Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research article

SERT-StructNet: Protein secondary structure prediction method based on multi-factor hybrid deep model

Benzhi Dong, Zheng Liu, Dali Xu, Chang Hou, Guanghui Dong, Tianjiao Zhang *, Guohua Wang *

College of Computer and Control Engineering, Northeast Forestry University, Harbin 150040, China

ARTICLE INFO

Keywords:
Protein secondary structure
Multi-factor features
Secondary structure propensity scores
Hybrid deep feature extraction

ABSTRACT

Protein secondary structure prediction (PSSP) is a pivotal research endeavour that plays a crucial role in the comprehensive elucidation of protein functions and properties. Current prediction methodologies are focused on deep-learning techniques, particularly focusing on multi-factor features. Diverging from existing approaches, in this study, we placed special emphasis on the effects of amino acid properties and protein secondary structure propensity scores (SSPs) on secondary structure during the meticulous selection of multi-factor features. This differential feature-selection strategy results in a distinctive and effective amalgamation of the sequence and property features. To harness these multi-factor features optimally, we introduced a hybrid deep feature extraction model. The model initially employs mechanisms such as dilated convolution (D-Conv) and a channel attention network (SENet) for local feature extraction and targeted channel enhancement. Subsequently, a combination of recurrent neural network variants (BiGRU and BiLSTM), along with a transformer module, was employed to achieve global bidirectional information consideration and feature enhancement. This approach to multi-factor feature input and multi-level feature processing enabled a comprehensive exploration of intricate associations among amino acid residues in protein sequences, yielding a Q_3 accuracy of 84.9% and an Sov score of 85.1%. The overall performance surpasses that of the comparable methods. This study introduces a novel and efficient method for determining the PSSP domain, which is poised to deepen our understanding of the practical applications of protein molecular structures.

1. Introduction

Proteins are fundamental components within living organisms, undertaking vital biological functions, such as signal transduction, information exchange, catalysis, and immune responses [1]. The primary structure of a protein is a sequence of amino acids arranged in a specific order [2]. This unique arrangement results in diverse protein types, reflecting the diversity of protein structures [3]. The secondary structure of proteins is the conformation formed by hydrogen-bonded stabilised local structures in polypeptide chains, or by the folding of peptide chain backbone atoms [4]. The secondary structure of proteins acts as a bridge between the primary and tertiary structures, constituting a pivotal focus within the field of protein structure prediction. Typically, secondary structures include three basic types: helices (H), sheets (E), and coils (C). These secondary structures are not uniformly distributed among different proteins and vary in type and quantity [5]. Initial research on secondary structures relied on wet experimental methods such as X-ray crystallography and nuclear magnetic resonance [6]. However, these methods are time consuming, expensive, and subject to uncertainties. Therefore, there is an urgent need to develop more efficient secondary structure prediction methods for proteins.

Various efficient methods have been explored to improve protein secondary structure prediction. Initially, Burkhard Rost et al. [7] proposed the PHD method, which uses a contour alignment algorithm to automatically process the sequence contour consisting of amino acids and connect the sequence to the structure through a feedforward network, thus predicting the secondary structure of proteins. Subsequently, machine learning methods have been gradually employed in this field. For example, Aydin et al. [8] proposed the IPSSP method in 2006, in which they constructed an improved Hidden Semi-Markov Model (HSMM) and introduced a residue-dependent model based on it to comprehensively consider amino acid correlations at the boundaries

* Corresponding authors.
E-mail addresses: tianjiaozhang@nefu.edu.cn (T. Zhang), ghwang@nefu.edu.cn (G. Wang).

https://doi.org/10.1016/j.csbj.2024.03.018
Received 5 December 2023; Received in revised form 20 March 2024; Accepted 21 March 2024
Available online 22 March 2024

of structural segments. Later, as support vector machines (SVM) demonstrated high performance in protein secondary structure prediction, nonlinear classification ability, and adaptability to learning with fewer samples, an increasing number of researchers began to adopt SVM in this field [9,10]. Recently, deep-learning methods have made significant progress in protein secondary structure prediction [11]. Wang et al. [12] proposed a deep convolutional neural network with a self-supervised conditional random field capable of modelling complex sequence structural relationships and considering the interdependence between structures. Furthermore, the RaptorX-SS web server proposed by Li et al. [13] is also suitable for proteins that lack close homologues or have sparse sequence profiles, which can consider a variety of factors that may affect the prediction from the data and the model itself. In addition to the above methods, other deep-learning methods, such as JPRED [14], Porter 5 [15], and CFLM [16] have also been proposed as efficient protein secondary structure prediction schemes and have made important contributions to this field.

However, as deep learning continues to advance, a model-centric approach alone is no longer sufficient to satisfy the demands of protein secondary structure prediction. More researchers are focusing on exploring the input features. The method proposed by Błazewicz et al. [17] analyses the importance of various amino acid properties in protein secondary structure prediction to address related tasks. Moreover, Li et al. [18] presented a hybrid encoding prediction approach that combined multiple amino acid physicochemical properties with secondary structure trend factors to form fused multidimensional encoding. They utilised an SVM to predict the secondary structures of proteins. More recently, Uzma et al. [19] introduced a novel protein secondary structure prediction model called the Protein Encoder, which employs an ensemble feature selection-based approach. This method utilises unsupervised autoencoders for feature extraction. By integrating information from multiple feature subsets, we select the optimal feature subset for classification prediction. Recently, analysis from the perspective of input features has become the primary approach for protein secondary structure prediction.

Despite the wide adoption of diversified data as input features, these methods often overlook the differential impacts of these distinct features on various protein secondary structures. These methods typically rely on various amino acid properties and other protein-related information but do not adequately consider the unique contributions of these features in predicting the secondary structure or the differences in the contributions from different feature combinations. The key to successful protein secondary structure prediction lies in understanding how different features capture and represent the uniqueness of protein secondary structures during the feature-selection process. Relying solely on diverse data inputs is insufficient; a deeper investigation into individual features and their combined correlations and contributions is required. Thus, when performing feature selection, we specifically considered the key influences of the features. Sequence-related features, such as protein sequences and the protein sequence position specificity matrix (PSSM) [20], provide sequence-based evolutionary information and amino acid alignment features; this combined use is sufficient to comprehensively capture the conserved nature of protein sequences and provide a rich dimension of information. Furthermore, amino acid-related features, such as properties obtained after selection [21] and secondary structure propensity scores (SSPs). We fully considered the differential impact of different amino acid properties on protein secondary structure, and this variability motivated us to make comprehensive considerations and select amino acid properties that may affect protein structure. For example, hydrophobicity may affect β-fold formation, while pKa may affect α-helix formation [22] and so on. Based on this, we selected features from various amino acid properties that have a significant effect on the formation of different secondary structures, which are important features that can be used for protein secondary structure prediction. Simultaneously, we applied the SSPs as input features and obtained scores for each factor using a computational method. This approach enhanced the predictive accuracy and robustness of the model in terms of protein structure, providing support for predicting various structural categories.

To address the issues raised above, this study addressed the following two points:

- In terms of input features, we integrated multiple factors including protein sequence, position-specific scoring matrix (PSSM), selected amino acid properties, and secondary structure propensity scores (SSPs). Our focus was not only on expanding the richness of the input features but also on investigating the effectiveness of different factors and the variations between different combinations of these factors to identify the optimal feature combination.
- In terms of the model design, we focused on developing a hybrid feature extraction model for better feature extraction of multiple factors. The model uses a combination of dilated convolution (D-Conv) [23] and a channel attention network (Squeeze-and-Excitation Network, SENet) [24] for local feature extraction. Two different recurrent neural network variants (BiGRU [25] and BiLSTM [26]) were employed in parallel for global feature extraction. Finally, feature enhancement is performed using a transformer [27] model. Feature extraction was performed at different levels and perspectives to fully utilise the effective information provided by the input features, thus improving the accuracy of the prediction results.

Through a series of experiments, we not only validated the influence of multi-factor inputs on prediction accuracy but also emphasised the significant role of selected amino acid properties and SSPs in protein secondary structure prediction tasks. These experimental results reflect the efficiency of the prediction model involved in data processing. Additionally, our findings demonstrated that the proposed method exhibited excellent performance on the test datasets.

2. Methods and materials

2.1. Datasets

The datasets utilised in this study included the benchmark dataset SCRATCH-1D [28], along with the publicly available datasets CB513 [29], CASP11 and CASP10 (https://predictioncenter.org/). The SCRATCH-1D dataset comprises secondary structure information of 8059 protein sequences. The structural information of these proteins is derived from X-ray crystallography, a process with a resolution of at least 2.5 angstroms and no chain breaks. The sequence similarity of the datasets was maintained at 25% to ensure fair performance evaluation. Additionally, CB513, CASP11, and CASP10 datasets are commonly used to test and compare protein secondary structure prediction methods. The CB513 dataset contained secondary structural information for 513 protein sequences. In the data preprocessing stage, we screened and eliminated unnatural residue data such as residue representations with " X " symbols and obtained 471 protein data points for our study. We conducted a similar procedure for the CASP11 and CASP10 datasets, the details of which are not explicitly described herein. Using these three datasets, we can ensure that our study is broadly applicable and comparable, and thus evaluate the performance of our methods more fully.

In addition to the protein sequence data mentioned above, we employed PSI-BLAST [30] to compare the target protein sequences with a database. Through multiple iterative comparisons with a threshold of 0.001, we aggregated and weighted the amino acid information from different positions to generate a corresponding position-specific scoring matrix. This PSSM matrix contains scores for each amino acid at different positions, reflecting their significance within a sequence. Based on the unique effects of different amino acid properties on the secondary structure of proteins, we carefully selected the corresponding amino acid property parameters from the AAindex database [31], including the acid-base properties of molecules with acidic protons (pKa1), acid-base properties of molecules with basic protons (pKb2), isoelectric point PH (pl4), and hydrophobicity (H). These property parameters were chosen based on their effects on different secondary structure types, and detailed information is provided in Table 1. During the computation of the secondary structure propensity scores (SSPs), we considered a certain regularity in the distribution of secondary structures. We used these data to determine the frequency of occurrence of each secondary structure in the datasets, which helped us quantify their relative importance. By computing the frequency of the appearance of different structures, we derived the propensity scores. The utilisation of these data will contribute to a more comprehensive understanding and evaluation of the performance and reliability of protein secondary structure prediction methods.

2.2. Overall architecture of SERT-StructNet

To process the introduced protein sequence data, amino acid attributes obtained after selection, and secondary structure propensity scores more efficiently, this study proposes a hybrid deep-learning model called SERT-StructNet. The model consists of three submodules: (1) multi-factor encoding and fusion, (2) hybrid deep feature extraction, and (3) a prediction output module. The overall architecture of the proposed model is shown in Fig. 1. First, in Module (1), we process the multi-factor data. Four types of data were processed: protein sequences, amino acid properties, secondary structure propensity scores (SSPs), and a protein position-specific scoring matrix. We One-hot-coded the protein sequences, selected the amino acid properties based on different secondary structure properties, computed and normalised the SSPs, and then fused them with the generated PSSM matrix to generate multi-factor input features. Subsequently, in Module (2), a multichannel feature extraction parallel mechanism and a transformer module are employed; hence, the model is termed a hybrid deep-learning model. In detail, to efficiently extract valuable information from a wide range of data, we use a multi-level hybrid feature extraction approach. Initially, we employed a dilated convolution (D-Conv) and channel attention network (Squeeze-and-Excitation Network, SENet) for local feature extraction, along with a bidirectional gated recurrent unit (BiGRU) and bidirectional long short-term memory (BiLSTM) for global feature extraction. In addition, to enhance the model's understanding of the data, we introduced a transformer module. Specifically, we first apply D-Conv and SENet during data incoming to capitalise on the expanded receptive field and selective channel enhancement. During the global feature extraction stage, two different variants of recurrent neural networks are used in parallel for bidirectional feature consideration and global feature extraction, and feature enhancement is processed by the transformer module. In Module (3), the output features from the aforementioned modules were introduced into an ((MLP) and employed for the structural classification output. The MLP abstracts and processes complex features, ultimately predicting the structural state of each amino acid residue, which is the result of the classification prediction for the structural state.

2.3. Multi-factor feature input

Diverse treatments are necessary owing to the introduction of various types of input data. First, we adopted One-hot encoding to process protein sequence data. In this encoding method, amino acid residues were converted into binary 0 s and 1 s, with each point in the vector assigned a distinct position. This code assigns specific positional information to each amino acid residue, and provides crucial inputs for subsequent data processing and fusion. This approach aids in a more intuitive understanding of the protein structure. Simultaneously, by leveraging the protein sequence information, we used PSI-BLAST to generate the corresponding n × 20 position-specific scoring matrix, where n represents the length of the protein sequence and 20 represents different categories of amino acid residues. Each position in this matrix indicates the specific positional score of the corresponding amino acid, reflecting the relative affinity and evolutionary conservation of the amino acid residues at particular positions. When choosing the nature of the amino acids, we fully considered the effects of different amino acid properties on the secondary structure. For example, polar amino acids maintain the folded structure and stability of proteins by participating in protein interactions, whereas hydrophobic amino acids promote protein folding by binding to water molecules on the surface of proteins, and acidic and basic amino acids affect the structure and stability of proteins by altering the distribution of charge in proteins. This selection process ensured that we effectively considered the effects of the amino acid properties. Ultimately, we chose four amino acid properties as the input data: acid-base properties of molecules with acidic protons (pKa1), acid-base properties of molecules with basic protons (pKb2), isoelectric point PH (pl4), and hydrophobicity (H). Finally, concerning the computation of secondary structure propensity scores (SSPs), our focus lies in comprehending the secondary structure patterns (commonly represented as H, E, C, denoting α-helix, β-strand, and coil) within protein sequences. Fig. 2 shows a schematic representation of the distribution of secondary structures.

We believe that these secondary structures are not randomly distributed within the given datasets but exhibit certain statistical regularities. Therefore, we used the following analytical approach. First, we determined the propensity factor for amino acid A, which consists of PH, PE, or PC, as shown in Eq. (1):

$$Pi = Ai/Ti, i \in \{H, E, C\}\# \tag{1}$$

Where *Ai* represents the score of the amino acid in the conformation of secondary structure i, *Ti* is the total score of the conformation of secondary structure i, and *Pi* denotes the secondary structure propensity scores.

Next, we normalised the SSPs and transformed this frequency information into a standardised encoding. Considering the propensity scores of amino acid A as an example, the normalisation process is shown in Eq. (2):

$$Pi = (Pi - P\min)/(P\max - P\min)\# \tag{2}$$

Table 1
Property parameters of 20 amino acids.

Amino acid	Properties				Amino acid	Properties			
	pK_{a1} (COOH)	pK_{b2} (NH^{3+})	pl4	H		pK_{a1} (COOH)	pK_{b2} (NH^{3+})	pl4	H
A	0.62	2.34	9.69	6	M	0.64	2.28	9.21	5.74
C	0.29	1.96	10.28	5.07	N	–0.78	2.02	8.8	5.41
D	–0.9	1.88	9.6	3.65	P	0.12	1.99	10.6	6.3
E	–0.74	2.19	9.67	4.25	Q	–0.85	2.17	9.13	5.65
F	1.19	1.83	9.13	5.48	R	–2.53	2.17	9.04	10.76
G	0.48	2.34	9.6	5.79	S	–0.18	2.21	9.15	5.68
H	–0.4	1.82	9.17	7.59	T	–0.05	2.09	9.1	5.6
I	1.38	2.36	9.6	5.97	V	1.08	2.32	9.62	5.96
K	–1.5	2.18	8.95	9.74	W	0.81	2.83	9.39	5.89
L	1.06	2.36	9.6	5.98	Y	0.26	2.2	9.11	5.66

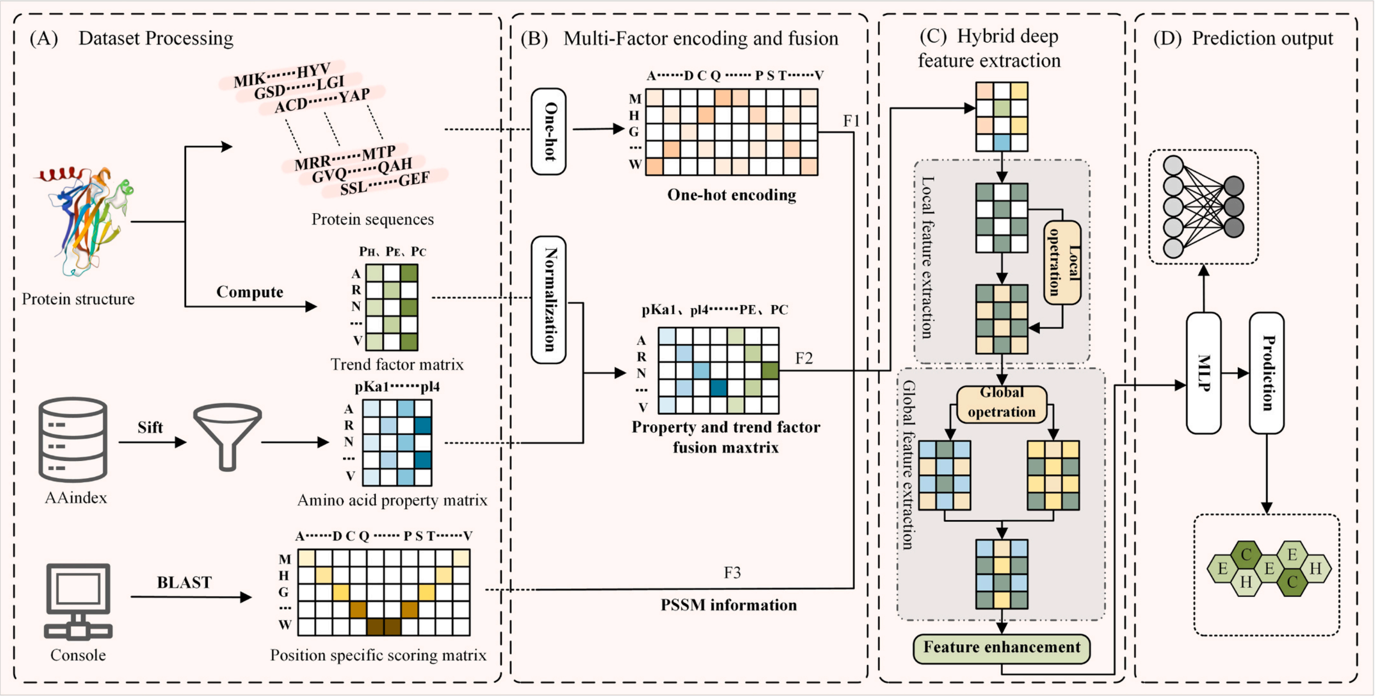

Fig. 1. Framework of SERT-StructNet. (A) Dataset processing. Various data sources are collected using diverse methods and passed on to subsequent modules. (B) Multi-factor encoding and fusion. Different encoding and computational methods are applied to process multi-factor data for further handling by the network model. (C) Extracting comprehensive features through the Hybrid deep feature extraction module and enhancing learning through the Transformer module. (D) Finally predicting the respective secondary structure conformations of amino acid residues via the MLP module.

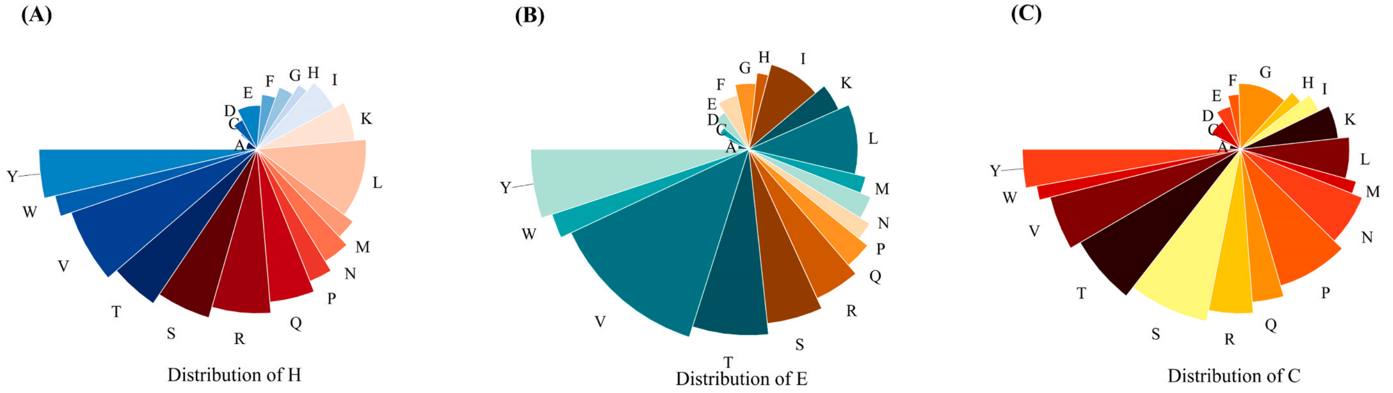

Fig. 2. Illustrative Diagram of Secondary Structure Distribution. (A) Distribution of secondary structure H in the dataset; (B) Distribution of secondary structure E in the dataset; (C) Distribution of secondary structure C in the dataset.

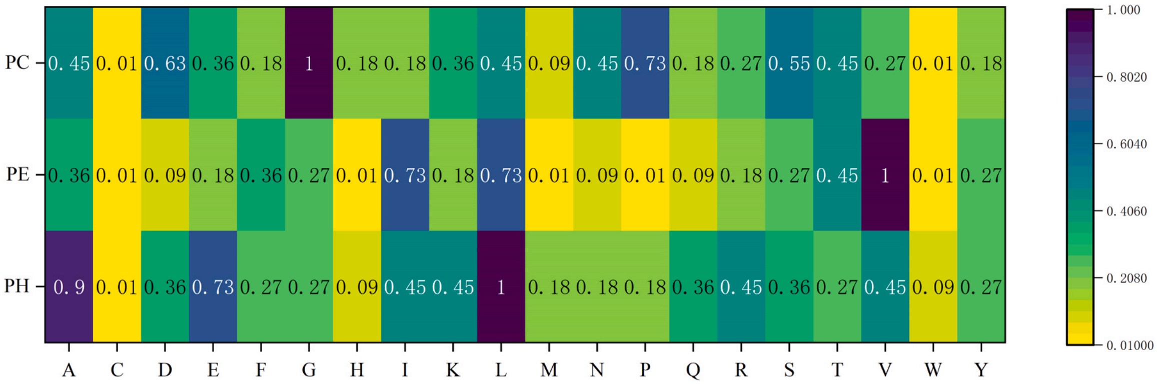

Fig. 3. Scores of Secondary Structure Propensity.

This encoding method simplifies data representation while preserving important structural information. Through this step, we can capture and compare secondary structure propensities across different protein sequences more effectively. Fig. 3 presents the secondary structure propensity scores.

2.4. Dilated convolution and channel attention module

In Fig. 4, we show an illustrative representation of local feature extraction. Emphasis is placed on demonstrating the utilisation of dilated convolution (D-Conv) and channel attention networks (SENet) for local feature extraction.

Specifically, after the data are fed into the network, the first step in data processing is the dilated convolution (D-Conv) module, which expands the sensory field of the model by continually updating the convolution kernel and injecting null values (also known as dilatation rates) into the convolution. Adjusting the dilation rate aids in capturing features at different scales, thereby obtaining a more comprehensive set of feature information. Dilated convolution is described by Eq. (3):

$$Y[s] = \left(X *_d f\right)(s) = \sum_{i=0}^{k-1} f(i) \bullet X_{s-d\bullet i}\# \tag{3}$$

where $Y[s]$ represents the dilated convolution result, X signifies the input element, w represents the convolutional kernel, d is the dilation rate, f is the filter size, k is the filter size, and s is the position of the sequence element.

After the calculation using D-Conv, while expanding the receptive field and extracting information more comprehensively, it is inevitable that some noise and irrelevant information might be introduced. To address this issue, we introduced a channel attention network (SENet) to achieve selective channel enhancement and selectively strengthen useful features to obtain effective information. SENet first performs the "Squeeze" operation, obtaining the importance weight for each channel by using global average pooling, i.e., obtaining the contribution ratio of each channel to the overall information. Then, in the "Excitation" stage, a small neural network is introduced to learn and generate the weights. Finally, the weights were applied to the original feature channels through element-wise multiplication to recalibrate the original features in the channel dimensions. The calculations of SENet are shown in Eqs. (4)–(6):

$$z = F_{sq}(Y_s) = \left(\sum_{i=1}^{M}\sum_{j=1}^{N} Y_s(i,j)\right) \Big/ (M \bullet N)\# \tag{4}$$

$$s = F_{ex} = \sigma(W \bullet z + b)\# \tag{5}$$

$$t = F_{scale} = s \bullet Y_s\# \tag{6}$$

where Y_s represents the output from the previous stage of dilated convolution, and z, s, and t denote the three stages of operation in SENet: Squeeze, Excitation, and Scale.

2.5. Parallel recurrent neural network variant module

Considering that the data cover a wide range of factors that contain a variety of different feature information, we performed global feature extraction to process and utilise the input features of these multi-factor data more effectively. Fig. 5 presents an overview of the two RNN variants of recurrent neural networks.

By performing parallel operations, we can input data features into both the bidirectional gated recurrent unit (BiGRU) and bidimensional long short-term memory (BiLSTM) modules. This combination can introduce diversity and complexity into the model, improving its generalisation, and making it better suited for different types of sequence data. Initially, the data after local feature extraction were input into the BiGRU, where the data underwent learning through both forward and backward recurrent units to learn data information. Because the information flows between these two gated units in opposite directions, the model can comprehensively capture dependencies within the data sequence. Finally, information from these two directions is concatenated to form a global feature representation.

Considering that the GRU unit has only two gating units, the update gate and the reset gate, it is more suitable for the analysis of straightforward amino acid sequence data. In Eqs. (7)–(10) represent the GRU calculation process.

$$Z_t = \sigma(W_Z \bullet [h_{t-1}, x_t])\# \tag{7}$$

$$r_t = \sigma(W_r \bullet [h_{t-1}, x_t])\# \tag{8}$$

$$\widetilde{h}t = \tanh(W_h \bullet [r_t \odot h_{t-1}, x_t])\# \tag{9}$$

$$h_t = (1 - Z_t) \odot h_{t-1} + Z_t \odot \widetilde{h}\# \tag{10}$$

Where Z_t represents the update gate, r_t denotes the reset gate, $\odot$ indicates element-wise multiplication, and h_t represents the final hidden state updated after passing through the gated units.

The data extracted for the local features were also input in parallel to the BiLSTM. The operational mechanism of BiLSTM is similar to that of gated units; however, unlike GRU, BiLSTM comprises three gating units: forget, input, and output gates. This characteristic makes BiLSTM more stable in data processing, allowing for finer information control and memory. Therefore, it is more suitable for handling sequences that require a deeper analysis, such as relatively complex amino acid sequences. In Eqs. (11)–(15) represent the computational process of LSTM:

$$f_t = \sigma\left(W_f \bullet [h_{t-1}, x_t] + b_f\right)\# \tag{11}$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i)\# \tag{12}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c \bullet [h_{t-1}, x_t] + b_c)\# \tag{13}$$

$$o_t = \sigma(W_O \bullet [h_{t-1}, x_t] + b_o)\# \tag{14}$$

$$h_t = o_t \odot \tanh(C_t)\# \tag{15}$$

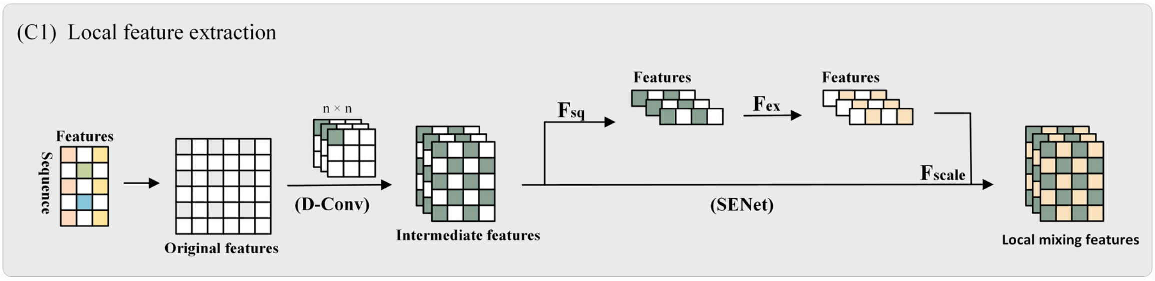

Fig. 4. Architecture of Dilated Convolution (D-Conv) and Channel Attention Network (SENet). Where n represents the size of the convolution kernel that keeps going through multiple scale changes.

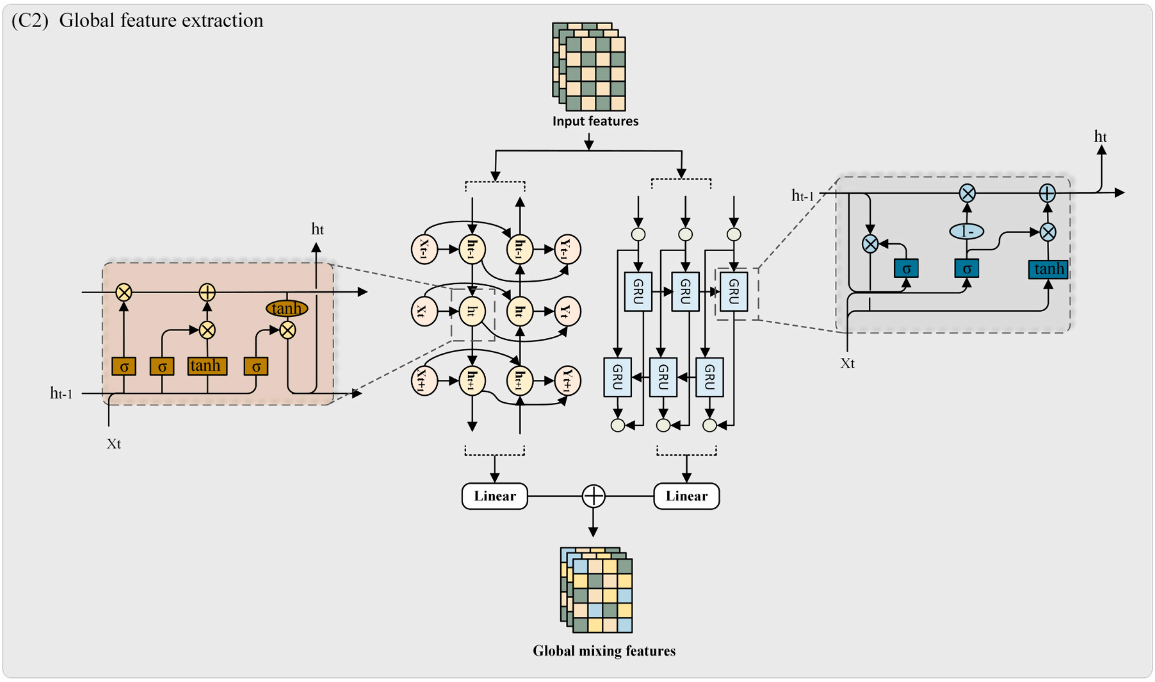

Fig. 5. Parallel Architecture of Bidirectional Gated Recurrent Units (BiGRU) and Bidirectional Long Short-Term Memory (BiLSTM) Networks.

In summary, through the parallel application of both recurrent processing mechanisms, the proposed model can analyse various types of data and capture feature information from different positions, thereby enhancing its performance.

2.6. Feature enhancement module

To enhance the effectiveness of the output information from feature extraction, we introduced a transformer module, the architecture of which is shown in Fig. 6. Within this module, we primarily utilised the transformer encoder module, which has found extensive applications in the field of natural language processing and has gradually been introduced to tasks related to protein secondary structure in recent years. The purpose of this module is to fully utilise the similarities between the tasks of predicting secondary protein structures and natural language processing. Both tasks primarily involve the processing of sequence data. Therefore, the core idea behind introducing the transformer encoder module is to exploit the self-attention mechanism, allowing for more efficient handling of sequence data (Eqs. Eqs. (16) and (17) show the calculation process for the attention mechanism.

$$\begin{cases} Q = h_t \bullet W_Q \\ K = h_t \bullet W_K \# \\ V = h_t \bullet W_V \end{cases} \tag{16}$$

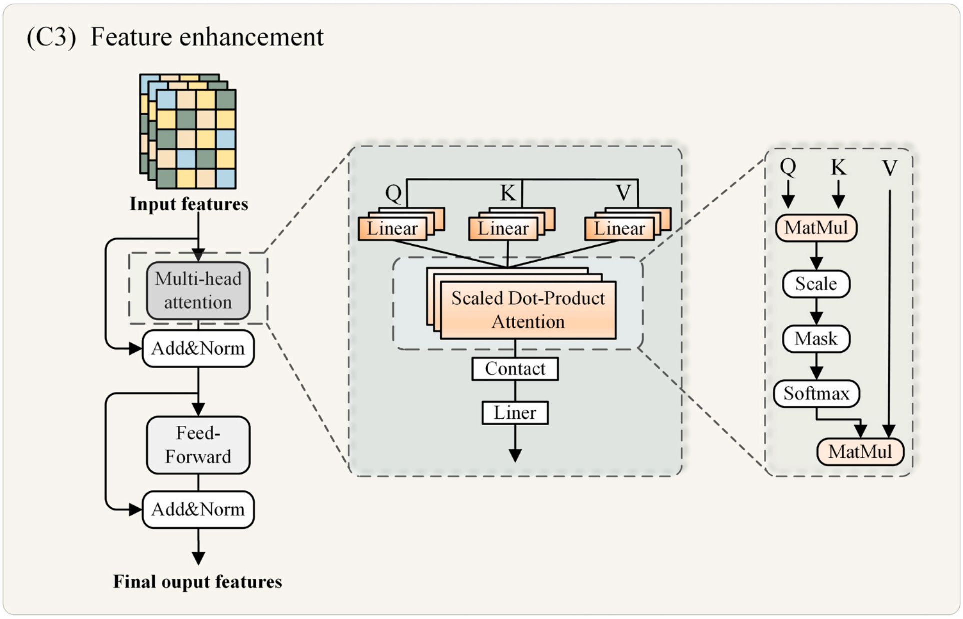

Fig. 6. Framework of Transformer Architecture.

$$Attention(Q,K,V) = Softmax\left(QK^{T}\Big/\sqrt{d_k}\right) \bullet V\# \tag{17}$$

Where Q represents the query vector; K denotes the key vector; V is the value vector; and $W_{Q,K,V}$ signifies the weight coefficients. Following the final attention-scaling calculation, weights were utilised to obtain the ultimate self-attention output.

Following the calculation of attention scores, processing through multi-head attention is essential. This step enables the model to handle different parts and features simultaneously, allowing each attention head to learn distinct representations. Moreover, the parallel nature of multi-head attention makes the computational process more efficient. Subsequently, the introduction of a non-linear activation function through a feedforward network is crucial for the model. This helps the model capture complex features and patterns, leading to a better fitting of the training data. Finally, by applying the residual connection and layer normalisation operations multiple times, the convergence speed of the model can be accelerated, enabling it to adapt to the training data more rapidly, Eqs. (18)–(21) reflects the overall operational process.

$$MultiHead(h_t) = Concat(head_1, head_2, \dots, head_n) \bullet W^{O}\# \tag{18}$$

$$H^{'} = Layer \quad norm(h_t + MultiHead(h_t))\# \tag{19}$$

$$FFN(h_t) = ReLU(h_t \bullet W_1 + b_1) \bullet W_2 + b_2\# \tag{20}$$

$$H = Layer \quad norm(FFN(h_t) + h_t)\# \tag{21}$$

In the mechanism of multi-head attention, a critical operation involves the matrix multiplication (MatMul) of queries (Q) and keys (K) to generate a similarity matrix, which measures the weight relationship between different positions. To stabilise the model training, a scaling operation (scale) was introduced by dividing the similarity matrix by a scaling factor ($\sqrt{d_k}$, where d represents the dimensions of Q and K). Additionally, in the data processing phase, a masking operation (mask) is employed to filter or shield data, helping the model discard irrelevant information and thereby enhancing its focus on crucial details.

2.7. Model training and predicting process

2.7.1. Prediction process

In the prediction module, we initially perform linear transformations using a multilayer perceptron (MLP) to maintain the dimensionality of intermediate representations obtained from the feature extraction stage in alignment with the dimensions of the labels. This action moulds the data in a high-dimensional space, harmonising their dimensions with those of the target labels. Subsequently, we used a loss function for the prediction. During this process, the model predictions were compared individually with the actual labels, resulting in predictions across various channels. Each channel represents the model's confidence score for different labels, where higher scores in the channel typically correspond to the most probable structural categories. By comparing and analysing the predicted results with real labels, we incrementally obtained accurate predictions.

The comparison process takes place at each time step in the amino acid sequence, allowing the model to continuously optimise its parameters based on the loss function (Eq. Eq. (22) shows the calculation principle of the loss function used in this study.

$$loss = -\sum_{t=1}^{T}\sum_{i=1}^{C} p_i^t \log(q_i^t)\# \tag{22}$$

where T represents the total time steps in the sequence, C represents the number of label categories, and p_i^t represents the probability of the ith category of the true label at time step t. First, the probabilities at each time step for each category are summed, and subsequently, an overall sum operation is carried out across all time steps.

The model parameters were continually updated through iterative cycles, indicating that the model progressively learned how to enhance its predictions, leading to sustained improvements in its predictive outcomes. This step-by-step training and optimisation process ensured that the model predicted the input data more accurately. Owing to the particular nature of protein secondary structure prediction tasks, in which each amino acid in a protein sequence corresponds to a specific secondary structure category, the model must precisely identify the category to which each amino acid residue belongs. Therefore, this iterative training method allows the model to steadily improve its performance and gradually approach real predictive results.

2.7.2. Experimental environment

Hardware equipment used in this study:

- CPU: Intel Xeon Gold 5218 R, 2.10 GHz;
- GPU: RTX 2080Ti (11 GB), cuda11.1;
- Memory: 32 GB.

This study is based on the ubuntu 16.04 operating system, Python 3.7, using the Pytorch framework to implement the model and complete the experiments.

2.7.3. Performance evaluation

In this study, two key metrics were used to evaluate the performance of the proposed SERT-StructNet model. First, we focused on the percentage of accurately predicted amino acid residues for the three-state secondary structure (H, E, and C), commonly referred to as the accuracy (Q_m), as shown in Eq. (23):

$$Q_m = \sum_{i=1}^{m} A_i \Big/ N\# \tag{23}$$

Where Q_m represents the accuracy, m has a value of three, denotes the number of categories for the three-state secondary structure, N represents the total number of amino acid residues, and A_i represents the number of correctly predicted amino acids.

The next was Segment Overlap (Sov), a significant metric used to compare predicted protein secondary structure results. Its primary purpose is to measure the similarity between the predicted and actual protein secondary structures, as shown in Eq. (24):

$$Sov = 100 * \sum_{S0}\left[\frac{minov(S_1,S_2) + \sigma(S_1,S_2)}{maxov(S_1,S_2)} \bullet length(S_1)\right]\Big/ N_{Sov} \tag{24}$$

Where N_{Sov} represents the total number of amino acid residues in the protein sequence, S_1 is the actual segment, S_2 is the predicted segment, S_0 represents the segments with the same structure in both S_1 and S_2, $maxov(S_1,S_2)$ denotes the maximum Sov between S_1 and S_2, $minov(S_1, S_2)$ denotes the minimum Sov between S_1 and S_2, and the boundary factor $\sigma(S_1,S_2)$ stands for the similarity score between S_1 and S_2, as shown in Eq. (25):

$$\sigma(S_1,S_2) = \min\left\{\begin{array}{c} maxov(S_1,S_2) - minov(S_1,S_2) \\ minov(S_1,S_2) \\ int[len(S_1)]/2 \\ int[len(S_2)]/2 \end{array}\right. \# \tag{25}$$

3. Results and discussions

3.1. Comparison with existing secondary structure prediction methods

To evaluate the performance of the proposed SERT-StructNet, we compared and assessed six existing protein secondary structure prediction methods on the same test dataset: DeepCNF [12], RaptorX-SS [13], JPRED [14], Porter 5 [15], Protein Encoder [19] and WGACSTCN[32]. In Table 2, we present the Q_3 accuracy and Sov of the SERT-StructNet method and other prediction methods on the same test set to obtain a more comprehensive understanding of the advantages or disadvantages of the performance of the SERT-StructNet method with respect to the other methods. Our research results in terms of Q_3 accuracy and Sov scores indicate that the proposed SERT-StructNet method consistently outperforms the six state-of-the-art methods. With Q_3 accuracy and Sov scores reaching 84.9%, 83.2%, 85.1%, 84.6%, 82.7%, and 82.5% in the three different test sets, respectively, it significantly surpasses other comparative models. For example, using the CB513 dataset, our method demonstrated a significant advantage over RaptorX-SS, showing a notable increase in almost all metrics, with substantial 6.6% and 5.6% increases in Q_3 accuracy and Sov scores, respectively. Similarly, compared with JPRED, our method displayed superior performance, outperforming JPRED across nearly all evaluation metrics. Specifically, it outperformed JPRED by 3.2% for Q_3 accuracy and 1.8% for Sov scores. Compared to DeepCNF, our method also exhibited significant improvements in various metrics, with a 2.6% increase in Q_3 accuracy and a 0.3% increase in Sov scores. Compared to the Porter 5 method, our method was 1.1% and 4.1% higher in terms of Q_3 accuracy and Sov score, respectively. Finally, compared to the WGACSTCN and Protein Encoder methods, the SERT-StructNet method demonstrated considerable performance enhancement, exceeding WGACSTCN by 0.2% in Q_3 accuracy and 4.9% in Sov scores, and surpassing the Protein Encoder by 0.8% in Q_3 accuracy and 4% in Sov scores.

The outstanding performance of our method can be attributed to the efficacy of the data utilised, and the robust data processing and feature extraction capabilities of the proposed model. This approach comprehensively considers the crucial data features required for protein secondary structure prediction, and analyses sequences from a bidirectional perspective to effectively capture their features. Furthermore, our model employs an attention mechanism to enhance feature representation, which improves the prediction performance. Moreover, data selection based on different structural features enhances the information content of datasets, thereby facilitating improved feature extraction. This method not only comprehensively considers data features but also amalgamates multiple information types, allowing a better capture of the complexity of protein structures and laying a solid foundation for accurate predictions.

3.2. 8-State protein secondary structure analysis

In this study, we investigated the 8-state protein secondary structure of an eight-state protein. The results in Table 3 show that our method performs equally well in 8-state protein secondary structure prediction. Our model takes full advantage of multi-factor data and better integrates multiple data features, leading to excellent results in the field of 8-state secondary structure. Notably, our method performs well in traditional 3-state protein secondary structure prediction and achieves good performance in the challenging 8-state secondary structure domain. This not only confirms the robustness and versatility of our method but also highlights its excellent performance in more complex protein secondary structure prediction.

Table 2
Comparison of the performance of our proposed SERT-StructNet with existing methods for 3-state PSSP on the test set.

Methods	CB513		CASP11		CASP10	
	Q_3 (%)	Sov (%)	**Q_3** (%)	Sov (%)	**Q_3** (%)	Sov (%)
RaptorX-SS	73.3	79.5	79.1	81.1	78.9	80.2
JPRED	81.7	83.3	80.4	82.0	81.6	82.4
DeepCNF	82.3	84.8	82.3	83.7	80.7	76.9
Porter 5	83.8	81.0	82.1	81.4	81.3	80.1
WGACSTCN	84.7	80.2	81.8	77.9	82.1	78.4
Protein Encoder	84.1	81.1	81.6	80.0	79.1	75.4
SERT-StructNet	**84.9**	**85.1**	**83.2**	**84.6**	**82.7**	**82.5**

Table 3
8-state PSSP performance comparison between our proposed SERT-StructNet and existing methods on the test set.

Methods	CB513		CASP11		CASP10	
	Q_8 (%)	Sov (%)	**Q_8** (%)	Sov (%)	**Q_8** (%)	Sov (%)
RaptorX-SS	64.9	71.5	65.1	73.6	64.8	72.4
JPRED	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
DeepCNF	68.3	76.8	72.3	77.5	71.8	70.4
Porter 5	68.3	64.8	71.0	66.7	70.1	72.0
WGACSTCN	75.1	73.2	71.0	69.8	70.3	69.3
Protein Encoder	73.1	66.9	70.6	68.8	71.1	72.4
SERT-StructNet	**75.6**	**73.9**	**73.4**	**72.3**	**72.2**	**74.5**

Simultaneously, we performed similar structural studies on the individual haplotypes of the 8-state secondary structure. Table 4 demonstrates that there were significant differences among the predicted 8-protein secondary structures. The characteristics of the dataset and the effect of model errors led to large differences in the frequency of occurrence of the 8-structure haplotypes, and the number of type I structures was almost zero. In the precision analysis of specific structure types, we observed that the precision of 3-state structures such as structure types H, E, and C, was higher. This suggests that the 3-state structures dominate the 8-state structures and constitute an important part of them. Interestingly, we performed detailed precision analyses for each structure type (HECTGISB), further deepening our understanding of the secondary structure of 8-state proteins.

3.3. Case study discussion for comparison with AlphaFold2

In this section, we compare and analyze the results of this study with those of AlphaFold2 (AF2). As the state-of-the-art protein structure prediction model, AF2 is known for its accurate running results and large amount of data, but it consumes a lot of running resources, has a high arithmetic demand, and has high requirements for operators. In contrast, our research model is characterized by its lightweight and low resource consumption. Despite the fact that AF2 leads in terms of performance, after comparative analyses, our research method still shows significant advantages in specific segments.

We processed the AlphaFold Protein Structure Database (AlphaFold DB) and the dataset used in this study, selecting data common to both, yielding a total of 491 case study data. The structure files predicted by AlphaFold2 were put through VADAR to derive the corresponding secondary structures, which were compared with the data of our study. The comparative analysis by Clustal algorithm shows that the overall accuracy of AF2 is 91.3% and the overall accuracy of our study is 84.7%. We can find that the performance of AF2 is perfect, although our prediction performance fails to exceed AF2 in the whole, this study still partially outperforms the prediction results of AF2 in some segments. The results of the comparative analysis of the overall dataset of this study with AF2 are presented in Fig. 7. Through statistical analysis, it is found that our

Table 4
Performance of our proposed SERT-StructNet for each structural haplotype in 8-state PSSP.

Dataset	CB513	CASP11	CASP10
QH	80.40	79.25	77.80
QE	82.29	82.26	84.89
QC	77.54	76.30	77.23
QT	61.34	63.09	59.26
QG	21.15	16.66	19.66
QI	0	0	0
QS	72.73	73.78	69.14
QB	74.78	75.75	62.90

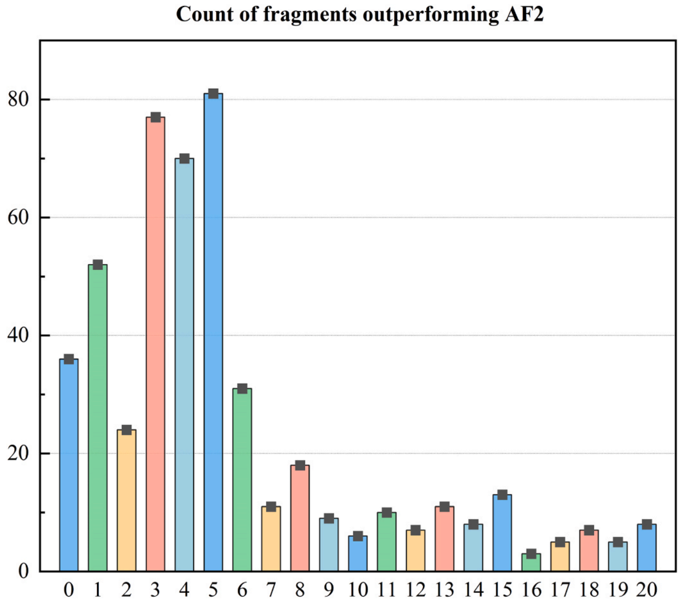

Fig. 7. Count of Fragments Outperforming AF2. The horizontal axis represents the number of segments in which the performance of our study surpasses AF2, while the vertical axis denotes the statistical count of sequences.

method is partially better than AF2 in several sequences within segments, which is attributed to the introduction of multifactorial features as well as the unique data processing and model design scheme, and it also demonstrates the value of this study in the field of protein secondary structure prediction.

Additionally, case studies were conducted, and two sequences were selected for demonstration purposes. As depicted in Fig. 8, it is evident that in certain segments, our predictive outcomes are more favorable compared to AF2.

3.4. Explore the optimal architecture of our model

To explore the optimal model configuration further, we conducted a series of hyperparametric experiments. To maintain consistency in the experimental setup, we fixed the learning rate at 0.0001, set the batch size to eight, and utilised the Adam optimiser. As illustrated in Fig. 9, different parameter settings for various components had a significant impact on model performance. This figure presents the optimal model parameter combinations, emphasising their crucial roles in performance optimisation. During the experimental process, we first conducted hyperparameter experiments with the number of layers in the dilated convolution, setting it to one, two, three, and four layers, and compared these variations to determine the optimal layer count. The results indicated that both smaller and larger numbers of layers resulted in decreased evaluation metrics, thereby affecting optimal model performance. Subsequently, we performed experiments on the number of SENet layers, also set to 1, 2, 3, and 4. The analysis showed that the effect of the SENet layers on the model was similar to that of the dilated convolution. The best model performance was achieved with three layers, and the number of layers increased, resulting in a lower model performance. We then explored the number of hidden units in the RNN variants of recurrent neural networks as hyperparameters, which were set to 32, 64, 128, and 256. The results indicated that as the number of hidden units gradually increased in powers of two, the model's performance improved until it reached 256, after which the performance began to gradually decline, validating the conclusion that the optimal

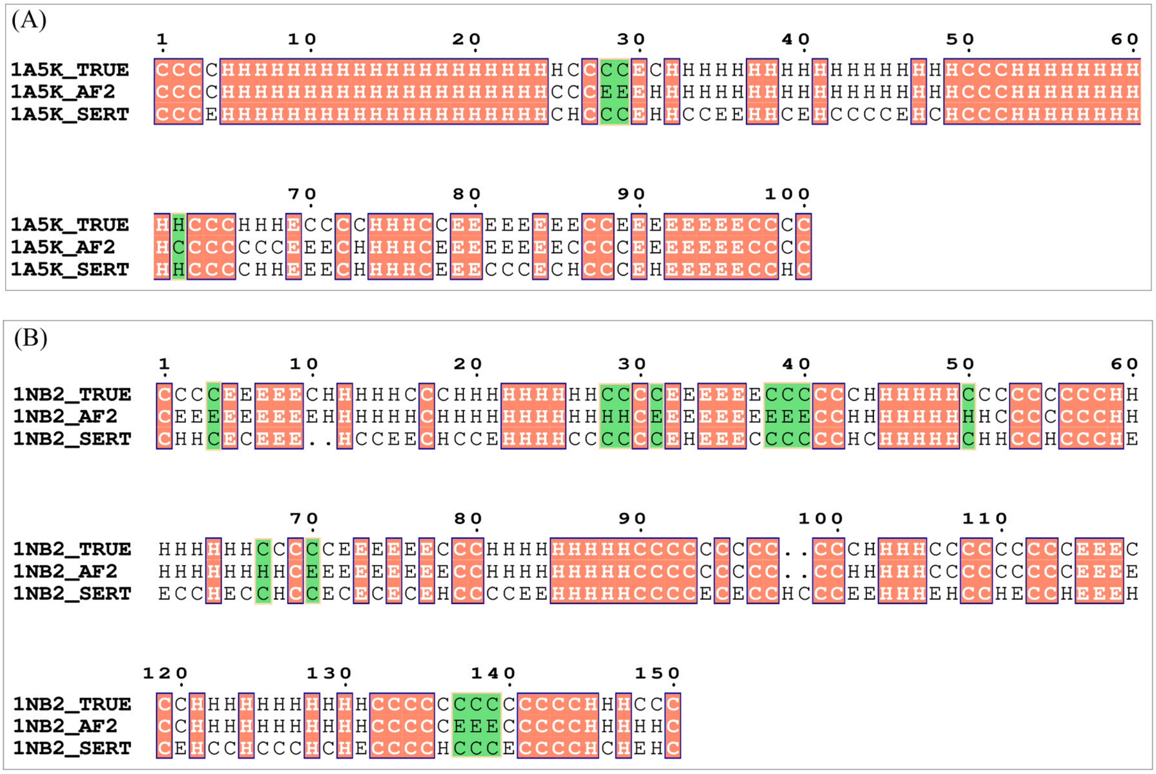
(A)
1
10
20
30
40
50
60
1A5K_TRUE CCCCHHHHHHHHHHHHHHHHHHHHHCCCCECHHHHHHHHHHHHHHHHHHCCCHHHHHHHH
1A5K_AF2 CCCCHHHHHHHHHHHHHHHHHHHHCCCEEEHHHHHHHHHHHHHHHHHHHCCCHHHHHHHH
1A5K_SERT CCCEHHHHHHHHHHHHHHHHHHHHCHCCCEHHCCEEHHCEHCCCCEHCHCCCHHHHHHHH
70
80
90
100
1A5K_TRUE HHCCCHHHECCCCHHHCCEEEEEEEECCEEEEEEEECCCC
1A5K_AF2 HCCCCCCCEEECHHHHCEEEEEEEECCCCEEEEEEECCCC
1A5K_SERT HHCCCCHHEEECHHHHCEEECCCECHCCCEHEEEEECCHC
(B)
1
10
20
30
40
50
60
1NB2_TRUE CCCCEEEEECHHHHHCCCHHHHHHHHHCCCCEEEEEECCCCCCHHHHHHCCCCCCCCCHH
1NB2_AF2 CEEEEEEEEEHHHHHHCHHHHHHHHHHCEEEEEEEEEECCHHHHHHHHHCCCCCCCHH
1NB2_SERT CHHCECEEE..HCCEECHCCEHHHHCCCCCCEHEEECCCCCCHCHHHHHCHHCCHCCCHE
70
80
90
100
110
1NB2_TRUE HHHHHHCCCCCEEEEEECCCHHHHHHHHHCCCCCCCCC..CCCHHHHCCCCCCCCCEEEC
1NB2_AF2 HHHHHHHHCEEEEEEEEECCHHHHHHHHHCCCCCCCCC..CCHHHHHCCCCCCCCCEEEE
1NB2_SERT ECCHECCHCCECECECEHCCCCEEHHHHHCCCCECECCHCCCEEHHHEHCCHECCHEEEH
120
130
140
150
1NB2_TRUE CCHHHHHHHHHHHCCCCCCCCCCCCCHHHCCC
1NB2_AF2 CCHHHHHHHHHHHCCCCCEEECCCCCHHHHHC
1NB2_SERT CEHCCHCCCHCHECCCCHCCCECCCCHCHEHC

Fig. 8. Two example analyses from the case data. Red-marked regions indicate fragments that were successfully predicted by both this study and AlphaFold2 (AF2), and green-marked regions indicate fragments that were successfully predicted by this study alone. (A) In the sequence of protein 1A5K, both this study's method and AF2 predicted a high percentage of successfully predicted fragments and successfully predicted 5 fragments that were not predicted successfully by AF2; (B) In the sequence of protein 1NB2, this study's method successfully predicted 13 fragments that were not predicted successfully by AF2, even though it successfully predicted a smaller percentage of fragments than AF2.

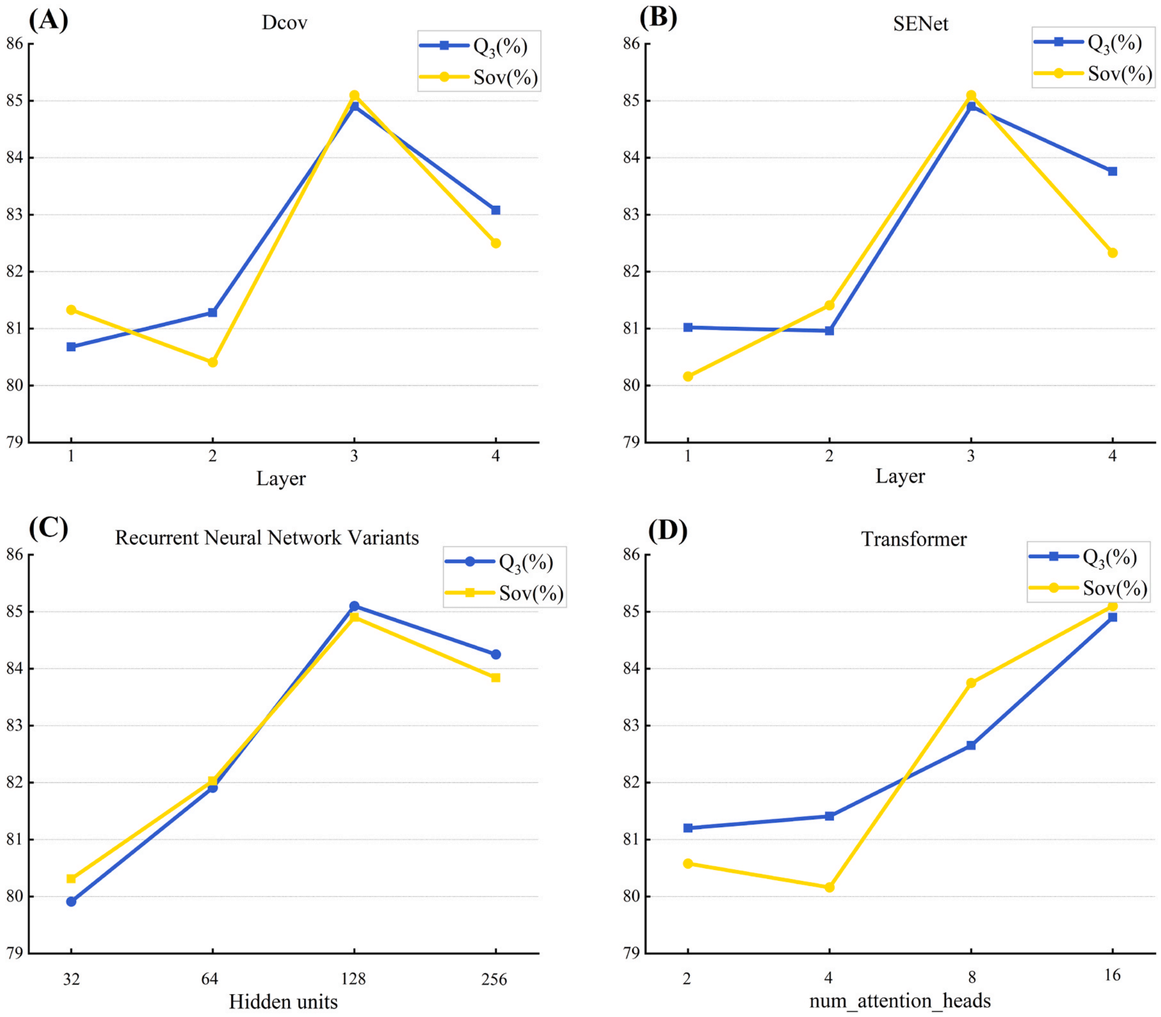

Fig. 9. Explore the optimal hyperparameter settings for each model experimental results are presented visually. (A) Hyperparametric experimental performance visualisation of D-Conv; (B) Hyperparametric experimental performance visualisation of SENet; (C) Hyperparametric experimental performance visualisation of recurrent neural network variant; (D) Hyperparametric experimental performance visualisation of Transformer.

model performance was at 128 hidden units. Finally, we conducted experiments on the number of heads in the transformer model's multi-head attention as a hyperparameter, setting it to 2, 4, 8, and 16. The experimental results revealed that, as the number of attention heads increased, the performance of the model gradually improved. However, considering the task characteristics and the overall model architecture, we selected 16 heads as the best setting after comprehensive consideration.

Finally, we found that the optimal parameters for each component typically fell within a moderate range, rather than blindly pursuing an increase in parameter values to enhance model performance. This phenomenon can be attributed to the nature of our task, which is protein secondary structure prediction involving the processing of protein sequences. As protein sequences are not overly complex in composition, blindly increasing the parameters may lead to difficulties in model training, resulting in overfitting issues. This increases the computational burden on the model and consumes excessive computational resources.

3.5. Ablation experiment

3.5.1. Data ablation

To validate the effect of various input data on protein secondary structure prediction using the SERT-StructNet model, we conducted multiple comparisons and evaluations of different data combinations using the same test set. Throughout the experiments, we maintained the model parameters and training settings while changing the input data features, including One-hot encoding, position-specific scoring matrix, and property features (physicochemical properties of amino acids and secondary structure propensity scores (SSPs)). The results in Table 5 show that the best performance was achieved by employing a combination of multiple elements, especially with the inclusion of selectively curated amino acid properties and SSPs, where Q_3 accuracy reached 84.9% and the Sov score reached 85.1%. These outcomes underscore the varying significance of different types of input data in predicting protein secondary structures. Each position in the PSSM represents the evolutionary information of the amino acids. This means that the PSSM can better reflect the relationships and weights between amino acid sequences, providing robust information for prediction. One-hot encoding is a representation method for each amino acid position in a sequence

Table 5
Experimental results for different combinations of input data.

Data	Network	Q_3 (%)	Sov (%)
One-hot	SERT-StructNet	77.51	78.58
One-hot + Properties		78.16	79.41
PSSM		80.41	82.25
PSSM+ Properties		81.54	83.58
One-hot + PSSM		83.13	84.26
One-hot + PSSM + Properties		**84.9**	**85.1**

using 0 s and 1 s. Despite its simplicity, it remains an indispensable representation method in protein secondary structure prediction tasks, enabling the model to identify amino acid positions. Property characterisation included the characterisation of amino acid properties and SSPs. The selection of amino acid properties is based on the variability in the effect of each property on different secondary structures, which is a comprehensive consideration of the relevant factors to ensure the validity of the amino acid properties. SSPs, on the other hand, indicate the proportions of different secondary structures in the data, which are also crucial for structure prediction and understanding. Overall, each type of input data has unique advantages and information content, contributing differently to model performance. This suggests that different types of input data can complement each other. These results emphasise the importance of considering multiple sources of information in protein secondary structure prediction tasks, which helps to capture a protein's secondary structure information comprehensively, thereby enhancing the model's performance and understanding.

3.5.2. *Model ablation*

In this subsection, we perform model ablation experiments by removing the key components of the model individually and performing a comprehensive comparison and evaluation of the models while keeping the input data and training parameters constant. Table 6 presents the prediction results of the SERT-StructNet model with different structures in the dataset. First, we compare the performance of the optimal model obtained in 3.2 with the case of removing SENet, and the results show that the Q_3 accuracy and Sov scores are improved by 1.42% and 1.02%, respectively. This shows that the introduction of SENet enhances the performance of the model by enhancing access to the expanded convolutional features, thereby allowing the next module to better understand and analyse the data. Next, we compared the performance of the optimal model with the case of removing the transformer and found that the Q_3 accuracy and Sov scores were improved by 2.29% and 2.69%, respectively. This is because the transformer processes the data with a multi-head attention mechanism, which allows it to focus on different subspaces in parallel, thus helping to process more features of the data. Consequently, the ability of the transformer to process data in this study exceeded that of SENet and contributed more to the prediction task. Finally, we focused on ablation experiments with recurrent network variants. We explored their impact in this study by controlling for the GRU and LSTM. We compared the optimal model with the model after removing the GRU, and the results showed that the Q_3 accuracy and Sov score improved by 4.13% and 5.49%, respectively, and compared the optimal model with the model after removing the LSTM, and the results showed that the Q_3 accuracy and Sov score improved by 3.04% and 2.39%, respectively. This significant improvement demonstrates the centrality of these two recurrent neural network variants in SERT-StructNet. This is because BiGRU and BiLSTM play crucial roles in data processing in the model. By comparing the impact of BiGRU and BiLSTM, we found that owing to the small percentage of long sequences in the dataset used, the properties of BiGRU are more fully exploited, that is, the importance of BiGRU is greater than that of BiLSTM. However, it is worth noting that both recurrent network variants were indispensable. Despite the small percentage of long sequences, they are not exclusive to long sequences; thus, BiLSTM and BiGRU can complement each other in the consideration of different data from different perspectives, leading to more comprehensive and effective feature extraction of the data.

Table 6
Experimental results for different model architectures.

Network	Data	Q_3 (%)	Sov (%)
RT-StructNet	One-hot	83.48	84.08
SET-StructNet	PSSM	80.15	79.43
SET-StructNet(no BiGRU)	Properties	80.77	79.61
SET-StructNet(no BiLSTM)		81.86	82.71
SER-StructNet		82.61	82.41
SERT-StructNet		**84.9**	**85.1**

4. Conclusion

In this study, we propose a protein secondary structure prediction model called SERT-StructNet. This model employs a hybrid deep feature extraction method that combines local extraction (dilated convolution and channel attention) with global extraction (variants of recurrent neural networks and transformers). In addition, we used multiple factors as inputs. This multi-level feature extraction method effectively exploits various depths of information between protein sequences, more accurately reflecting the complex mapping between sequences and structures, thereby enhancing the model's performance. Through a series of experiments, our predictive model outperformed existing methods in terms of evaluation metrics such as Q_3 accuracy and Sov. After analysing the 8-state category of protein secondary structures with the same precision and evaluation metrics as Sov, we further explored each single structure type in depth to deepen our understanding of secondary structures. Furthermore, our experimental results confirm the contribution of each input feature and different feature combinations to the model predictions. The incorporation of multi-factor features further enhanced the feature representation of the original data, particularly the introduction of carefully selected amino acid properties and secondary structure propensity scores, effectively enhancing the model's understanding of the input features. Moreover, we conducted a detailed analysis of the contributions of various components to the model, revealing its extensive potential for adjustments and improvements. Future research will focus on exploring additional deep-learning algorithms, feature extraction, and optimisation techniques to delve deeper into protein secondary structure prediction using comprehensive data. We also plan to broaden the research scope and delve into the tasks and learning related to protein secondary structure prediction to advance the development of this field. This field has extensive application prospects and we anticipate continuous research to enhance its performance and expand its application range.

WebServer and data availability

We establish a webserver to implement the proposed method, which is currently accessible via https://bioinfor.nefu.edu.cn/SERT-StructNet/. Moreover, the source code and dataset of SERT-StructNet have been uploaded to https://github.com/LindaEdu/SERT-StructNet/.

CRediT authorship contribution statement

Guohua Wang: Funding acquisition, Resources, Supervision, Writing – review & editing. **Benzhi Dong:** Conceptualization, Methodology, Writing – original draft. **Zheng Liu:** Investigation, Validation, Writing – original draft. **Dali Xu:** Data curation, Investigation, Visualization. **Chang Hou:** Data curation, Validation. **Guanghui Dong:** Data curation, Visualization. **Tianjiao Zhang:** Formal analysis, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Key R&D Program of China [2021YFC2100101], National Natural Science Foundation of China [62272095, 62172087, 62072095] and Key R&D Program of

Heilongjiang Province [2022ZX01A29].

References

[1] Bepler T, Berger B. Learning the protein language: evolution, structure, and function. Cell Syst 2021;12:654–69. e653.

[2] Detlefsen NS, Hauberg S, Boomsma W. Learning meaningful representations of protein sequences. Nat Commun 2022;13.

[3] Monzon AM, Fornasari MS, Zea DJ, Parisi G. Exploring protein conformational diversity. Comput Methods Protein Evol 2019:353–65.

[4] Hassan M, Coutsias EA. Protein secondary structure motifs: a kinematic construction. J Comput Chem 2020;42:271–92.

[5] Saghapour E, Sehhati M. Physicochemical position-dependent properties in the protein secondary structures. Iran Biomed J 2019;23:253–61.

[6] Koehler Leman J, Künze G. Recent advances in NMR protein structure prediction with ROSETTA. Int J Mol Sci 2023;24.

[7] Rost B, Sander C, Schneider R. PHD–an automatic mail server for protein secondary structure prediction. Comput Appl Biosci: CABIOS 1994;10:53–60.

[8] Aydin Z, Altunbasak Y, Borodovsky M. Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. BMC Bioinforma 2006;7.

[9] Chen C, Tian Y, Zou X, Cai P, Mo J. Prediction of protein secondary structure content using support vector machine. Talanta 2007;71:2069–73.

[10] Sun XD, Huang RB. Prediction of protein structural classes using support vector machines. Amino Acids 2006;30:469–75.

[11] Peng Z, Wang W, Han R, Zhang F, Yang J. Protein structure prediction in the deep learning era. Curr Opin Struct Biol 2022;77.

[12] Wang S, Peng J, Ma J, Xu J. Protein secondary structure prediction using deep convolutional neural fields. Sci Rep 2016;6:18962.

[13] Wang S, Li W, Liu S, Xu J. RaptorX-property: a web server for protein structure property prediction. Nucleic Acids Res 2016;44:W430–5.

[14] Drozdetskiy A, Cole C, Procter J, Barton GJ. JPred4: a protein secondary structure prediction server. Nucleic Acids Res 2015;43:W389–94.

[15] Torrisi M, Kaleel M, Pollastri G. Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. Sci Rep 2019;9.

[16] S G, R VE. Protein secondary structure prediction using cascaded feature learning model. Appl Soft Comput 2023;140.

[17] Błazewicz J, Hammer PL, Lukasiak P. Predicting secondary structures of proteins. Recognizing properties of amino acids with the logical analysis of data algorithm. IEEE Eng Med Biol Mag: Q Mag Eng Med Biol Soc 2005;24:88–94.

[18] Li Z, Wang J, Zhang S, Zhang Q, Wu W. A new hybrid coding for protein secondary structure prediction based on primary structure similarity. Gene 2017;618:8–13.

[19] Uzma U, Manzoor Z, Halim. Protein encoder: an autoencoder-based ensemble feature selection scheme to predict protein secondary structure. Expert Syst Appl 2023:213.

[20] de Brevern AG, Chen T-R, Juan S-H, Huang Y-W, Lin Y-C, Lo W-C. A secondary structure-based position-specific scoring matrix applied to the improvement in protein secondary structure prediction. Plos One 2021;16.

[21] Raicar G, Saini H, Dehzangi A, Lal S, Sharma A. Improving protein fold recognition and structural class prediction accuracies using physicochemical properties of amino acids. J Theor Biol 2016;402:117–28.

[22] Dwyer DS. Electronic properties of the amino acid side chains contribute to the structural preferences in protein folding. J Biomol Struct Dyn 2001;18:881–92.

[23] Park S, Chang DE. Multipath lightweight deep network using randomly selected dilated convolution. Sensors 2021;21.

[24] Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell 2020;42:2011–23.

[25] Krishnamurthy K, Can T, Schwab DJ. Theory of gating in recurrent neural networks. Phys Rev X 2022;12.

[26] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80.

[27] Li Z, Zhang Z, Zhao H, Wang R, Chen K, Utiyama M, Sumita E. Text compression-aided transformer encoding. IEEE Trans Pattern Anal Mach Intell 2021:1. -1.

[28] Magnan CN, Baldi P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics 2014;30:2592–7.

[29] Cuff JA, Barton GJ. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Protein: Struct, Funct, Genet 1999;34:508–19.

[30] Hu G, Kurgan L. Sequence similarity searching. Curr Protoc Protein Sci 2018;95.

[31] Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M. AAindex: amino acid index database, progress report 2008. Nucleic Acids Res 2007;36:D202–5.

[32] Yuan L, Ma Y, Liu Y. Protein secondary structure prediction based on Wasserstein generative adversarial networks and temporal convolutional networks with convolutional block attention modules. Math Biosci Eng 2023;20:2203–18.

Benzhi Dong is an associate professor at the College of Computer and Control Engineering, Northeast Forestry University. His research interests include bioinformatics and computer vision.

Zheng Liu is a master candidate at the College of Computer and Control Engineering, Northeast Forestry University. His research interests include bioinformatics.

Dali Xu is a lecturer at the College of Computer and Control Engineering, Northeast Forestry University. Her research interests include bioinformatics and deep learning.

Chang Hou is a lecturer at the College of Computer and Control Engineering, Northeast Forestry University. Her research interests include bioinformatics.

Guanghui Dong is an associate professor at the College of Computer and Control Engineering, Northeast Forestry University. Her research interests include bioinformatics and pattern recognition.

Tianjiao Zhang is an associate professor at the College of Computer and Control Engineering, Northeast Forestry University. His research interests include bioinformatics.

Guohua Wang is a professor at the College of Computer and Control Engineering, Northeast Forestry University. He is also a Principal Investigator at the Key Laboratory of Tree Genetics and Breeding, at Northeast Forestry University. His research interests are bioinformatics and machine learning.