.59	78.69	76.41	76.71	75.69	81.30
50	80.08	78.26	75.81	76.17	75.11	80.75
70	79.88	77.91	75.20	75.81	74.92	80.33
120	79.80	77.88	75.11	75.72	74.61	80.18

The CNN and bidirectional TCN are aggregated for prediction because in recent years, many PSSPs have been developed using a combination of CNN and LSTM [17]. The advantage of these models is that they can not only extract the local features near the target amino acids by CNN, but also obtain the interaction between amino acid residues by LSTM, effectively improving the accuracy of PSSP. TCN can better replace the work of LSTM. The problem that TCN cannot be used for bidirectional prediction sequence has also been solved. Therefore, C-BITCN can obtain better results than other methods.

III. EXPERIMENT

A. DATA SETS

In this study, seven public datasets: CullPDB [18], CB513 [19] and CASP [20], [21], [22], [23], [24]: CASP10, CASP11, CASP12, CASP13, CASP14. The CullPDB dataset contain 11,650 proteins. The CullPDB dataset was selected based on a 25% percentage identification cut-off, a 3 angstroms resolution cut-off and an R-factor cut-off of 0.25. The CullPDB and duplicate proteins were removed from the test set. In this study, the models were tested using the public datasets CASP10, CASP11, CASP12, CASP13, CASP14 and CB513 with less than 25% sequence identity. The number of protein sequences are listed in Table 1.

B. EXPERIMENTAL ENVIRONMENT

The experimental environment for this study was as follows: The cluster hardware consists of four NF5280M5s, one NF5288M5, and two gigabit switches. The host template was an Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz with 24 CPUs and a maximum of 176G of memory. NF5280M5 was equipped with a Tesla V100 GPU (16G of memory) computing card and NF5288M5 was equipped with a Tesla V100 GPU (32G of memory) computing card. Centos 7.4 X64 standard system, MATLAB running version R2019b.

C. EVALUATION INDICATORS

The DSSP subdivides the three classes of PSS into eight classes: H (alpha helix), G (310 helix), I (pi-helix), E (beta-strand), B (beta-bridge), T (beta-turn), S (bend) and L (loop

or irregular). For the purposes of this study, G, H and I were uniformly classified as H, B and E as E and the rest as C.

Q_3 and Q_8 are often used as the evaluation criteria in this study. These values were obtained using the following equation.

$$Q_i = \frac{TP_i}{TP_i + FP_i} \quad i \in \{H, E, C\}. \quad (11)$$

where TP_i is the number of residues correctly predicted in class i and FP_i is the number of residues incorrectly predicted in class i . Q_3 can be obtained from the following equation:

$$Q_3 = \frac{TP_H + TP_E + TP_C}{TP} \quad (12)$$

$$Q_8 = \frac{TP_G + TP_H + TP_I + TP_B + TP_E + TP_T + TP_S + TP_L}{TP} \quad (13)$$

where TP is the total number of amino acid residues.

The segment overlap score (SOV) is an evaluation of PSS based on protein structure fragments. Unlike Q_3 , which targets only a single residue, SOV was used to evaluate whether the predicted protein sequence fragments were correct. Then, the formula for the SOV is:

$$SOV = \frac{100}{N_{SOV}} \sum_{S_0} \left[\frac{\min(S_1, S_2) + \delta(S_1, S_2)}{\max(S_1, S_2)} \text{length}(S_1) \right]. \quad (14)$$

The length of S_1 is marked as $\text{length}(S_1)$, the union of the S_1 and S_2 sequences is marked as $\max(S_1, S_2)$, and the intersection of the S_1 and S_2 sequences is marked as $\min(S_1, S_2)$.

δ in the above equation was set to allow for variation of the fragments at the edges of the protein structure and $\delta(S_1, S_2)$ was taken to be consistent with the following definition:

$$\delta(S_1, S_2) = \min \begin{cases} (\max(S_1, S_2) - \min(S_1, S_2)) \\ \min(S_1, S_2) \\ \text{int}[\text{length}(S_1) \div 2] \\ \text{int}[\text{length}(S_2) \div 2] \end{cases}. \quad (15)$$

TABLE 4. Q8 results for C-BITCN with different feature input dimensions.

Features dimension	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
28	69.67	69.81	68.30	67.38	66.28	74.13
40	70.19	70.08	68.86	67.99	66.67	74.60
48	70.62	70.53	69.40	68.50	67.00	75.03
88	70.22	69.91	68.88	67.96	66.71	74.62

TABLE 5. Q3 commissioning results for the C-BITCN tandem residual block.

Block Number	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
6	81.66	78.73	77.93	77.87	77.13	83.34
7	82.13	79.24	78.30	78.35	77.69	84.19
8	82.69	79.80	78.91	78.83	78.26	84.84
9	83.21	80.56	79.53	79.52	78.89	85.34
10	82.76	79.88	79.07	79.09	78.37	84.72

TABLE 6. Q8 commissioning results for the C-BITCN tandem residual block.

Block Number	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
2	69.01	68.77	67.71	66.91	65.72	73.67
3	69.28	69.41	68.20	67.22	66.09	74.19
4	69.80	69.88	68.81	67.89	66.57	74.70
5	70.62	70.53	69.40	68.50	67.00	75.03
6	69.93	69.83	68.80	67.76	66.54	74.49

TABLE 7. 3class Tri-fold cross-validation results for CNN, TCN and C-BITCN.

Experiment	CNN	TCN	C-BITCN
Experiment 1	80.52	81.28	84.68
Experiment 2	80.26	80.64	84.59
Experiment 3	80.74	81.53	84.63
Experiment results	80.50	81.15	84.63

TABLE 8. 8class Tri-fold cross-validation results for CNN, TCN and C-BITCN.

Experiment	CNN	TCN	C-BITCN
Experiment 1	70.59	71.63	74.56
Experiment 2	70.43	70.84	74.72
Experiment 3	70.36	70.89	74.60
Experiment results	70.46	71.12	74.62

D. EFFECT OF DIFFERENT PARAMETERS ON RESULTS

In the experiment, we found that many parameters in the model have a significant impact on the final prediction results, such as the size of the CNN sliding window, the dimension of features extracted from the CNN fully connected layer, the number of series, residual blocks in the bidirectional TCN, and filter size. The tuning process of the sliding window size is presented in Table 2.

Different network structures also affected the final results. There are two types of pooling layers, the maximum pooling layer and the average pooling layer, and the maximum pooling layer is used in this study.

In this model, the input data of the bidirectional TCN module contains the features extracted from the fully connected layer of the CNN and processed by the CNN. Therefore, the

TABLE 9. Q3 results for C-BITCN.

Dataset	Q3	QC	QE	QH	SOV
CASP10	83.21	82.16	83.43	86.57	76.59
CASP11	80.56	79.74	80.26	82.45	66.09
CASP12	79.53	76.20	75.54	85.21	74.34
CASP13	79.52	79.38	77.44	84.29	74.41
CASP14	78.89	76.42	79.69	83.10	72.97
CB513	85.34	82.79	83.17	89.10	80.89

size of the fully connected layer will affect the final results of the experiment. The process of adjusting the appropriate fully connected layer is presented in Table 3 and 4. (the input dimension of C-BITCN is $23 \times N$ means the dimension of the extracted CNN features is $3 \times N$, N is the length of the sequence.)

The number of series residual of blocks in the TCN determines the number of layers in the model. The number of layers and the size of the filter jointly affect the final perceived field size. Therefore, the number of series of residual blocks has a significant effect on the final result. The debugging results of the series residual block are presented in Tables 5 and 6, respectively.

E. CROSS-VALIDATION

To better evaluate the effects of the model, we also used three-fold cross-validation. In this experiment, there were 11650 proteins in the training set, containing a total of 2778501 training samples, 1852334 of which were used as the training set and 926167 as the test set in the three-fold cross-validation, with a sliding window of size 20×19 . Three three-fold cross-validation experiments were

TABLE 10. Q8 results for C-BITCN.

Dataset	Q8	QG	QH	QI	QB	QE	QT	QS	QL	SOV
CASP10	70.62	33.26	88.03	0	17.34	79.70	32.38	56.18	65.96	67.39
CASP11	70.53	39.04	87.13	NAN	22.56	77.82	38.86	52.93	53.91	67.23
CASP12	69.40	38.83	87.44	NAN	16.00	75.96	34.52	44.86	51.63	66.26
CASP13	68.50	35.13	83.72	NAN	17.93	70.51	31.09	42.63	51.13	65.51
CASP14	67.00	21.56	85.03	NAN	28.00	71.56	30.91	45.37	50.97	64.80
CB513	75.03	52.34	87.26	89.10	0	55.37	82.63	85.98	65.78	67.12

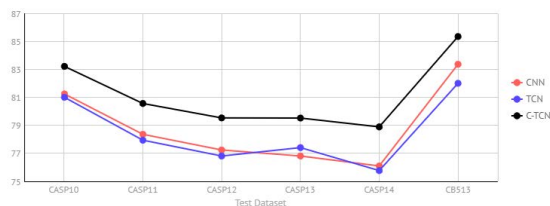


FIGURE 7. Q3 prediction results for CNN, TCN and C-TCN.

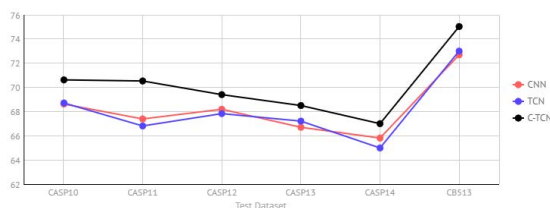


FIGURE 8. Q8 prediction results for CNN, TCN and C-TCN.

conducted to ensure that all data were used as test samples. The cross-validation results of the CNN and C-BITCN are shown in Table 7 and 8.

F. EXPERIMENTAL RESULTS

To evaluate the accuracy and stability of the model in predicting PSS, we used cullpdb containing 11650 proteins as the training set and CASP10, CASP11, CASP12, CASP13, CASP14 and CB513 as the test set. There were no duplicate proteins during training of the test sets.

1) RESULTS FOR C-BITCN

The Q3 and Q8 accuracies of C-BITCN for each dataset are listed in Table 9 and Table 10, respectively.

2) COMPARISON OF C-BITCN WITH CNN AND BIDIRECTIONAL TCN

Comparison of the experimental results of the CNN, bidirectional TCN, and C-BITCN used in the experiments is shown in Fig 7 and 8.

C-BITCN achieved Q3 accuracy of 83.21%, 80.56%, 79.53%, 79.52%, 78.89%, 85.34% and Q8 accuracy of 70.62%, 70.53%, 69.40%, 68.50%, 67.00%, 75.03% on the test sets CASP10, CASP11, CASP12, CASP13, CASP14 and CB513. Compared with the CNN, it is improved by 2.64%, 2.45%, 2.25%, 1.52%, 1.83%, 3.27% and 1.64%, 1.45%, 1.25%, 1.52%, 0.83%, 2.19% respectively.

These experiments indicate that C-BITCN can indeed combine the advantages of CNN and bidirectional TCN, and can achieve higher PSSP results.

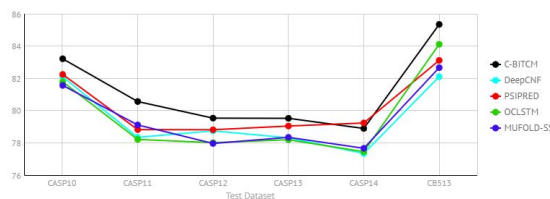


FIGURE 9. Q3 comparison results.

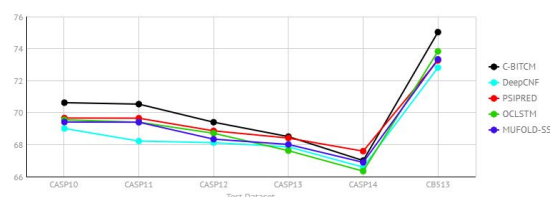


FIGURE 10. Q8 comparison results.

3) COMPARISON OF C-BITCN WITH CNN AND BIDIRECTIONAL TCN

We used Q3 and Q8 as the evaluation criteria and compare the model with PSIPRED [25], DeepCNF [26], OCLSTM [27], and MUFOLD-SS [28]. The comparison results for Q3 and Q8 are presented in Fig 9 and 10, respectively. In Fig 9, the results of PSIPRED, DeepCNF and OCLSTM in the test set are taken from this paper [24] and [25].

IV. CONCLUSION

To analyze the local and global interactions between amino acids in protein sequences, we propose C-BITCN, an artificial neural network based on CNN and bidirectional TCN, for the first time and apply it to protein secondary structure prediction. It can be seen from the experimental results that the accuracy of C-BITCN is significantly improved compared with CNN and bidirectional TCN, indicating that C-BITCN is indeed effective. Compared to other methods, the results show that C-BITCN outperformed the other methods for predicting PSSs.

In our experiment, we used cullpdb after removing duplicate proteins as the training set and CASP10, CASP11, CASP12, CASP13, CASP14 and CB513 as the test set and achieved better results than other methods for PSSP. Although there are many ways to predict the secondary structure of proteins using the binding of CNN and LSTM, few have used CNN and bidirectional TCN to predict the secondary structure of proteins. C-BITCN can achieve good results because it can extract local features from CNN, analyse the interaction and connection between amino acid sequences,

determine the type of amino acids, predict the amino acids on either side of the position, and analyse the interaction of amino acids in protein sequences with large length through bidirectional TCN. The two factors that have a great influence on the prediction of secondary structures are extracting the local characteristics of the amino acid chain and observing the interaction of amino acids in the protein sequence. C-BITCN has these two characteristics simultaneously; therefore, it achieves better results than the other methods.

Because this model adopts maximum pooling, amino acid mutation or information error will affect the prediction results. We will continue to optimize the model and test it on more datasets in the future.

REFERENCES

- [1] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 142–148, May 2018.
- [2] C. Chen, L. Chen, X. Zou, and P. Cai, "Prediction of protein secondary structure content by using the concept of chous pseudo amino acid composition and support vector machine," *Protein Peptide Lett.*, vol. 16, no. 1, pp. 27–31, Jan. 2009.
- [3] M. N. Islam, S. Iqbal, A. R. Katebi, and M. T. Hoque, "A balanced secondary structure predictor," *J. Theor. Biol.*, vol. 389, pp. 60–71, Jan. 2016.
- [4] B. Yang, "Predicting protein secondary structure using a mixed-modal SVM method in a compound pyramid model," *Knowl.-Based Syst.*, vol. 24, no. 2, pp. 304–313, 2011.
- [5] N. P. Bidargaddi, M. Chetty, and J. Kamruzzaman, "Combining segmental semi-Markov models with neural networks for protein secondary structure prediction," *Neurocomputing*, vol. 72, nos. 16–18, pp. 3943–3950, Oct. 2009.
- [6] X.-Q. Yao, H. Zhu, and Z.-S. She, "A dynamic Bayesian network approach to protein secondary structure prediction," *BMC Bioinf.*, vol. 9, no. 1, p. 49, Dec. 2008.
- [7] S. A. Malekpour, S. Naghizadeh, H. Pezeshk, M. Sadeghi, and C. Eslahchi, "Protein secondary structure prediction using three neural networks and a segmental semi Markov model," *Math. Biosciences*, vol. 217, no. 2, pp. 145–150, Feb. 2009.
- [8] Y. T. Tan and B. A. Rosdi, "FPGA-based hardware accelerator for the prediction of protein secondary class via fuzzy K-nearest neighbors with Lempel–Ziv complexity based distance measure," *Neurocomputing*, vol. 148, pp. 409–419, Jan. 2015.
- [9] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, no. 1, pp. 1–11, May 2016.
- [10] Y. Liu and J. Cheng, "Protein secondary structure prediction based on wavelets and 2D convolutional neural network," in *Proc. 7th Int. Conf. Comput. Syst.-Biol. Bioinf.*, Dec. 2016, pp. 53–57.
- [11] C. Schuld, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, 2004, pp. 32–36.
- [12] Z. Li and Y. Yu, "Protein secondary structure prediction using cascaded convolutional and recurrent neural networks," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 1–8.
- [13] R. Heffernan, Y. Yang, K. Paliwal, and Y. Zhou, "Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility," *Bioinformatics*, vol. 33, no. 18, pp. 2842–2849, Sep. 2017.
- [14] Y. Guo, W. Li, B. Wang, H. Liu, and D. Zhou, "DeepACLSTM: Deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction," *BMC Bioinf.*, vol. 20, no. 1, p. 341, Dec. 2019, doi: [10.1186/s12859-019-2940-0](https://doi.org/10.1186/s12859-019-2940-0).
- [15] S. K. Sønderby and O. Winther, "Protein secondary structure prediction with long short term memory networks," 2014, *arXiv:1412.7828*.
- [16] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.
- [17] J. Cheng, Y. Liu, and Y. Ma, "Protein secondary structure prediction based on integration of CNN and LSTM model," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102844.
- [18] G. Wang and R. L. Dunbrack, "PISCES: Recent improvements to a PDB sequence culling server," *Nucleic Acids Res.*, vol. 33, pp. W94–W98, Jul. 2005.
- [19] J. A. Cuff and G. J. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," *Proteins, Struct., Function, Genet.*, vol. 34, no. 4, pp. 508–519, Mar. 1999.
- [20] J. Moul, K. Fidelis, A. Kryshtafovych, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP)—round IX," *Proteins, Struct., Function, Bioinf.*, vol. 79, no. S10, pp. 1–5, 2011.
- [21] J. Moul, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP)—Round x," *Proteins, Struct., Function, Bioinf.*, vol. 82, pp. 1–6, Feb. 2014.
- [22] J. Moul, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (CASP)—Round XII," *Proteins, Struct., Function, Bioinf.*, vol. 86, pp. 7–15, Mar. 2018, doi: [10.1002/prot.25415](https://doi.org/10.1002/prot.25415).
- [23] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moul, "Critical assessment of methods of protein structure prediction (CASP)—Round XIII," *Proteins*, vol. 87, pp. 1011–1020, Dec. 2019.
- [24] J. Moul and A. Kryshtafovych, "Special issue: CASP14: Critical assessment of methods of protein structure prediction, 14th round," *Proteins*, vol. 89, pp. 1603–1997, Jan. 2021.
- [25] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol.*, vol. 292, no. 2, pp. 195–202, Sep. 1999.
- [26] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, no. 1, pp. 1–11, May 2016.
- [27] Y. Zhao and Y. Liu, "OCLSTM: Optimized convolutional and long short-term memory neural network model for protein secondary structure prediction," *PLoS ONE*, vol. 16, no. 2, pp. 1–14, 2021.
- [28] C. Fang, Y. Shang, and D. Xu, "MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction," *Proteins, Struct., Function, Bioinf.*, vol. 86, no. 5, pp. 592–598, Mar. 2018.



YUNQING ZHANG is currently pursuing the M.Eng. degree in computer application technology with the Shandong Academy of Sciences, Qilu University of Technology. His research interests include deep learning, intelligent information processing, and biomedical information processing.



YUMING MA received the master's degree in automatic control from Harbin Engineering University. She is currently an Associate Professor with the Shandong Academy of Sciences, Qilu University of Technology. Her research interests include bioinformatics and machine learning.



YIHUI LIU received the Ph.D. degree in computer science from the University of Nottingham, U.K., in 2004. She is currently a Professor with the Shandong Academy of Sciences, Qilu University of Technology, China. She has published more than 40 articles. Her research interests include biomedical information and medical image analysis, including protein structure prediction, and microarray data.

...