Received: 18 November 2024 | Revised: 8 January 2025 | Accepted: 23 January 2025 | Published online: 4 March 2025

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2025, Vol. 00(00) 1–15
DOI: 10.47852/bonviewJDSIS52024821

BON VIEW PUBLISHING

iACP-SEI: An Anticancer Peptide Identification Method Incorporating Sequence Evolutionary Information

Bowen Zheng[1], Rujun Li[1], Haotian Wang[1], Sheng Wang[1], Shiyu Peng[1], Mingxin Li[1], Liangzhen Jiang[2,3] and Zhibin Lv[1,*]

[1]*College of Biomedical Engineering, Sichuan University, China*

[2]*College of Food and Biological Engineering, Chengdu University, China*

[3]*Country Key Laboratory of Coarse Cereal Processing, Ministry of Agriculture and Rural Affairs, China*

Abstract: Anticancer peptides (ACPs) are a promising focus in clinical oncology due to their ability to inhibit tumor cell proliferation with minimal side effects. Nevertheless, large-scale, expeditious and efficacious identification of ACPs is hindered by the high cost and time demands of conventional wet-lab experiments. Therefore, we introduced a new method called iACP-SEI to identify ACPs using sequence evolution information. iACP-SEI method utilizes the ESM2 protein language model, based on Transformer architecture, to extract feature vectors that encapsulate evolutionary information from peptide sequences. These vectors underwent feature selection via the light gradient boosting machine and used in an ensemble learning approach. Using the AntiCP2.0 main and alternate datasets, iACP-SEI model achieved independent test accuracies of 77.78% and 94.82%, respectively. Furthermore, it outperformed current methods on an unbalanced dataset, achieving a cross-validation accuracy of 90.39%, demonstrating improved robustness in handling imbalanced class samples. Although iACP-SEI demonstrated higher predictive performance and robustness than other methods, some limitations of it are also discussed.

Keywords: anticancer peptides, ESM, deep representation learning, feature selection, ensemble learning

1. Introduction

Cancer continues to pose a substantial worldwide health challenge, resulting in millions of fatalities each year [1]. To address this issue, it is necessary to use inventive treatment approaches. Mainstream cancer treatments such as chemotherapy, radiotherapy, and targeted therapies effectively inhibit tumor growth but often damage normal cells, leading to severe side effects. These limitations have heightened the need for safer and more targeted treatment options [2]. Given these challenges, ACPs have attracted significant scientific interest due to their unique advantages [3]. ACPs are a class of small molecule peptides, typically consisting of no more than 50 amino acids, which have demonstrated specific anti-tumor activity against cancer cells [4]. Compared with conventional therapies, ACPs offer a high degree of safety and selectivity due to their naturally derived bioactive molecules and natural cationic properties, enabling them to selectively bind to the anionic elements on the surface of tumor cells [5]. With advances in clinical research, there has been a growing discovery and confirmation of ACPs derived from proteins [6]. For example, Peelle et al. [7] confirmed the validity of a randomized peptide library mediated by the protein backbone through phenotypic screening of mammalian cells. Meanwhile, Norman et al. [8] used genetic techniques to select and suppress relevant biological pathway peptides. Although these methods proved to be effective, they are often characterized by time and cost constraints, posing challenges for widespread implementation [9]. Consequently, developing new computational methods and bioinformatics tools for the efficient identification and optimization of ACPs is crucial to advancing research and enhancing clinical applications in this field [10–13].

In recent years, many computational methods have been created to identify potential ACPs by analyzing peptide sequences [14]. These models include iACP, PEPred-Suite, ACPred-Fuse, AntiCP 2.0, iACP-DRLF, ACP-check, ACP-BC, and ACP-DRL [5, 6, 15–21]. iACP uses an approach based on pseudo amino acid composition (PseAAC) and g-gap dipeptide [6] patterns to extract sequence features, followed by classification using support vector machines (SVM). PEPred-Suite uses a flexible technique for learning feature representation, which improves the accuracy and robustness of prediction through feature selection and feature fusion [22]. ACPred-Fuse applies the Random Forest (RF) classifier to enhance the accuracy and robustness of prediction by fusing multi-view information and sequential features [21]. AntiCP 2.0 utilizes the characteristics of amino acid composition and dipeptide composition to predict ACPs using multiple

*Corresponding author: Zhibin Lv, College of Biomedical Engineering, Sichuan University, China. Email: lvzhibin@pku.edu.cn

methods of machine learning, such as SVM and RF. The iACP-DRLF method uses two deep representation learning feature extraction techniques (soft symmetric alignment embedding and uniform representation embedding) [18], which results in extraction of deeper features containing more sequence information, and finally combines with the light gradient boosting machine (LGBM) algorithm for feature optimization. ACP-check utilizes a Bidirectional Long Short Term Memory (Bi-LSTM) network to capture temporal information in peptides and integrates it with features of amino acid sequences (e.g., dipeptide composition, amino acid composition, etc.,) [19] to enhance the identification accuracy of ACPs. ACP-BC is a three-channel end-to-end [20] structure, which utilizes Bi-LSTM, BERT (Bidirectional Encoder Representations from Transformers), and manual approaches to extract features and combine them for processing. The ACP-DRL model utilizes in-domain pre-training of language models and Bi-LSTM [20]. Additionally, it introduces BERT-based protein macrolanguage models for ACP recognition, eliminating the limitation of sequence length and the reliance on manual features.

In addition to these advancements, many biomedical researchers have further validated the efficacy of these model-predicted ACPs through experiments or theoretical calculations. For example, Grisoni et al. [23] developed an ensemble machine learning model to design and identify ACPs. They synthesized 14 candidate peptides, and in vitro experiments on breast cancer (MCF7) and lung cancer (A549) cell lines demonstrated that six of these peptides exhibited anticancer activity, with five showing inhibitory effects on both MCF7 and A549 cell lines. Charoenkwan et al. [24] developed iACP-FSCM based on the primary sequence and confirmed the potential binding efficacy of model-predicted ACPs to HIF-1α through molecular docking simulations. Ma et al. [25] combined existing antimicrobial peptide prediction models and metagenomic mining techniques to screen 40 potential ACPs from multiple datasets. Through in vitro experiments, they confirmed that 39 of these ACPs exhibited inhibitory effects on at least one cancer cell line. These studies underscore the effectiveness of integrating AI tools with experimental assays to discover novel and efficacious ACPs, demonstrating the practical feasibility of AI-driven peptide design in biomedical research. However, given that these models predominantly rely on experimentally annotated ACP datasets—which remain limited—their effectiveness is limited. Despite achieving promising results in ACP recognition, there is substantial scope for further refinement and improvement.

In the last few years, the field of protein and peptide sequence research has seen a growing use of large language models due to the development of deep learning. Some of the more typical models include ProtTrans [26], UniRep [27], AlphaFold [28], RoseTTA-Fold [29], and ESM [30–32]. These models employ deep learning techniques and use large-scale datasets trained in an unsupervised or semi-supervised manner. Although this deep representation learning approach enables models to understand a wider range of biological patterns and complex sequence properties [33], the requisite large training datasets and extensive training time pose significant demands on computational resources. Fortunately, these problems can be easily overcome by implementing transfer learning, where pre-trained models are used to accurately identify ACPs in the study.

ESM2 (Evolutionary Scale Modeling 2) [32] is a pre-trained protein language model based on BERT and Transformer [34]. Compared to its predecessor model, ESM-1b [30], ESM2 has been improved in terms of architecture and training parameters with additional computational resources and data. Through masked training techniques, ESM2 effectively learns long-distance dependencies and contextual nuances in protein and peptide sequences and outputs a high-dimensional feature vector containing structural and functional information of the protein. This information can be manipulated by linear projection or other downstream models to enable various predictions and analyses of proteins. However, it is important to note that deep neural networks (DNNs), including the Transformer architecture used in ESM2, exhibit an implicit bias known as the Frequency Principle (F-Principle) [35, 36]. This principle suggests that DNNs tend to learn functions from low to high frequencies during training, capturing general patterns before finer details. Consequently, ESM2 might face challenges in learning high-frequency components of protein sequences, such as subtle local motifs or rare patterns, which are essential for certain protein functions or interactions. This potential deficiency underscores the need for further research into the training dynamics of ESM2 and the development of methods to enhance its capacity to capture high-frequency features in protein sequences.

This paper presents the development of a novel recognition model for ACPs, namely iACP-SEI, which uses ESM2 as its basic pre-trained model. For the purpose of assessing the impact of multiple architectural complexities on model accuracy, we utilized ESM2 pre-trained models with two configurations: a 33-layer architecture containing 650 million parameters and a 36-layer architecture with 3 billion parameters, to extract features from peptide sequences. Afterwards, these features were employed to train models utilizing three distinct machine learning algorithms: Logistic Regression (LR) [37], SVM [13, 38, 39], and LGBM [40, 41]. In order to obtain higher prediction accuracy, we input the feature vectors into the LGBM for feature selection [42]. Ultimately, our method incorporated a stacked ensemble learning approach [43] with LGBM and SVM as base learners to optimize prediction. Compared with the current advanced ACP recognition methods such as ACP-DRL [20] and ACP-BC [6], the optimized iACP-SEI model achieves better results in both five-fold cross-validation (5CV) and independent tests.

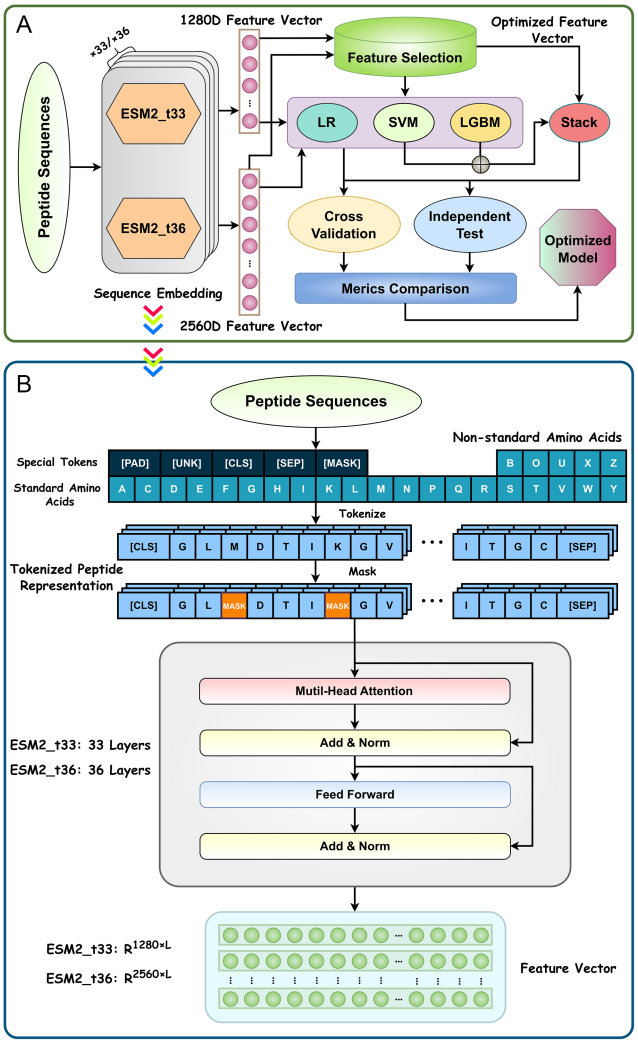
2. Materials and Methods

2.1. Overall framework

The modeling flowchart is shown in Figure 1-A. The primary procedures are as follows:

1) Initially, peptide sequences were characterized as vectors with evolutionary information using the model ESM2. This process resulted in two types of features: the ESM2_t33 feature (1280 dimensions) and the ESM2_t36 feature (2560 dimensions).
2) Subsequently, the space of feature vectors was optimized using the LGBM algorithm.
3) To boost model performance, the stacked ensemble learning model was constructed with SVM and LGBM as the base learners.
4) The optimized feature vectors were then input into four machine learning models: LR, SVM, LGBM, and a Stacked Ensemble model.
5) Finally, 5CV and independent tests were used to compare the effectiveness of different models. The model with the greatest accuracy was chosen to be developed as the final iACP-SEI predictor.

Figure 1
General overview of modeling



Further details of the modeling process will be provided in the next section.

2.2. Dataset

For model training and subsequent evaluation, we selected the dataset produced by Agrawal et al. as the basic dataset. The basic dataset comprises data aggregated from multiple sources, including ACP-DL, ACPP, ACPred-FL, AntiCP, iACP [17], and CancerPPD [44]. It is divided into two parts: the main dataset and the alternate dataset. The main dataset consists of 861 experimentally validated ACPs [45], matched with an equal number of non-ACPs, where all negative samples are antimicrobial peptides (AMPs). Meanwhile, the alternate dataset comprises 970 experimentally confirmed ACPs and 970 non-ACPs. These non-ACPs are random peptides sourced from Swiss-Prot.

To investigate the robustness of the model when dealing with minority samples, we performed 5CV on the unbalanced dataset created by Xu et al. [20]. The unbalanced dataset consists of 845 ACPs and 3800 non-ACPs. It includes all the information from the main and alternate datasets, together with supplementary data from ACPred-Fuse [20]. Redundant sequences in this dataset were removed using the CD-HIT algorithm to ensure data quality.

2.3. Feature extraction

2.3.1. Self-attention module

Feature extraction is used to obtain sequence evolutionary information by utilizing the ESM2 pre-trained model. Its core mechanism is the stacked self-attention module, which is used to determine the features of amino acids. Detailed computational steps of the self-attention mechanism are as follows:

First, the peptide input sequence is represented as a matrix $X \in R^{n\times d}$, where n denotes the peptide sequence length and d denotes the embedding dimension. To compute self-attention, the input embedding matrix X is transformed into the Query, Key, and Value spaces using projection [46]:

$$Q = W_q X, K = W_k X, V = W_v X \tag{1}$$

where $W_q, W_k, W_v \in R^{d\times d_k}$ are the learned weight matrices, with d_k typically set to d/h, where h represents the number of attention heads. Next, by computing the dot product of the queries and keys, the attention scores are obtained:

$$\text{scores} = \frac{QK^T}{\sqrt{d_k}} \tag{2}$$

where d_k is the scaling factor to mitigate the problem of excessively large dot product values at large matrix dimensions. Then, the Softmax function is used to normalize the scores in order to generate the attention weight matrix:

$$W = \text{Softmax}(\text{scores}) \tag{3}$$

The Softmax function is defined as follows:

$$\text{Softmax}(A_{ij}) = \frac{\exp(A_{ij})}{\sum_{k=1}^{n}\exp(A_{ik})} \tag{4}$$

where A_{ij} represents the specific element located at the intersection of row i and column j in the input vector matrix. Afterward, the value matrix is subjected to weighting and summation by utilizing the attention weights, resulting in the final output:

$$\text{Attention}(Q, K, V) = WV \tag{5}$$

In a multi-head self-attention mechanism, this process [47] is replicated across several "heads", each using a unique set of parameters to calculate attention independently. Subsequently, the results from individual heads are combined and sent into a linear transformation layer:

$$\text{Multi Head}(Q, K, V) = \text{concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W_o \tag{6}$$

where $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$, and W_o is the weight matrix of the linear transformation. Finally, residual connections and layer normalization are employed to stabilize the training process and hasten convergence [48]:

$$Y = \text{LayerNorm}(X + \text{MultiHead}(Q, K, V)) \tag{7}$$

The ESM2 model superimposes these attentional mechanisms by means of Transformer architectures with different numbers of

layers and is thus able to capture long-range dependencies and rich feature representations in peptide sequences.

2.3.2. ESM2 pre-trained model

The ESM2 model is a sophisticated protein language model created by Meta AI [32] that uses deep learning techniques to predict protein function and structure [49]. Based on the Transformer architecture, this model captures the evolutionary patterns and sequence-structure-function relationships of proteins through pre-training on millions of protein sequences from different biological species that have been processed by Multiple Sequence Alignment (MSA) and evolutionary analysis. The model then performs self-supervised learning by predicting masked-out amino acids in proteins and finally generates latent vectors that represent their structural and functional attributes. These latent vectors are versatile and support various downstream bioinformatics tasks including protein function prediction, structure modeling, and protein-protein interaction prediction [49].

Figure 1-B displays the structure of the ESM2 model. Initially, the peptide sequences are converted into sequence matrices with masks via tokenization, encoding, and concatenation. The mask enables the pre-trained ESM2 model to develop predictive capabilities [50]. These matrices are subsequently fed into the Transformer encoder network containing a self-attention module, producing feature vector matrices that characterize the peptide sequences. Depending on the number of layers of self-attention modules, multiple versions of the ESM2 model exist. Specific parameters for each version are detailed in Table 1. In this study, we used esm2_t33_650M_UR50D and esm2_t36_3B_UR50D, featuring 33 and 36 self-attention layers with output dimensions of 1280 and 2560, respectively.

2.4. Feature selection method

The main objective of feature selection is to increase model efficiency and performance by removing irrelevant or redundant features, which helps reduce overfitting and accelerate training [51]. Common feature selection methods include analysis of variance (ANOVA), which calculates the *F*-value between features and the target variable for classification problems; recursive feature elimination (RFE), which recursively reduces the feature set to identify key features; minimum redundancy maximum relevance (mRMR), which select features highly correlated with the target and minimize redundancy; and Lasso Regression, which applies regularization to shrink less important feature coefficients to zero [52–55]. These methods can help filter out the most meaningful information from the data and enhance the efficiency of model processing. In our study, we choose the LGBM algorithm, which employs a decision tree to find the optimal feature space and is known for its efficiency in handling large data and achieving high accuracy. The LGBM procedure is as follows. Firstly, all the candidate features and target variables are fed into the LGBM model for learning [56]. Through the built-in feature importance evaluation mechanism based on the Gain metric, LGBM measures the contribution of each feature to the model's predictive power. Therefore, features are ranked and those that exceed a pre-defined threshold, which is determined empirically are selected for the optimized feature set based on the resulting feature importance score.

2.5. Machine learning methods

In this study, we initially picked three different machine learning algorithms—LR, SVM, and LGBM—for comparative analysis. LR is a statistical model widely used for classification, particularly effective in addressing binary classification problems. In this model, outputs from linear regression are mapped to the (0,1) interval using a sigmoid activation function, representing the probability of category membership. SVM classifies samples in a high-dimensional domain by constructing one or more hyperplanes. The training goal for SVM is to discover an optimal hyperplane that maximizes the gap between distinct data categories. Its core mechanism involves kernel techniques that project data into a higher-dimensional domain, enabling linearly inseparable data to be separated. LGBM is a decision tree technique that leverages the gradient boosting framework to provide efficient training speed and improved accuracy. By optimizing the input data using a histogram, LGBM greatly increases the efficiency of data splitting. This makes it particularly well-suited for handling datasets with complex patterns.

To further enhance the model's overall performance, we applied an ensemble learning method that combined the results of several base learners. Ensemble learning more effectively captures diverse data features and patterns, reduces overfitting risk, and enhances generalization than a single model [57, 58]. Common ensemble learning methods include Bagging, Boosting, and Stacking [59]. Specifically, Stacking utilizes the outputs of various base learners as inputs for secondary learners, which then make the final predictions. By fusing the predictions of multiple models, Stacking can completely use the strengths of each base learner and further enhance the whole model's performance.

In the Stacking method, assuming there are n base learners with $h_i(x)$ denoting the prediction of the $\mathrm{i_{th}}$ base learner, the secondary learner can be expressed as:

$$\mathrm{H(x)} = \sum_{i=1}^{\mathrm{n}} \alpha_{\mathrm{i}} \mathrm{h_i(x)} \tag{8}$$

where α_{i} denotes the weight of the base learner $\mathrm{h_i}$, which satisfies the condition that the total of all base learner weights is equal to one, $\sum_{i=1}^{\mathrm{n}} \alpha_{\mathrm{i}} = 1$. By optimizing the weights α_{i}, we can optimize

Table 1
Configuration table for the ESM-2 model

ESM-2 model	Params	Layers	Dataset	Embedding Dim
esm2_t48_15B_UR50D	15B	48	UR50/D 2021_04	5120
esm2_t36_3B_UR50D	3B	36	UR50/D 2021_04	2560
esm2_t33_650M_UR50D	650M	33	UR50/D 2021_04	1280
esm2_t30_150M_UR50D	150M	30	UR50/D 2021_04	640
esm2_t12_35M_UR50D	35M	12	UR50/D 2021_04	480
esm2_t6_8M_UR50D	8M	6	UR50/D 2021_04	320

the performance of the secondary learner H. Furthermore, the secondary learners can also be in the form of more complex non-linear combinations, as demonstrated below:

$$\mathrm{H(x)} = \sigma\left(\sum_{i=1}^{n} \beta_{\mathrm{i}} \mathrm{h_i(x)} + \mathrm{b}\right) \tag{9}$$

Here, β_{i} stands for the weight, b for the bias parameters, and σ for the activation function (e.g., sigmoid or ReLU). Due to the increased computational complexity, the explanatory and predictive power of such non-linear combinations is generally greater than that of linear combinations

In the first phase of the Stacked Ensemble model, we used the SVM and LGBM models as base learners and performed 5CV on them using the training set (please see 2.6 for details on how to do this). After training, each base learner produced an output, and we concatenated these two outputs to form a new training set. Subsequently, we fed the test set into both models to generate their respective predictions and spliced these prediction sets together to form a new test set. In the second stage, we used the SVM model as the meta-learner, training it on the new training set and performing classification on the new test set.

2.6. Evaluation metrics and methods

For the purpose of assessing the effectiveness of our model, we employed several metrics: accuracy (ACC), sensitivity (Sn), specificity (Sp), Matthews correlation coefficient (MCC), and the area under the receiver operating characteristic curve (AUC) [18]. The equations for the first four metrics are as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{10}$$

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{12}$$

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{13}$$

Here, TP represents the number of true-positive samples, TN represents the number of true-negative samples, FP represents the number of false-positive samples, and FN represents the number of false-negative samples [60]. With respect to the AUC, it is a metric that quantifies the performance of a model by measuring the area under the receiver operating characteristic curve (ROC), which plots the true-positive rate against the false-positive rate across various thresholds. A higher AUC value implies better model performance, with a value closer to 1 being optimal. For 5CV on unbalanced datasets, we introduced an additional evaluation metric known as the area under the precision-recall (PR) curve (AUPR) [20], ranging from 0 to 1. A value closer to 1 means that the model has higher precision and recall in identifying positive class samples [61]. Compared to AUC, which is suitable for dealing with more balanced datasets, AUPR focuses on the prediction ability of positive classes (minority classes), and therefore more effectively reflects the performance under unbalanced conditions.

In terms of the evaluation method, we chose the widely used K-fold cross-validation (with K set to 5 in this study) as well as independent tests. In K-fold cross-validation, the dataset is first evenly partitioned into K copies. Next, the algorithm will sequentially select K-1 copies of the K copies as the training set, while the residual one is designated as the validation set, until each copy of the data has been utilized as the validation set. In other words, the model will go through a total of K training sessions, and the evaluation result of the model is determined by taking the average metric value from the K validations. The advantage of this approach lies in its ability to provide a more stable evaluation of model's performance, since the validation set during training will cover all the data in the original dataset. For independent tests, the model is trained only once. In addition, the test data are not included in the training set. For this reason, independent tests are effective in assessing the generalizability of the model.

3. Results

In this section, we assess six initial models developed using two sequence embedding models, ESM2_t33 and ESM2_t36, combined with three machine learning methods: LR, SVM, and LGBM. The models are designated as T33_LR_F1280, T36_LR_F2560, T33_SVM_F1280, T36_SVM_F2560, T33_LGBM_F1280, and T36_LGBM_F2560. For example, in the model name T33_SVM_F1280, 'T33' indicates the employment of the ESM2_t33 model for feature extraction, 'SVM' denotes the SVM machine learning method, and 'F1280' represents a feature dimension of 1280. We then apply the LGBM feature selection technique to optimize the models. To further enhance model performance, we construct stacked ensemble learning models with SVM and LGBM serving as base learners, subsequently evaluating each model's performance. Finally, we select the best-performing model (iACP-SEI) and compare its results with current methods.

3.1. Preliminary performance of the models

In Figure 2 below, we can see the results of 5CV and independent tests of the six models on the main dataset. It is apparent that the SVM-based models achieved the best results in both evaluations. Specifically, T33_SVM_F1280 has the most superior performance in 5CV, outperforming other models in ACC (75.8%), MCC (0.515), Sn (75.7%), and Sp (75.8%), except for a slightly lower performance in AUC (0.804). Furthermore, the performance of T36_SVM_F2560 in independent tests also validates the superiority of the SVM model, which outperformed the second-ranked performance model T33_LR_F1280 on all evaluation metrics, with improvements of 1.75% in ACC, 3.60% in MCC, 2.92% in Sn, 0.58% in Sp and 0.25% in AUC.

For the purpose of assessing the generality of each model across various datasets, we trained the six models on the alternate dataset. The results of their evaluation are presented in Figure 3. In 5CV, mirroring its top performance on the main dataset, T33_SVM_F1280 again achieved optimal results, excelling in ACC (94.1%), MCC (0.882), Sn (91.7%), and AUC (0.979), while the performance of Sp (96.4%) was second only to that of the T33_LGBM_F1280 model of 96.9%. In independent tests, T33_SVM_F1280 ranked first in ACC (93.78%), MCC (0.877), and Sn (90.67%), and was lower than T36_LGBM_F2560's 97.93% in Sp (96.89%), and slightly lower than T33_LR_F1280's 0.973 in AUC (0.970). From Tables 2 and 3, it can be found that the average ACC optimum of 5CV and independent tests on the main dataset was obtained by T36_SVM_F2560, and the average

Figure 2
Preliminary performance of LR, SVM, and LGBM on the main dataset

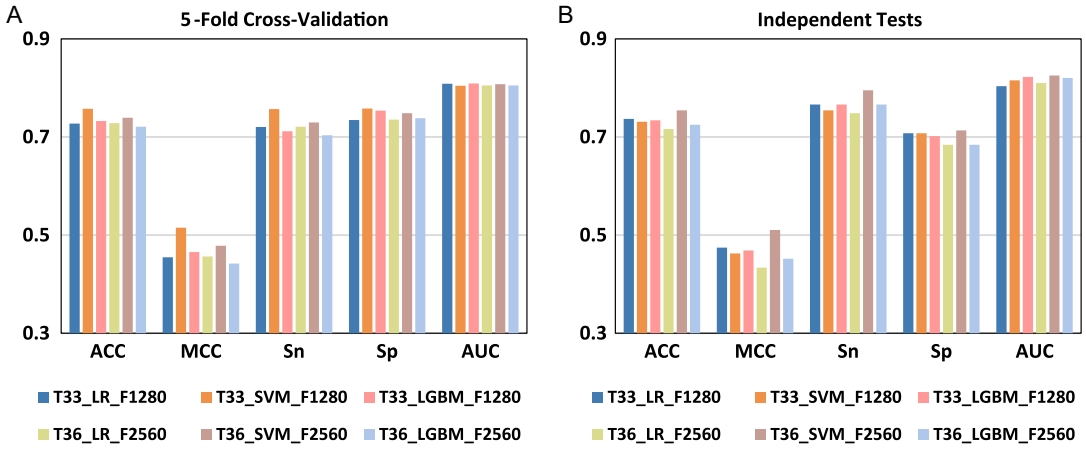
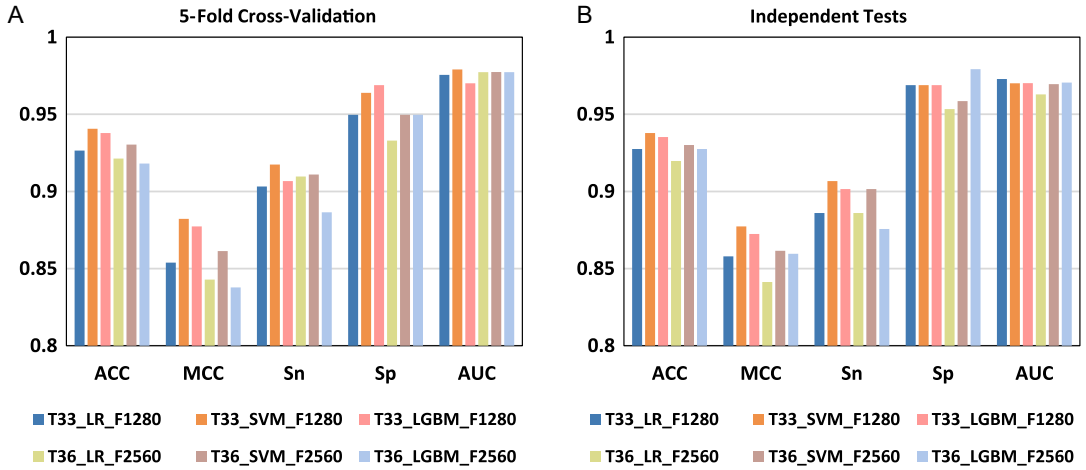


Figure 3
Preliminary performance of LR, SVM, and LGBM on the alternate dataset



ACC optimum on the alternate dataset was obtained by T33_SVM_F1280. These results suggest that recognition models combining ESM2 and SVM may have greater potential to handle complex datasets and capture features and patterns in the data more effectively.

3.2. Performance after feature selection

As discussed in the previous subsection, we found that in the model based on the LGBM algorithm, T33_LGBM_F1280 significantly outperformed T36_LGB-M_F2560, which we speculated was due to feature redundancy resulting from increased feature dimensions in the latter. For LGBM algorithms, an excess of low-information features can detract from model learning rather than adding value. Therefore, we optimized the feature space, utilizing the LGBM model to assess feature importance and arrange features in descending order of importance. Subsequently, we used these optimized feature vectors to train both SVM and LGBM models. For different datasets, we adjusted the dimensionality of feature selection. On the main dataset, we obtained four optimized models: t33_SVM_F155, t36_SVM_F205, t33_LGBM_F295, and t36_LGBM_F165. For instance, 't33_SVM_F155' indicates that only the 155 most critical dimensions were selected from the 1280 features extracted from ESM2_t33. On the alternate dataset, we obtained four optimized models, T33_SVM_F50, T36_SVM_F50, T33_LGBM_F255, and T36_LGBM_F175. Results from 5CV and independent tests are presented in Figures 4 and 5, with corresponding accuracy values listed in Tables 4 and 5.

On the main dataset, it is evident that feature selection considerably improves the performance of the LGBM model, especially in 5CV, as shown in Figure 4 and Table 4. Taking T36_LGBM_F165 as an example, compared with the model T36_LGBM_F2560 without feature selection, its performance exhibited significant improvement across all metrics: ACC increased by 5.60% to 77.69%, MCC by 11.28% to 0.555, Sn by 4.07% to 74.42%, Sp by 7.12% to 80.96%, and AUC by 7.58% to 0.881. In independent tests, the T36_LGBM_F165 also outperformed T36_LGBM_F2560, showing increases of 3.51% in ACC to 76.02%, 7.09% in MCC to 0.523, 4.09% in Sn to 80.70%, and 2.93% in Sp to 71.35%, although it trailed by 0.76% in AUC at 0.813. However, for the SVM-based models, feature

Table 2
Table of preliminary performance metrics for LR, SVM, and LGBM on the main dataset, with the best score for each metric bolded and underlined

			5-Fold cross-Validation					Independent tests					
Feature	Model	Dim	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AVG_ACC(%)
ESM_t33	LR	1280	72.74	0.455	72.04	73.45	0.808	73.68	0.474	76.61	70.76	0.803	73.21
	SVM	1280	**75.75**	**0.515**	**75.70**	**75.81**	0.804	73.10	0.462	75.44	70.76	0.816	74.43
	LGBM	1280	73.26	0.466	71.16	75.37	**0.809**	73.39	0.469	76.61	70.18	0.823	73.32
ESM_t36	LR	2560	72.82	0.456	72.09	73.55	0.805	71.64	0.434	74.85	68.42	0.810	72.23
	SVM	2560	73.91	0.478	72.97	74.85	0.808	**75.44**	**0.510**	**79.53**	**71.35**	**0.825**	**74.67**
	LGBM	2560	72.09	0.442	70.35	73.84	0.805	72.51	0.452	76.61	68.42	0.820	72.30

Table 3
Table of preliminary performance metrics for LR, SVM, and LGBM on the alternate dataset, with the best score for each metric bolded and underlined

			5-Fold cross-Validation					Independent tests					
Feature	Model	Dim	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AVG_ACC(%)
ESM_t33	LR	1280	92.65	0.854	90.32	94.97	0.976	92.75	0.858	88.60	96.89	**0.973**	92.70
	SVM	1280	**94.06**	**0.882**	**91.74**	96.39	**0.979**	**93.78**	**0.877**	**90.67**	96.89	0.970	**93.92**
	LGBM	1280	93.78	0.877	90.67	**96.89**	0.970	93.52	0.872	90.16	96.89	0.970	93.65
ESM_t36	LR	2560	92.13	0.843	90.97	93.29	0.977	91.97	0.841	88.60	95.34	0.963	92.05
	SVM	2560	93.03	0.861	91.10	94.97	0.977	93.01	0.862	90.16	95.85	0.970	93.02
	LGBM	2560	91.81	0.838	88.65	94.97	0.977	92.75	0.860	87.56	**97.93**	0.971	92.28

Figure 4
Comparison of metrics after feature selection optimization on the main dataset

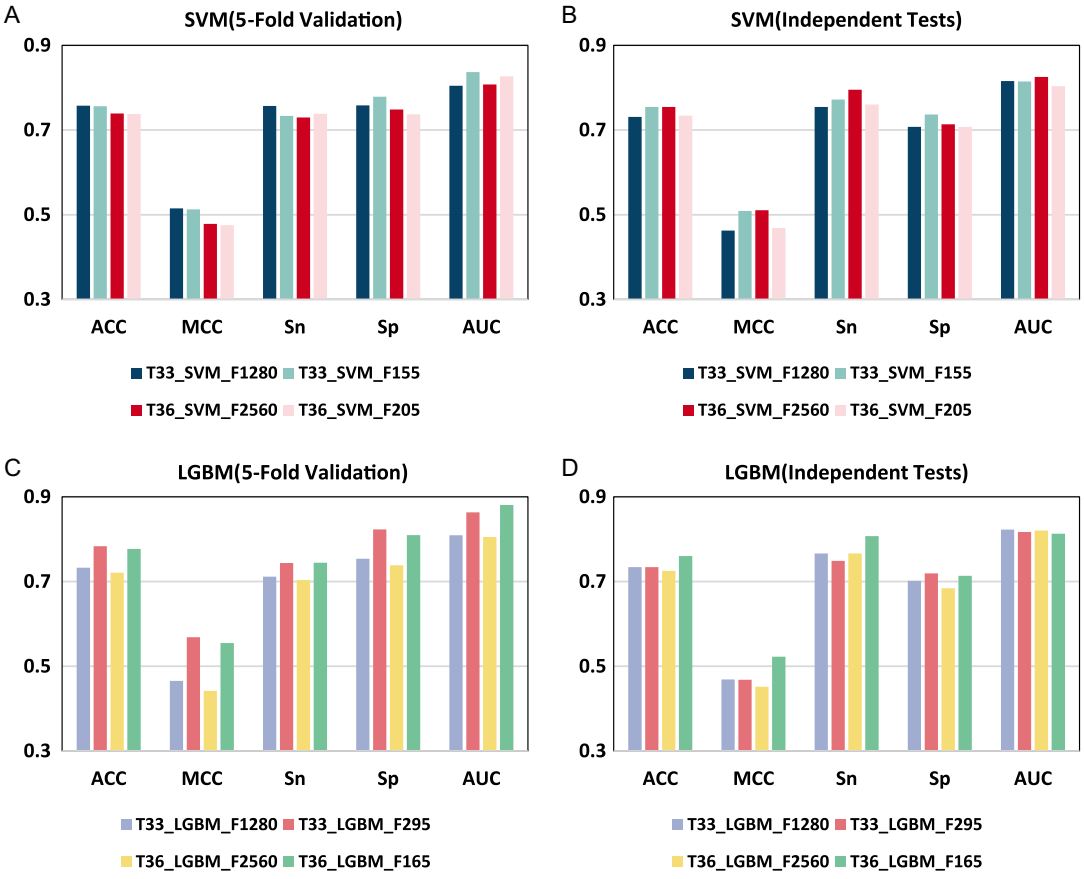


Figure 5
Comparison of metrics after feature selection optimization on the alternate dataset

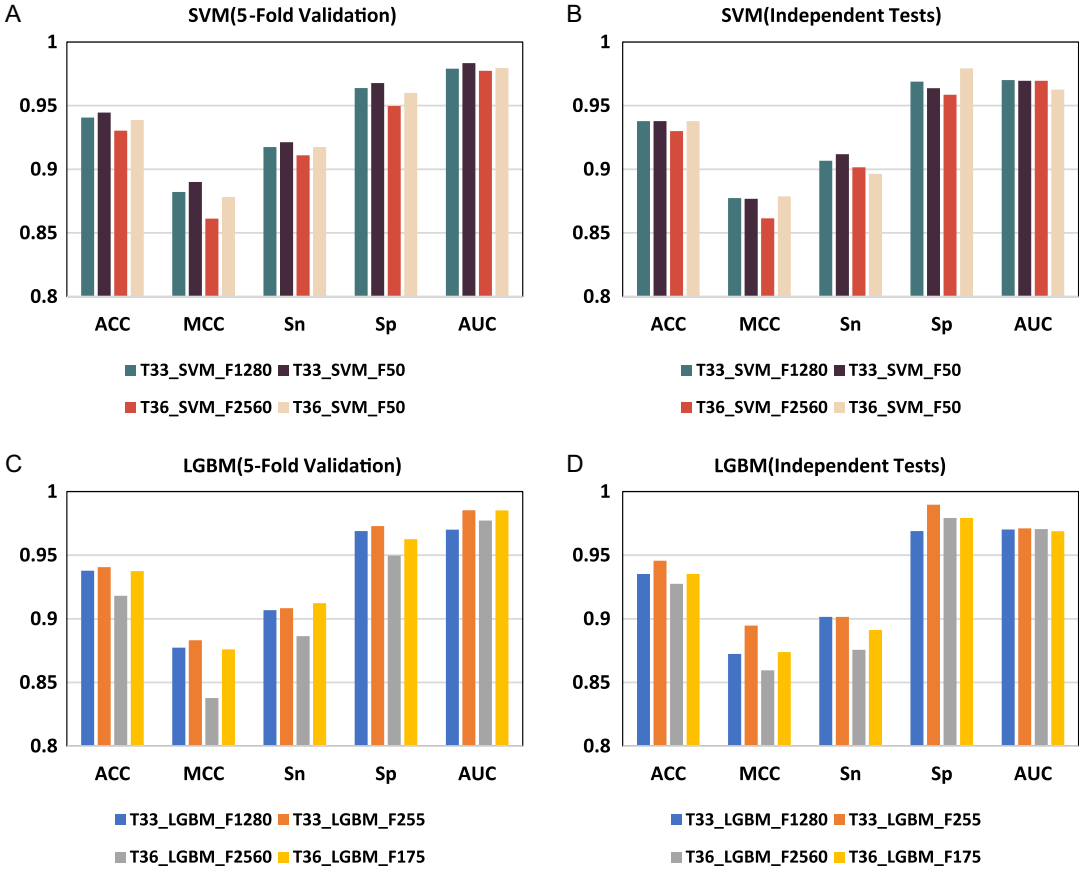


Table 4
Table of performance metrics for SVM and LGBM after feature selection on the main dataset, with the best score for each metric bolded and underlined

			5-Fold cross-Validation					Independent tests					
Feature	Model	Dim	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AVG_ACC(%)
ESM_t33	SVM	1280	75.75	0.515	**75.70**	75.81	0.804	73.10	0.462	75.44	70.76	0.816	74.43%
		155	75.61	0.513	73.35	77.88	0.837	75.44	0.509	77.19	**73.68**	0.815	75.52%
	LGBM	1280	73.25	0.466	71.16	75.37	0.809	73.39	0.469	76.61	70.18	0.823	73.32%
		295	**78.32**	**0.568**	74.38	**82.30**	0.863	73.39	0.468	74.85	71.93	0.817	75.86%
ESM_t36	SVM	2560	73.91	0.478	72.97	74.85	0.808	75.44	0.510	79.53	71.35	**0.825**	74.67%
		205	73.76	0.475	73.84	73.69	0.827	73.39	0.468	76.02	70.76	0.804	73.58%
	LGBM	2560	72.09	0.442	70.35	73.84	0.805	72.51	0.452	76.61	68.42	0.820	72.30%
		165	77.69	0.555	74.42	80.96	**0.881**	**76.02**	**0.523**	**80.70**	71.35	0.813	**76.86%**

Table 5
Table of performance metrics for SVM and LGBM after feature selection on the alternate dataset, with the best score for each metric bolded and underlined

			5-Fold cross-Validation					Independent tests					
Feature	Model	Dim	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AVG_ACC(%)
ESM_t33	SVM	1280	94.06	0.882	91.74	96.39	0.979	93.78	0.877	90.67	96.89	0.970	93.92%
		50	**94.45**	**0.890**	**92.13**	96.77	0.983	93.78	0.877	**91.19**	96.37	0.970	94.12%
	LGBM	1280	93.78	0.877	90.67	96.89	0.970	93.52	0.872	90.16	96.89	0.970	93.65%
		255	94.06	0.883	90.84	**97.29**	**0.985**	**94.56**	**0.895**	90.16	**98.96**	**0.971**	**94.31%**
ESM_t36	SVM	2560	93.03	0.861	91.10	94.97	0.977	93.01	0.862	90.16	95.85	0.970	93.02%
		50	93.87	0.878	91.74	96.00	0.979	93.78	0.879	89.64	97.93	0.963	93.83%
	LGBM	2560	91.81	0.838	88.65	94.97	0.977	92.75	0.860	87.56	97.93	**0.971**	92.28%
		175	93.74	0.876	91.23	96.26	**0.985**	93.52	0.874	89.12	97.93	0.969	93.63%

selection on the main dataset did not enhance performance and instead led to a decline. For example, T36_SVM_F205 was lower than the original model T36_SVM_F2560 in all five metrics in independent tests. While T33_SVM_F155 outperforms T33_SVM_F1280 in the first four metrics, it still did not surpass T36_SVM_F2560 overall, and this advantage was not maintained in 5CV.

On the alternate dataset, the same improvement was observed with the LGBM model from Figure 5 and Table 5: the feature-selected model outperformed the original model in both 5CV and independent tests. Among them, T33_LGBM_F255 performed best, achieving 5CV scores of ACC (94.06%), MCC (0.883), Sp (97.29%), and AUC (0.986). These represented improvements of 0.28%, 0.58%, 0.40%, and 0.01%, respectively, compared to the next best model. Although Sn for T33_LGBM_F255 (90.84%) was slightly (0.39%) behind that of T36_LGBM_F175 (91.23%), it still surpassed other LGBM models. In independent tests, T33_LGBM_F255 lead in ACC (94.56%), MCC (0.895), Sn (90.16%), Sp (98.96%), and AUC (0.971), outperforming the second-best model by margins of 1.04%, 2.08%, 1.04%, 1.03%, and 0.05%, respectively. In particular, it also had the best values of ACC, MCC, Sp, and AUC among all models (including SVM models). For SVM-based models, feature selection on the alternate dataset yielded partial improvements in performance. These enhancements were mainly in 5CV. For example, T33_SVM_F50 exhibited small improvements over the original model T33_SVM_F1280 in all five metrics: ACC (94.45%) improved by 0.39%, MCC (0.890) by 0.78%, Sn (92.13%) by 0.39%, Sp (96.77%) by 0.38% and AUC (0.983) by 0.44%. However, there were no such improvements in the independent tests.

Overall, T36_LGBM_F165 and T33_LGBM_F255 were the top performers on the main and alternate datasets, respectively, as shown in Tables 4 and 5. This is attributed to both models not only achieving the highest number of leading metrics but also the highest average ACC values: 76.86% for T36_LGBM_F165 and 94.31% for T33_LGBM_F255.

3.3. Performance of ensemble learning models

For maximum effectiveness in recognizing ACPs, we adopted an optimization using a stacked ensemble learning method with SVM and LGBM as base learners and developed four such ensemble models. Assessments of the different evaluation methods are shown in Figures 6 and 7, with corresponding accuracy values listed in Tables 6 and 7. For the purpose of precisely assessing the behavior of the ensemble learning models, we selected the top-performing models in the feature selection optimization section for comparison.

Figure 6
Performance of SVM, LGBM, and Stacked ensemble learning models on the main dataset

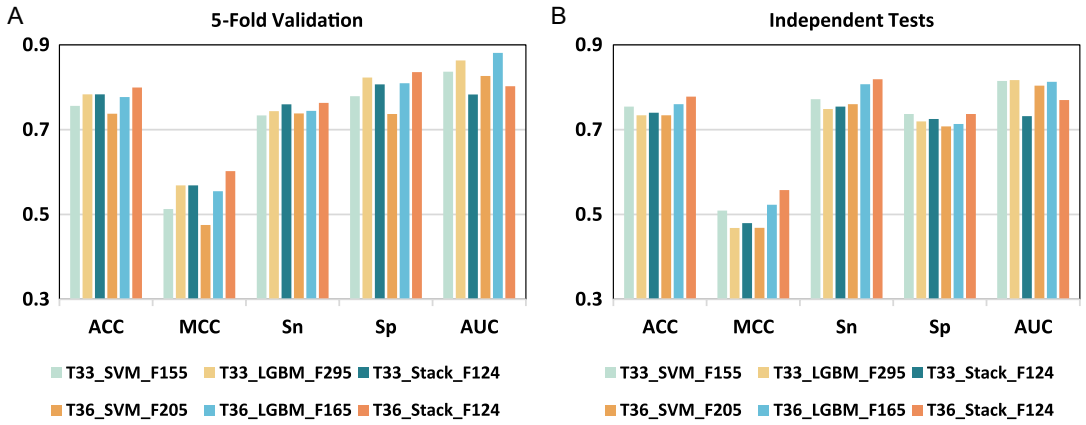


Figure 7
Performance of SVM, LGBM, and Stacked ensemble learning models on the alternate dataset

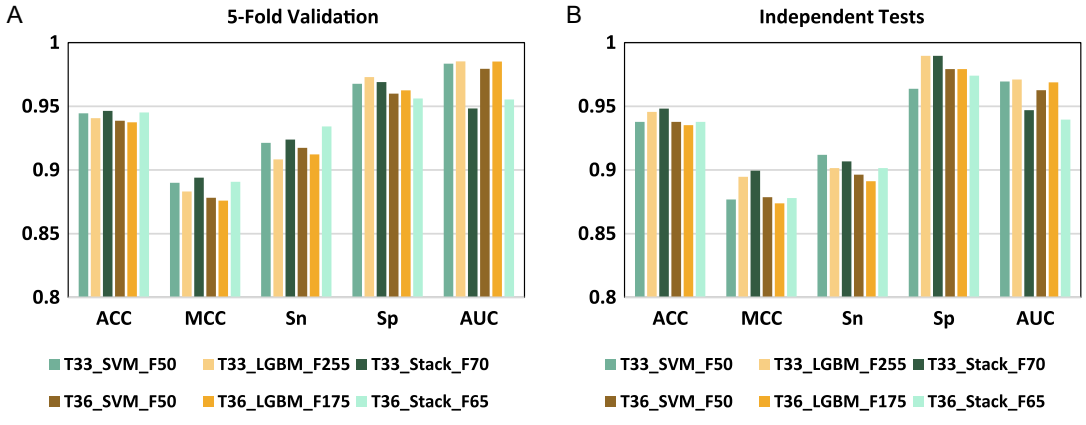


Table 6
Table of performance metrics for SVM, LGBM, and Stacked ensemble learning models on the main dataset, with the best score for each metric bolded and underlined

			5-Fold cross-Validation					Independent tests					
Feature	Model	Dim	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AVG_ACC(%)
ESM_t33	SVM	155	75.61	0.513	73.35	77.88	0.837	75.44	0.509	77.19	**73.68**	0.815	75.52
	LGBM	295	78.32	0.568	74.38	82.30	0.863	73.39	0.468	74.85	71.93	**0.817**	75.86
	Stack	124	78.33	0.569	75.99	80.68	0.783	73.98	0.480	75.44	72.51	0.732	76.15
ESM_t36	SVM	205	73.76	0.475	73.84	73.69	0.827	73.39	0.468	76.02	70.76	0.804	73.58
	LGBM	165	77.69	0.555	74.42	80.96	**0.881**	76.02	0.523	80.70	71.35	0.813	76.86
	Stack	124	**79.94**	**0.602**	**76.32**	**83.58**	0.802	**77.78**	**0.557**	**81.87**	**73.68**	0.770	**78.86**

Table 7
Table of performance metrics for SVM, LGBM, and Stacked ensemble learning models on the alternate dataset, with the best score for each metric bolded and underlined

			5-Fold cross-Validation					Independent tests					
Feature	Model	Dim	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AVG_ACC(%)
ESM_t33	SVM	50	94.45	0.890	92.13	96.77	0.983	93.78	0.877	**91.19**	96.37	0.970	94.12
	LGBM	255	94.06	0.883	90.84	**97.29**	**0.986**	94.56	0.895	90.16	**98.96**	**0.971**	94.31
	Stack	70	**94.65**	**0.894**	92.39	96.90	0.948	**94.82**	**0.899**	90.67	**98.96**	0.947	**94.73**
ESM_t36	SVM	50	93.87	0.878	91.74	96.00	0.979	93.78	0.879	89.64	97.93	0.963	93.83
	LGBM	175	93.74	0.876	91.23	96.26	0.985	93.52	0.874	89.12	97.93	0.969	93.63
	Stack	65	94.52	0.891	**93.42**	95.61	0.955	93.78	0.878	90.16	97.41	0.940	94.15

In cross-validation on the main dataset (Figure 6-A and Table 6), it is apparent that the general effectiveness of model T36_Stack_F124 is better than the other models, which effectively illustrates the applicability of the stacking ensemble learning method. Specifically, it achieved scores of 79.94% for ACC, 0.602 for MCC, 76.32% for Sn, and 83.58% for Sp, an improvement of 1.61%, 3.35%, 0.33%, and 1.28%, respectively, over the second-place models. With respect to AUC, the T36_Stack_F124 did not perform as well as the remaining four single models, being outperformed by the best-performing model T36_LGBM_F165 (88.09%) by 6.07%. However, T36_Stack_F124 still performed better than another ensemble model, T33_Stack_F124. A similar improvement was also observed in independent tests (Figure 6-B and Table 6), where T36_Stack_F124 ranked first on ACC, MCC, Sn, and Sp with 77.78%, 0.557, 81.87%, and 73.68%, respectively.

In tests conducted on the alternate dataset, the ensemble learning models consistently demonstrated superior performance (Figure 7 and Table 7). As an example, T33_Stack_F70 had an ACC of 94.65% and 94.82%, and an MCC of 0.894 and 0.899, which ranked first among all models in the two evaluation methods. In terms of the remaining metrics, the ensemble learning models also exhibited good performance. For instance, T36_Stack_F65 had the highest Sn score of 93.42% in 5CV. Meanwhile, T33_Stack_F70 and T33_LGBM_F255 jointly lead with a Sp of 98.96% in independent tests. Although some ensemble models occasionally underperformed single models in certain metrics, they consistently achieve the highest ACC scores, which were considered the most indicative of overall effectiveness in 5CV and independent tests. In particular, T36_Stack_F124 leads with an average ACC of 78.86% on the main dataset, 2% higher than the runner-up, while T33_Stack_F70 registered an average ACC of 94.73% on the alternate dataset, outperforming the second best by 0.42%.

The discussion illustrates that applying an ensemble learning approach significantly strengthens model performance in predicting ACPs. This is particularly evident in the performance of models utilizing T36_Stack with 124 dimensions (124D) and T33_Stack with 70 dimensions (70D), which have proven to be the optimal choices for predicting ACPs on the main and alternate datasets, respectively.

3.4. Comparison with advanced methods

With the aim of comprehensively evaluating the effectiveness of iACP-SEI, we compared it with other machine learning or deep learning models, including iACP, PEPred-Suite, ACPred-Fuse, AntiCP 2.0, iACP-DRLF, ACP-check, ACP-BC, and ACP-DRL. Table 8 shows the independent testing performance of each model on AntiCP 2.0's benchmark datasets, with the top score for each metric in bold and underlined. Although iACP-SEI did not surpass the most advanced model, ACP-DRL, in ACC (77.78% vs. 78.96%) and Sp (73.68% vs. 78.39%) on the main dataset, it excelled in MCC and Sn metrics. Specifically, its MCC score of 0.56 matched that of ACP-DRL and ACP-check, and its Sn score of 81.87% exceeded the second-ranked iACP-DRLF by approximately 1.17%. On the alternate dataset, iACP-SEI significantly outperforms other models, leading in ACC (94.82%), MCC (0.90), and Sp (98.96%).

To further illustrate the excellence of our model, we compared it on the unbalanced dataset constructed by ACP-DRL. The latest 5CV scores of ACP-check, ACP-BC, and ACP-DRL were sourced from ACP-DRL [20]. Table 9 shows the model performance on this unbalanced dataset, with the top score for each metric bolded and

Table 8
Comparison of iACP-SEI with current ACP predictors on the main and alternate datasets for independent tests, with the best score for each metric bolded and underlined

	Main dataset				Alternate dataset			
Method	ACC(%)	MCC	Sn(%)	Sp(%)	ACC(%)	MCC	Sn(%)	Sp(%)
iACP	55.10	0.11	77.91	32.16	77.58	0.55	78.35	76.80
PEPred-Suite	53.49	0.08	33.14	73.84	57.47	0.16	40.21	74.74
ACPred-Fuse	68.90	0.38	69.19	68.60	78.87	0.60	64.43	93.30
AntiCP 2.0	75.43	0.51	77.46	73.41	92.01	0.84	92.27	91.75
iACP-DRLF	77.50	0.55	80.70	74.30	93.00	0.86	89.60	96.40
ACP-check	78.00	**0.56**	80.00	77.00	93.00	0.86	**93.00**	93.00
ACP-BC	75.16	0.50	72.61	77.71	91.05	0.82	92.14	89.96
ACP-DRL	**78.96**	**0.56**	79.53	**78.39**	94.43	0.89	92.22	96.64
iACP-SEI(ours)	77.78	**0.56**	**81.87**	73.68	**94.82**	**0.90**	90.67	**98.96**

Table 9
Comparison of iACP-SEI with current ACP predictors on the unbalanced dataset for five-fold cross-validation, with the best score for each metric bolded and underlined

Method	ACC(%)	MCC	Sn(%)	Sp(%)	AUC	AUPR
ACP-check	82.00	0.40	49.22	89.21	0.76	0.44
ACP-BC	88.53	0.60	**64.58**	93.87	0.89	0.71
ACP-DRL	89.82	0.64	62.47	95.89	0.91	0.78
iACP-SEI(ours)	**90.39**	**0.66**	61.70	**96.93**	**0.93**	**0.80**

underlined. As we can see, iACP-SEI was the top performer in all five metrics, achieving superior scores for ACC (94.82%), MCC (0.90), Sp (98.96%), AUC (0.93), and AUPR (0.80). The Sn score (61.70%) is only slightly below those of ACP-BC (64.58%) and ACP-DRL (62.47%), which ranked third. These results indicate that iACP-SEI is able to identify and process minority class samples more effectively without being overwhelmed by the majority class, and is more robust in the face of data imbalance. In summary, iACP-SEI is one of the most advanced ACP predictors that are based on machine learning, especially for unbalanced data, where it consistently and accurately differentiates between ACPs and non-ACPs.

4. Conclusion

In conclusion, we developed a potent ACP recognition model named iACP-SEI. iACP-SEI leverages the ESM2 protein language model to construct optimal ensemble learning models for the main and alternate AntiCP 2.0 datasets using feature selection and stacking ensemble learning method. To test the robustness of iACP-SEI, we also conducted detailed experiments on the unbalanced dataset of ACP-DRL. The results demonstrated that iACP-SEI outperformed the existing advanced models across all evaluated datasets. In particular, on both the alternate and unbalanced datasets, iACP-SEI achieved enhanced performance in ACC and MCC. By extracting evolutionary information features of peptides using a protein language model, iACP-SEI showed an improved ability to identify ACPs and non-ACPs.

Although iACP-SEI demonstrated higher predictive performance and robustness than other methods, there are still some limitations. For example, the large difference in the distribution of certain anticancer peptide samples leads to difficulties in the model recognizing these samples. Some ACPs with cyclic structures or unconventional amino acid compositions may not align with the main patterns the model has learned, making them more challenging to identify accurately. In addition, the deep feature extraction process of the model requires significant computational resources, which can limit its application. However, these shortcomings do not prevent us from applying the pre-trained protein language models to analyze peptide or protein sequences and develop more efficient methods. In recent years, some studies have begun to focus on the prediction of multifunctional peptides. For example, the ETFC model developed by Fan et al. utilizes positional encoding and text encoding combined with a feedforward neural network for prediction [62]. These models are all based on traditional sequence information. Therefore, in future research, we will attempt to apply the modeling methods used in iACP-SEI to the prediction of multifunctional peptides.

Funding Support

This project has received support from the National Natural Science Foundation of China (No. 62371318, No. 32302083), 2024 Foundation Cultivation Research—Basic Research Cultivation Special Funding (No. 20826041H4211), and the Chengdu Science and Technology Bureau (No. 2024-YF08-00022-GX).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data available on request from the corresponding author upon reasonable request.

Author Contribution Statement

Bowen Zheng: Validation, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Supervision, Project administration. **Rujun Li:** Investigation, Data curation, Visualization. **Haotian Wang:** Validation, Investigation, Visualization. **Sheng Wang:** Investigation. **Shiyu Peng:** Data curation. **Mingxin Li:** Formal analysis. **Liangzhen Jiang:** Writing – review & editing. **Zhibin Lv:** Conceptualization, Methodology, Software, Supervision, Project administration, Funding acquisition.

References

[1] Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R. L., Soerjomataram, I., ..., & Global cancer statistics 2022. (2024). GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, *74*, 229–263.

[2] Gottlieb, E., & Tomlinson, I. P. (2005). Mitochondrial tumour suppressors: A genetic and biochemical update. *Nature Reviews Cancer*, *5*, 857–866.

[3] Hollingsworth, R. E., & Jansen, K. (2019). Turning the corner on therapeutic cancer vaccines. *npj Vaccines*, *4*, 7.

[4] Gaspar, D., Veiga, A. S., & Castanho, M. A. (2013). From antimicrobial to anticancer peptides. A review. *Frontiers in Microbiology*, *4*, 63880.

[5] Rao, B., Zhou, C., Zhang, G., Su, R., & Wei, L. (2020). ACPred-Fuse: Fusing multi-view information improves the prediction of anticancer peptides. *Briefings in Bioinformatics*, *21*, 1846–1855.

[6] Sun, M., Hu, H., Pang, W., & Zhou, Y. (2023). ACP-BC: A model for accurate identification of anticancer peptides based on fusion features of bidirectional long short-term memory and chemically derived information. *International Journal of Molecular Sciences*, *24*, 15447.

[7] Peelle, B., Lorens, J., Li, W., Bogenberger, J., Payan, D. G., & Anderson, D. (2001). Intracellular protein scaffold-mediated display of random peptide libraries for phenotypic screens in mammalian cells. *Chemistry & Biology*, *8*, 521–534.

[8] Norman, T. C., Smith, D. L., Sorger, P. K., Drees, B. L., O'Rourke, S. M., Hughes, T. R., ..., & Murray, A. W. (1999). Genetic selection of peptide inhibitors of biological pathways. *Science*, *285*, 591–595.

[9] Zhang, Z., Cui, F., Su, W., Dou, L., Xu, A., Cao, C., & Zou, Q. (2022). webSCST: An interactive web application for single-cell RNA-sequencing data and spatial transcriptomic data integration. *Bioinformatics*, *38*, 3488–3489.

[10] Müller, A. T., Gabernet, G., Hiss, J. A., & Schneider, G. (2017). modlAMP: Python for antimicrobial peptides. *Bioinformatics*, *33*, 2753–2755.

[11] Cui, F., Li, S., Zhang, Z., Sui, M., Cao, C., El-Latif Hesham, A., & Zou, Q. (2022). DeepMC-iNABP: Deep learning for multiclass identification and classification of nucleic

acid-binding proteins. *Computational and Structural Biotechnology Journal*, *20*, 2020–2028.

[12] Cao, C., Wang, J., Kwok, D., Cui, F., Zhang, Z., Zhao, D., … , & Zou, Q. (2022). webTWAS: A resource for disease candidate susceptibility genes identified by transcriptome-wide association study. *Nucleic Acids Research*, *50*, D1123–D1130.

[13] Wang, Y., Zhai, Y., Ding, Y., & Zou, Q. (2024). SBSM-Pro: Support bio-sequence machine for proteins. *Science China Information Sciences*, *67*, 212106.

[14] Gu, Z. F., Hao, Y. D., Wang, T. Y., Cai, P. L., Zhang, Y., Deng, K. J., … , & Lv, H. (2024). Prediction of blood–brain barrier penetrating peptides based on data augmentation with Augur. *BMC Biology*, *22*, 86.

[15] Chen, W., Ding, H., Feng, P., Lin, H., & Chou, K. C. (2016). iACP: A sequence-based tool for identifying anticancer peptides. *Oncotarget*, *7*, 16895.

[16] Wei, L., Zhou, C., Su, R., & Zou, Q. (2019). PEPred-Suite: Improved and robust prediction of therapeutic peptides using adaptive feature representation learning. *Bioinformatics*, *35*, 4272–4280.

[17] Agrawal, P., Bhagat, D., Mahalwal, M., Sharma, N., & Raghava, G. P. S. (2021). AntiCP 2.0: An updated model for predicting anticancer peptides. *Briefings in Bioinformatics*, *22*, bbaa153.

[18] Lv, Z., Cui, F., Zou, Q., Zhang, L., & Xu, L. (2021). Anticancer peptides prediction with deep representation learning features. *Briefings in Bioinformatics*, *22*, bbab008.

[19] Zhu, L., Ye, C., Hu, X., Yang, S., & Zhu, C. (2022). ACP-check: An anticancer peptide prediction model based on bidirectional long short-term memory and multi-features fusion strategy. *Computers in Biology and Medicine*, *148*, 105868.

[20] Xu, X., Li, C., Yuan, X., Zhang, Q., Liu, Y., Zhu, Y., & Chen, T. (2024). ACP-DRL: An anticancer peptides recognition method based on deep representation learning. *Frontiers in Genetics*, *15*, 1376486.

[21] Liu, M., Wu, T., Li, X., Zhu, Y., Chen, S., Huang, J., … , & Liu, H. (2024). ACPPfel: Explainable deep ensemble learning for anticancer peptides prediction based on feature optimization. *Frontiers in Genetics*, *15*, 1352504.

[22] Chen, J., Cheong, H., & Siu, S. (2021). XDeep-AcPEP: Deep learning method for anticancer peptide activity prediction based on convolutional neural network and multitask learning. *Journal of Chemical Information and Modeling*, *61*, 3789–3803.

[23] Grisoni, F., Neuhaus, C. S., Hishinuma, M., Gabernet, G., Hiss, J. A., Kotera, M., & Schneider, G. (2019). De novo design of anticancer peptides by ensemble artificial neural networks. *Journal of Molecular Modeling*, *25*, 112.

[24] Charoenkwan, P., Chiangjong, W., Lee, V. S., Nantasenamat, C., Hasan, M. M., & Shoombuatong, W. (2021). Improved prediction and characterization of anticancer activities of peptides using a novel flexible scoring card method. *Scientific Reports*, *11*, 3017.

[25] Ma, Y., Liu, X., Zhang, X., Yu, Y., Li, Y., Song, M., & Wang, J. (2023). Efficient mining of anticancer peptides from gut metagenome. *Advanced Science*, *10*, 2300107.

[26] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., … , & Rost, B. (2022). ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*, 7112–7127.

[27] Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., & Church, G. M. (2019). Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, *16*, 1315–1322.

[28] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., … , & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*, 583–589.

[29] Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., … , & Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, *373*, 871–876.

[30] Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., … , & Ma, J. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*, e2016239118.

[31] Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., & Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, *34*, 29287–29303.

[32] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., … , & Shmueli, Y. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, *379*, 1123–1130.

[33] Cao, C., Shao, M., Zuo, C., Kwok, D., Liu, L., Ge, Y., … , & Zou, Q. (2024). RAVAR: A curated repository for rare variant–trait associations. *Nucleic Acids Research*, *52*, D990–D997.

[34] Chen, J., Wu, H., & Wang, N. (2024). KEGG orthology prediction of bacterial proteins using natural language processing. *BMC Bioinformatics*, *25*, 146.

[35] Xu, Z. Q. J., Zhang, Y., & Luo, T. (2022). Overview frequency principle/spectral bias in deep learning. *arXiv:2201.07395*.

[36] Xu, Z. Q. J. (2020). Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, *28*, 1746–1767.

[37] Dai, R., Zhang, W., Tang, W., Wynendaele, E., Zhu, Q., Bin, Y., … , & Xia, J. (2021). BBPpred: Sequence-based prediction of blood-brain barrier peptides with feature representation learning and logistic regression. *Journal of Chemical Information and Modeling*, *61*, 525–534.

[38] Wan, Y., Wang, Z., & Lee, T. Y. (2021). Incorporating support vector machine with sequential minimal optimization to identify anticancer peptides. *BMC Bioinformatics*, *22*, 286.

[39] Manganaro, L., Sabbatini, G., Bianco, S., Bironzo, P., Borile, C., Colombi, D., … , & Vittorio Scagliotti, G. (2023). Non-small cell lung cancer survival estimation through multi-omic two-layer SVM: A multi-omics and multi-sources integrative model. *Current Research in Bioinformatics*, *18*, 658–669.

[40] Zhang, Y., Yu, S., Xie, R., Li, J., Leier, A., Marquez-Lago, T. T., … , & Wang, J. (2020). PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. *Bioinformatics*, *36*, 704–712.

[41] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*, 1937–1967.

[42] Zhang, L., Hu, X., Xiao, K., & Kong, L. (2024). Effective identification and differential analysis of anticancer peptides. *BioSystems*, *241*, 105246.

[43] Arif, M., Ahmed, S., Ge, F., Kabir, M., Khan, Y. D., Yu, D. J., & Thafar, M. (2022). StackACPred: Prediction of anticancer peptides by integrating optimized multiple feature descriptors with stacked ensemble approach. *Chemometrics and Intelligent Laboratory Systems*, *220*, 104458.

[44] Tyagi, A., Tuknait, A., Anand, P., Gupta, S., Sharma, M., Mathur, D., …, & Raghava, G. P. (2015). CancerPPD: A database of anticancer peptides and proteins. *Nucleic Acids Research*, *43*, D837–D843.

[45] Liu, J., Li, M., & Chen, X. (2022). AntiMF: A deep learning framework for predicting anticancer peptides based on multi-view feature extraction. *Methods*, *207*, 38–43.

[46] Wang, Z., Meng, J., Dai, Q., Li, H., Xia, S., Yang, R., & Luan, Y. (2024). DeepPepPI: A deep cross-dependent framework with information sharing mechanism for predicting plant peptide-protein interactions. *Expert Systems with Applications*, *252*, 124168.

[47] Wang, H., Zhao, J., Zhao, H., Li, H., & Wang, J. (2021). CL-ACP: A parallel combination of CNN and LSTM anticancer peptide recognition model. *BMC Bioinformatics*, *22*, 512.

[48] Zhang, X., Wei, L., Ye, X., Zhang, K., Teng, S., Li, Z., …, & Wei, L. (2023). SiameseCPP: A sequence-based Siamese network to predict cell-penetrating peptides by contrastive learning. *Briefings in Bioinformatics*, *24*, bbac545.

[49] Song, H., Lin, X., Zhang, H., & Yin, H. (2024). ACP-ESM2: The prediction of anticancer peptides based on pre-trained classifier. *Computational Biology and Chemistry*, *110*, 108091.

[50] Zhang, F., Li, J., Wen, Z., & Fang, C. (2024). FusPB-ESM2: Fusion model of ProtBERT and ESM-2 for cell-penetrating peptide prediction. *Computational Biology and Chemistry*, *111*, 108098.

[51] He, S., Ye, X., Sakurai, T., & Zou, Q. (2023). MRMD3.0: A python tool and webserver for dimensionality reduction and data visualization via an ensemble strategy. *Journal of Molecular Biology*, *435*, 168116.

[52] Tang, H., Zhao, Y. W., Zou, P., Zhang, C. M., Chen, R., Huang, P., & Lin, H. (2018). HBPred: A tool to identify growth hormone-binding proteins. *International Journal of Biological Sciences*, *14*, 957.

[53] Sanz, H., Valim, C., Vegas, E., Oller, J. M., & Reverter, F. (2018). SVM-RFE: Selection and visualization of the most relevant features through non-linear kernels. *BMC Bioinformatics*, *19*, 1–18.

[54] Jo, I., Lee, S., & Oh, S. (2019). Improved measures of redundancy and relevance for mRMR feature selection. *Computers*, *8*, 42.

[55] Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*, 185–205.

[56] Ge, G., & Zhang, J. (2023). Feature selection methods and predictive models in CT lung cancer radiomics. *Journal of Applied Clinical Medical Physics*, *24*, e13869.

[57] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*, e1249.

[58] Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*, 241–258.

[59] González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, *64*, 205–237.

[60] Cai, C., Lin, H., Wang, H., Xu, Y., Ouyang, Q., Lai, L., & Pei, J. (2023). miDruglikeness: Subdivisional drug-likeness prediction models using active ensemble learning strategies. *Biomolecules*, *13*, 29.

[61] Wang, H., Guo, F., Du, M., Wang, G., & Cao, C. (2022). A novel method for drug-target interaction prediction based on graph transformers model. *BMC Bioinformatics*, *23*, 459.

[62] Fan, H., Yan, W., Wang, L., Liu, J., Bin, Y., & Xia, J. (2023). Deep learning-based multi-functional therapeutic peptides prediction with a multi-label focal dice loss function. *Bioinformatics*, *39*, btad334.

How to Cite: Zheng, B., Li, R., Wang, H., Wang, S., Peng, S., Li, M., …, & Lv, Z. (2025). iACP-SEI: An Anticancer Peptide Identification Method Incorporating Sequence Evolutionary Information. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS52024821