RESEARCH

Open Access

DeepBP: Ensemble deep learning strategy for bioactive peptide prediction

Ming Zhang[1*], Jianren Zhou[1], Xiaohua Wang[1], Xun Wang[1] and Fang Ge[2*]

*Correspondence:
zhangming@just.edu.cn;
gfang0616@njupt.edu.cn

[1] School of Computer, Jiangsu University of Science and Technology, 666 Changhui Road, Zhenjiang 212100, China
[2] State Key Laboratory of Organic Electronics and Information Displays & Institute of Advanced Materials (IAM), Nanjing University of Posts & Telecommunications, 9 Wenyuan Road, Nanjing 210023, China

Abstract

Background: Bioactive peptides are important bioactive molecules composed of short-chain amino acids that play various crucial roles in the body, such as regulating physiological processes and promoting immune responses and antibacterial effects. Due to their significance, bioactive peptides have broad application potential in drug development, food science, and biotechnology. Among them, understanding their biological mechanisms will contribute to new ideas for drug discovery and disease treatment.

Results: This study employs generative adversarial capsule networks (CapsuleGAN), gated recurrent units (GRU), and convolutional neural networks (CNN) as base classifiers to achieve ensemble learning through voting methods, which not only obtains high-precision prediction results on the angiotensin-converting enzyme (ACE) inhibitory peptides dataset and the anticancer peptides (ACP) dataset but also demonstrates effective model performance. For this method, we first utilized the protein language model—evolutionary scale modeling (ESM-2)—to extract relevant features for the ACE inhibitory peptides and ACP datasets. Following feature extraction, we trained three deep learning models—CapsuleGAN, GRU, and CNN—while continuously adjusting the model parameters throughout the training process. Finally, during the voting stage, different weights were assigned to the models based on their prediction accuracy, allowing full utilization of the model's performance. Experimental results show that on the ACE inhibitory peptide dataset, the balanced accuracy is 0.926, the Matthews correlation coefficient (MCC) is 0.831, and the area under the curve is 0.966; on the ACP dataset, the accuracy (ACC) is 0.779, and the MCC is 0.558. The experimental results on both datasets are superior to existing methods, demonstrating the effectiveness of the experimental approach.

Conclusion: In this study, CapsuleGAN, GRU, and CNN were successfully employed as base classifiers to implement ensemble learning, which not only achieved good results in the prediction of two datasets but also surpassed existing methods. The ability to predict peptides with strong ACE inhibitory activity and ACPs more accurately and quickly is significant, and this work provides valuable insights for predicting other functional peptides. The source code and dataset for this experiment are publicly available at https://github.com/Zhou-Jianren/bioactive-peptides.

Keywords: ACE inhibitory peptides, Anticancer peptides, Protein language model, Gated recurrent unit, Generative adversarial capsule network

Background

Bioactive peptides are molecules composed of short-chain amino acids, usually consisting of 2–50 amino acid residues. They can interact with receptors in organisms through specific mechanisms, thereby regulating a variety of physiological functions, such as anti-inflammatory, hypoglycemic, antioxidant and anti-cancer [1]. Bioactive peptides are widely found in food, dairy products, marine organisms and plants, and can be obtained by enzymatic hydrolysis, fermentation or chemical synthesis. Among them, anticancer peptides (ACP) have been one of the hot topics of research in recent years, and have attracted attention because of their ability to induce tumor cell apoptosis, inhibit tumor growth and metastasis. At the same time, Angiotensin-Converting Enzyme (ACE) inhibitory peptides, as an important bioactive peptide, can effectively lower blood pressure, improve cardiovascular health, and are widely used in the prevention and treatment of hypertension and heart disease. However, traditional wet experimental methods have many defects in the screening and optimization of functional peptides [2]. These methods are usually time-consuming and labor-intensive, and require a large amount of reagents and experimental samples, resulting in high costs [3]. In addition, the results of wet experiments are often affected by experimental conditions and operator skills, and have certain uncertainties [4]. In contrast, the use of computational methods and deep learning models can efficiently predict potential bioactive peptides, thereby improving screening efficiency and reducing costs, providing a more reliable and rapid alternative for the study of functional peptides [5].

At present, computational methods have achieved important success in the prediction of bioactive peptides. In the prediction of bioactive peptides, existing computational methods mainly include sequence alignment, machine learning and deep learning techniques [6]. Sequence alignment methods, mainly based on similarity search, such as BLAST, can be used to identify peptides with potential bioactivity [7]. Machine learning-based methods mainly use a variety of machine learning algorithms, such as random forests and support vector machines, to classify and extract features of bioactive peptides [8]. For example, methods such as AVPpred [9], PredAPP [10], AIPpred [11], and THPep [12] use SVM and RF algorithms based on various feature engineering techniques (including PSSM matrix, pseudo amino acid composition (PseAAC), physicochemical properties, etc.) to identify peptides [13]. Although effective, traditional machine learning-based methods are usually difficult to generalize to other peptide data sets. In recent years, algorithmic models based on deep learning, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), have been widely used in the functional prediction of peptides [14]. Compared with machine learning, deep learning algorithm models have the advantages of automatically extracting features and fully utilizing high-performance hardware [15]. Different deep learning frameworks are used to identify bioactive peptides [16–21], such as DeepAFP [5], Deep-AmPEP30 [22], and ITP-Pred [23].

Among many experimental methods, the pLM4ACE method is used to detect the activity of ACE inhibitory peptides. In pLM4ACE, Du et al. [24]. established a state-of-the-art dataset by manually collecting peptides that exhibit ACE inhibition and divided them into high-active and low-active/non-active groups based on half-maximal inhibitory concentration (IC50). The pLM4ACE model was developed using the protein language model (pLM)-evolutionary scaling model (ESM-2). They built an optimized model based on logistic regression (LR) combined with ESM-2 embedding. On the test dataset, pLM4ACE achieved a balanced accuracy (BACC) of 0.883 ± 0.017, a Matthews correlation coefficient (MCC) of 0.77 ± 0.032, and an area under the curve (AUC) of 0.96 ± 0.009. In ACP prediction, Jiang et al. [25]. used the pre-trained model RoBERTa to predict ACPs. They fine-tuned the pre-trained model so that RoBERTa could better predict ACPs and ultimately achieved excellent performance. The accuracy (ACC) reached 0.762 and the MCC was 0.528.

In existing research methods, single models are used. Although single models have excellent performance in certain environments, they also have certain limitations. Specifically, a single model can usually only capture a certain type of features in the data and cannot fully cover the diversity of the data. For example, the LR model is a linear model that is good at processing linear relationship features, but performs poorly for complex nonlinear patterns; while RoBERTa is good at processing sequence and language data, it may not be accurate enough when capturing spatial or local features in specific biological sequences. In addition, single models are easily affected by data noise, and the prediction results may not be stable enough. In order to overcome the limitations of a single model, ensemble learning methods can capture different information in the data more comprehensively by combining the advantages of multiple models. Using three models, generative adversarial capsule networks (CapsuleGAN), gated recurrent unit (GRU) and CNN, for voting integration can effectively integrate the strengths of each model: CapsuleGAN can capture the spatial hierarchy of the data, GRU is good at processing the temporal information in the sequence, and CNN can effectively extract local features. Through weighted average voting, different models play their respective advantages in the final prediction, thereby improving the accuracy and robustness of the overall model and reducing the bias and instability that may exist in a single model. The use of ensemble learning not only improves the accuracy of predictions, but also enhances the adaptability of the model to different types of data features. The main contributions of this work are as follows: 1) Introducing an integrated learning framework: A voting integration method based on CapsuleGAN, GRU and CNN improves the performance of peptide function prediction. 2) Innovative use of CapsuleGAN: CapsuleGAN was applied to peptide prediction for the first time, and the feature extraction capability was enhanced by combining it with the attention mechanism. 3) Specialized optimization for bioactive peptides: This study combined the characteristics of deep learning models to design a model architecture optimized for the prediction tasks of ACE inhibitory peptides and ACPs, achieving efficient prediction of these bioactive peptides. 4) Comprehensive comparative analysis: Detailed experimental comparison with existing methods verifies the performance advantages of the new method. 5) Scalable framework: The proposed model framework has good scalability and can be applied to other biological sequence prediction tasks.

Material and methods

Benchmark datasets

For ACE inhibitory peptides, we used the same benchmark dataset created by Du et al. to test the experimental method [24]. This dataset contains 1020 experimentally verified

ACE inhibitory peptide sequences. Among them, positive samples are peptides that exhibit high ACE inhibitory activity, while negative samples are peptides that exhibit low and inactive ACE inhibitory activity. In the end, there were 394 positive samples and 626 negative samples. Subsequently, all positive and negative samples were randomly divided into training datasets and independent test datasets in a ratio of 8:2. For the ACP dataset, we obtain it from the study of Jiang et al. In the ACP dataset [25], the number of positive and negative samples is the same, and the training set and test set are also divided in a ratio of 8:2. The distribution of the number of peptide sequences in the two datasets is shown in Table 1, the distribution of peptide sequence lengths is shown in Fig. 1, and the amino acid composition is shown in Fig. 2.

Feature representation method

In this study, we mainly use protein language models for feature extraction work. The protein language model is a deep learning-based tool that can automatically extract rich feature information from protein sequences [26]. Compared with traditional feature extraction methods, protein language models have significant advantages. Traditional methods often rely on manually designed features, which may not fully capture complex

Table 1 Distribution of datasets

Dataset	Positive	Negative	Train data	Test data
ACE inhibitory peptides	394	626	816	204
ACP	861	861	1378	344

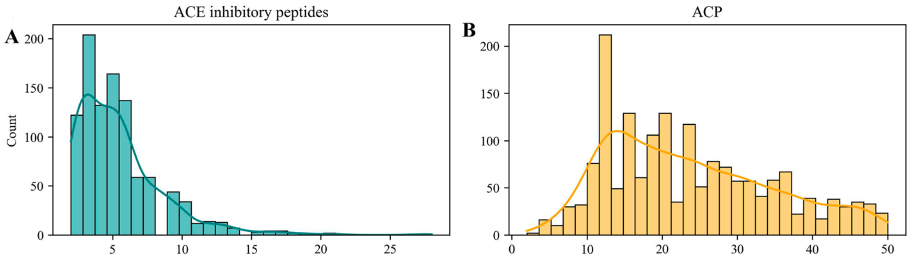

Fig. 1 Sequence length distribution, the X-axis represents the length of the peptide sequence, and the y-axis represents the number of corresponding sequences. **A** Sequence length distribution of ACE inhibitory peptide dataset. **B** Sequence length distribution of ACP dataset

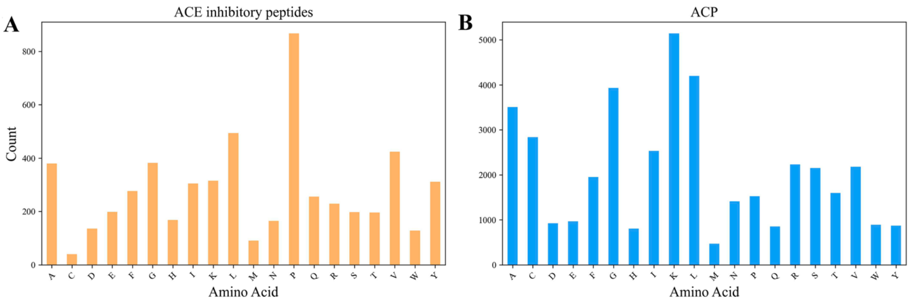

Fig. 2 Amino acid composition. **A** Amino acid composition of ACE inhibitory peptide sequences. **B** Amino acid composition of ACP sequences

sequence information and contextual relationships [27]. By training massive sequence data, protein language models can learn potential patterns and semantic associations in sequences, thereby extracting more comprehensive and accurate features. This model can not only handle diverse amino acid sequences, but also identify potential functional domains and important biological information, improving the accuracy of peptide classification and functional prediction. In this study, the protein language model used is ESM-2.

The ESM-2 model [28], released by the Meta AI Research (FAIR) Protein team in 2022, represents a further development in their work on language models. ESM-2 demonstrates a good understanding of protein sequences and has a faster inference speed compared to AlphaFold2 [29]. The ESM-2 used in this study is a pLM with 320 output embeddings. The pre-trained pLM along with its implementation code are available at https://github.com/facebookresearch/esm. ESM-2 is a protein language model improved based on the Bidirectional Encoder Representations from Transformers (BERT) architecture [30]. It uses a large number of protein sequences for fine-tuning. The last hidden state of the ESM-2 model is its output. In this context, the feature dimensions generated by the model typically do not vary with the length of the peptide sequence [3]. Every peptide sequence is depicted by a vector of identical dimensions, with features incorporating information from neighboring residues as well as the overall sequence context. Therefore, features derived from ESM-2 can be regarded as employing a global description strategy, guaranteeing a consistent feature dimension irrespective of the peptide's length [31].

Performance evaluation methods

This experiment uses the test set to evaluate the effectiveness and robustness of the model. At the same time, ACC, BACC, MCC, sensitivity (Sn) and specificity (Sp) are used to evaluate the performance of the model. The parameters are calculated using the counts of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN), as shown in the following formula:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$Sn = \frac{TP}{TP + FN} \tag{2}$$

$$Sp = \frac{TN}{TN + FP} \tag{3}$$

$$BACC = 0.5 * Sn + 0.5 * Sp \tag{4}$$

$$MCC = \frac{(TP * TN) - (FN * FP)}{\sqrt{(TP + FN) * (TP + FP) * (TN + FP) * (TN + FN)}} \tag{5}$$

The AUC is determined using the "auc" function available in the Sklearn library [32]. The AUC quantifies the area under the receiver operating characteristic (ROC) curve.

This curve illustrates the model's capacity to differentiate between positive and negative instances across various threshold settings.

Model

Overview of DeepBP

The purpose of this experimental method is to identify functional peptides, mainly to detect the activity of ACE inhibitory peptides and detect ACPs. The dataset is derived from the studies of Ref [24] and Ref [25]. (Fig. 3A). This experimental method mainly consists of two parts: feature extraction module and classification module (Fig. 3B). In the feature extraction module, the protein language model used is mainly ESM-2, and the features with a dimension of ($n \times 320$) are obtained through the 6-layer BERT model. In the classification module, three different models are mainly used: CapsuleGAN, CNN, and GRU. In CapsuleGAN, the attention mechanism-convolutional block attention module (CBAM) is added to improve the accuracy (Fig. 3C). The CNN model can better capture local features and increase the robustness of the model (Fig. 3D). The GRU model is a model that specializes in processing sequence data (Fig. 3E). By using the gating mechanism, it can better capture the medium and long-range information in the

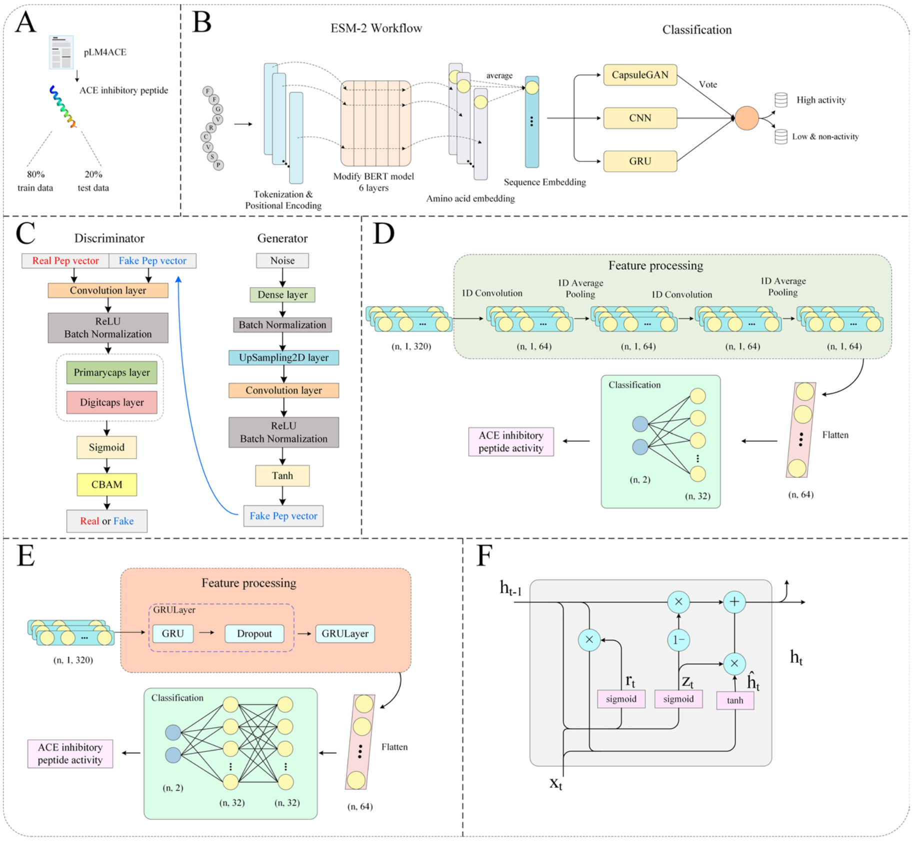

Fig. 3 Overall experimental process of DeepBP. **A** Dataset collection and processing. **B** Overall experimental process. Including feature extraction module and classification module. The feature extraction module is mainly composed of ESM-2, and its main content is to extract high-quality peptide representations from primary sequences. The classification module is composed of CapsuleGAN, GRU, and CNN models, and the classification method adopts the voting method. **C** CapsuleGAN model structure diagram. **D** CNN model structure diagram. **E** GRU model structure diagram. **F** GRULayer internal structure design

features (Fig. 3F). By training these three models and testing the test set, the prediction results of different models on the test set are obtained, and then these prediction results are weighted averaged as the final prediction results of the test set.

Generative adversarial capsule networks

In this study, we used the CapsuleGAN model [33]. This model is a generative adversarial network (GAN) based on the capsule network, which can effectively capture the spatial information of the input data and the complex relationship between features [34]. CapsuleGAN reduces the possibility of feature loss by maintaining the interdependence of information in the feature dimension. It can better characterize the local structure and global information in the sequence data, thereby improving the generalization ability of the model.

The CapsuleGAN model primarily consists of two components: the generator and the discriminator [33]. In the generator, a random vector (noise) is provided as input, and through a neural network, it generates synthetic data of the same dimensionality as the real data [35]. In the discriminator, both the synthetic data samples generated by the generator and real data samples are inputted, and the discriminator attempts to differentiate between them, thereby enhancing the model's learning ability with respect to real data [36]. In the process of the generator generating fake peptide vectors, the discriminator strengthens the model's understanding of peptide characteristics by identifying "real peptide vectors" and "fake peptide vectors". This adversarial learning mechanism aims to enhance the generalization ability of the model and make it more accurate in predicting active peptides.

In this study, the generator follows the architecture of GAN to learn data representation. It initially takes a random noise vector as input and employs fully connected layers and batch normalization layers to generate an initial data representation. Subsequently, the quality of the generated data is enhanced through upsampling. This process involves two layers of convolution and activation, followed by batch normalization operations. Finally, the generator employs 3×3 convolutional kernels and a tanh activation function to produce synthetic data of the same size as the real data.

In the discriminator part, a combination of CNN and Capsule Neural Network [37, 38] structure is employed, along with the integration of residual connections and CBAM [39] to enhance performance. Initially, both real and synthetic data inputs into the discriminator undergo two layers of convolution and activation, followed by batch normalization operations, before being fed into the capsule network structure.

The capsule network structure comprises two main components: the Primarycaps layer and the Digitcaps layer [40]. In the Primarycaps layer, input features are received and represented in the form of capsules, with each capsule associated with a specific feature [41]. The Digitcaps layer receives capsules as input and employs the "dynamic routing" algorithm to determine the connection weights between the Primarycaps and Digitcaps [42]. In this study, the dynamic routing process is repeated three times to enhance the model's learning capability. Residual connections and CBAM modules are added to the obtained output to adjust the weights between features and enhance the model's capability, followed by classification using a Sigmoid activation function.

Gated recurrent unit

GRU [43] is an improved RNN specifically designed to process sequence data. Compared with traditional RNN, GRU can effectively solve the long-term dependency problem by introducing a gating mechanism, thereby better capturing long-distance information in the sequence.

In this study, the preprocessed feature data is processed by GRU. The processed features are regularized by a Dropout layer to avoid overfitting. Subsequently, the features are further abstracted and processed by GRU again. Finally, the output features are flattened for the final classification task.

In this model, GRU is the most important. Specifically, GRU contains two main gates: reset gate and update gate. When data is input, the GRU model will gradually process the input of each time step. For each time step t, the reset gate and update gate need to be calculated. The reset gate r_t determines how to combine the previous hidden state h_{t-1} with the current input x_t:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{6}$$

where W_r is the weight matrix of the reset gate and x_t is the feature vector. The update gate z_t controls the degree of update of the current hidden state:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{7}$$

where W_z is the weight matrix of the update gate. Then the candidate hidden state is calculated based on the reset gate and the current input:

$$\hat{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t]) \tag{8}$$

Finally, the final hidden state h_t of the GRU can be obtained:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \tag{9}$$

The final hidden state h_t can be used as a representation of the entire sequence.

In this process, the reset gate controls how much previous information the model retains in the current time step, while the update gate determines the degree of fusion of the current input with past information [44]. This design gives GRU a stronger memory capacity and faster convergence speed when processing tasks such as time series, language modeling, and biological sequences. Due to its relatively simple structure, GRU is also superior to the more complex long short-term memory network (LSTM) in computational efficiency, so it is widely used in practical applications, especially in fields such as bioinformatics and natural language processing [45].

Convolutional neural networks

CNN [46] is a deep learning model widely used in image processing and sequence data analysis. CNN can effectively extract local features from input data through local connections and weight sharing. In this experiment, the CNN architecture is used to process the sequence features extracted from ESM-2. The model mainly consists of multiple convolutional layers, pooling layers and fully connected layers. Through the gradual extraction and dimensionality reduction of sequence features, the model can effectively capture the local pattern of features and use them for classification tasks.

In the CNN model, the input data is the preprocessed data of the sequence features extracted by ESM-2, with a dimension of (n, 1, 320). Through the convolution layer, the local pattern of the feature is extracted and reduced to (n, 1, 64), which helps to reduce redundant information and enhance the capture of local correlation. The subsequent pooling operation further reduces the complexity of the feature, prevents the model from overfitting and improves the generalization ability. The combined operation of convolution and pooling is repeatedly applied, and finally the feature is processed into a vector with 64 dimensions. After feature extraction, the flattening operation maps the multi-dimensional features into a one-dimensional vector and enters the fully connected layer. This step maps the extracted features to 32 neurons through linear combination to further refine the high-level feature representation. Finally, the classification layer outputs the 32-dimensional vector into two categories, representing the high and low ACE inhibitory activity of the peptide.

The advantage of this CNN architecture is that it can effectively process sequence data through convolution operations and extract local patterns with biological significance [47]. The superposition structure of multi-layer convolution and pooling enhances the model's perception of patterns at different scales, making it perform well when processing complex sequence data [48]. Finally, the combination of flattening and fully connected layers provides a more refined feature representation for classification, ensuring the classification performance of the model.

Results

Performance evaluation of single classifier and ensemble learning

In the classification study of functional peptides, choosing a suitable classifier is crucial to improving the prediction performance of the model. Although traditional single classifiers perform well in some cases, they often have difficulty processing complex biological data, which may lead to overfitting or insufficient generalization of the model. To solve this problem, this study uses an ensemble learning method to combine the prediction results of multiple base learners in order to improve classification accuracy and stability [49].

This study systematically evaluated the performance difference between single classifiers and ensemble learning for ACE inhibitory peptides and ACPs datasets. Specifically, we first preprocessed and extracted features for the dataset, then applied different single classifiers (CapsuleGAN, GRU, CNN) and recorded their performance indicators. Subsequently, the prediction results of multiple base classifiers were fused through ensemble learning methods to further improve the classification effect. In addition, in ensemble learning, we used two different methods for testing and comparison, mainly voting and stacking. By comparing the performance of the two methods, the method with the best performance was obtained.

Experimental results on ACE inhibitory peptide dataset

We evaluated the classification performance of different methods on the ACE inhibitory peptide dataset, and the results are shown in Table 2. As can be seen from the

Table 2 Performance evaluation of each method in the ACE inhibitory peptide dataset

Method	BACC	Sn	Sp	MCC	AUC
GRU	0.913	0.949	0.877	0.802	0.965
CNN	0.911	0.949	0.874	0.799	0.963
CapsuleGAN	0.887	0.916	0.859	0.748	0.937
Stacking	0.919	**0.971**	0.866	0.805	0.964
Voting	**0.926**	0.960	**0.891**	**0.831**	**0.966**

Bold values indicate the highest values for each respective indicator

The stacking method uses CapsuleGAN, GRU, and CNN as base classifiers and LR as meta-classifier to get the highest predicted value. The weight ratio of GRU, CNN, and CapsuleGAN in the voting method is 6:3:1

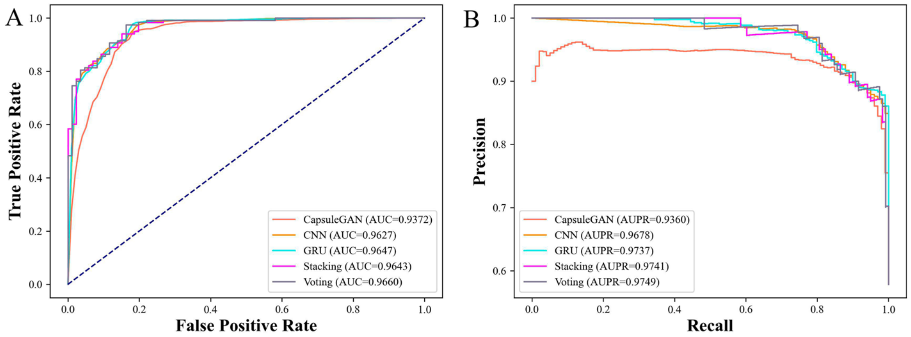

Fig. 4 Model performance on ACE inhibitory peptides. **A** ROC curve. **B** PR curve

table, the ensemble learning methods (Voting and Stacking) performed well in most indicators, surpassing single classifiers. In multiple indicators such as Sn, Sp, BACC, MCC and AUC, the Voting method performed particularly well, achieving a BACC of 0.926 and an AUC of 0.966, demonstrating its advantages in functional peptide classification.

Among the single classifiers, the GRU model performed best, with an AUC of 0.965, followed by the CNN model with an AUC of 0.963. Although CapsuleGAN performed slightly worse in sensitivity, achieving an AUC of 0.937, its overall performance is still competitive.

The performance differences of different methods can be intuitively seen from the ROC and PR curves (Fig. 4). The ROC curve shows that the curves of the GRU, Voting and Stacking methods are almost close to the ideal state, indicating that they can maintain a high true positive rate with a low false positive rate. Similarly, the PR curve also shows the excellent performance of the GRU and ensemble methods, especially when the recall rate is high, they still maintain a high precision.

We also visualized the loss function of the model and plotted the loss curves of the three base classifiers (CapsuleGAN, CNN, and GRU), as shown in Fig. 5. As can be seen from the figure, during the training process, the loss values of all models gradually decreased with the increase in the number of training rounds, and tended to stabilize near the end of the training, without a significant upward trend, indicating that the model did not have obvious overfitting. These loss curves can further verify the stability of the model during the training process, and the final convergence state is relatively ideal.

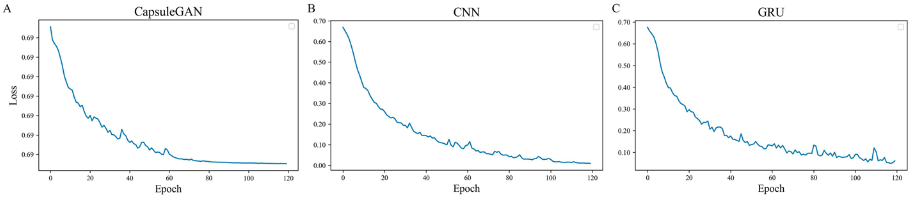

Fig. 5 Loss function of the model training on the ACE inhibitory peptide dataset. **A** Loss function of the CapsuleGAN model. **B** Loss function of the CNN model. **C** Loss function of the GRU model

Experimental results on ACP dataset

We also evaluated the classification performance of different methods on the ACP dataset, and the results are shown in Table 3. As can be seen from the table, the ensemble learning methods (Voting and Stacking) still perform well in terms of performance indicators. In multiple indicators such as Sn, Sp, ACC, and MCC, the Voting method performed well, achieving an ACC of 0.779 and an MCC of 0.558. Among single classifiers, the CapsuleGAN model performed best, with an ACC of 0.75 and an MCC of 0.501. It can be intuitively seen from the ROC and PR curves (Fig. 6) that the Voting and Stacking methods still maintain high accuracy in the ROC curve and PR curve. Among them, Vote performed particularly well, with an AUC of 0.8325 and an AUPR of 0.8517.

Similar to the ACE inhibitory peptide dataset, we also visualized the loss function of the model on the ACP dataset. As shown in Fig. 7. As can be seen from the curves, the loss values of the three models gradually decreased with the training iterations and tended to stabilize near the end of training. The loss value of the CapsuleGAN model decreased rapidly in the first few iterations and then entered a stable stage. The loss

Table 3 Performance evaluation of each method in the ACP dataset

Method	ACC	Sn	Sp	MCC
GRU	0.733	0.728	0.740	0.467
CNN	0.740	0.753	0.730	0.482
CapsuleGAN	0.750	0.753	0.749	0.501
Stacking	0.776	**0.791**	0.762	0.553
Voteing	**0.779**	0.786	**0.773**	**0.558**

Bold values indicate the highest values for each respective indicator

The stacking method uses CapsuleGAN, GRU, and CNN as base classifiers and CNN as meta-classifier to obtain the highest prediction value. The weight ratio of GRU, CNN, and CapsuleGAN in the voting method is 1:7:2

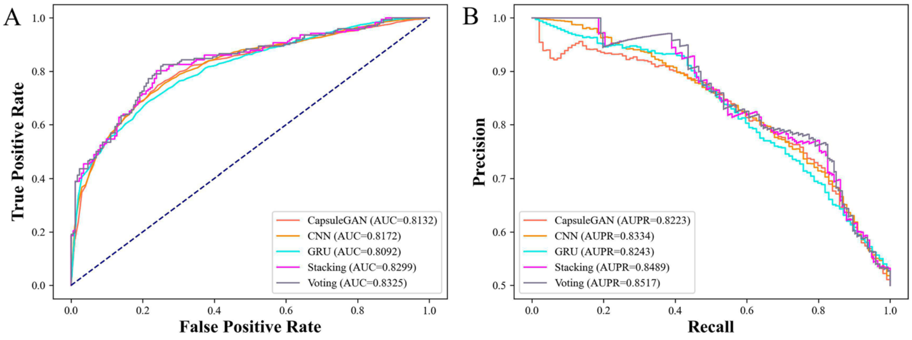

Fig. 6 Model performance on the ACP dataset. **A** ROC curve. **B** PR curve

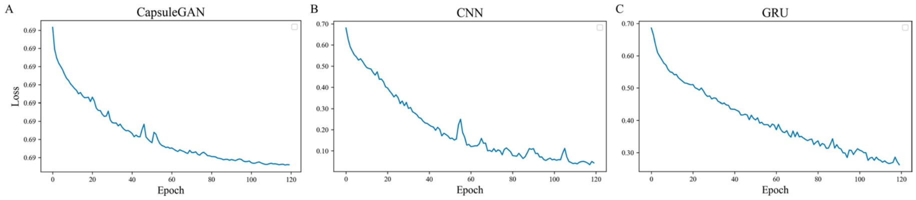

Fig. 7 Loss function of model training on the ACP dataset. **A** Loss function of the CapsuleGAN model. **B** Loss function of the CNN model. **C** Loss function of the GRU model

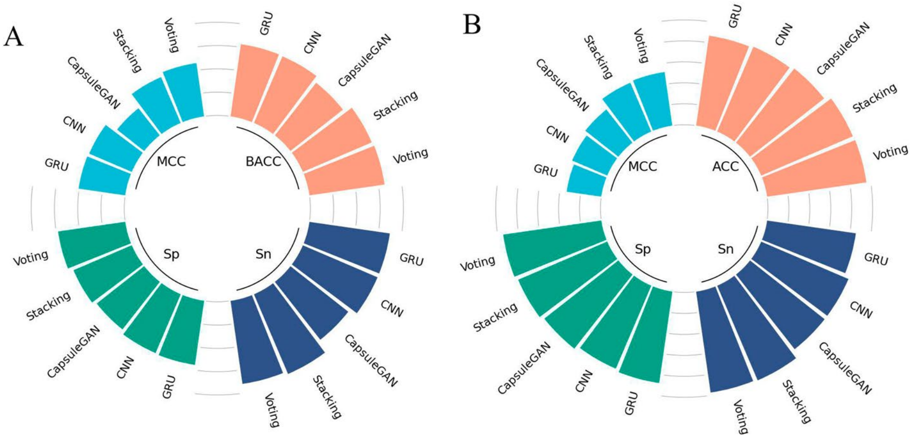

Fig. 8 Model performance analysis. **A** Performance of each model on the ACE inhibitory peptide dataset. **B** Performance of each model on the ACP dataset

curve of the CNN model has an obvious downward trend, which gradually stabilized after 50 iterations. There were slight fluctuations, but the overall trend remained convergent. The GRU model also showed a gradual downward trend, with a lower fluctuation amplitude in the final stage. Overall, the loss curves of the three models showed good convergence, without obvious overfitting or loss value recovery. This shows that the model maintains stability during the training process on the ACP dataset and successfully avoided the overfitting problem caused by the complexity of the model.

Comprehensive analysis

A comprehensive analysis of different methods is conducted on the two data sets. Figure 8 shows the performance of different methods under multiple performance indicators on the ACE inhibitory peptide and ACP datasets, including AUC, BACC, MCC, Sn and Sp. According to the performance comparison analysis, the ensemble learning method shows obvious advantages on both datasets. Whether it is on core performance indicators such as AUC, BACC or MCC, Stacking and Voting are significantly better than a single classifier. In particular, Voting effectively improves the accuracy and robustness of classification by combining the results of multiple classifiers. These results demonstrate that ensemble learning methods have significant application prospects in the functional peptide classification task of ACE inhibitory peptides and ACP datasets. In addition, although GRU and CNN perform better on some indicators as a

Table 4 Comparison with existing methods on ACE inhibitory peptide dataset

Method	BACC	Sn	Sp	MCC	AUC
pLM4ACE (LR)#	0.883	0.845	**0.920**	0.770	0.960
pLM4ACE (SVM)#	0.867	0.825	0.910	0.740	0.955
pLM4ACE (MLP)#	0.855	0.815	0.895	0.711	0.951
DeepBP	**0.926**	**0.960**	0.891	**0.831**	**0.966**

Bold values indicate the highest values for each respective indicator

Indicates that the experimental results come from Ref. [24]

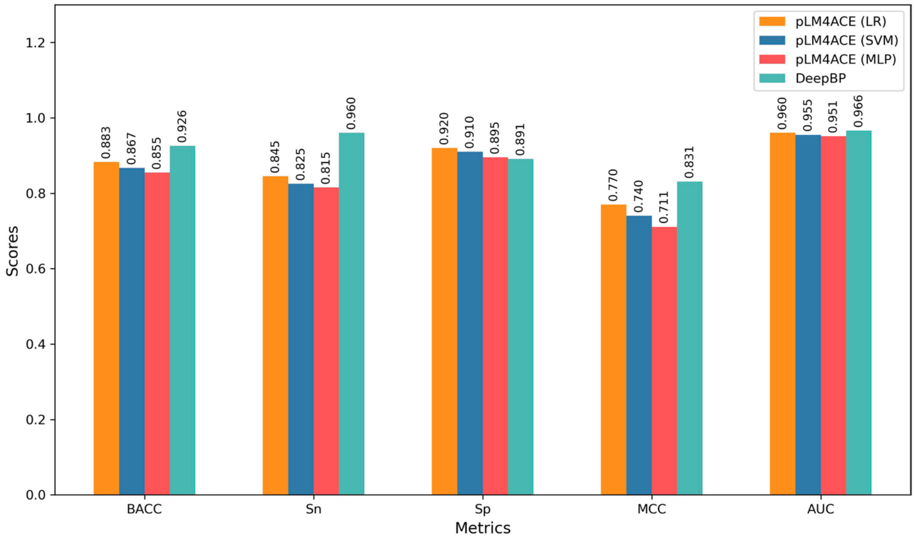

Fig. 9 Comparison with existing methods on ACE inhibitory peptide dataset

single classifier, they still cannot surpass the comprehensive effect of ensemble learning in terms of overall performance.

Evaluate the effectiveness of experimental methods

In order to verify the effectiveness of this experimental method, we collected some existing experimental methods for predicting ACPs and the activity of ACE inhibitory peptides, and compared them with our experimental results.

Performance comparison on ACE inhibitory peptide dataset

The first is the prediction of ACE inhibitory peptide activity. The experimental results are shown in Table 4 and Fig. 9. In Table 4 and Fig. 9, we compare the proposed ensemble learning method (DeepBP) with the existing pLM4ACE method (including three variants of LR, SVM and MLP). The overall results show that DeepBP shows clear advantages on multiple key performance indicators.

The ensemble learning model surpasses existing methods in core indicators such as BACC, Sn, MCC and AUC. The improvement in BACC and Sn indicators is particularly significant, indicating that ensemble learning can handle data imbalance and identify positive samples. Have better performance. Although the LR variant of pLM4ACE has a slight advantage in Sp, DeepBP still shows stronger robustness performance.

Table 5 Comparison with existing methods on the ACP dataset

Method	ACC	Sn	Sp	MCC
AntiCP_2.0#	0.754	0.775	0.734	0.510
AntiCP#	0.506	**1.000**	0.012	0.070
ACPred#	0.535	0.856	0.214	0.090
ACPred-FL#	0.448	0.671	0.225	-0.120
ACPred-Fuse#	0.689	0.692	0.686	0.380
AEPred-Suite#	0.535	0.331	0.738	0.080
iACP#	0.551	0.779	0.322	0.110
RoBERTa#	0.762	0.732	**0.800**	0.528
DeepBP	**0.779**	0.786	0.773	**0.558**

Bold values indicate the highest values for each respective indicator

Indicates that the experimental results come from Ref. [25]

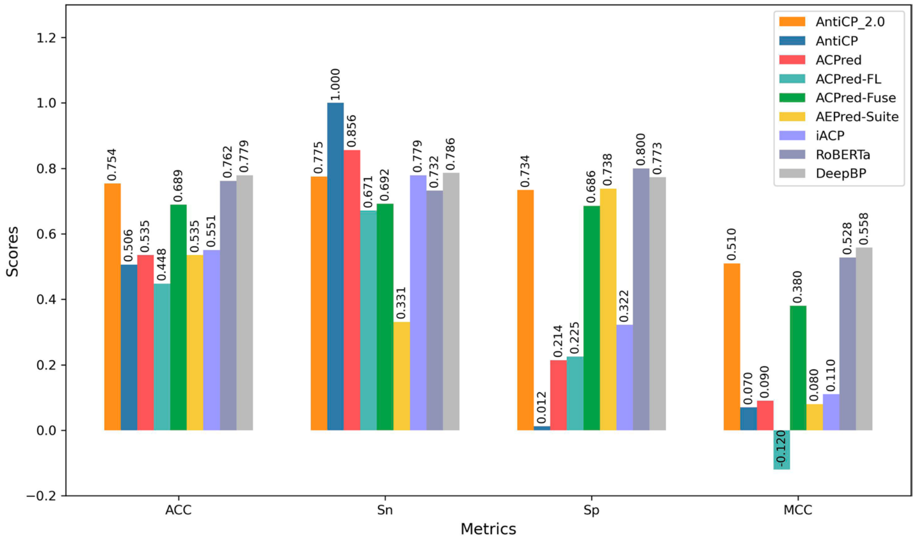

Fig. 10 Comparison with existing methods on the ACP dataset

Performance comparison on ACP dataset

The experimental comparison of the ACP data set is shown in Table 5 and Fig. 10. Table 5 and Fig. 10 show the comparison between DeepBP and various existing algorithms on the ACP dataset. Through these results, we can see that DeepBP shows superiority in multiple indicators, especially in important indicators such as ACC, Sn and MCC, which are superior to existing methods.

On the ACC and Sn metrics, DeepBP achieves scores of 0.779 and 0.786, respectively, which are higher than most existing algorithms. Especially when compared with the better-performing AntiCP_2.0 (ACC = 0.754, Sn = 0.775) and ACP-NLP (ACC = 0.762, Sn = 0.732), DeepBP shows significant performance improvement. This shows that DeepBP has higher accuracy and sensitivity in identifying positive samples, further validating its effectiveness in ACP identification tasks. On the MCC index, DeepBP also achieves outstanding results (MCC = 0.558), which is significantly better than most comparison methods. This result shows that the ensemble learning model has advantages in handling sample imbalance and providing more balanced classification results. Although some existing algorithms (such as AntiCP and AEPred-Suite) have advantages in specific tasks in terms of Sp indicators, DeepBP is more stable in overall performance based on various indicators. DeepBP can ensure high sensitivity while also taking into account classification accuracy and robustness, providing strong support for the prediction of ACP functional peptides. Through this comparison, the advantages of ensemble learning are further verified on the ACP dataset, demonstrating its broad applicability in functional peptide prediction tasks.

Discussion

Bioactive peptides refer to small molecule fragments generated during protein decomposition. They play an important role in regulating physiological functions, enhancing immunity, antibacterial, anticancer and lowering blood pressure. Among them, ACE inhibitory peptides are a class of bioactive peptides that can inhibit the activity of ACE. They mainly block ACE from converting angiotensin I into angiotensin II with a pressor effect in the body, thereby playing a role in lowering blood pressure. ACPs are a class of bioactive peptides that work by inhibiting the proliferation of cancer cells or inducing apoptosis of cancer cells. These peptides usually inhibit tumor growth and spread by destroying cancer cell membranes or interfering with intracellular signal transduction pathways.

The significance of this study is to use modern deep learning and ensemble learning technologies to develop a new algorithm model that can effectively predict peptides. Peptide molecules have broad application potential in the field of biomedicine, but due to their wide variety and complex functions, traditional experimental screening methods are time-consuming and labor-intensive, and difficult to carry out on a large scale. By introducing computational models, especially feature extraction methods based on protein language models (such as ESM-2), this study greatly improved the accuracy and efficiency of prediction. This can not only accelerate the discovery of functional peptides, but also provide important theoretical support and technical guarantees for related drug development and disease treatment. In addition, this study also has certain clinical relevance, mainly reflected in the practical application of peptide function prediction. Through the accurate prediction of ACE inhibitory peptide activity and ACPs, the study provides important support for the development of new drugs. ACE inhibitory peptides have potential application value in the treatment of hypertension, while ACPs may become an auxiliary tool for cancer treatment. Accurately predicting the active functions of these peptides can accelerate the new drug screening and development process, reduce experimental costs, and improve the efficiency of drug development. Therefore, this study has important academic value and application prospects in the fields of bioinformatics and drug development.

Although this study significantly improved the prediction accuracy of functional peptides through ensemble learning and deep learning methods, there are still some limitations. First, the dataset used is relatively limited, especially the sample size of peptides (such as ACE inhibitory peptides and ACPs) may be insufficient, which to some extent affects the generalization ability of the model. The model may show a decrease in prediction performance when dealing with new or under-characterized peptides. Second, the protein language model (such as ESM-2) used in this study is mainly trained based on existing large-scale protein sequence databases, so the performance of the model depends on the quality and coverage of the training data. For some peptides that have not been widely studied or whose functional properties are not yet clear, the model may not be able to effectively extract features or make accurate predictions. In addition, although the ensemble learning method showed good prediction results in this study, its complexity also brings about an increase in computational costs. Large-scale data training and model optimization require high computing resources, which may become a bottleneck in practical applications, especially in resource-constrained environments. Finally, although the model in this study performed well in many performance indicators, its prediction of the biological activity of peptides is still based on sequence information, and fails to fully consider the structural information of peptides and the actual performance in experimental environments. Therefore, future research may need to combine more biological characteristics and experimental data to further improve the reliability and practicality of the model.

Conclusion

This study used an ensemble learning method to systematically analyze and evaluate the prediction of peptides. By combining multiple deep learning models (such as GRU, CNN, CapsuleGAN) and ensemble strategies such as voting, the study verified the effectiveness of ensemble learning in processing functional peptide prediction tasks. Experimental results show that the ensemble learning method has achieved good results in multiple performance indicators. On the ACE inhibitory peptide dataset, the BACC is 0.926, the MCC is 0.831, and the AUC is 0.966. On the ACP dataset, the ACC is 0.779 and the MCC is 0.558. All are better than the existing experimental methods.

In addition, this experiment also made full use of the features extracted by the protein language model-ESM-2 to further improve the prediction ability of the model. By comparing the existing prediction methods, the ensemble model proposed in this study has shown superior performance in terms of accuracy, sensitivity, specificity, etc., proving the practicality and effectiveness of this method in peptide screening.

In general, the ensemble learning method used in this study not only provides an efficient and accurate way to predict functional peptides, but also provides new ideas and methods for future research on peptide molecules in the biomedical field.

Abbreviations

ACE	Angiotensin converting enzyme
ACP	Anticancer peptides
ESM	Evolutionary scale modeling
CNN	Convolutional neural network
GRU	Gated recurrent unit
CapsuleGAN	Generative adversarial capsule network
CBAM	Convolutional block attention module

Acknowledgements

We sincerely thank the anonymous reviewers for their valuable feedback, and we would also like to express our deep gratitude to Watshara Shoombuatong for his contributions during the early stages of the manuscript preparation.

Author contributions

MZ and FG conceived the study, manuscript writing and data analysis. JRZ method development, manuscript writing and data analysis. XW and XHW method development and data analysis. and FG performed final edits and finalized the manuscript. All authors read and approved the final manuscript.

Funding
This work is supported by the Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (Grant No. NY223062).

Availability of data and materials
The dataset supporting the conclusions of this article is available in the GitHub repository, https://github.com/Zhou-Jianren/bioactive-peptides.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 9 April 2024 Accepted: 4 November 2024
Published online: 11 November 2024

References

1. Charoenkwan P, Nantasenamat C, Hasan MM, Moni MA, Manavalan B, Shoombuatong W. StackDPPIV: a novel computational approach for accurate prediction of dipeptidyl peptidase IV (DPP-IV) inhibitory peptides. Methods. 2022;204:189–98.
2. Ge F, Zhu Y-H, Xu J, Muhammad A, Song J, Yu D-J. MutTMPredictor: robust and accurate cascade XGBoost classifier for prediction of mutations in transmembrane proteins. Comput Struct Biotechnol J. 2021;19:6400–16.
3. Ge F, Muhammad A, Yu D-J. DeepnsSNPs: accurate prediction of non-synonymous single-nucleotide polymorphisms by combining multi-scale convolutional neural network and residue environment information. Chemom Intell Lab Syst. 2021;215:104326.
4. Charoenkwan P, Chumnanpuen P, Schaduangrat N, Shoombuatong W. Accelerating the identification of the allergenic potential of plant proteins using a stacked ensemble-learning framework. J Biomol Struct Dyn. 2024. https://doi.org/10.1080/07391102.2024.2318482.
5. Yao L, Zhang Y, Li W, Chung C, Guan J, Zhang W, et al. DeepAFP: an effective computational framework for identifying antifungal peptides based on deep learning. Protein Sci. 2023;32:e4758.
6. Charoenkwan P, Chumnanpuen P, Schaduangrat N, Oh C, Manavalan B, Shoombuatong W. PSRQSP: an effective approach for the interpretable prediction of quorum sensing peptide using propensity score representation learning. Comput Biol Med. 2023;158:106784.
7. Ye J, McGinnis S, Madden TL. BLAST: improvements for better sequence analysis. Nucleic Acids Res. 2006;34:W6–9. https://doi.org/10.1093/nar/gkl164.
8. Salem M, Keshavarzi Arshadi A, Yuan JS. AMPDeep: hemolytic activity prediction of antimicrobial peptides using transfer learning. BMC Bioinformatics. 2022;23:389.
9. Thakur N, Qureshi A, Kumar M. AVPpred: collection and prediction of highly effective antiviral peptides. Nucleic Acids Res. 2012;40:W199-204.
10. Zhang W, Xia E, Dai R, Tang W, Bin Y, Xia J. PredAPP: predicting anti-parasitic peptides with undersampling and ensemble approaches. Interdiscip Sci Comput Life Sci. 2022; 1–11.
11. Manavalan B, Shin TH, Kim MO, Lee G. AIPpred: sequence-based prediction of anti-inflammatory peptides using random forest. Front Pharmacol. 2018;9:276.
12. Shoombuatong W, Schaduangrat N, Pratiwi R, Nantasenamat C. THPep: a machine learning-based approach for predicting tumor homing peptides. Comput Biol Chem. 2019;80:441–51.
13. Dong Y, Chang Y, Wang Y, Han Q, Wen X, Yang Z, et al. MFSynDCP: multi-source feature collaborative interactive learning for drug combination synergy prediction. BMC Bioinform. 2024;25:140.
14. Li C, Zhi D, Wang K, Liu X. MetaRNN: differentiating rare pathogenic and rare benign missense SNVs and InDels using deep learning. Genome Med. 2022;14:115.
15. Ge F, Hu J, Zhu Y-H, Arif M, Yu D-J. TargetMM: accurate missense mutation prediction by utilizing local and global sequence information with classifier ensemble. Comb Chem High Throughput Screen. 2022;25:38–52.
16. Guan J, Yao L, Xie P, Chung C-R, Huang Y, Chiang Y-C, et al. A two-stage computational framework for identifying antiviral peptides and their functional types based on contrastive learning and multi-feature fusion strategy. Brief Bioinform. 2024;25:bbae208.
17. Yao L, Li W, Zhang Y, Deng J, Pang Y, Huang Y, et al. Accelerating the discovery of anticancer peptides through deep forest architecture with deep graphical representation. Int J Mol Sci. 2023;24:4328.
18. Yao L, Guan J, Xie P, Chung C-R, Deng J, Huang Y, et al. AMPActiPred: a three-stage framework for predicting antibacterial peptides and activity levels with deep forest. Protein Sci. 2024;33:e5006.
19. Pang Y, Yao L, Jhong J-H, Wang Z, Lee T-Y. AVPIden: a new scheme for identification and functional prediction of antiviral peptides based on machine learning approaches. Brief Bioinform. 2021;22:bbab263.
20. Yao L, Guan J, Li W, Chung C-R, Deng J, Chiang Y-C, et al. Identifying antitubercular peptides via deep forest architecture with effective feature representation. Anal Chem. 2024;96:1538–46.

21. Guan J, Yao L, Chung C-R, Chiang Y-C, Lee T-Y. Stackthpred: identifying tumor-homing peptides through GBDT-based feature selection with stacking ensemble architecture. Int J Mol Sci. 2023;24:10348.
22. Yan J, Bhadra P, Li A, Sethiya P, Qin L, Tai HK, et al. Deep-AmPEP30: improve short antimicrobial peptides prediction with deep learning. Mol Ther-Nucleic Acids. 2020;20:882–94.
23. Cai L, Wang L, Fu X, Xia C, Zeng X, Zou Q. ITP-Pred: an interpretable method for predicting, therapeutic peptides with fused features low-dimension representation. Brief Bioinform. 2021;22:bbaa367.
24. Du Z, Ding X, Hsu W, Munir A, Xu Y, Li Y. pLM4ACE: a protein language model based predictor for antihypertensive peptide screening. Food Chem. 2024;431:137162.
25. Jiang L, Sun N, Zhang Y, Yu X, Liu X. Bioactive peptide recognition based on NLP pre-train algorithm. IEEE/ACM Trans Comput Biol Bioinf. 2023;20:3809–19.
26. Madani A, McCann B, Naik N, Keskar NS, Anand N, Eguchi RR, et al. Progen: language modeling for protein generation. arXiv preprint arXiv:200403497. 2020;16:1315.
27. Chandra A, Sharma A, Dehzangi I, Tsunoda T, Sattar A. PepCNN deep learning tool for predicting peptide binding residues in proteins using sequence, structural, and language model features. Sci Rep. 2023;13:20882.
28. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. 2022; 2022.07.20.500902.
29. Qi L, Du J, Sun Y, Xiong Y, Zhao X, Pan D, et al. Umami-MRNN: deep learning-based prediction of umami peptide using RNN and MLP. Food Chem. 2023;405:134935.
30. Gui Y-M, Wang R-J, Wang X, Wei Y-Y. Using deep neural networks to improve the performance of protein–protein interactions prediction. Int J Pattern Recognit Artif Intell. 2020;34:2052012.
31. Zhang M, Gong C, Ge F, Dong-Jun Yu. FCMSTrans: accurate prediction of disease-associated nsSNPs by utilizing multiscale convolution and deep feature combination within a transformer framework. J Chem Inf Model. 2024;64(4):1394–406. https://doi.org/10.1021/acs.jcim.3c02025.
32. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. 2018.
33. Jaiswal A, AbdAlmageed W, Yue Wu, Natarajan P. Capsulegan: generative adversarial capsule network. In: Leal-Taixé L, Roth S, editors. Computer vision—ECCV 2018 workshops: Munich, Germany, September 8–14, 2018, Proceedings, Part III. Cham: Springer International Publishing; 2019. p. 526–35.
34. Wang Y, Wang X, Chen C, Gao H, Salhi A, Gao X, et al. RPI-CapsuleGAN: predicting RNA-protein interactions through an interpretable generative adversarial capsule network. Pattern Recogn. 2023;141:109626.
35. Gan Y, Xiang T, Ouyang D, Zhou M, Ye M. SPGAN: siamese projection generative adversarial networks. Knowl-Based Syst. 2024;285:111353.
36. Sun G, Ding S, Sun T, Zhang C. SA-CapsGAN: using capsule networks with embedded self-attention for generative adversarial network. Neurocomputing. 2021;423:399–406.
37. Huang Y, Huang H-Y, Chen Y, Lin Y-C-D, Yao L, Lin T, et al. A robust drug–target interaction prediction framework with capsule network and transfer learning. Int J Mol Sci. 2023;24:14061.
38. Yao L, Xie P, Guan J, Chung C-R, Huang Y, Pang Y, et al. CapsEnhancer: an effective computational framework for identifying enhancers based on chaos game representation and capsule network. J Chem Inf Model. 2024;64:5725–36.
39. Woo S, Park J, Lee J-Y, Kweon IS. Cbam: convolutional block attention module. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, editors. Computer vision—ECCV 2018: 15th European conference, Munich, Germany, September 8–14, 2018, proceedings, Part VII. Cham: Springer International Publishing; 2018. p. 3–19.
40. Zhang N, Ruan J, Duan G, Gao S, Zhang T. The interstrand amino acid pairs play a significant role in determining the parallel or antiparallel orientation of β-strands. Biochem Biophys Res Commun. 2009;386:537–43.
41. Rajasegaran J, Jayasundara V, Jayasekara S, Jayasekara H, Seneviratne S, Rodrigo R. Deepcaps: going deeper with capsule networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019. p. 10725–33.
42. Hahn T, Pyeon M, Kim G. Self-routing capsule networks. Advances in neural information processing systems. 2019; 32.
43. Dey R, Salem FM. Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE; 2017. p. 1597–600.
44. Mahjoub S, Chrifi-Alaoui L, Marhic B, Delahoche L. Predicting energy consumption using LSTM, multi-layer GRU and drop-GRU neural networks. Sensors. 2022;22:4062.
45. Irie K, Tüske Z, Alkhouli T, Schlüter R, Ney H, Others. LSTM, GRU, highway and a bit of attention: an empirical overview for language modeling in speech recognition. In: Interspeech. 2016. p. 3519–23.
46. Chua LO. CNN: a vision of complexity. Int J Bifurc Chaos. 1997;7:2219–425.
47. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J big Data. 2021;8:1–74.
48. Shin H-C, Roth HR, Gao M, Lu L, Xu Z, Nogues I, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging. 2016;35:1285–98.
49. Dong X, Yu Z, Cao W, Shi Y, Ma Q. A survey on ensemble learning. Front Comp Sci. 2020;14:241–58.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ming Zhang received the PhD degree in pattern recognition and machine intelligence from the Nanjing University of Science and Technology, in 2013. He is currently a full professor in the School of Computer, Jiangsu University of Science and Technology. His research interests include granular computing, machine learning, pattern recognition, and bioinformatics.

Jianren Zhou received his B.S. degree from Anyang University. He is currently pursuing a M.S. degree in Electronic Information at Jiangsu University of Science and Technology, with an expected graduation in 2025. His reserch interests include bioinformatics and deep learning.

Xiaohua Wang received his B.S. degree from Qingdao University of Science and Technology. He is currently pursuing a M.S. degree in Electronic Information at Jiangsu University of Science and Technology, with an expected graduation in 2025. His reserch interests include bioinformatics and deep learning.

Xun Wang is an associate professor in the School of Computer, Jiangsu University of Science and Technology. Her research interests include machine learningdata mining, and big data processing.

Fang Ge received her Ph.D. degree from Nanjing University of Science and Technology. She is currently an associate professor in Nanjing University of Posts and Telecommunications. Her research interests include bioinformatics, pattern recognition, and data mining.