Current Medicinal Chemistry, 2022, *29*, 881-893

REVIEW ARTICLE

Better Performance with Transformer: CPPFormer in the Precise Prediction of Cell-penetrating Peptides

Yuyang Xue[1], Xiucai Ye[1,*], Lesong Wei[1], Xin Zhang[2], Tetsuya Sakurai[1] and Leyi Wei[2,*]

[1]*Department of Computer Science, University of Tsukuba, Tsukuba, Japan;* [2]*School of Software, Shandong University, Jinan, China*

ARTICLE HISTORY

Received: March 11, 2021
Revised: July 28, 2021
Accepted: August 07, 2021

DOI:
10.2174/0929867328666210920103140

Abstract: Owing to its superior performance, the Transformer model, based on the 'Encoder-Decoder' paradigm, has become the mainstream model in natural language processing. However, bioinformatics has embraced machine learning and has led to remarkable progress in drug design and protein property prediction. Cell-penetrating peptides (CPPs) are a type of permeable protein that is a convenient 'postman' in drug penetration tasks. However, only a few CPPs have been discovered, limiting their practical applications in drug permeability. CPPs have led to a new approach that enables the uptake of only macromolecules into cells (*i.e.*, without other potentially harmful materials found in the drug).

Most previous studies have utilized trivial machine learning techniques and hand-crafted features to construct a simple classifier. CPPFormer was constructed by implementing the attention structure of the Transformer, rebuilding the network based on the characteristics of CPPs according to their short length, and using an automatic feature extractor with a few manually engineered features to co-direct the predicted results. Compared to all previous methods and other classic text classification models, the empirical results show that our proposed deep model-based method achieves the best performance, with an accuracy of 92.16% in the CPP924 dataset, and passes various index tests.

Keywords: Cell-penetrating peptides, deep learning, drug penetration, transformer, feature extractor, classification.

1. INTRODUCTION

1.1. Aims and Goals

The design of drugs has rapidly changed in recent decades. Drug designers have transitioned from traditional hand-crafted engineering to computer-aided design (CAD), thereby adopting model-based drug prediction with new artificial intelligence tools [1, 2]. Despite peerless progress in drug design, drug absorption remains a major issue for cells [3]. Owing to poor cell penetration, the absorption capacity of most existing drugs is markedly compromised. Peptide drugs have gradually become a competitor to replace protein macromolecular drugs and have gradually entered the main stage of drug design due to their short peptide chain and strong cell affinity. Cell-penetrating peptides (CPPs) [4, 5], also known as cell-permeable proteins [6], are peptides consisting of only 5 to 50 amino acids that have become the centerpiece of the drug industry as they are harmless to cell membranes and can act as drugs, proteins, and nanoparticle transporters.

Moreover, peptide-based treatments are beneficial for drug manufacturers, including their low cost and high efficiency in synthesis [7]. However, verifying CPP through manual experimental methods is extremely expensive, time-consuming, and laborious [8]. Accordingly, the ability to identify the correct CPPs from a large number of peptide sequences is of critical importance. From the interest of pharmaceutical manufacturers to the affirmation of the medical community to jointly drive the discovery and identification of CPPs, more efficient automatic identification methods have been developed.

*Address correspondence to these authors at the Department of Computer Science, University of Tsukuba, Tsukuba, Japan; E-mail: yexiucai@cs.tsukuba.ac.jp; School of Software, Shandong University, Jinan, China; E-mail: weileyi@sdu.edu.cn

1875-533X/22 $65.00+.00

Here, we present a variation of Transformer, CPPFormer. First, we selected a high-quality dataset, CPP924, as our main training and validation sample. A low-redundancy dataset enables easier data cleaning and thus is suitable for training. Thereafter, we analyzed the distribution of amino acids in the dataset comprising both positive and negative samples and constructed the CPPFormer model by tweaking the Transformer model to fit the characteristics of CPP sequences. Notably, the attention mechanism in Transformer enhances the token importance in the FASTA sequences. Finally, we conducted several experiments, including a comparison of both manual feature extractor and automatic feature, and comparison of SOTA models and NLP models, and found that the CPPFormer obtained the best result for the testing dataset with an accuracy of 92.16%, ultimately highlighting the remarkable progress made in Transformer development.

Our study highlights include:

1. The proposal, a Transformer variant, CPPFormer, that is suitable for CPP classification. Briefly, we modified and fine-tuned the original model according to the specific situation and ultimately achieved the state-of-the-art prediction effect.
2. A comparison of the proposed model with multiple language models was made. We concluded that CPPFormer can achieve better results than the RNN and LSTM series models in sequence analysis and classification.
3. A comparison of manual peptide feature extraction methods revealed that the features automatically extracted by the CPPFormer's attention module can better find hidden information in the peptide sequence.

1.2. Related Works

With the introduction of machine learning [9], bioinformatics has kept up with the pace of the era and has led to a new round of innovation and improvement *via* the data-driven model. The most important part of building a robust machine learning model is selecting suitable datasets. Since the discovery of the first CPP, the Tat peptide, in the 1980s [10], scientists have identified hundreds of CPPs. Some mature datasets and benchmarks have also been developed and achieved. In the early stage of research, a small CPP benchmark was proposed by Sanders *et al.* [11], which contained only 111 positive samples. The first CPP-specific database, CPPSite [12], has 843 manually collected CPPs validated *via* experiments. In 2015, CPPSite 2.0 [13], an updated version of the CPPSite database, comprised nearly double the number of its predecessor, with nearly 1700 unique CPPs, including their secondary and tertiary structures. However, the imbalance of positive and negative samples in the dataset and high redundancy have a serious impact on the machine learning performance. To solve this problem, CPP924 [14] was developed to balance the data and reduce sequence similarity, thereby enabling any two positive sequences to differ by more than 80%. KELM-CPPpred [15] also comprised a smaller volume containing 826 sequences with an equal number of CPPs and non-CPPs. The emergence of effective datasets has markedly accelerated the development of machine learning models and has enabled researchers with less biological backgrounds to bridge the gap in professional knowledge to further participate in the discovery and identification of CPP.

The objectives of machine learning are to extract useful or meaningful features (both for human and computational models) [16, 17], learn from the distribution of data, and make a prediction on the unseen data. Most researchers believe that the peptide structure is clustered by an intrinsic feature, such as sequence combination of spatial structure, and machine learning is good at recognizing its pattern. The prediction of CPP is a classification task [18] for determining permeable peptide cells. Regarding the construction of machine learning classifiers, dating back to 2008, Hansen *et al.* [19] developed a z-scale physiochemical descriptor on 87 non-redundant CPPs, which served as a start-up and foundation for later works. The first artificial neural network (ANN) combined with the prediction of CPP, which was proposed by Dobchev *et al.* [20], engineered the feature using biochemical properties, together with principal component analysis (PCA) dimension reduction techniques. By constructing a support vector machine (SVM) [21, 22], Sanders *et al.* [11] achieved an overall accuracy of 75% using a more objective dataset. Gautam *et al.* [12] further studied the capability of SVM, combined with more manual features, such as dipeptide combination frequency (DPF), and highlighted the CPPsite dataset. Holton *et al.* [23] conducted a trial on ANNs and proposed a network model, CPPpred. Random forest is a good choice for ensembling multiple decision trees, as implemented by Chen *et al.* [24] and Qiang *et al.* [25]. Furthermore, SkipCPP-pred [14] was constructed as a public web server to exploit CPP sequences online using an adaptive k-skip-n-gram feature extractor with a fine-tuned random forest. Arif *et al.* [26] implemented a gradient boost decision tree using optimized multiscale features to construct TargetCPP. Su *et al.* [27] reviewed and

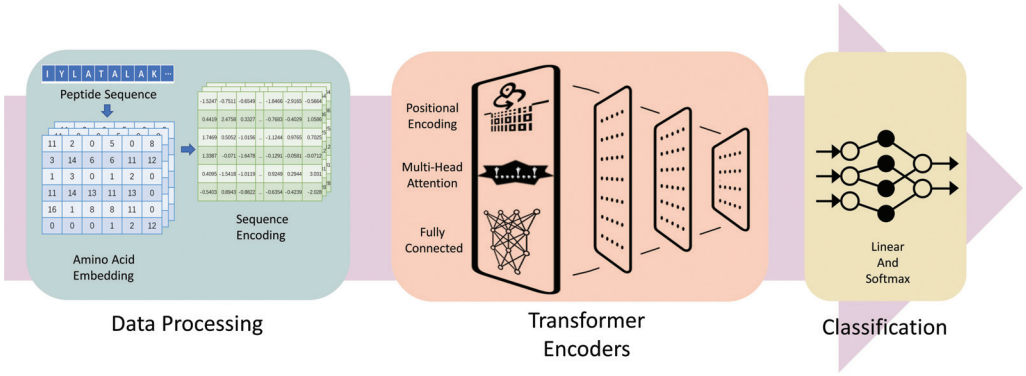

Fig. (1). The workflow of CPPFormer framework. *(A higher resolution/colour version of this figure is available in the electronic copy of the article).*

assembled multiple predictors on a web server and provided a statistical comparison with previous work. However, most of the previous works were mainly based on manually curated features such as simple amino acid composition (AAC) [28], binary pattern profile, and pseudo amino acid composition (PAAC) [29], which may not be able to explore the characteristic structure inside the peptide chain.

With the increasing development of natural language processing (NLP), many efforts have been made to construct advanced models and complex feature extractors to teach computers the meaning and grammar of human languages, especially using deep and wide neural networks [30]. As a result, CPP sequence prediction can be treated as a text classification problem, similar to sentiment analysis [31] and toxic comment classification [32]. Since 2017, the Transformer model [33, 34] has subverted the traditional neural network architecture and compensated for the shortcomings of convolutional neural networks (CNN) [35, 36], recurrent neural networks (RNN) [37], and the long-short term memory model [38, 39]. This model outperforms most of its rivals in natural language processing, such as neural machine translation [40], speech recognition [41], and generative modeling [42]. Even in bioinformatics, Transformers have become prominent, enabling the performance of multiple tasks by achieving state-of-the-art (SOTA) performance [43-47]. Transformers are good for semantic feature extraction, long-distance feature capture, and comprehensive feature extraction. The core module of the Transformer architecture is the self-attention module [33]. When the language model examines each element in the sequence, the self-attention module calculates the similarity scores of all position pairs to help encode the clues of the word well. However, as the length of the input sequence increases, the model requires a quadratic calculation time to generate all similarity scores. Furthermore, an increase in the required calculation memory is also observed. The efficiency problem faced by the attention mechanism has become increasingly prominent. Accordingly, for applications that require long-distance attention, some researchers have proposed several fast and space-efficient improvement methods based on Transformer [48-50]; however, most of the common methods rely on the sparse attention mechanism [51]. Although minor problems may be found, Transformer is competent in most NLP tasks, even in other related fields.

2. MATERIALS AND METHODS

2.1. Machine Learning Workflow

Here, we present the framework of our training and prediction workflow, which is illustrated in Fig. (**1**), providing a concise summary of this procedure. For the first part of data processing, the peptide data were decrypted from the encapsulated FASTA format sequence type, with each peptide sequence possessing a label that indicates whether it is a positive or negative sample. After trivial data processing, we formatted the peptide sequence. To allow the input sequence to be recognized by the computer, we embedded 20 amino acids and then encoded the entire sequence into a matrix that can be easily recognized by the neural network. The encoding matrix was then sent to the transformer encoders for further feature extraction. The encoder is a multiple-encoder set, which can easily tune the hyperparameter of the number of encoder combinations. Each encoder has a positional encoding module, a multi-head attention module, and a fully connected layer. Finally, the calculated features were sent to the last linear predictor, and Softmax was used to determine the probability of predictions. More details for each module are provided below.

2.2. Data Preparation

We prepared data from the famous benchmark CPP924. Briefly, 462 sequences were true CPPs veri-

fied by experiments, while the remaining sequences were non-CPPs. CPP924 was selected as our main training dataset because 1. it has a good balance between positive and negative samples. An imbalance between positive and negative samples leads to deviations in machine learning tasks and will markedly affect the accuracy of predictions. 2. although the amount of data is less than that in some other databases, the de-redundancy of CPP924 makes the data distinct and purer. Any two sequences in the positive group differed by more than 80%. Furthermore, we carefully checked the dataset and found some unrecognized characters and some artificial amino acids, besides the usual 20 types. Therefore, we further organized the data entries, which comprised a total of 915 available samples.

We processed our peptide sequences in a natural language manner as there were 20 known non-synthetic amino acids. We can construct a glossary and add <*UNK*> and <*PAD*>, which is a total of 22 words. <*UNK*> represents unknown characters, such as artificial amino acids or some of the wrong characters and numbers. <*PAD*> represents the padding bits and the placeholder. Because the length of each peptide sequence is different, the bits must be filled with a fixed length to facilitate batch processing by the network.

We tokenized the amino acid and then constructed the embedding. The length of the longest sequence in CPP924 was found to be 61; thus, all sequences were padded to 64 (which is an integer multiple of 2). Thereafter, the word vectors were built for the training, validation, and test datasets, which were randomly split in an 8: 1: 1 ratio. The constructed word vectors could then be fed into the embedding layers of the network model.

2.3. Manual Feature Engineering Techniques

Here, we introduce some traditional bioinformatics peptide (and protein) feature-encoding algorithms. Before the automatic feature extraction network was widely used, the feature extraction techniques proposed by biological researchers markedly accelerated past machine learning models. On the one hand, manual features have strengthened the interpretability of machine learning models. On the other hand, these features can help new researchers to better understand the characteristics learned by machine learning models.

2.3.1. Amino Acid Composition (AAC)

The AAC might be the simplest feature extracted from the peptide (protein) sequence. The concept is to represent the frequency of each amino acid occurring in the sequence, which can be calculated using the following formula:

$$aac(i) = \frac{n_i}{N} \quad (1)$$

where i denotes the 20 amino acid residues, n_i is the frequency of each residue, and i and N are the total number of residues in the sequence. The returned result is a 20-dimensional feature vector for each amino acid frequency.

2.3.2. Pseudo Amino Acid Composition (PAAC)

Similar to AAC, this feature is primarily used to characterize the amino acid frequency matrix, which contributes to processing without salient homology to a peptide sequence with other peptides [52, 53]. However, additional information is included in the matrix to represent some local features, such as the correlation between residues at a certain distance. For each amino acid, hydrophobicity, hydrophilicity, and side-chain mass values may have an impact on the properties of the peptides:

$$P(i) = \frac{P^o(i) - \frac{1}{20}\sum_{i=1}^{20} P^o(i)}{\sqrt{\frac{\sum_{i=1}^{20}\left[P^o(i) - \frac{1}{20}\sum_{i=1}^{20} P^o(i)\right]^2}{20}}} \quad (2)$$

where $\mathrm{P}(i)$ denotes the converted property (hydrophobicity, hydrophilicity, or side chain mass) of the amino acid i, and $P^o(i)$ denotes the original value of these properties.

A correlation function that averages the values of the three physicochemical properties is defined as:

$$\Theta(J_k, J_l) = \frac{1}{3}[H_1(J_k) - H_1(J_l)]^2 + [H_2(J_k) - H_2(J_l)]^2 + [M(J_k) - M(J_l)]^2 \quad (3)$$

where H_1, H_2, and M refer to hydrophilicity, hydrophilicity, and side-chain mass, respectively, and J_k and J_l denote the amino acids at positions k and l.

Subsequently, a set of sequence order-correlated factors is defined as follows:

$$\theta_\lambda = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} \Theta(J_k, J_{k+\lambda}) \quad (4)$$

where λ is an integer parameter to be selected, which must be smaller than the sequence length N.

With f_i as the normalized occurrence frequency of amino acid i in the sequence, and a set of $20 + \lambda$ descriptors, called the pseudo amino acid, the composition can be defined as:

$$X_c = \begin{cases} \frac{f_c}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j} & 1<c<20 \\ \frac{w\theta_{c-20}}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j} & 21<c<20+\lambda \end{cases} \quad (5)$$

where w is the weighting factor for the sequence-order effect and is set to 0.05, as suggested by Chou *et al.* [28].

2.3.3. N-gram Composition

N-gram [54] is an algorithm based on statistical language models. The basic concept of this algorithm is to perform a sliding window operation of size N on the content of the text according to bytes, forming a sequence of byte fragments of length N. This function computes the di- or tri-peptide composition of amino acid sequences. Therefore, the function parameter n can only take arguments 2 and 3. In this study, we used dipeptides in our experiments [9, 55].

$$di(i,j) = \frac{N_{ij}}{N-1}, i,j = 1,2,\dots,20 \quad (6)$$

where N_{ij} denotes the dipeptides comprising amino acids ij, respectively. N denotes the total sequence length.

Some variations of the N-gram, such as the k-skip-n-gram, used by Wei *et al.* [14], show great potential at extracting intrinsic features from sequences. Only the original version of the N-gram was employed herein to test its ability.

2.3.4. BLOSUM62

BLOSUM (BLOcks SUbsititution Matrix), first proposed by Henikoff *et al.* [56], is a substitution matrix used in bioinformatics to compare alignments between two protein sequences. The BLOSUM62 matrix was developed to analyze amino acid frequencies in clusters of related peptides. For the substitution of amino acid i for amino acid j, the score is expressed as:

$$S_{ij} = \frac{1}{\lambda} log \frac{p_{ij}}{q_i q_j} \quad (7)$$

where p_{ij} is the frequency of the substitution in homologous proteins, and q_i and q_j are the frequencies of amino acid i and amino acid j. $\frac{1}{\lambda}$ is the scaling factor for obtaining the matrix in an integer form.

2.4. CPPFormer Model

In this section, we introduce our CPP classification prediction model based on Transformer. First, we reveal the advantages of the attention mechanism, then present the internal structure of the Transformer, and finally share details of our model.

2.4.1 Attention Mechanism

The development of deep learning [39] in natural language processing is comprised of RNN and LSTM. However, RNN and LSTM play an important role in feature extraction and semantic understanding even in some applications, but the accompanying problems are still pending. One of the most serious shortcomings of RNNs is their long-term dependence. Although LSTM has been introduced to alleviate this problem, the LSTM in the decoding stage cannot be well targeted at the earliest input sequence when the sequence is very long. Bahdanau *et al.* [57] first implemented this method in machine translation. In the process of translating texts, we need to refer to the context in which each word is appropriately translated. Similarly, in the attention model, when a human translates the current word, he or she will look for the corresponding words in the source sentence and combine the previously translated parts to do the corresponding translation. The hidden states of the RNN encoder are: $(h_1, h_2, \dots, h_t)$. Asumming that the hidden state of the current decoder hidden state is s_{t-1}, each input position j and the current output position should be computed as:

$$\overrightarrow{c_t} = \left(a(s_{t-1}, h_1),\ \cdots, a(s_{t-1}, h_T)\right) \quad (8)$$

where a is dot product or any other correlation. The score for each attention can be represented by the following formula:

$$a_{tj} = \frac{exp(c_{tj})}{\sum_{k=1}^{T} exp(c_{tk})} \quad (9)$$

The model uses attention as the weight and gives different attention to each element. A higher weight represents a higher degree of attention; the element can play a decisive role in this sequence. The attention mechanism can provide the encoder with more information than the previous feature input with equal weight, which enables its focus on high-attention elements and enhances the model's interpretive ability. Accordingly, a natural question arises: as the attention mechanism is remarkably effective, can we remove the RNN part of the model and simply use attention?

2.4.2. Transformer with Multi-Head Attention

Transformer no longer uses recurrent neural network layers like other language models; however, the attention module and feedforward layer are required. The structure of the Transformer encoder is displayed in Fig. (**2**).

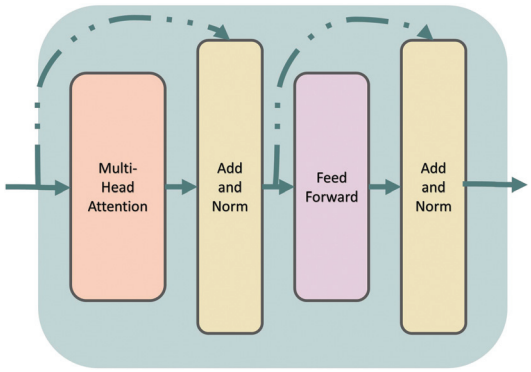

Fig. (2). One simplified encoder structure in transformer encoders. *(A higher resolution/colour version of this figure is available in the electronic copy of the article).*

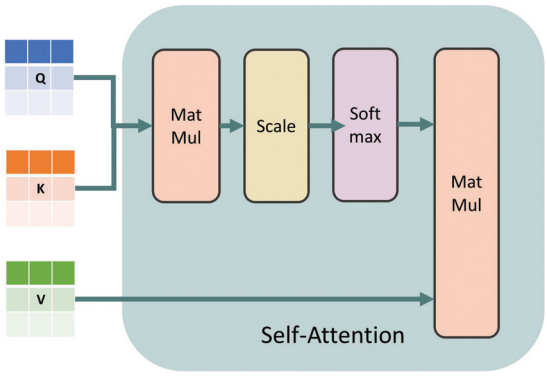

Fig. (3). The self-attention mechanism. *(A higher resolution/colour version of this figure is available in the electronic copy of the article).*

The transformer model cannot capture sequential information, which means that despite extensive disruptions in the sequence, Transformer will obtain the same results. To solve this problem, the feature of position embedding is introduced when encoding the word vectors. Specifically, position-coding adds the position information of the word to the word vector, enabling the distinction of words in different positions by Transformer.

In sequence analysis tasks, in addition to the absolute position of the word, the relative position of the word is also very important. Thus, the position encoding function can be expressed as:

$$PE_{(pos,2i)} = sin\left(pos/10000^{2i/d_{model}}\right) \quad (10)$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{2i/d_{model}}\right) \quad (11)$$

where pos is the position of each token, i is the dimensionality of the word, d_{model} is the dimensionality of the word vector.

A multi-head attention module instead of a single self-attention module is used because the weighting of a single self-attention module reduces the effective resolution; that is, it cannot fully reflect the information from different representation subspaces. The use of a multi-head attention mechanism is similar to that of multiple convolution kernels in the same convolution layer in a CNN to a certain extent. It can enhance the model's different characteristics of the token in different subspaces and avoid the suppression of this characteristic by average pooling. To explain the mechanism of self-attention, the calculation process is depicted in Fig. (**3**). Multi-head attention is equivalent to the ensemble of multiple different self-attentions. The self-attention function can be written as:

$$SA(Q,K,V) = softmax\left(\frac{QK}{\sqrt{d_k}}\right)V \quad (12)$$

For each input vector, we first need to generate three new vectors, Q, K, and V, representing the query vector, key vector, and value vector, respectively. Q means that in order to encode the current element, we need to attend to other elements (including itself) and acquire a query vector. The K key vector can be considered the key element for the information to be retrieved, and the V value vector is the real content. We calculated a score for each vector, QK. Thereafter, to stabilize the gradient, Transformer uses a score normalization. After applying the Softmax activation function to the normalized score, the activated result is multiplied by V to obtain the weighted score of each input vector. For multi-head attention, Qs, Ks, and Vs can be different.

Each attention layer adds a residual connection [58] and a LayerNorm layer [59]. The 'Add' operation borrows from the ResNet model and is mainly used to make the multilayer overlay of the Transformer without degrading the effect. The layer normalization operation normalizes the vector, which simplifies the difficulty of learning. Both methods are important for training a very deep neural network.

As our task is to make a classification prediction, only the encoder part in Transformer is needed to complete feature extraction. Accordingly, the decoder part is directly replaced by the Softmax module. Herein, additional details regarding the subsequent fully connected layer and the Softmax classification module are also provided.

3. EXPERIMENTS, RESULTS, AND DISCUSSION

3.1. Data Analysis

Before conducting the experiment, we performed a preprocessing analysis of the data to gain a preliminary

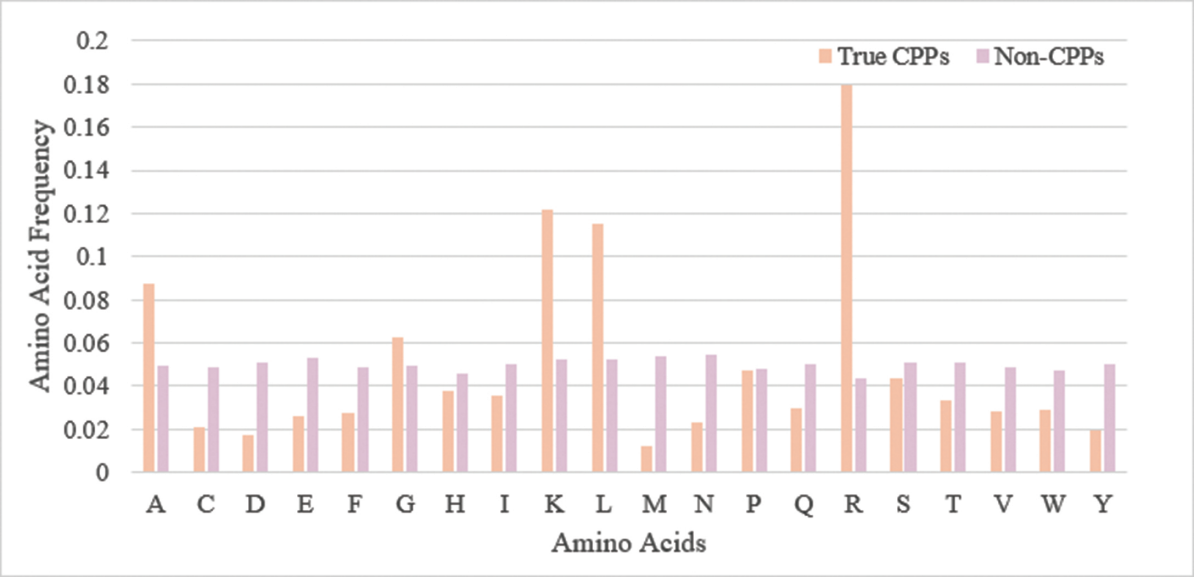

Fig. (4). Data analysis on amino acid appearance. *(A higher resolution/colour version of this figure is available in the electronic copy of the article).*

understanding of the overall data. We compared the True CPP samples and non-CPP sequence amino acid frequencies in the CPP924, as shown in Fig. (**4**). Evidently, the arginine (R) occupies a very large proportion in the positive CPP sequences, followed by lysine (K) and leucine (L). Therefore, these three amino acid structures may play an important role in cell penetration. Furthermore, methionine (M) and aspartate (D) were small, indicating that these two amino acids do not have a strong relationship with CPPs. Nonetheless, the distribution of the amino acid frequency of non-CPP sequences revealed that negative samples were carefully and equally collected, strictly based on the distribution of true CPPs in the dataset. The CPP924 dataset was confirmed to be one of the most stringent benchmarks. We compared the true CPP samples and non-CPP sequence amino acid frequencies in the CPP924 dataset.

3.2. Evaluation Metrics

To provide a detailed assessment of our proposed method and compare other feature extraction methods and state-of-the-art models, we employed six commonly used evaluation metrics to measure the performance [60-68]. Accuracy (ACC) [69] indicates the proximity of the prediction results to the ground truth. Sensitivity and specificity are statistical measures of the performance of a binary classification test that is widely used in medicine. Sensitivity (Sen) measures the proportion of true positives correctly identified, while specificity (Spe) [70] measures the proportion of true negatives. The Matthews correlation coefficient (MCC) [71] by Matthews *et al.* [72], also known as Pearson's phi coefficient, is used as a measure of the quality of binary classifications. MCC is essentially the correlation coefficient between the observed and predicted binary classification and returns a value between -1 and +1. The geometric mean (G-Mean) is a metric that measures the balance between classification performances in both the negative and positive classes. The F1 score corresponds to the harmonic mean of precision and recall, also known assensitivity, where precision is the positive predictive value. All six measurements are shown below [73-75]:

$$ACC = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

$$\mathrm{Sen} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (14)$$

$$\mathrm{Spe} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TP+TN}} \quad (15)$$

$$\mathrm{MCC} = \frac{\mathrm{TP\times TN - FP\times FN}}{\sqrt{(\mathrm{TP+FN})(\mathrm{TP+FP})(\mathrm{TN+FP})(\mathrm{TN+FN})}} \quad (16)$$

$$\mathrm{G-mean} = \sqrt{\mathrm{Sen\times Spe}} \quad (17)$$

$$\mathrm{F1} = 2\frac{\mathrm{Precision\times Sensitivity}}{\mathrm{Precision+Sensitivity}} = \frac{2\mathrm{TP}}{2\mathrm{TP+FP+FN}} \quad (18)$$

where, in all equations, TP represents correctly recognized positive CPP samples, TN represents the negative samples correctly predicted, FP represents the wrongly identified positive samples, and NP represents the wrongly predicted non-CPP peptide sequences.

To avoid the overfitting problem during the training process, we adopted the k-fold cross-validation and split the training set into k small sets; the test set was retained for final evaluation [25, 76, 77]. In our experiment, the k value was set to 10. In cross-validation, we used the test dataset for each sample of the test set one after another and used the remaining dataset for training.

4. COMPARISON ANALYSIS

4.1. Performance Analysis of Feature Extractor

As mentioned in the previous section, we used manual features as the network input to train and predict CPP using the CPP924 dataset. We also compared the automatic word embedding feature extractor by calculating these six evaluation metrics. A simple multi-perceptron was employed in the baseline model, consisting of only two fully connected layers and a dropout layer. This model was compared with our proposed CPPFormer to identify the performance of manual and automatic feature extraction in neural networks. The results are presented in Table **1**.

For all manual features, BLOSUM62 achieved the best results, followed by PAAC. However, the results of N-gram and AAC were somewhat unsatisfactory. For both models, automatic feature extraction had the best performance in all six metrics, indicating that the network has sufficient ability to extract features of linear peptide sequences, especially the multi-head attention module in CPPFormer, which achieved an accuracy of 92.19% and could distinguish the importance of linear sequences to a certain extent. In our experimental settings, the length of word embedding was set to 200. As the peptide sequences are markedly shorter than most of the protein sequences, we believe this is a more appropriate value.

By combining the above conclusion, we can determine whether the combination of manual features and automatic feature extraction can achieve better results. Accordingly, the machine learning model can be more explanatory based on automatic feature extraction, supplemented by manual features. We will carry out more related experiments in future work to prove this notion.

4.2. Performance Analysis Models

To further demonstrate the efficiency and accuracy of our proposed method, we selected some public CPP prediction methods to compare our test set. Table **2** shows a detailed display of the scores of the various indicators in all test models. The SVM model [11] did not perform very well on the existing data, exhibiting only 75.86% accuracy, 75.90% sensitivity, and 23.2% specificity. Recently published techniques, including CellPPD-DC, CellPPD-BP, and SkipCPP-Pred [14], had accuracies of 87.02%, 83.70%, and 90.62%, respectively, highlighting their great improvement on each of these indices. In addition, SOTA CPPred_RF [78] and CPPred_FL [25] almost matched the accuracy of our model, with accuracies of 91.6% and 91.2%, respectively. Nonetheless, our CPPFormer model outperforms these models with higher sensitivity, specificity, and F1 scores.

As the classic Transformer is a text model, we used some well-known text classification models for training the CPP classification tasks. TextCNN [79] is a text classification model that uses a convolutional layer and pooling layer. The convolution operation is equivalent to extracting the 2-gram, 3-gram, and 4-gram information in the sentence. Multiple convolutions are used to extract multiple features, and maximum pooling extracts the most important information to retain. TextRNN [80] abandoned the convolutional layer and used the classic LSTM layer to better capture long-distance semantic relations; however, owing to its recursive structure, it cannot be calculated in parallel and is slow. TextRNN + Attention [81] used a bidirectional LSTM and added an attention layer after it. The output of the LSTM is activated and then multiplied by the attention matrix to obtain the score for each time series. This operation is a weighted average of the hidden layer of each time segment of the LSTM. In our experiments, the result of TextRNN was not better than that of TextCNN. In fact, the addition of the attention mechanism brings the accuracy of TextRNN to 90.62%. CPPFormer retains the title as the best model owing to its advancing architecture and structure.

CPPFormer obtained better results because 1. We carefully designed the network according to the characteristics of the peptide sequence, abandoned the original Transformer 6-layer Encoder-Decoder structure, and only used one layer of an encoder for feature extraction, which is effective at solving the problem of underfitting short peptide sequences and small datasets on large models. 2. We used the attention mechanism, which effectively solves the problem that sequence features do not use different weights for important parts. 3. Based on the structure of the neural network, compared to kernel-based or other machine learning models, our model can achieve better performance in sequence classification tasks.

Table 1. Comparison of feature extractors in the baseline model and CPPFormer.

	Features	ACC	Spe	Sen	MCC	G-mean	F1-score
MLP	AAC	0.5652	0.5909	0.5417	0.1326	0.5658	0.5652
	PAAC	0.8623	0.8841	0.8406	0.7253	0.862	0.8593
	N-gram	0.7899	0.8413	0.7467	0.5863	0.7926	0.7943
	BLOSUM62	0.8696	0.8857	0.8529	0.7393	0.8692	0.8657
	Automatic	**0.8913**	**0.9014**	**0.8806**	**0.7824**	**0.8909**	**0.8872**
CPPFormer	AAC	0.7754	0.8154	0.7397	0.5547	0.7766	0.777
	PAAC	0.8841	0.9	0.8676	0.7683	0.8837	0.8806
	N-gram	0.8551	0.8824	0.8286	0.7115	0.855	0.8529
	BLOSUM62	0.8768	0.8767	0.8769	0.7531	0.8768	0.8702
	Automatic	**0.9219**	**0.9219**	**0.9256**	**0.8317**	**0.9237**	**0.9217**

Table 2. Analysis between SOTA CPP models and NLP models.

Models	AAC	Sen	Spe	F1-score
SVM	0.7586	0.7590	0.2320	0.8627
CellPPD-DC	0.8702	0.8330	0.9070	0.8682
CellPPD-BP	0.8370	0.7810	0.8920	0.8310
SkipCPP-Pred	0.9062	0.8851	0.9260	0.9143
CPPred_RF	0.9160	0.9050	0.9260	0.8931
CPPred_FL	0.9120	0.9200	0.9050	0.9129
CPPFormer	**0.9219**	**0.9219**	**0.9261**	**0.9217**
TextCNN	0.8906	0.8906	0.8941	0.8904
TextRNN	0.8594	0.8710	0.8685	0.8593
TextRNN+Att	0.9062	0.9081	0.9113	0.9062

4.3. Future Directions

The field of research on CPPs is still advancing, and substantial work is still required for researchers. AlphaFold2 [82] comprises a model that can reproduce protein folding to a large extent through artificial intelligence, combining the digital world in a computer with the complex real world. Compared to proteins, peptide sequences are shorter, and their structure is relatively simple. Therefore, investigations of polypeptide structure remain promising. In future work, we plan to combine more updated manual features and adopt innovative network structures and mechanisms to further establish an ideal system for CPP prediction [77, 83-89]. The proposed method uses the attention mechanism. Furthermore, different amino acid relationships with different weights have achieved significant progress. The specificity in different sequences should be considered in future assessments. The method of searching for the peptide space through reinforcement learning will be a more powerful choice. In addition, we also hope to use cell-penetrating peptides for more genomic applications in the future, such as drug design, new use of old drugs, and other issues.

CONCLUSION

Currently, the development of CPPs is ongoing. Machine learning can accelerate the development of bioinformatics, and CPP can be gradually used by more researchers in the pharmaceutical field to benefit humankind, cure diseases, and rescue wounds. Experiments have proven that the proposed CPPFormer method can achieve the best results ever in CPP sequence prediction. When selecting a model, the use of special hyperparameters for the CPP dataset to adapt to its characteristics and careful comparison experiments are key to achieving good results.

LIST OF ABBREVIATIONS

CPP	=	Cell-penetrating Peptides
CAD	=	Computer-aided Design
SOTA	=	State-of-the-art
NLP	=	Natural Language Processing
RNN	=	Recurrent Neural Network
LSTM	=	Long-Short Term Memory
PCA	=	Principal Component Analysis
SVM	=	Support Vector Machine
DPF	=	Dipeptide Combination Frequency
PAAC	=	Pseudo Amino Acid Composition
CNN	=	Convolutional Neural Networks
AAC	=	Amino Acid Composition
BLOSUM62	=	BLOcks SUbsititution Matrix
MCC	=	Matthews Correlation Coefficient
Spe	=	Specificity
Sen	=	Sensitivity

CONSENT FOR PUBLICATION

Not applicable.

FUNDING

This work was supported, in part, by the New Energy and Industrial Technology Development Organization 265 (NEDO) Grant (AJD30064) and the Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Scientific Research under Grant 18H03250, and the Natural Science Foundation of China (Nos. 62072329, and 62071278).

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

The authors have expressed their gratitude to the New Energy and Industrial Technology Development Organization (NEDO), Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Scientific research, and the Natural Science Foundation of China for providing financial grants to their paper.

REFERENCES

[1] Schneider, P.; Walters, W.P.; Plowright, A.T.; Sieroka, N.; Listgarten, J.; Goodnow, R.A. Jr.; Fisher, J.; Jansen, J.M.; Duca, J.S.; Rush, T.S.; Zentgraf, M.; Hill, J.E.; Krutoholow, E.; Kohler, M.; Blaney, J.; Funatsu, K.; Luebkemann, C.; Schneider, G. Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discov.,* **2020**, *19*(5), 353-364. http://dx.doi.org/10.1038/s41573-019-0050-3 PMID: 31801986

[2] Chen, L.; Chu, C.; Zhang, Y-H.; Zheng, M.; Zhu, L.; Kong, X. Identification of drug-drug interactions using chemical interactions. *Curr. Bioinform.,* **2017**, *12*(6), 526-534. http://dx.doi.org/10.2174/1574893611666160618094219

[3] Khalili, P.; Arakelian, A.; Chen, G.; Plunkett, M.L.; Beck, I.; Parry, G.C.; Doñate, F.; Shaw, D.E.; Mazar, A.P.; Rabbani, S.A. A non-RGD-based integrin binding peptide (ATN-161) blocks breast cancer growth and metastasis *in vivo. Mol. Cancer Ther.,* **2006**, *5*(9), 2271-2280. http://dx.doi.org/10.1158/1535-7163.MCT-06-0100 PMID: 16985061

[4] Fonseca, S.B.; Pereira, M.P.; Kelley, S.O. Recent advances in the use of cell-penetrating peptides for medical and biological applications. *Adv. Drug Deliv. Rev.,* **2009**, *61*(11), 953-964. http://dx.doi.org/10.1016/j.addr.2009.06.001 PMID: 19538995

[5] Lakshmanan, M.; Kodama, Y.; Yoshizumi, T.; Sudesh, K.; Numata, K. Rapid and efficient gene delivery into plant cells using designed peptide carriers. *Biomacromolecules,* **2013**, *14*(1), 10-16. http://dx.doi.org/10.1021/bm301275g PMID: 23215041

[6] Rüter, C.; Buss, C.; Scharnert, J.; Heusipp, G.; Schmidt, M.A. A newly identified bacterial cell-penetrating peptide that reduces the transcription of pro-inflammatory cytokines. *J. Cell Sci.,* **2010**, *123*(Pt 13), 2190-2198. http://dx.doi.org/10.1242/jcs.063016 PMID: 20554895

[7] Otvos, L. *Peptide-based drug design: here and now*; Springer, **2008**, pp. 1-8. http://dx.doi.org/10.1007/978-1-59745-419-3

[8] Gao, S.; Simon, M.J.; Hue, C.D.; Morrison, B. III.; Banta, S. An unusual cell penetrating peptide identified using a plasmid display-based functional selection platform. *ACS Chem. Biol.,* **2011**, *6*(5), 484-491. http://dx.doi.org/10.1021/cb100423u PMID: 21291271

[9] Yang, W.; Zhu, X-J.; Huang, J.; Ding, H.; Lin, H. A brief survey of machine learning methods in protein sub-Golgi localization. *Curr. Bioinform.,* **2019**, *14*(3), 234-240. http://dx.doi.org/10.2174/1574893613666181113131415

[10] Frankel, A.D.; Pabo, C.O. Cellular uptake of the tat protein from human immunodeficiency virus. *Cell,* **1988**, *55*(6), 1189-1193. http://dx.doi.org/10.1016/0092-8674(88)90263-2 PMID: 2849510

[11] Sanders, W.S.; Johnston, C.I.; Bridges, S.M.; Burgess, S.C.; Willeford, K.O. Prediction of cell penetrating peptides by support vector machines. *PLOS Comput. Biol.,* **2011**, *7*(7), e1002101. http://dx.doi.org/10.1371/journal.pcbi.1002101 PMID: 21779156

[12] Gautam, A.; Singh, H.; Tyagi, A.; Chaudhary, K.; Kumar, R.; Kapoor, P.; Raghava, G.P. CPPsite: A curated database of cell penetrating peptides. *Database (Oxford),* **2012**, *2012*, bas015. http://dx.doi.org/10.1093/database/bas015 PMID: 22403286

[13] Agrawal, P.; Bhalla, S.; Usmani, S.S.; Singh, S.; Chaudhary, K.; Raghava, G.P.; Gautam, A. CPPsite 2.0: A repository of experimentally validated cell-penetrating peptides. *Nucleic Acids Res.,* **2016**, *44*(D1), D1098-D1103. http://dx.doi.org/10.1093/nar/gkv1266 PMID: 26586798

[14] Wei, L.; Tang, J.; Zou, Q. SkipCPP-Pred: An improved and promising sequence-based predictor for predicting cell-

penetrating peptides. *BMC Genomics,* **2017**, *18*(Suppl. 7), 742.
http://dx.doi.org/10.1186/s12864-017-4128-1 PMID: 29513192

[15] Pandey, P.; Patel, V.; George, N.V.; Mallajosyula, S.S. KELM-CPPpred: Kernel extreme learning machine based prediction model for cell-penetrating peptides. *J. Proteome Res.,* **2018**, *17*(9), 3214-3222.
http://dx.doi.org/10.1021/acs.jproteome.8b00322 PMID: 30032609

[16] Zhang, J.; Liu, B. A review on the recent developments of sequence-based protein feature extraction methods. *Curr. Bioinform.,* **2019**, *14*(3), 190-199.
http://dx.doi.org/10.2174/1574893614666181212102749

[17] Dao, F.Y.; Lv, H.; Zulfiqar, H.; Yang, H.; Su, W.; Gao, H.; Ding, H.; Lin, H. A computational platform to identify origins of replication sites in eukaryotes. *Brief. Bioinform.,* **2021**, *22*(2), 1940-1950.
http://dx.doi.org/10.1093/bib/bbaa017 PMID: 32065211

[18] Tang, H.; Su, Z.D.; Wei, H.H.; Chen, W.; Lin, H. Prediction of cell-penetrating peptides with feature selection techniques. *Biochem. Biophys. Res. Commun.,* **2016**, *477*(1), 150-154.
http://dx.doi.org/10.1016/j.bbrc.2016.06.035 PMID: 27291150

[19] Hansen, M.; Kilk, K.; Langel, U. Predicting cell-penetrating peptides. *Adv. Drug Deliv. Rev.,* **2008**, *60*(4-5), 572-579.
http://dx.doi.org/10.1016/j.addr.2007.09.003 PMID: 18045726

[20] Dobchev, D.A.; Mager, I.; Tulp, I.; Karelson, G.; Tamm, T.; Tamm, K.; Janes, J.; Langel, U.; Karelson, M. Prediction of cell-penetrating peptides using artificial neural networks. *Curr. Comput. Aided Drug Des.,* **2010**, *6*(2), 79-89.
http://dx.doi.org/10.2174/157340910791202478 PMID: 20402661

[21] Tahir, M.; Idris, A. MD-LBP: An Efficient computational model for protein subcellular localization from HeLa cell lines using SVM. *Curr. Bioinform.,* **2020**, *15*(3), 204-211.
http://dx.doi.org/10.2174/1574893614666190723120716

[22] Kuo, J-H.; Chang, C-C.; Chen, C-W.; Liang, H-H.; Chang, C-Y.; Chu, Y-W. Sequence-based structural B-cell Epitope prediction by using two layer SVM model and association rule features. *Curr. Bioinform.,* **2020**, *15*(3), 246-252.
http://dx.doi.org/10.2174/1574893614666181123155831

[23] Holton, T.A.; Pollastri, G.; Shields, D.C.; Mooney, C. CPPpred: Prediction of cell penetrating peptides. *Bioinformatics,* **2013**, *29*(23), 3094-3096.
http://dx.doi.org/10.1093/bioinformatics/btt518 PMID: 24064418

[24] Chen, L.; Chu, C.; Huang, T.; Kong, X.; Cai, Y-D. Prediction and analysis of cell-penetrating peptides using pseudo-amino acid composition and random forest models. *Amino Acids,* **2015**, *47*(7), 1485-1493.
http://dx.doi.org/10.1007/s00726-015-1974-5 PMID: 25894890

[25] Qiang, X.; Zhou, C.; Ye, X.; Du, P.F.; Su, R.; Wei, L. CPPred-FL: A sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning. *Brief. Bioinform.,* **2018**. Online ahead of print.
http://dx.doi.org/10.1093/bib/bby091 PMID: 30239616

[26] Arif, M.; Ahmad, S.; Ali, F.; Fang, G.; Li, M.; Yu, D.J. TargetCPP: Accurate prediction of cell-penetrating peptides from optimized multi-scale features using gradient boost decision tree. *J. Comput. Aided Mol. Des.,* **2020**, *34*(8), 841-856.
http://dx.doi.org/10.1007/s10822-020-00307-z PMID: 32180124

[27] Su, R.; Hu, J.; Zou, Q.; Manavalan, B.; Wei, L. Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. *Brief. Bioinform.,* **2020**, *21*(2), 408-420.
http://dx.doi.org/10.1093/bib/bby124 PMID: 30649170

[28] Huang, G.; Li, J. Feature extractions for computationally predicting protein post-translational modifications. *Curr. Bioinform.,* **2018**, *13*(4), 387-395.
http://dx.doi.org/10.2174/1574893612666170707094916

[29] Chou, K.C. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins,* **2001**, *43*(3), 246-255.
http://dx.doi.org/10.1002/prot.1035 PMID: 11288174

[30] Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine,* **2018**, *13*(3), 55-75.
http://dx.doi.org/10.1109/MCI.2018.2840738

[31] Liu, B. Sentiment analysis and opinion mining. *Synth. Lectures Hum. Lang. Technol.,* **2012**, *5*(1), 1-167.
http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016

[32] van Aken, B.; Risch, J.; Krestel, R.; Löser, A. In: *Challenges for toxic comment classification: An in-depth error analysis*, Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, October **2018**; Association for Computational Linguistics: Stroudsburg, Pennsylvania, United States, pp. 33-42.
http://dx.doi.org/10.18653/v1/W18-5105

[33] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N. Attention is all you need. *Adv. Neural Inf. Process. Syst.,* **2017**, *30*, 5998-6008.

[34] Dehghani, M.; Gouws, S.; Vinyals, O.; Uszkoreit, J.; Kaiser, Ł. Universal transformers. *arXiv,* **2018**. Preprint Papers.

[35] LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In: *The Handbook of Brain Theory and Neural Networks*; MIT Press: Pennsylvania, **1995**. Vol. *3361* (10).

[36] Zhang, L.; He, Y.; Song, H.; Wang, X.; Lu, N.; Sun, L. Elastic net regularized softmax regression methods for multi-subtype classification in cancer. *Curr. Bioinform.,* **2020**, *15*(3), 212-224.
http://dx.doi.org/10.2174/1574893613666181112141724

[37] Jordan, M.I. Attractor dynamics and parallelism in a connectionist sequential machine. In: *Artificial Neural Networks: Concept Learning*; ACM Digital: NewYork City, **1990**; pp. 112-127.

[38] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.,* **1997**, *9*(8), 1735-1780.
http://dx.doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276

[39] Long, H.; Sun, Z.; Li, M.; Fu, H.Y.; Lin, M.C. Predicting protein phosphorylation sites based on deep learning. *Curr. Bioinform.,* **2020**, *15*(4), 300-308.
http://dx.doi.org/10.2174/1574893614666190902154332

[40] Chen, M.X.; Firat, O.; Bapna, A.; Johnson, M.; Macherey, W.; Foster, G.; Jones, L.; Schuster, M.; Shazeer, N.; Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Chen, Z.; Wu, Y.; Hughes, M. In: *The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation*, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, July, **2018**; Association for Computational Linguistics, Stroudsburg, Pennsylvania, United States, 2018; pp. 76-86.

http://dx.doi.org/10.18653/v1/P18-1008

[41] Luo, H.; Zhang, S.; Lei, M.; Xie, L. Simplified self-attention for transformer-based end-to-end speech recognition. *arXiv,* **2020**. Preprint paper.

[42] Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, Ł.; Shazeer, N.; Ku, A. Image transformer. *arXiv,* **2018**. Preprint paper.

[43] Du, Y.; Meier, J.; Ma, J.; Fergus, R.; Rives, A. Energy-based models for atomic-resolution protein conformations. *arXiv,* **2020**. Preprint paper.

[44] Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rost, B End-to-end multitask learning, from protein language to protein features without alignments *bioRxiv,* **2020**, 864405. Preprint paper.
http://dx.doi.org/10.1101/864405

[45] Madani, A.; McCann, B.; Naik, N.; Keskar, N.S.; Anand, N.; Eguchi, R.R ProGen: Language modeling for protein generation *bioRxiv,* **2020**, 982272. Preprint paper.
http://dx.doi.org/10.1101/2020.03.07.982272

[46] Rives, A.; Goyal, S.; Meier, J.; Guo, D.; Ott, M.; Zitnick, C.L Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences *bioRxiv,* **2020**, 622803. Preprint paper.
http://dx.doi.org/10.1101/622803

[47] Ingraham, J.; Garg, V.; Barzilay, R.; Jaakkola, T. Generative models for graph-based protein design. In: *Advances in Neural Information Processing Systems*, **2019**, 15820-15831. Article No.: 1417.

[48] Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention augmented convolutional networks. *arXiv,* **2019**. Preprint paper.

[49] Gulati, A.; Qin, J.; Chiu, C-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; Pang, R. Conformer: Convolution-augmented transformer for speech recognition. *arXiv,* **2020**. Preprint paper.

[50] Stuart, T.; Butler, A.; Hoffman, P.; Hafemeister, C.; Papalexi, E.; Mauck, W.M. III.; Hao, Y.; Stoeckius, M.; Smibert, P.; Satija, R. Comprehensive integration of single-cell data. *Cell,* **2019**, *177*(7), 1888-1902.e21.
http://dx.doi.org/10.1016/j.cell.2019.05.031 PMID: 31178118

[51] Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating long sequences with sparse transformers. *arXiv,* **2019**. Preprint paper.

[52] Yang, H.; Tang, H.; Chen, X.X.; Zhang, C.J.; Zhu, P.P.; Ding, H.; Chen, W.; Lin, H. Identification of secretory proteins in mycobacterium tuberculosis using pseudo amino acid composition. *BioMed Res. Int.,* **2016**, *2016*, 5413903.
http://dx.doi.org/10.1155/2016/5413903 PMID: 27597968

[53] Chen, X.X.; Tang, H.; Li, W.C.; Wu, H.; Chen, W.; Ding, H.; Lin, H. Identification of bacterial cell wall lyases *via* pseudo amino acid composition. *BioMed Res. Int.,* **2016**, *2016*, 1654623.
http://dx.doi.org/10.1155/2016/1654623 PMID: 27437396

[54] Broder, A.Z.; Glassman, S.C.; Manasse, M.S.; Zweig, G. Syntactic clustering of the web. *Comput. Netw. ISDN Syst.,* **1997**, *29*(8-13), 1157-1166.
http://dx.doi.org/10.1016/S0169-7552(97)00031-7

[55] Tang, H.; Zhao, Y.W.; Zou, P.; Zhang, C.M.; Chen, R.; Huang, P.; Lin, H. HBPred: a tool to identify growth hormone-binding proteins. *Int. J. Biol. Sci.,* **2018**, *14*(8), 957-964.
http://dx.doi.org/10.7150/ijbs.24174 PMID: 29989085

[56] Henikoff, S.; Henikoff, J.G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA,* **1992**, *89*(22), 10915-10919.
http://dx.doi.org/10.1073/pnas.89.22.10915 PMID: 1438297

[57] Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv,* **2014**. Preprint paper.

[58] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv,* **2016**. Preprint paper.

[59] Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv,* **2016**. Preprint paper.

[60] Boukelia, A.; Boucheham, A.; Belguidou, M.; Batouche, M.; Zehraoui, F.; Tahi, F. A novel integrative approach for non-coding RNA classification based on deep learning. *Curr. Bioinform.,* **2020**, *15*(4), 338-348.
http://dx.doi.org/10.2174/1574893614666191105160633

[61] Jin, Q.; Meng, Z.; Tuan, D.P.; Chen, Q.; Wei, L.; Su, R. DUNet: A deformable network for retinal vessel segmentation. *Knowl. Base. Syst.,* **2019**, *178*, 149-162.
http://dx.doi.org/10.1016/j.knosys.2019.04.025

[62] Manavalan, B.; Basith, S.; Shin, T.H.; Wei, L.; Lee, G. Meta-4mCpred: A Sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol. Ther. Nucleic Acids,* **2019**, *16*, 733-744.
http://dx.doi.org/10.1016/j.omtn.2019.04.019 PMID: 31146255

[63] Manavalan, B.; Basith, S.; Shin, T.H.; Wei, L.; Lee, G. mAHTPred: A sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics,* **2019**, *35*(16), 2757-2765.
http://dx.doi.org/10.1093/bioinformatics/bty1047 PMID: 30590410

[64] Hong, Z.; Zeng, X.; Wei, L.; Liu, X. Identifying enhancer-promoter interactions with neural network based on pre-trained DNA vectors and attention mechanism. *Bioinformatics,* **2020**, *36*(4), 1037-1043.
http://dx.doi.org/10.1093/bioinformatics/btz694 PMID: 31588505

[65] Wei, L.; Liao, M.; Gao, Y.; Ji, R.; He, Z.; Zou, Q. Improved and promising identification of human MicroRNAs by incorporating a high-quality negative set. *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* **2014**, *11*(1), 192-201.
http://dx.doi.org/10.1109/TCBB.2013.146 PMID: 26355518

[66] Wei, L.; Wan, S.; Guo, J.; Wong, K.K.L. A novel hierarchical selective ensemble classifier with bioinformatics application. *Artif. Intell. Med.,* **2017**, *83*, 82-90.
http://dx.doi.org/10.1016/j.artmed.2017.02.005 PMID: 28245947

[67] Wei, L.; Xing, P.; Shi, G.; Ji, Z.; Zou, Q. Fast prediction of protein methylation sites using a sequence-based feature selection technique. *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* **2019**, *16*(4), 1264-1273.
http://dx.doi.org/10.1109/TCBB.2017.2670558 PMID: 28222000

[68] Wei, L.; Xing, P.; Zeng, J.; Chen, J.; Su, R.; Guo, F. Improved prediction of protein-protein interactions using novel negative samples, features, and an ensemble classifier. *Artif. Intell. Med.,* **2017**, *83*, 67-74.
http://dx.doi.org/10.1016/j.artmed.2017.03.001 PMID: 28320624

[69] Amanat, S.; Ashraf, A.; Hussain, W.; Rasool, N.; Khan, Y.D. Identification of lysine carboxylation sites in proteins by integrating statistical moments and position relative features *via* general PseAAC. *Curr. Bioinform.,* **2020**, *15*(5), 396-407.
http://dx.doi.org/10.2174/1574893614666190723114923

[70] Niu, M.; Zhang, J.; Li, Y.; Wang, C.; Liu, Z.; Ding, H.; Zou, Q.; Ma, Q. CirRNAPL: A web server for the identification of circRNA based on extreme learning machine. *Comput. Struct. Biotechnol. J.,* **2020**, *18*, 834-842. http://dx.doi.org/10.1016/j.csbj.2020.03.028 PMID: 32308930

[71] Li, Y.; Niu, M.; Zou, Q. ELM-MHC: An improved MHC identification method with extreme learning machine algorithm. *J. Proteome Res.,* **2019**, *18*(3), 1392-1401. http://dx.doi.org/10.1021/acs.jproteome.9b00012 PMID: 30698979

[72] Matthews, B.W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta,* **1975**, *405*(2), 442-451. http://dx.doi.org/10.1016/0005-2795(75)90109-9 PMID: 1180967

[73] Lv, H.; Dao, F-Y.; Guan, Z-X.; Yang, H.; Li, Y-W.; Lin, H. Deep-Kcr: Accurate detection of lysine crotonylation sites using deep learning method. *Brief. Bioinform.,* **2021**, *22*(4), bbaa255. http://dx.doi.org/10.1093/bib/bbaa255 PMID: 33099604

[74] Zhu, X.J.; Feng, C.Q.; Lai, H.Y.; Chen, W.; Lin, H. Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowl. Base. Syst.,* **2019**, *163*, 787-793. http://dx.doi.org/10.1016/j.knosys.2018.10.007

[75] Lin, H.; Liang, Z.Y.; Tang, H.; Chen, W. Identifying sigma70 promoters with novel pseudo nucleotide composition. *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* **2019**, *16*(4), 1316-1321. http://dx.doi.org/10.1109/TCBB.2017.2666141 PMID: 28186907

[76] Wei, L.; Ding, Y.; Su, R.; Tang, J.; Zou, Q. Prediction of human protein subcellular localization using deep learning. *J. Parallel Distrib. Comput.,* **2018**, *117*, 212-217. http://dx.doi.org/10.1016/j.jpdc.2017.08.009

[77] Wei, L.; Hu, J.; Li, F.; Song, J.; Su, R.; Zou, Q. Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. *Brief. Bioinform.,* **2018**, *21*(1), 106-119. http://dx.doi.org/10.1093/bib/bby107 PMID: 30383239

[78] Wei, L.; Xing, P.; Su, R.; Shi, G.; Ma, Z.S.; Zou, Q. CPPred-RF: A Sequence-based predictor for identifying cell-penetrating peptides and their uptake efficiency. *J. Proteome Res.,* **2017**, *16*(5), 2044-2053. http://dx.doi.org/10.1021/acs.jproteome.7b00019 PMID: 28436664

[79] Kim, Y. In: *Convolutional neural networks for sentence classification*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October, **2014**; Stroudsburg, Pennsylvania, United States, **2014**; pp. 1746–1751. http://dx.doi.org/10.3115/v1/D14-1181

[80] Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv,* **2016**. Preprint paper.

[81] Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. In: *Attention-based bidirectional long short-term memory networks for relation classification*, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, August, 2016; Association for Computational Linguistics: Stroudsburg, Pennsylvania, United States, **2016**; pp. 207–212. http://dx.doi.org/10.18653/v1/P16-2034

[82] Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S.A.A.; Ballard, A.J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A.W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature,* **2021**, *596*(7873), 583-589. http://dx.doi.org/10.1038/s41586-021-03819-2 PMID: 34265844

[83] Su, R.; Liu, X.; Wei, L.; Zou, Q. Deep-Resp-Forest: A deep forest model to predict anti-cancer drug response. *Methods,* **2019**, *166*, 91-102. http://dx.doi.org/10.1016/j.ymeth.2019.02.009 PMID: 30772464

[84] Su, R.; Liu, X.; Xiao, G.; Wei, L. Meta-GDBP: A high-level stacked regression model to improve anticancer drug response prediction. *Brief. Bioinform.,* **2020**, *21*(3), 996-1005. http://dx.doi.org/10.1093/bib/bbz022 PMID: 30868164

[85] Su, R.; Wu, H.; Xu, B.; Liu, X.; Wei, L. Developing a multi-dose computational model for drug-induced hepatotoxicity prediction based on toxicogenomics data. *IEEE/ACM Trans. Comput. Biol. Bioinformatics,* **2019**, *16*(4), 1231-1239. http://dx.doi.org/10.1109/TCBB.2018.2858756 PMID: 30040651

[86] Wei, L.; Chen, H.; Su, R. M6APred-EL: A sequence-based predictor for identifying N6-methyladenosine sites using ensemble learning. *Mol. Ther. Nucleic Acids,* **2018**, *12*, 635-644. http://dx.doi.org/10.1016/j.omtn.2018.07.004 PMID: 30081234

[87] Su, R.; Liu, X.; Wei, L. MinE-RFE: Determine the optimal subset from RFE by minimizing the subset-accuracy-defined energy. *Brief. Bioinform.,* **2020**, *21*(2), 687-698. http://dx.doi.org/10.1093/bib/bbz021 PMID: 30860571

[88] Dai, C.; Feng, P.; Cui, L.; Su, R.; Chen, W.; Wei, L. Iterative feature representation algorithm to improve the predictive performance of N7-methylguanosine sites. *Brief. Bioinform.,* **2021**, *22*(4), bbaa278. http://dx.doi.org/10.1093/bib/bbaa278

[89] Wei, L.; He, W.; Malik, A.; Su, R.; Cui, L.; Manavalan, B. Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework. *Brief. Bioinform.,* **2021**, *22*(4), 2020-Nov-05. http://dx.doi.org/10.1093/bib/bbaa275 PMID: 33152766