OXFORD

Briefings in Bioinformatics, 2023, **24(1)**, 1–10
https://doi.org/10.1093/bib/bbac630
Problem solving protocol

Prediction of anticancer peptides based on an ensemble model of deep learning and machine learning using ordinal positional encoding

Qitong Yuan, Keyi Chen, Yimin Yu, Nguyen Quoc Khanh Le and Matthew Chin Heng Chua

Corresponding author. Nguyen Quoc Khanh Le, Professional Master Program in Artificial Intelligence in Medicine, College of Medicine, Taipei Medical University, Taipei, Taiwan; Research Center for Artificial Intelligence in Medicine, Taipei Medical University, Taipei, Taiwan. Tel: +886-02-66382736; #1992; Fax: +886-02-27321956; E-mail: khanhlee@tmu.edu.tw

Abstract

Anticancer peptides (ACPs) are the types of peptides that have been demonstrated to have anticancer activities. Using ACPs to prevent cancer could be a viable alternative to conventional cancer treatments because they are safer and display higher selectivity. Due to ACP identification being highly lab-limited, expensive and lengthy, a computational method is proposed to predict ACPs from sequence information in this study. The process includes the input of the peptide sequences, feature extraction in terms of ordinal encoding with positional information and handcrafted features, and finally feature selection. The whole model comprises of two modules, including deep learning and machine learning algorithms. The deep learning module contained two channels: bidirectional long short-term memory (BiLSTM) and convolutional neural network (CNN). Light Gradient Boosting Machine (LightGBM) was used in the machine learning module. Finally, this study voted the three models' classification results for the three paths resulting in the model ensemble layer. This study provides insights into ACP prediction utilizing a novel method and presented a promising performance. It used a benchmark dataset for further exploration and improvement compared with previous studies. Our final model has an accuracy of 0.7895, sensitivity of 0.8153 and specificity of 0.7676, and it was increased by at least 2% compared with the state-of-the-art studies in all metrics. Hence, this paper presents a novel method that can potentially predict ACPs more effectively and efficiently. The work and source codes are made available to the community of researchers and developers at https://github.com/khanhlee/acp-ope/.

Keywords: Anticancer peptide, Ordinal positional encoding, Handcrafted feature, Feature fusion, Model ensemble, Deep learning, Machine learning

Introduction

Cancer is a deadly disease that kills millions of people worldwide each year. Cancer treatment is a severe medical problem that humanity faces [1]. The most common cancer treatments in recent years are radiotherapy, chemotherapy and targeted therapy [2]. These medicines are intended to kill cancer cells, but they also destroy normal cells. These procedures have apparent adverse effects and, in the meantime, are out of reach for many people [3].

Therefore, the research direction started to turn to anticancer peptides (ACPs) [4]. ACPs have several advantages over many other cancer treatments. For instance, they seem safer since they are natural biological targets. Furthermore, they kill cancer cells with better selectivity due to their inherent anion properties, which preferentially connect with the anionic biological membrane parts of the cancer cell [5].

ACP therapy has been widely researched and implemented at various clinical phases over the years, but only a limited amount has been utilized in clinical treatment. Nonetheless, computational prediction approaches are becoming increasingly relevant in the screening, discovery and prediction of ACPs. Researchers have been concentrating on developing a faster and less expensive method for discovering and identifying new ACPs [1].

Plenty of articles focused on ACP prediction over the past decade. For instance, Tyagi *et al.* [6] created AntiCP, an indicator related to support vector machine (SVM) to predict ACPs. Hajisharifi *et al.* also created an SVM-based model utilizing Chou's pseudo amino acid composition (PseAAC) and local alignment kernel [7–9], leading to the high precision of anticipating ACPs [10]. Vijayakumar and Lakshmi [11] developed ACPP, a prediction technique based on compositional information centroidal and distributional metrics of amino acids to predict ACPs more precisely. With the help of the dipeptide combination, Chen *et al.* [12] proposed a new

Qitong Yuan is a student with the Institute of Systems Science, National University of Singapore, Singapore. His research interests are artificial intelligence, intelligent system and bioinformatics.
Keyi Chen is a student with the Institute of Systems Science, National University of Singapore, Singapore. His research interests are artificial intelligence, intelligent system and bioinformatics.
Yimin Yu is a student with the Institute of Systems Science, National University of Singapore, Singapore. His research interests are artificial intelligence, intelligent system and bioinformatics.
Nguyen Quoc Khanh Le is an assistant professor with the Professional Master Program in Artificial Intelligence in Medicine, College of Medicine, Taipei Medical University, Taiwan. His research interests are artificial intelligence, bioinformatics and radiomics.
Matthew Chin Heng Chua is an Assistant Director of Data Science at the Ministry of Defence of Singapore and an Assistant Professor at National University of Singapore (NUS) School of Medicine. His research interests are artificial intelligence, intelligent system and bioinformatics.
Received: August 3, 2022. **Revised:** December 1, 2022. **Accepted:** December 28, 2022

predicting method called iACP. To represent ACPs, Li and Wang [13] used the AAC, average chemical shifts and decreased AAC. Rao *et al.* [14] presented ACPred-Fuse, a predictor that increased feature ability by merging multi-view data. Deep neural networks (DNNs) have shown promising results in bioinformatics in recent years [15–17], and they have been also used to predict ACP. For instance, Yu *et al.* [18] examined three different deep learning architectures and found that the bidirectional long short-term memory (BiLSTM) performed well in ACP prediction [18]. A novel DNN architecture was employed by Ahmed *et al.* [19] that learned and combined three different features utilizing parallel convolution groups.

After screening, we found four latest articles that are more targeted and constructive to our research content. Regarding a benchmark paper, Lv *et al.* [1] proposed an ACP predictor using deep representation learning features. This study used two types of sequence embedding strategies, including soft symmetric alignment and unified representation (UniRep) embeddings. In addition, Cao *et al.* [20] proposed an efficient algorithm using a dual-channel ensemble learning algorithm on multi-modal sequence features. The first channel used a convolutional neural network (CNN) architecture, and the second aimed to extract optimal handcraft features. According to Chen *et al.* [21], peptide sequences are depicted by combining binary profile and biochemical properties features, before augmenting the samples in the higher dimensional space. Finally, Yi *et al.* [22] presented an effective feature extraction strategy to fully utilize peptide sequence information by assimilating binary profile and K-mer sparse matrix of the reduced amino acid letters.

According to the latest article [1], the current results reached an accuracy of 77.5% on a benchmark dataset. Then we aim to propose a novel model that performs better than that. The main contributions of this work are summarized as follows: (1) Proposing a novel method for protein sequence encoding, which may be a general solution for similar problems related to protein learning; (2) Proposing an ensemble architecture of machine learning and deep learning to excellently capture features from protein sequences and (3) Proposing a novel model that achieves a better performance than the latest predictor [1].

Materials and methods

Dataset

We used the same dataset as in the benchmark paper [1] for subsequent comparison convenience. The dataset contains 1718 data (including 859 ACPs and 859 non-ACPs) which were experimentally validated. Among the data, the ACPs were retrieved from CancerPPD database [23], and the negative samples were from the antimicrobial peptides. The data were divided into two subsets for training (conducting cross-validation) and independent datasets. They can be freely accessed and downloaded at AntiCP 2.0 server (https://webs.iiitd.edu.in/raghava/anticp2/).

Peptide sequence encoding

Ordinal encoding

We divided amino acids into 21 categories and encoded them like those in the benchmark paper [1]. Since the longest sequence in the data set is 50, if the sequence length is less than 50, we fill it with 0.

Ordinal positional encoding

We designed another encoding method adding positional information. Each amino acid is encoded according to its species and position in the sequence (Figure 1). For example, a sequence *ACD* is converted to vector form [[1], [52], [103], [0], [0], ..., [0]]. If the sequence length is less than 50, we fill it with zero values. In this way, we put the positional information of the amino acids in the encoding to some extent. Using this idea, we trained the model using not only amino acid encoding but also amino acid position in the sequence. This encoding method can significantly improve the performance of the CNN model. Our research found that the encoding method with positional information could improve the cross-validation accuracy.

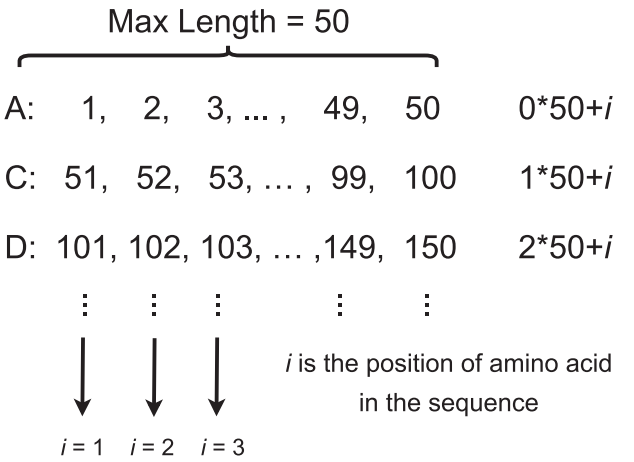

Figure 1. Ordinal positional encoding. Each amino acid is encoded by 50 values according to its position in sequence length of 50.

In addition, we added some widely used features of amino acid sequences to increase the input information: AAC, dipeptide composition (DPC), the composition of k-spaced amino acid group pairs (CKSAAGP) and k-mer sparse matrix. The features were described by Cao *et al.* [20] and Yi *et al.* [22] in their papers as below:

Amino acid composition

AAC represents the normalized frequency of 20 standard amino acids present in the protein sequence, which generates 20-dimensional feature vectors. It can be calculated as follows:

$$f(a) = \frac{N(a)}{L}, a \in \{A, C, D, ..., Y\}, \quad (1)$$

where $f(a)$ denotes the frequency of the occurrence of amino acid a, $N(a)$ denotes the total number of amino acids a appearing in the peptide sequences and L denotes the length of the peptide sequences.

Dipeptide composition

DPC generates 400-dimensional feature vector, which can be calculated as follows:

$$D(r, s) = \frac{N_{rs}}{L - 1}, r, s \in \{A, C, D, ..., Y\}, \quad (2)$$

where N_{rs} is the number of dipeptides represented by amino acid types r and s.

Composition of K-Spaced amino acid group pairs

For this feature set, all amino acids can be divided into five classes based on their physicochemical properties. The CKSAAGP is used to calculate the frequency of amino acid group pairs separated by any k residues. Using $k = 0$ as an example, there are 25 0-spaced

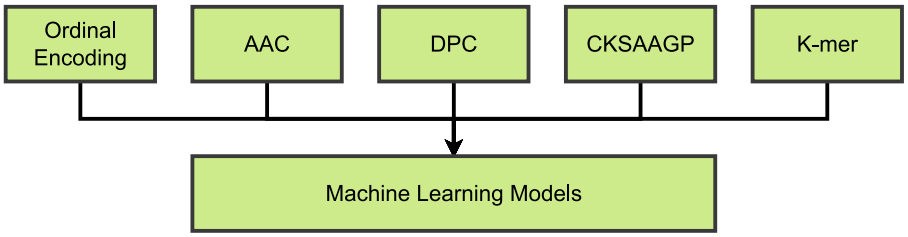

Figure 2. Flowchart of machine learning models. All features (including ordinal encoding and handcraft features) are inserted into machine learning algorithms directly.

group pairs (i.e., G1G1, G1G2, G1G3,..., G5G5). The features can be defined as

$$\left(\frac{N_{G1G1}}{N}, \frac{N_{G1G2}}{N}, \frac{N_{G1G3}}{N}, \ldots, \frac{N_{G5G5}}{N}\right)_{25} \tag{3}$$

Each value denotes the number of times the residual group pair appears in the peptide sequence. For a peptide sequence with length L, when $k = 0, 1, 2, 3, 4, 5$, the corresponding values of N are L-1, L-2, L-3, L-4, L-5 and L-6, respectively.

K-mer sparse matrix

In this feature, $k-1$ consecutive nucleotides and k consecutive nucleotides are regarded as a unit. The amino acids were reduced into seven groups based on their dipole moments and side chain volume. Then each peptide sequence was scanned from left to right, stepping one amino acid at a time, which is considered the characteristics of each amino acid. This study set the value of k to 3 to process the peptide sequence. The k-mer sparse matrix M can be defined as follows:

$$M = (a_{ij})_{7k} \times (L - K + 1) \tag{4}$$

$$a_{ij} = \begin{cases} 1, & \text{if } m_j m_{j+1} m_{j+2} = k - mer(\mathrm{i}) \\ 0, & \text{else} \end{cases} \tag{5}$$

Feature selection

Feature selection can eliminate irrelevant or redundant features to prevent overfitting and improve model accuracy [24, 25]. Since the total number of handcrafted features is more than 700, we used Random Forest (RF) and Light Gradient Boosting Machine (LGBM) to select important features for the deep learning model. Taking the handcrafted features and labels as input, the built-in functions of Random Forest and LGBM can obtain the importance of the features. Then we ranked the features in order of their importance. Through experimenting using different numbers of features, we find it performs best when selecting about 150 features. If the CNN architecture was used, Random Forest helped to establish better features than LGBM. For BiLSTM and recurrent neural network (RNN), LGBM selected better ones. Finally, we decided to use both methods and directly applied them based on their performance on each architecture.

Modeling

We used both machine learning and deep learning models [26, 27] to learn features extracted from peptide sequences. All models were implemented using Python programming language on a workstation with Intel core i9-10900X CPU and NVIDIA GeForce RTX 3080 GPU. *scikit-learn* package (https://scikit-learn.org/) was employed to implement machine learning algorithms and *tensorflow* package (https://www.tensorflow.org/) was employed for deep learning models. To support replication, we also released our model implementation on GitHub (https://github.com/khanhlee/acp-ope/). The detailed modeling architecture is shown as follows.

Machine learning

We used SVM, RF, XGBoost and LightGBM models as supervised machine learning classification algorithms. Among them, SVM is a classification algorithm to classify data based on their representation on hyperplane. The other three are based on the ensemble learning approach. We concatenated ordinal encoding and other handcrafted features together as input to the machine learning model. All machine learning models undergo hyperparameter tuning to find optimal parameters. The machine learning flowchart is shown in Figure 2.

Deep learning

This study proposes a deep learning architecture based on a dual-channel idea. The architecture was a modified version from the study of Cao *et al.* [20], where they developed a DNN-based ACP predictor by fusing multi-view features based on dual-channel ensemble learning. There were two channels here: the first was a CNN to extract the deep features of the sequence, and the second was to learn the generated features. Our CNN architecture was built using a 1D CNN layer (256 filters and kernel size of 2) followed by a max-pooling layer (pooling size of 2). In addition to the CNN architecture, we evaluated the performance of different deep learning architectures, i.e. RNN, Bi-LSTM, and transformers. Moreover, the attention layer was added after the neural network layers to see its potential to improve performance. The flowchart can be seen in Figure 3. First, we input the ordinal encoding into the deep learning layers, then concatenated to the other handcraft features via a dense layer. At the final stage, another classifier was added to learn the abovementioned features. We also compared the model with and without sequence data augmentation. All deep learning models were run to 25 epochs.

Ensemble model

We ensemble the better performing models to improve the predictive performance. In general, the final model comprises of two modules: deep learning and machine learning. After several experiments, the best deep learning module contained two channels: Bi-LSTM channel and CNN channel. Thus, the flowchart has three paths in total. The ensemble model structure is shown in Figure 4.

In the first path, the peptide sequence was inputted into the ordinal encoding layer, then flowed into the Bi-LSTM channel. After the embedding layer and Bi-LSTM layer, the output was

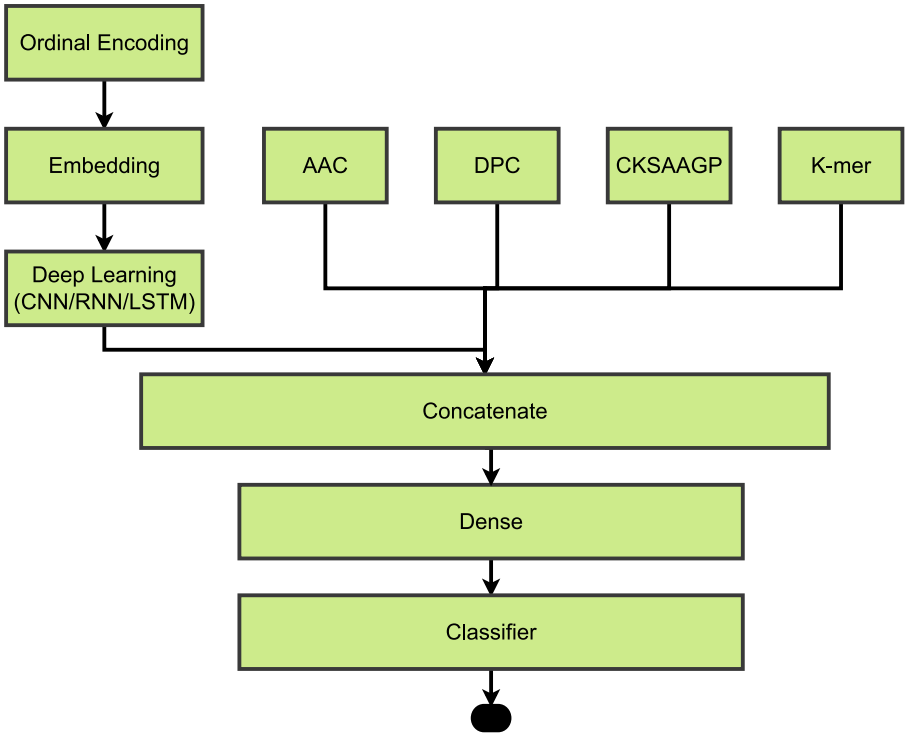

Figure 3. Flowchart of deep learning models. The ordinal encoding feature is inserted into deep learning algorithms and then concatenated with other handcrafted features. A final classifier is added in this step to learn all features and generate the prediction outcomes.

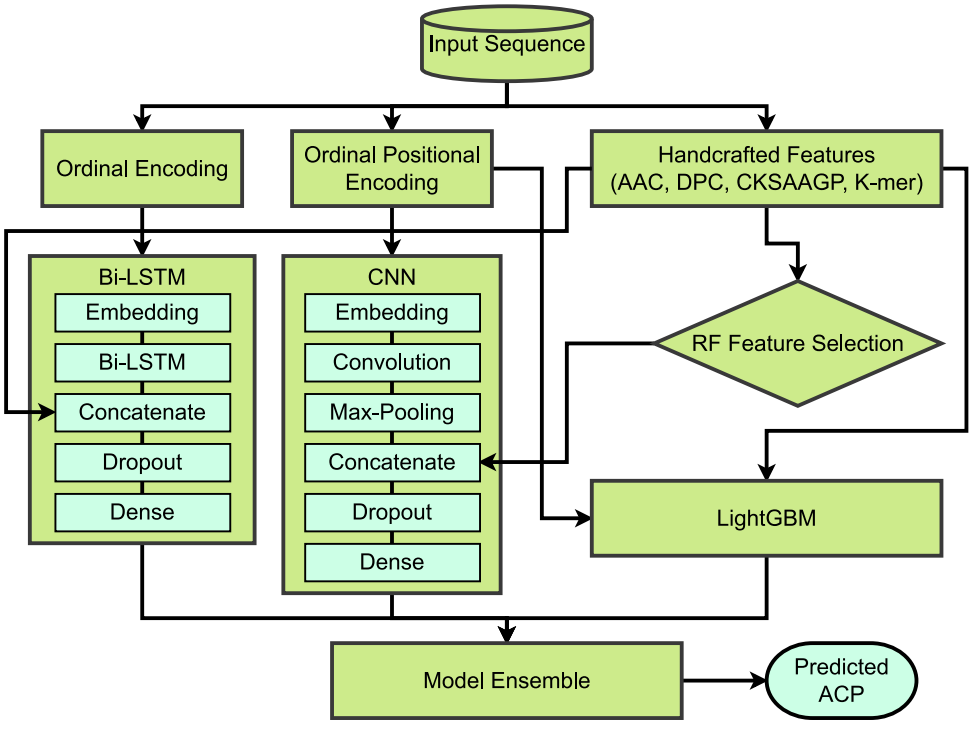

Figure 4. Flowchart of ensemble model. It is a model that combines Bi-LSTM, CNN and LightGBM together to achieve the best prediction outcomes of ACPs.

concatenated with handcrafted features and inputted into the dropout layer and dense layer in sequence.

In the second path, there were two main differences. First, another encoding method that contained positional information was used to make up for the shortage of the CNN model that it did not remember sequential relation. Second, we selected the handcrafted features before being input into the CNN channel.

In the third path, the sequence data were inputted into the feature extraction layer and generated 745 dimensions of handcrafted features. Then, these features were concatenated with ordinal encoding features before inputting into the LightGBM model. Finally, we voted these three models' classification results with the same weights for a final result in the model ensemble layer.

Table 1. Cross-validation performance of different machine learning models on all features

Algorithm	Sens	Spec	Accuracy
SVM	0.6775	0.6776	0.6775
RF	0.7246	0.7251	0.7246
XGBoost	0.7500	0.7501	0.7500
LightGBM	0.7790	0.7797	0.7790

Results

Evaluation methods and measurement metrics

We used a 5-fold cross-validation technique similar to the benchmark paper to ensure consistency [1]. After evaluating and selecting the final model, we tested it on the independent test set. To compare the results, we also adopted similar evaluation metrics used in the benchmark paper [1]: accuracy (ACC), sensitivity (Sens), specificity (Spec) and area under the receiver operating characteristic curves (AUC). Lv *et al.* described these metrics in their paper as shown in Equation 6–8. True positive sample number (TP), true negative sample number (FN), false positive sample number (FP) and false negative sample number (FN) were used to compute the metrics [25, 28].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Sens = \frac{TP}{TP + FN} \tag{7}$$

$$Spec = \frac{TN}{TN + FN} \tag{8}$$

Performance results of machine learning models

Based on the model assumptions of machine learning models, four classifiers (including SVM, RF, XGBoost and LightGBM) were chosen to train and compare. As shown in Table 1, with all 745 features set as input, LightGBM and XGBoost models performed better with accuracy scores of 0.7790 and 0.7500. RF, XGBoost and LightGBM models have a certain degree of overfitting. To address this problem, we have tried to adjust the number of estimators and control the depth of trees. Adding regularization term also helps reduce the overfitting problem. By optimizing the parameters, training accuracies have been reduced by 3–5%.

A more detailed comparison between XGBoost and LightGBM was made for further discussion. Figure 5 shows the performance of these two models with different features as input, where some findings could be obtained. First, LightGBM models have higher accuracy scores than XGBoost models in all seven groups. Second, the model with only ordinal encoding features already had an accuracy of over 0.71, which is not bad. All models with a distinct handcrafted feature (ACC, DPC, CKSAAGP and K-mer) have a significant accuracy of around 0.67. Among them, the LightGBM model with CKSAAGP has the highest accuracy of 0.7464. Meanwhile, the model performs better with all features involved as input than with any single feature. Thus, we can conclude that all four kinds of handcrafted features and ordinal encoding features provided valuable information for classification.

Performance results of deep learning models

Convolutional neural network

For deep learning models, different CNN and RNN models were built and compared using 5-folds cross validation evaluations. As shown in Table 2, the CNN model with 745 features as auxiliary input has a relatively good accuracy (0.7515) on the validation set. We tried to add an attention layer and use a data augmentation method to optimize this model. Combining CNN with SVM or Bi-LSTM was also considered. However, complex models have not achieved as good results as expected.

Considering that training accuracy was a bit high, we used RF and LGBM methods to perform feature selection to prevent overfitting. After getting the importance score of all features, we applied feature increment strategy to determine how many features to keep [1]. Specifically, 300 models were constructed using the top 1,2...300 features. Then we compared the accuracy of all these models and found the best one. As a result, the model with 148 features selected by the RF method and the model with 149 features determined by the LGBM method have relatively higher performances. Furthermore, the RF method helped improve the model performance to the greatest extent, which reduced the training accuracy by 0.33% and increased the validation accuracy by 0.65%.

Figure 6 shows this model's change in training and validation results in 25 epochs. The dark lines represent the average of 5-fold

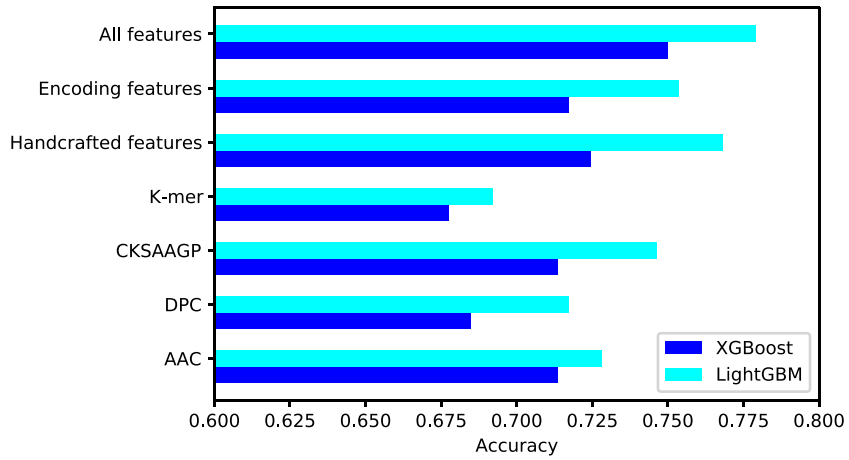

Figure 5. Cross-validation accuracy of XGBoost and LightGBM with different features.

Table 2. Cross-validation results of CNN models

Model	Features	Sens	Spec	Accuracy
CNN	AAC+DPC+CKSAAGP+Kmer	0.7522	0.7152	0.7339
CNN+DA	AAC+DPC+CKSAAGP+Kmer	0.6425	0.7982	0.7200
CNN+SVM	AAC+DPC+CKSAAGP+Kmer	0.7251	0.7650	0.7448
CNN+SVM+DA	AAC+DPC+CKSAAGP+Kmer	0.7155	0.7671	0.7404
CNN	AAC+DPC+CKSAAGP+Kmer+RF Feature selection (148)	0.7479	0.7630	0.7580
CNN	AAC+DPC+CKSAAGP+Kmer+LGBM Feature Selection (149)	0.6662	0.8015	0.7346
CNN+Bi-LSTM	AAC+DPC+CKSAAGP+Kmer	0.6646	0.8233	0.7447
CNN+Bi-LSTM+Attention	AAC+DPC+CKSAAGP+Kmer+RF Feature Selection (134)	0.5985	0.7731	0.6844

DA: data augmentation.

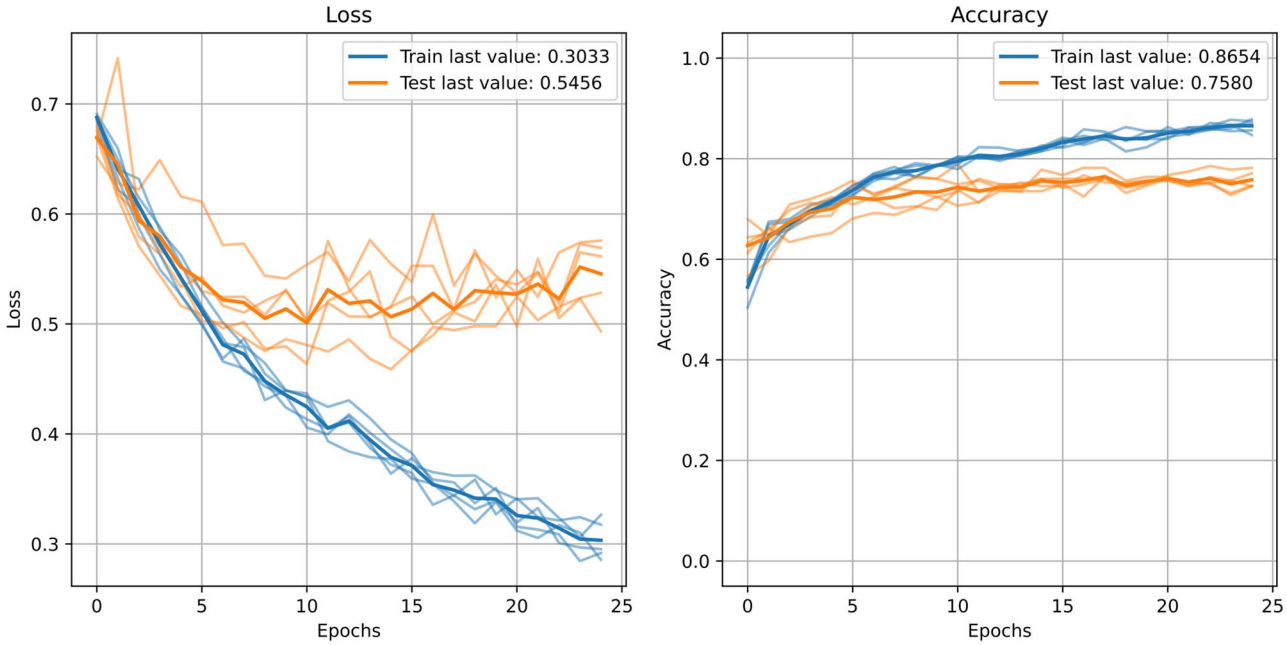

Figure 6. Training and validation results for CNN model with 148 RF selected features.

cross-validation results. After feature selection, there still exists a gap between training and validation accuracy. However, the differences between the training accuracy (0.8654) and validation accuracy (0.7580) were still acceptable at 25 epochs. This shows the feature selection method does control the overfitting problem to some degree.

Recurrent neural network

Comparison work of RNN models was started by trying models with no features appended (e.g. simple RNN, BRNN, DBRNN, GRU, BIGRU, LSTM and Bi-LSTM). It showed that the DBRNN and Bi-LSTM models had higher accuracies of over 0.71, so we mainly made further exploration into these two models.

Table 3 shows the accuracy scores of improved DBRNN models. With 150 features from the LGBM features selection layer, the validation accuracy increases to 0.7304, which is about 3% higher than the RF-selected features and attention layer model. The training and validation results of this model in 25 epochs are also shown in Figure 7.

Bidirectional long short-term memory

By adding 745 features as an auxiliary input, the accuracy of the Bi-LSTM model on 5-fold validation sets has achieved 0.7551. Like the previous models, data augmentation and attention layer are unsuitable for this dataset. Then, as shown in Table 4, RF, LGBM and PCA methods were used to perform feature selection. It is observed that with 150 selected features, each method helps reduce training accuracy by over 6%, but validation accuracy declines simultaneously. Compared with the 300 selected features, the accuracy reduction of both training and validation sets are relatively smaller. Considering the reduction of validation accuracy, 745 features eventually remained in the Bi-LSTM model. Figure 8 shows the model's training and validation results in 25 epochs.

From the results of deep learning models, some conclusions were arrived at. First, feature selection is an excellent way to prevent overfitting and improve model performance, especially for CNN, but it sometimes reduces validation accuracies in RNN and Bi-LSTM models. Second, data augmentation does not work well in this case. The augmented data are highly similar to the original data, which may cause the model to overlearn the features of the training set. Thus, almost all the model results witness an increase in training accuracy and a decline in validation accuracy. Finally, due to the size and characteristics of the dataset, compared with complex models, like CNN + Bi-LSTM, CNN + SVM, or models with attention layers, simple models with useful auxiliary features are more likely to achieve better performance.

Performance results of ensemble model

Among these machine learning models and deep learning models, the top five models with higher cross-validation performance

Table 3. Cross-validation results of DBRNN models

Model	Features	Sens	Spec	Accuracy
DBRNN	AAC+DPC+CKSAAGP+kmer	0.7662	0.7022	0.7339
DBRNN	AAC+DPC+CKSAAGP+K-mer+RF Feature Selection (150)	0.7196	0.7612	0.7397
DBRNN	AAC+DPC+CKSAAGP+K-mer+LGBM Feature Selection (150)	0.7328	0.7534	0.7420
DBRNN+Attention	-	0.6419	0.7864	0.7121
DBRNN+Attention	AAC+DPC+CKSAAGP+K-mer	0.6870	0.7500	0.7179
DBRNN+DA	AAC+DPC+CKSAAGP+K-mer	0.6465	0.7761	0.7106

DA: data augmentation

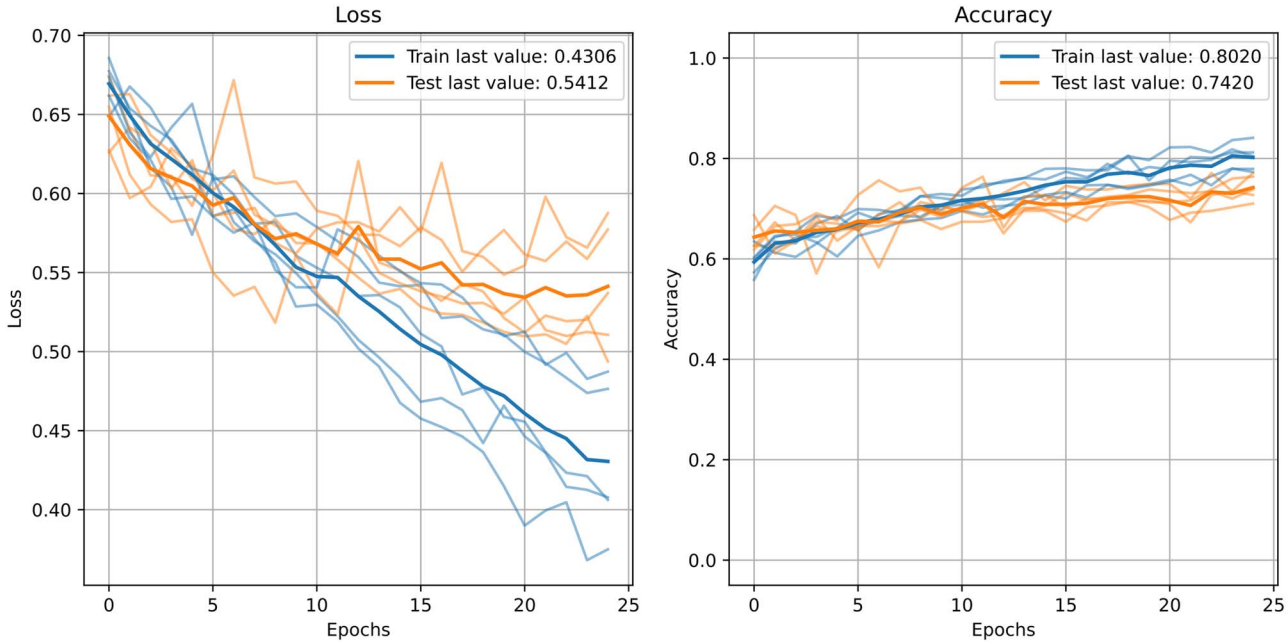

Figure 7. Training and validation results of DBRNN model with 150 LGBM selected features.

Table 4. Cross-validation results of Bi-LSTM models

Model	Features	Sens	Spec	Accuracy
Bi-LSTM	AAC+DPC+CKSAAGP+K-mer	0.7676	0.7523	0.7595
Bi-LSTM	AAC+DPC+CKSAAGP+K-mer+RF Feature Selection (150)	0.7553	0.7358	0.7464
Bi-LSTM	AAC+DPC+CKSAAGP+K-mer+LGBM Feature Selection (150)	0.7409	0.7420	0.7420
Bi-LSTM+DA	AAC+DPC+CKSAAGP+K-mer+LGBM Feature Selection (150)	0.7781	0.7353	0.7573
Bi-LSTM	AAC+DPC+CKSAAGP+K-mer+LGBM Feature Selection (300)	0.7583	0.7482	0.7537
Bi-LSTM	AAC+DPC+CKSAAGP+K-mer+PCA (150)	0.8015	0.6455	0.7231
Bi-LSTM+DA	AAC+DPC+CKSAAGP+K-mer	0.7224	0.7635	0.7428
Bi-LSTM+Attention	-	0.7021	0.7481	0.7260
Bi-LSTM+Attention	AAC+DPC+CKSAAGP+K-mer	0.6994	0.7842	0.7427

DA: data augmentation

are shown in Table 5. After optimizing the hyperparameters, this study tested each model in the independent test set. We also included another model using transformer architecture [29] on peptide sequences for comparison. Subsequently, it was noticed that LightGBM, CNN and Bi-LSTM have a higher accuracy on both cross-validation and testing sets, so this study chose these three models to do the further model ensemble.

Finally, we selected the three best models (CNN, Bi-LSTM and LightGBM) to vote for a better result. The final ensemble model then had a sensitivity of 0.8153, specificity of 0.7676 and accuracy of 0.7895, which increased by about 0.3% compared with our best separate model. Figure 9 illustrates the ROC curves of the three best individual models and the final ensemble model. It could be observed that AUC increases to 0.876 by over 0.007 after the model ensemble.

As shown in the literature review, there were some previous predictors for ACP prediction, such as iACP [12], PEPred-Suite [30], ACPpred-Fuse [14], ACPred-FL [31], ACPred [32], AntiCP [6], AntiCP_2.0 [33] and iACP-DRLF [1]. Therefore, to highlight the efficiency of our model, we compared our performance with previous predictors on the same dataset (Table 6). Considering the latest predictor (iACP-DRLF) [1] as the benchmark result (accuracy of 0.7749), our work increased the accuracy by 1.46%. Also, both sensitivity and specificity increased as well. It is also observed that our predictor outperformed the other predictors in most measurement metrics.

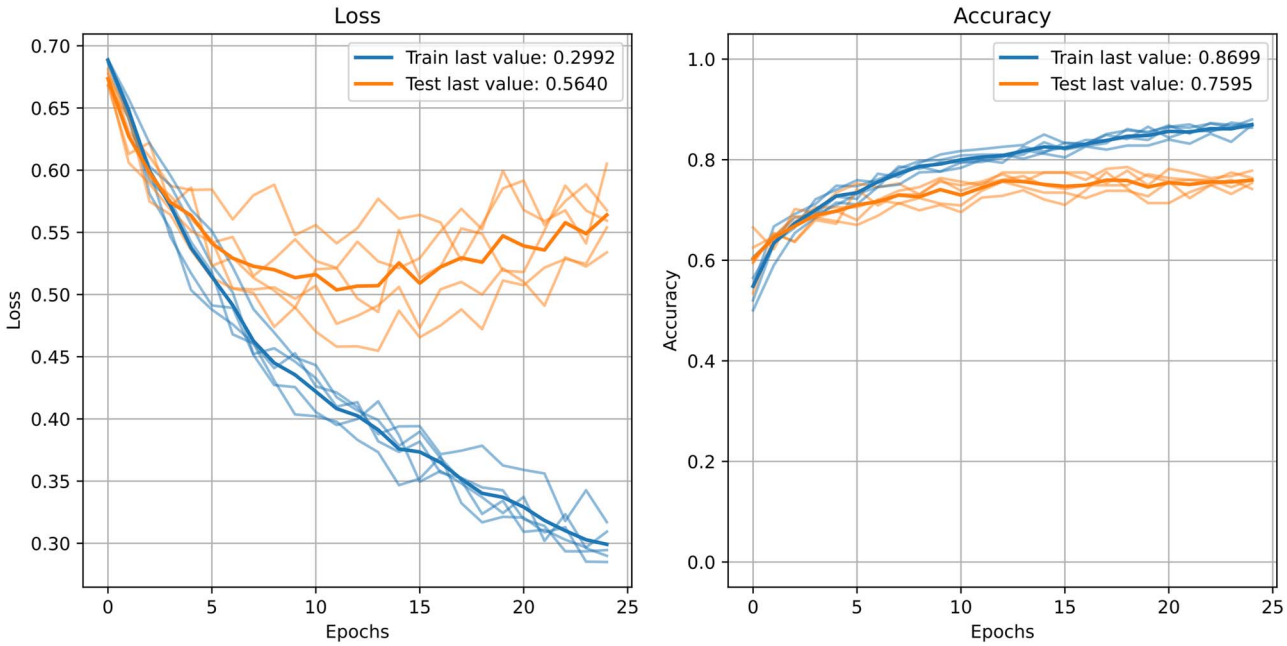

Figure 8. Training and validation results of Bi-LSTM Model with 745 features.

Table 5. Performance results of the best models

Model	Features	Cross-validation accuracy	Testing accuracy
XGBoost	AAC+DPC+CKSAAGP+K-mer	0.7500	0.7661
LightGBM	PC+CKSAAGP+K-mer	0.7826	0.7865
CNN	AAC+DPC+CKSAAGP+K-mer+RF Feature Selection (148)	0.7558	0.7719
DBRNN	AAC+DPC+CKSAAGP+K-mer+LGBM Feature Selection (150)	0.7420	0.7579
Bi-LSTM	AAC+DPC+CKSAAGP+K-mer	0.7551	0.7836
Transformers	-	0.7427	0.7449

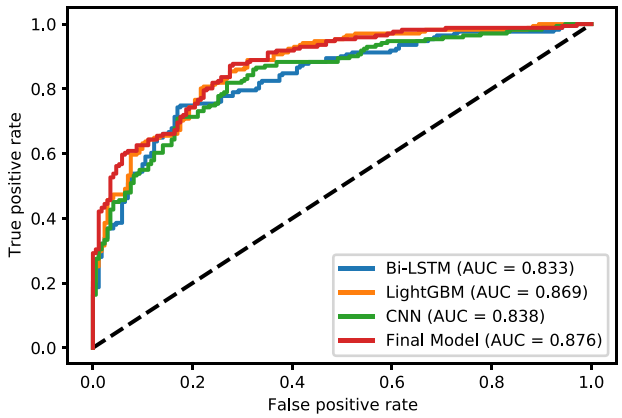

Figure 9. ROC curves of the best individual models and ensemble model.

Table 6. Comparison between our final model and previous predictors on the same independent dataset

Model	Sens	Spec	Accuracy
ACP-OPE*	0.8153	0.7676	0.7895
iACP-DRLF	0.8070	0.7427	0.7749
AntiCP_2.0	0.775	0.734	0.754
AntiCP	1	0.12	0.506
ACPred	0.856	0.214	0.535
ACPred-FL	0.671	0.225	0.448
ACPpred-Fuse	0.692	0.686	0.689
PEPred-Suite	0.331	0.738	0.535
iACP	0.779	0.332	0.551

* Our final model.

Discussion

Most studies in ACP prediction used one-hot or ordinal encoding methods to encode the peptide sequences. For instance, Rao *et al.* [34] was the first to apply graph convolutional networks in ACPs prediction [34]. They used one-hot encoding in the embedding layer. Cao *et al.* [20] used ordinal encoding in the dual-channel DNN. However, the CNN model will naturally lose some positional information in the max pooling layer, which may cause overall underfitting of the model [35, 36], considering that the positional information of peptides sequence is critical when identifying ACPs. To solve this problem, we tried to use a new encoding method containing the positional information to embed the sequence. Compared with simple ordinal encoding, this new method helps improve the validation accuracy by 9.06%. In comparison, RNN and Bi-LSTM models can remember positional information in the structure, so the ordinal encoding method performs just as well as this new encoding method. However, another limitation of our encoding method is that it leaves the dependency on two adjacent amino acids [37]. Hence, further studies, which take this point into account, will need to be undertaken.

Many studies explored computational models to predict ACPs, such as SVM [6, 10] or LSTM [18, 22]. Furthermore, recent studies tend to improve the model using more complex components. For example, Cao *et al.* [20] fused features from different

channels. Lv *et al.* [1] used deep learning networks to extract pre-trained features and use machine learning models to classify them. However, in the research process, it was observed that, given a small dataset size, simpler models actually perform better and have less risk of overfitting. Thus, we shifted the focus from the complexity of the models to utilizing model ensemble techniques to improve the model performance and increase generalization ability by having the three best models vote for a final result. As a result, the ensemble model has outstanding performance in every evaluation metric (accuracy, sensitivity, specificity and AUC) compared with any separate model and our benchmark model.

Conclusion

To conclude, our final model has an accuracy of 0.7895, thus increasing the accuracy by more than 1.5% compared with previous studies. In addition, it is noted that using deep learning models combined with machine learning methods opens a new avenue in bioinformatics modeling. Meanwhile, simply increasing the complexity of the model can lead to decrease in performance, and sometimes simple models can have better performance. Furthermore, scientists will be able to utilize our architecture in their future studies to deal with the prediction problems of ACPs, thus contributing to promoting progress in bioinformatics.

Key Points

- A novel model to predict anticancer peptides from sequence information
- Features are extracted using ordinal encoding with positional information
- Model is constructed by integrating machine learning and deep learning
- Compared with previous models, our proposed method had a significant improvement in both 5-fold cross-validation and independent test.
- A useful resource for biologists and scientists who would like to perform research on anticancer peptides.

Data availability

The work and source codes are made available to the community of researchers and developers at https://github.com/khanhlee/acp-ope/.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions.

Funding

This work was supported by the National Science and Technology Council, Taiwan [grant numbers MOST110-2221-E-038-001-MY2 and MOST111-2628-E-038-002-MY3].

References

1. Lv Z, Cui F, Zou Q, *et al.*Anticancer peptides prediction with deep representation learning features. *Brief Bioinform* 2021;**22**(5):bbab008.
2. Cheng L, Zhao H, Wang P, *et al.*Computational methods for identifying similar diseases. *Mol Ther Nucleic Acids* 2019;**18**:590–604.
3. Thakkar S, Sharma D, Kalia K, *et al.*Tumor microenvironment targeted nanotherapeutics for cancer therapy and diagnosis: a review. *Acta Biomater* 2020;**101**:43–68.
4. Maeda H, Khatami M. Analyses of repeated failures in cancer therapy for solid tumors: poor tumor-selective drug delivery, low therapeutic efficacy and unsustainable costs. *Clin Transl Med* 2018;**7**(1):e11.
5. Ge R, Feng G, Jing X, *et al.*Enacp: an ensemble learning model for identification of anticancer peptides. *Front Genet* 2020;**11**: 760.
6. Tyagi A, Kapoor P, Kumar R, *et al.*In silico models for designing and discovering novel anticancer peptides. *Sci Rep* 2013;**3**(1): 1–8.
7. Amanat S, Ashraf A, Hussain W, *et al.*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.
8. Hasan MAM, Ben MK, Islam JR, *et al.*Citrullination site prediction by incorporating sequence coupled effects into pseaac and resolving data imbalance issue. *Curr Bioinform* 2020;**15**(3): 235–45.
9. Naseer S, Hussain W, Khan YD, *et al.*Sequence-based identification of arginine amidation sites in proteins using deep representations of proteins and pseaac. *Curr Bioinform* 2020;**15**(8): 937–48.
10. Hajisharifi Z, Piryaiee M, Beigi MM, *et al.*Predicting anticancer peptides with chou's pseudo amino acid composition and investigating their mutagenicity via Ames test. *J Theor Biol* 2014;**341**: 34–40.
11. Vijayakumar S, Ptv L. Acpp: a web server for prediction and design of anti-cancer peptides. *Int J Pept Res Ther* 2015;**21**(1): 99–106.
12. Chen W, Ding H, Feng P, *et al.*Iacp: a sequence-based tool for identifying anticancer peptides. *Oncotarget* 2016;**7**(13): 16895.
13. Li F-M, Wang X-Q. Identifying anticancer peptides by using improved hybrid compositions. *Sci Rep* 2016;**6**(1): 1–6.
14. Rao B, Zhou C, Zhang G, *et al.*Acpred-fuse: fusing multi-view information improves the prediction of anticancer peptides. *Brief Bioinform* 2020;**21**(5): 1846–55.
15. Le NQK, Nguyen V-N. Snare-cnn: a 2d convolutional neural network architecture to identify snare proteins from high-throughput sequencing data. *PeerJ Comput Sci* 2019;**5**:e177.
16. Li F, Zhu F, Ling X, *et al.*Protein interaction network reconstruction through ensemble deep learning with attention mechanism. *Front Bioeng Biotechnol* 2020;**8**:390.
17. Yan J, Bhadra P, Li A, *et al.*Deep-ampep30: improve short antimicrobial peptides prediction with deep learning. *Mol Ther Nucleic Acids* 2020;**20**:882–94.
18. Lezheng Y, Jing R, Liu F, *et al.*Deepacp: a novel computational approach for accurate identification of anticancer peptides by deep learning algorithm. *Mol Ther Nucleic Acids* 2020;**22**: 862–70.
19. Ahmed S, Muhammod R, Khan ZH, *et al.*Acp-mhcnn: an accurate multi-headed deep-convolutional neural network to predict anticancer peptides. *Sci Rep* 2021;**11**(1): 1–15.
20. Cao R, Wang M, Bin Y, *et al.*Dlff-acp: prediction of acps based on deep learning and multi-view features fusion. *PeerJ* 2021;**9**:e11906.
21. Chen X-G, Zhang W, Yang X, *et al.*Acp-da: improving the prediction of anticancer peptides using data augmentation. *Front Genet* 2021;**12**:1131.
22. Yi H-C, You Z-H, Zhou X, *et al.*Acp-dl: a deep learning long short-term memory model to predict anticancer peptides using high-efficiency feature representation. *Mol Ther Nucleic Acids* 2019;**17**: 1–9.

23. Tyagi A, Tuknait A, Anand P, *et al.*Cancerppd: a database of anticancer peptides and proteins. *Nucleic Acids Res* 2015;**43**(D1): D837–43.
24. Dhall A, Patiyal S, Sharma N, *et al.*Computer-aided prediction and design of il-6 inducing peptides: Il-6 plays a crucial role in covid-19. *Brief Bioinform* 2021;**22**(2): 936–45.
25. Basith S, Manavalan B, Shin TH, *et al.*Sdm6a: a web-based integrative machine-learning framework for predicting 6ma sites in the rice genome. *Mol Ther Nucleic Acids* 2019;**18**:131–41.
26. Razzaghi P, Abbasi K, Shirazi M, *et al.*Multimodal brain tumor detection using multimodal deep transfer learning. *Appl Soft Comput* 2022;**129**:109631.
27. Razzaghi P, Abbasi K, Shirazi M, *et al.*Modality adaptation in multimodal data. *Expert Syst Appl* 2021;**179**:115126.
28. Chen K, Wei Z, Zhang Q, *et al.*Whistle: a high-accuracy map of the human n 6-methyladenosine (m6a) epitranscriptome predicted using a machine learning approach. *Nucleic Acids Res* 2019;**47**(7): e41–1.
29. Le NQK, Ho Q-T. Deep transformers and convolutional neural network in identifying dna n6-methyladenine sites in cross-species genomes. *Methods* 2022;**204**:199–206.
30. Wei L, Zhou C, Ran S, *et al.*Pepred-suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. *Bioinformatics* 2019;**35**(21): 4272–80.
31. Wei L, Zhou C, Chen H, *et al.*Acpred-fl: a sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics* 2018;**34**(23): 4007–16.
32. Schaduangrat N, Nantasenamat C, Prachayasittikul V, *et al.*Acpred: a computational tool for the prediction and analysis of anticancer peptides. *Molecules* 2019;**24**(10): 1973.
33. Agrawal P, Bhagat D, Mahalwal M, *et al.*Anticp 2.0: an updated model for predicting anticancer peptides. *Brief Bioinform* 2021;**22**(3):bbaa153.
34. Rao B, Zhang L, Zhang G. Acp-gcn: the identification of anticancer peptides based on graph convolution networks. *IEEE Access* 2020;**8**:176005–11.
35. Lv H, Dao F-Y, Guan Z-X, *et al.*Deep-kcr: accurate detection of lysine crotonylation sites using deep learning method. *Brief Bioinform* 2021;**22**(4):bbaa255.
36. Lv H, Dao F-Y, Zulfiqar H, *et al.*Deepips: comprehensive assessment and computational identification of phosphorylation sites of sars-cov-2 infection using a deep learning-based approach. *Brief Bioinform* 2021;**22**(6):bbab244.
37. Liu S, Xiang X, Gao X, *et al.*Neighborhood preference of amino acids in protein structures and its applications in protein structure assessment. *Sci Rep* 2020;**10**(1): 1–11.