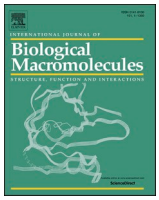
International Journal of Biological Macromolecules 282 (2024) 136940

Contents lists available at ScienceDirect

International Journal of Biological Macromolecules

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



# Voting-ac4C:Pre-trained large RNA language model enhances RNA N4-acetylcytidine site prediction

Yanna Jia [a], Zilong Zhang [a], Shankai Yan [a], Qingchen Zhang [a], Leyi Wei [b,c], Feifei Cui [a,*]

[a] School of Computer Science and Technology, Hainan University, Haikou 570228, China
[b] Centre for Artificial Intelligence driven Drug Discovery, Faculty of Applied Science, Macao Polytechnic University, Macao SAR, China
[c] School of Informatics, Xiamen University, Xiamen, China

ARTICLE INFO

*Keywords:*
N4-acetylcytidine
Feature extraction
RNAErnie
Soft voting
Ensemble learning

ABSTRACT

RNA N4-acetylcytidine (ac4C) modification plays a crucial role in gene expression regulation. However, existing prediction methods face limitations in capturing RNA sequence features, particularly in handling sequence complexity and long-range dependencies. To enhance the accuracy of RNA-ac4C modification sites prediction, this study introduces, for the first time, the transformer-based RNAErnie pre-trained model, which deeply extracts semantic information from RNA sequences. This model is combined with six traditional feature extraction methods (such as One-hot, ENAC, etc.) to form a multidimensional feature set. On this basis, we propose the Voting-ac4C model, which utilizes a deep neural network for feature selection. The selected features are then fed into a soft voting ensemble learning model, integrating the strengths of various machine learning algorithms to predict RNA-ac4C modification sites. Experimental results demonstrate that compared to the state-of-the-art methods, Voting-ac4C achieves significant improvements across multiple metrics, including AUC, SN, SP, ACC, and MCC. This study provides a novel approach for RNA modification sites prediction and highlights the potential applications of pre-trained models in biological sequence analysis.

## 1. Introduction

RNA modification refers to the post-transcriptional chemical alteration process of RNA molecules. Among these modifications, N4-acetylcytidine (ac4C) is a common type catalyzed by the enzyme *N*-acetyltransferase 10 (NAT10) [1,2], which adds an acetyl group to the nitrogen at position 4 of the cytidine base. Initially discovered in eukaryotic and prokaryotic tRNA and rRNA, recent studies have also identified ac4C in human mRNA [3]. This discovery demonstrated its involvement in regulating gene expression, maintaining mRNA stability, and its association with various diseases. In summary, ac4C modification serves as a critical post-transcriptional modification of RNA, playing essential roles in cellular functions and disease processes [4]. Investigating the functions and mechanisms of RNA ac4C modification sites is crucial for elucidating its biological significance and developing therapeutic approaches for associated diseases.

In recent years, numerous bioinformatic tools have been developed for identifying ac4C modification sites in mRNA. Initially, Zhao et al. introduced a predictor named PACES [5], which utilized position-specific dinucleotide sequence profile and K-nucleotide frequencies as encoding methods. Random Forest (RF) was employed as the machine learning algorithm to train the model and generate results. Subsequently, Alam et al. proposed an ensemble model named XG-ac4C for predicting ac4C modification sites [6]. This model incorporated six different types of feature encodings, including Nucleotide chemical property (NCP), Nucleotide density (ND), K-mer, One-hot encoding, EIIP+PseEIIP. Extreme Gradient Boosting (XGboost) was used as the classification algorithm. Following that, several models for predicting ac4C modification sites were developed. Su et al. developed the iRNA-ac4C model [7], which employed three feature extraction methods: K-mer, Nucleotide chemical property (NCP), and Accumulated nucleotide frequency (ANF). They used the GBDT algorithm combined with the mRMR-IFS strategy for prediction. Lou et al. utilized three encoding methods (Kmer, PseKNC, and PseEIIP) and employed a stacking-based algorithm in their classification approach to identify ac4C modification sites, termed Stacking-ac4C [8]. Li et al. developed MetaAC4C [9], which leveraged pre-trained Bidirectional Encoder Representations from Transformers (BERT). The model is based on a Bidirectional LSTM architecture, incorporating attention mechanisms and residual connections. To address data imbalance, they also utilized a generative

* Corresponding author.
*E-mail address:* feifeicui@hainanu.edu.cn (F. Cui).

https://doi.org/10.1016/j.ijbiomac.2024.136940
Received 3 September 2024; Received in revised form 11 October 2024; Accepted 24 October 2024
Available online 28 October 2024

adversarial network to generate synthetic feature samples. Wang et al. introduced DeepAC4C [10], a model that used Convolutional Neural Networks (CNN) for prediction. This network combined hybrid features composed of physicochemical patterns and nucleotide distribution representations. Most recently, Nhat Truong Pham et al. proposed the ac4C-AFL model [11], which utilized an adaptive feature representation strategy to extract the most representative features from RNA sequences. It also employs a novel ensemble feature importance scoring method to rank and select the most relevant features for accurate ac4C modification sites prediction. The above studies indicate that with the rapid development of deep learning and machine learning methods, many technical approaches can effectively improve the accuracy of RNA modification site prediction [12–15]. However, current methods still exhibit two major limitations: (1) These methods primarily relied on traditional feature extraction techniques (such as Kmer, PseEIIP, PseKNC) and simple machine learning models (such as SVM [16–18], Random Forest [19,20]). Although these approaches have improved prediction performance to some extent, they still face challenges related to insufficient feature representation and inadequate capture of contextual information. While traditional feature extraction methods can extract certain statistical information from sequences, they often fail to capture the higher-order structures and complex relationships within the sequences [21–23]; (2) Furthermore, existing machine learning models tend to suffer from overfitting when dealing with high-dimensional features [24], and their interpretability is limited, making them less effective in RNA modification sites prediction tasks.

To overcome the aforementioned limitations, we propose Voting-ac4C. This study is the first to introduce the Transformer-based RNAErnie pre-trained model for feature encoding [25]. RNAErnie, trained on large-scale RNA sequence data, effectively captures deep semantic information and contextual dependencies within the sequences. At the same time, we combined six traditional feature encoding methods to comprehensively capture the diverse characteristics of the sequences. The integrated features were subjected to dimension reduction using a deep neural network, and the resultant features were then fed into a soft voting ensemble learning model. This approach further enhanced the prediction performance and robustness of the model. The innovation of this study lies in the first combination of the RNAErnie pre-trained model with traditional feature encoding methods for RNA sequence feature encoding. This approach not only improves the accuracy of RNA-ac4C modification sites prediction but also offers new perspectives and tools for future RNA modification sites research.

## 2. Materials and methods

### 2.1. Benchmark datasets

The construction of reliable benchmark datasets is fundamental for developing robust predictors and understanding the underlying mechanisms of ac4C modification sites. Arango et al. utilized acRIP-seq to identify 4250 candidate ac4C peaks, revealing that the majority of acetylated genes possess 1–2 ac4C peaks [5]. However, due to the limited resolution of acRIP-seq, the specific sites corresponding to these ac4C peaks may not necessarily be cytidine. To establish dependable datasets, Su et al. selected the 100 nucleotides surrounding the cytidine closest to each ac4C peak as positive samples, while negative samples consisted of 201 nucleotides centered on cytidine randomly chosen from non-peak regions. Redundant sequences were subsequently removed using the CD-HIT tool with a similarity threshold of 0.8.

In this study, we utilized datasets derived from the research of Su et al. [7], which included an equal number of positive and negative samples, totaling 2758 each. All samples were divided into training and testing datasets using stratified sampling at a ratio of 4:1. The training set comprised 2206 positive and 2206 negative samples, while the independent testing set consisted of 552 positive and 552 negative samples. Detailed information regarding the datasets is presented in Table 1.

**Table 1**
Details of the benchmark datasets.

| Data types | Training datasets | Testing datasets |
|---|---|---|
| Positive | 2206 | 552 |
| Negative | 2206 | 552 |

### 2.2. Feature encoding

The first step in constructing a machine learning model for ac4C modification sites recognition is to encode RNA sequences while preserving as much of the original RNA information as possible. This is crucial for developing an accurate and robust model [26–30]. In this study, aimed at enhancing the accuracy and reliability of ac4C modification sites prediction, Voting-ac4C combines the Transformer-based RNAErnie pre-trained model with six traditional feature extraction methods, including One-hot, ENAC, C2, ND, TPCP, and Ksnpf [31]. This integrative approach leverages the RNAErnie model's strengths in capturing complex RNA sequence features while incorporating the established expertise of traditional methods, resulting in a more comprehensive feature representation. The next section will provide a detailed description of these methods.

#### 2.2.1. RNAErnie model

In this study, the RNAErnie model is employed for feature encoding in ac4C modification sites prediction. RNAErnie is a pre-trained model based on the Transformer architecture specifically designed for RNA sequences. Built upon the Enhanced Representation through Knowledge Integration (ERNIE) framework, this model incorporates multiple Transformer layers and multi-head self-attention mechanisms, with each Transformer block having a hidden state dimension of 768 [25]. These design choices enable RNAErnie to capture complex patterns and deep biological information within RNA sequences.

Firstly , RNAErnie is based on the Transformer architecture, with one of its core components being the multi-head attention mechanism. This mechanism enhances the model's ability to understand RNA sequences by capturing different aspects of the input sequence through parallel computation of multiple attention heads. For each attention head, the input sequence X is mapped into Query, Key, and Value matrices through different linear transformation matrices [32], as shown in Eqs. (1)–(3).

$$Q = XW^{Q} \tag{1}$$

$$K = XW^{K} \tag{2}$$

$$V = XW^{V} \tag{3}$$

Each attention head independently computes attention scores using the self-attention mechanism to assess the interrelationships among elements of the input sequence. The specific computation is defined by Eq. (4), where the dimension $d_k$ of the key vectors is utilized to scale the dot product of the query matrix Q and the key matrix K, preventing gradient issues. This scaled dot product generates raw attention scores, which are then transformed into a probability distribution through the softmax function. The distribution is used to weight the vectors in the value matrix V, yielding a contextual representation of the sequence.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V \tag{4}$$

The multi-head attention aggregates the outputs of each attention head by concatenating them and subsequently applying a linear transformation to derive the final representation, as shown in Eq. (5). Here, the concatenation operator Concat merges the individual head outputs, and $W^{O}$ is a trainable transformation matrix that maps the combined features into the desired output space.

$$\text{MultiHead}\,(Q, K, V) = \text{Concat}\,(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \quad (5)$$

Secondly, the RNAErnie model employs three masking strategies to enhance RNA sequence representation [25]. Base-level masking involves masking individual nucleotides to capture local features. Subsequence-level masking targets contiguous segments of the RNA sequence, aiming to capture long-range dependencies and global features. Meanwhile, motif-level random masking introduces randomness by masking nucleotides irrespective of their position or continuity, which improves the model's robustness and generalization capabilities.

Additionally, RNAErnie incorporates the coarse-grained types of RNA (e.g., mRNA, miRNA, lncRNA) as special vocabulary tokens, appending these type tokens to the end of each RNA sequence during pretraining [25]. This strategy enables the model to recognize and utilize RNA type-specific features when handling various downstream tasks, thereby enhancing the model's domain adaptation and task generalization capabilities.

RNAErnie leverages the multi-head attention mechanism to capture multidimensional features of RNA sequences, effectively addressing sequence complexity and long-range dependencies, thereby enhancing the accuracy of RNA-ac4C modification sites prediction. By adopting a multi-level masking strategy and incorporating RNA type tokens, RNAErnie strengthens feature representation, providing robust support for accurate prediction. This highlights the model's strong capabilities and broad applicability in RNA sequence analysis.

#### 2.2.2. One-hot encoding

One-hot encoding is a simple and effective feature extraction method, widely used in bioinformatics due to its ability to represent nucleotide sequences in an efficient and straightforward manner. It represents the four RNA bases, adenine (A), cytosine (C), guanine (G), and uracil (U), in RNA molecules as binary vectors composed of 0 s and 1 s. Specifically, this means that the nucleotides A, C, G, and U are represented by the four vectors (1,0,0,0), (0,1,0,0), (0,0,1,0), and (0,0,0,1), respectively [33].

#### 2.2.3. ND encoding

Nucleotide Density (ND) encoding is a common method in bioinformatics that represents each RNA sequence by combining nucleotide frequency with the position of individual nucleotides within the sequence. The main principle is to treat one or several bases in the RNA sequence as an element and calculate the frequency of that element within the RNA sequence. The frequency and positional distribution of each nucleotide are captured in the Nucleotide Density (ND) [31]. The density ($d_i$) of a nucleotide is calculated according to Eq. (6).

$$d_i = \frac{n}{i} \quad (6)$$

where n is the number of times the nucleotide appears before the i-th position (including the i-th position). For example, in the sequence "AUGCUCGAU", the density of U at positions 2, 5, and 9 would be 0.50, 0.40, and 0.33, respectively. Similarly, the density of C at positions 4 and 6 would be 0.25 and 0.33, respectively.

#### 2.2.4. C2 encoding

C2 encoding is a common sequence characterization model that converts the elements of a biological sequence into specific values from a global sequence perspective. Specifically, C2 encoding transforms the RNA bases on the nucleotide chain of RNA molecules into 2-bit binary values. For example, adenine (A) is encoded as (0,0), cytosine (C) as (1,1), guanine (G) as (1,0), and uracil (U) as (0,1) [34].

#### 2.2.5. ENAC encoding

ENAC (Encoding Nucleic Acid Composition) encoding is an effective method used to analyze and represent nucleic acid sequences. It employs a sliding window approach to calculate the nucleic acid composition within a fixed-length window, generating feature vectors for each window. This method captures local structural information within the sequence and provides useful feature representations for subsequent analysis and modeling. ENAC encoding is particularly effective for nucleic acid sequences of fixed lengths [35]. ENAC can be computed using Eq. (7).

$$V = \left[\frac{N_{A,win1}}{S}, \frac{N_{C,win1}}{S}, \frac{N_{G,win1}}{S}, \frac{N_{U,win1}}{S}, \frac{N_{A,win2}}{S}, \ldots, \frac{N_{G,win_{L-S+1}}}{S}, \frac{N_{U,win_{L-S+1}}}{S}\right] \quad (7)$$

S represents the size of the sliding window. $N_{t,winr}$ is the number of nucleotides t in the r-th sliding window. t represents the type of nucleotide, which can be A, C, G, or U. r is the index of the sliding window, ranging from 1 to $L - S + 1$, where L is the length of the sequence.

#### 2.2.6. TPCP encoding

Eleven physicochemical properties of TPCP were obtained from recent studies. These properties, which are listed in Table S1, were normalized according to the methods described in the article. For each sequence window containing a TPCP, a 1375-dimensional vector was created (comprising 125 trinucleotides × 11 physicochemical properties). Any tri-nucleotides containing the nucleobase N were assigned a value of zero [36].

#### 2.2.7. KSNPF encoding

KSNPF (k-spaced nucleotide pair frequencies) quantifies the occurrence of 16 nucleotide pairs that are separated by k arbitrary nucleotides within a sequence. By setting k to values of 0, 1, 2, 3, and 4, the sequence is transformed into various feature representations that reflect the frequency of these nucleotide pairs at different spacing intervals, as demonstrated in Eq. (8).

$$[f(AA)\ldots f(AXA)\ldots f(AXXA)\ldots f(AXXXA)\ldots f(AXXXXA)\ldots] \quad (8)$$

Here, X represents any nucleotide, and AXA refers to two adenines (A) with any nucleotide X in between. The term f(AXA) denotes the frequency of AXA in the sequence [37]. By calculating the frequency of all nucleotide pairs, the sequence is encoded into an 80-dimensional vector (16 pairs × 5 spacing values = 80).

### 2.3. Model framework

To effectively predict RNA-ac4C modification sites, we developed the model Voting-ac4C. The model consists of five parts: data collection, feature encoding, dimension reduction and modeling, model evaluation and web server, as illustrated in Fig. 1.

Firstly, we utilized benchmark datasets derived from the research of Su et al., which comprised 2758 positive and 2758 negative samples. The samples were carefully selected, with positive samples corresponding to the nucleotides surrounding ac4C peaks and negative samples sourced from non-peak regions. Using stratified sampling, the dataset was split into training (2206 positive and 2206 negative) and independent testing sets (552 positive and 552 negative) to ensure robust evaluation. Subsequently, a multi-dimensional feature encoding approach is employed for each RNA sequence derived from these datasets. This study introduces the use of the Transformer-based RNAErnie pre-trained large model for the first time in the field of RNA modification site prediction, enabling the extraction of global contextual features from RNA sequences. Additionally, it combines six traditional encoding methods including One-hot, ENAC, C2, ND, TPCP and Ksnpf to extract diverse features of RNA sequences, encompassing physicochemical properties and positional specificity. This hybrid representation of features effectively reflects both local and global information of the sequences, significantly enhancing the model's capacity for feature representation. Next, the generated high-dimensional features are input into a Deep Neural Network (DNN) for dimension reduction. Through

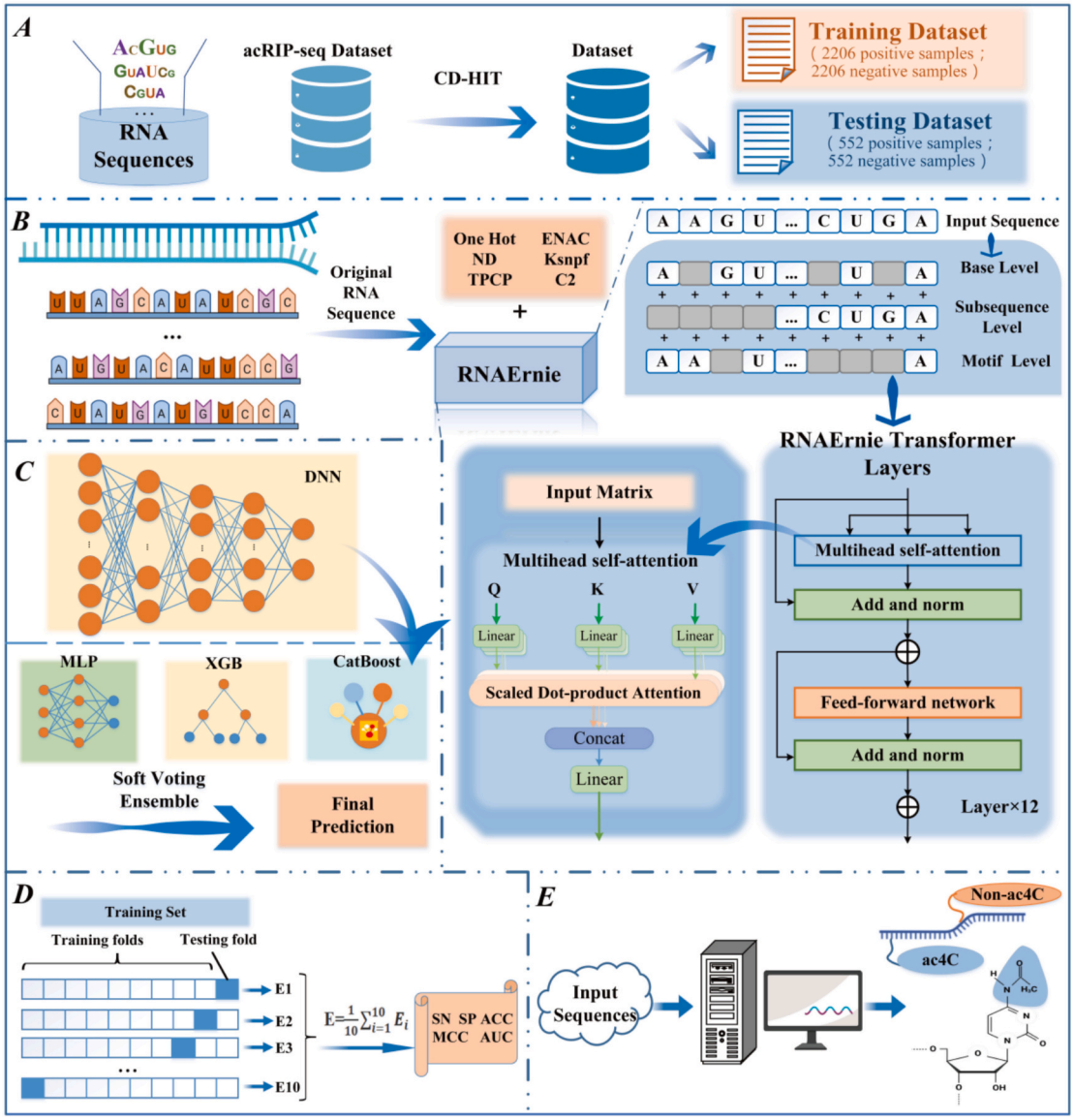



**Fig. 1. Overview of the proposed model. A. Data collection.** Utilizing datasets from Su et al., with an equal number of 2758 positive and 2758 negative samples, stratified into training (2206 each) and testing sets (552 each). **B. Feature encoding.** Processing RNA sequences with the RNAErnie model and six traditional methods, including One-hot, ENAC, C2, ND, TPCP and Ksnpf. **C. Dimension reduction and modeling.** Using a Deep Neural Network (DNN) for dimension reduction, followed by inputting the processed features into a soft voting ensemble model comprising Multi-layer Perceptron (MLP), XGBoost and CatBoost. **D. Model evaluation.** Employing ten-fold cross-validation to assess model's performance. **E. Web-sever.** Providing a user-friendly interface for inputting RNA sequences and obtaining predictions.

the multi-layer nonlinear mapping of the DNN, the features are processed from a high-dimensional space to a lower-dimensional space, effectively reducing computational complexity and storage requirements. The features processed by the DNN are then fed into a soft voting ensemble model constructed using XGBoost, MLP and CatBoost classifiers [38]. Afterward, we employed ten-fold cross-validation to ensure robustness and prevent overfitting. Independent testing datasets are also used to assess performance through metrics like accuracy, sensitivity, specificity, MCC and AUC. The final step is creating a user-friendly web server with an intuitive interface where users can input RNA sequences and obtain predictions for ac4C modification sites effectively.

### 2.3.1. Deep neural network

High-dimensional feature vectors can lead to the curse of dimensionality, resulting in overfitting in predictive models and prolonged computational times. To mitigate these challenges, Voting-ac4C integrates a hybrid feature set that combines a Transformer-based RNAErnie pre-trained large language model with traditional feature encoding methods, which is then input into a Deep Neural Network (DNN) for dimension reduction.

Deep Neural Network (DNN) consist of multiple interconnected layers, with each layer processing input data through non-linear functions to learn complex feature representations [39,40]. This deep architecture enables DNN to capture intricate patterns within high-dimensional data, which is particularly significant in the context of RNA modification site prediction [41]. Through a multi-layer non-linear mapping process, DNN can transform high-dimensional features into lower-dimensional representations, effectively reducing computational complexity and storage requirements while retaining key features crucial for enhancing prediction accuracy [42]. In the process of dimension reduction, we ultimately selected the second-to-last layer of the DNN as the output, which contains rich feature information.

Moreover, DNN is capable of capturing complex interactions

between different features, allowing for a higher level of feature representation. This dimension reduction method not only reduces the interference of redundant information but also enhances the model's understanding of feature importance, significantly improving predictive accuracy. Detailed parameters utilized in the DNN are provided in Supplementary Material S2.

#### 2.3.2. Model selection

Voting-ac4C employs three different machine learning algorithms, namely CatBoost, XGBoost (XGB), and Multi-Layer Perceptron (MLP), all implemented using the Scikit-Learn package. CatBoost is a Gradient Boosting Decision Tree (GBDT) framework that utilizes symmetric decision trees, requiring fewer hyperparameters and supporting categorical variables with high accuracy. CatBoost addresses challenges such as efficiently handling categorical features, gradient bias, and prediction shift. By mitigating these issues, CatBoost reduces overfitting, thereby enhancing the algorithm's accuracy and generalization ability [43]. XGBoost (eXtreme Gradient Boosting) is an efficient library that implements the Gradient Boosting algorithm. It trains decision tree models using gradient boosting, combining loss functions, regularization terms, and gradient information to optimize the model effectively [44]. MLPClassifier is a powerful classification algorithm that achieves complex pattern recognition through multiple layers of non-linear transformations. Its core idea is to learn intricate features within the data using a deep network structure. The MLPClassifier consists of an input layer, one or more hidden layers, and an output layer [45].

#### 2.3.3. Soft voting

Soft Voting is an ensemble learning method used to combine the predictions of multiple classifiers to improve classification performance [46]. This method calculates the final prediction probability by taking a weighted average of the prediction probabilities from all classifiers. The core formula of the soft voting model is shown in Eq. (9).

$$\hat{y} = arg\,max_c \sum_{i=1}^{n} w_i . P_i(y = c|x) \tag{9}$$

The final class prediction is based on the maximum value of the weighted average probabilities. In other words, the class with the highest predicted probability is selected as the final prediction result. Here, $\hat{y}$ represents the final prediction, c denotes the class, n is the number of base models, $w_i$ is the weight of the i-th model, and $P_i\,(y = c|x)$ is the probability predicted by the i-th model that the sample belongs to class c.

#### 2.3.4. Model evaluation metrics

To comprehensively evaluate our models' performance, we considered five key evaluation metrics based on previous research: sensitivity (SN), specificity (SP), overall accuracy (ACC), Matthews correlation coefficient (MCC), and area under the curve (AUC) [47–49]. These metrics serve as fundamental indicators reflecting the predictor's performance across various aspects [50]. The calculation formulas are as follows:

$$SN = \frac{TP}{TP + FN} \tag{10}$$

$$SP = \frac{TN}{TN + FP} \tag{11}$$

$$ACC = \frac{TP + FN}{TP + TN + FP + FN} \tag{12}$$

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

Here, TP, FN, TN, and FP represent the counts of true positives, false negatives, true negatives, and false positives, respectively. ACC (accuracy) evaluates the model's proficiency in correctly predicting both positive and negative samples. SN (sensitivity) and SP (specificity) depict the predictor's aptitude in identifying positive and negative samples, respectively. MCC (Matthews correlation coefficient) provides a balanced assessment of the predictor's performance, considering the sample distribution. AUC (area under the curve) reflects the overall efficacy of the model, where higher values signify superior predictive performance [51].

Cross-validation, a widely employed statistical strategy for model evaluation across various classification problems, entails partitioning the datasets into multiple subsets for iterative training and testing. In this study, to optimize computational efficiency without compromising accuracy, we adopted a 10-fold cross-validation approach. This method involved dividing the datasets into ten equally sized folds, iteratively training the model on nine folds while using the remaining fold for validation, and repeating this process until each fold had served as both training and validation data [52]. Subsequently, we assessed the performance of our models and compared them with other methodologies using independent datasets, ensuring robustness and reliability in our evaluation.

#### 2.3.5. Web server development

To make Voting-ac4C more convenient for users and improve its practicality, we have developed a user-friendly online server. Biologists can use this web server to obtain ac4C modification sites prediction results without any complex mathematical calculations. In this process, users simply input RNA sequences and submit them to get the desired results. The web server is available at http://www.bioai-lab.com/ac4C and is open and free to everyone.

## 3. Results and discussion

### 3.1. Evaluating the impact of RNAErnie on prediction results

In current bioinformatics research, feature encoding methods are crucial for the accuracy and effectiveness of RNA sequence analysis. The Voting-ac4C model employs RNAErnie along with six traditional feature encoding methods to process the original RNA data. To investigate the potential advantages of RNAErnie in feature extraction, we compared the methods that utilize RNAErnie with those that rely solely on the six traditional feature encoding methods. The experimental results, as shown in Table 2, the model incorporating RNAErnie exhibits significant improvements across all key performance metrics, including sensitivity (SN), specificity (SP), accuracy (ACC), Matthews correlation coefficient (MCC), and area under the ROC curve (AUC).

These results demonstrate that RNAErnie excels at capturing the complex features of RNA sequences, thereby enhancing the model's predictive performance. This advantage is likely due to RNAErnie's ability to leverage large-scale pre-trained representations of RNA sequences, which more effectively capture the underlying patterns of RNA modifications, surpassing the performance of traditional methods alone. Moreover, integrating RNAErnie allows the model to generalize better, as it captures both global and local structural features of RNA sequences, a task that traditional encoding methods struggle to achieve.

**Table 2**
Performance metrics comparison between models using RNAErnie and those not using RNAErnie. "Non-RNAErnie" refers to the model using only six traditional feature encoding methods (One-hot, ENAC, C2, ND, TPCP and Ksnpf). "RNAErnie" refers to the model using a combination of RNAErnie and the six traditional encoding methods (One-hot, ENAC, C2, ND, TPCP and Ksnpf). Bold values indicate the method that achieves the best performance.

| Method | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
|---|---|---|---|---|---|
| Non-RNAErnie | 80.97 | 79.89 | 80.43 | 60.87 | 87.92 |
| RNAErnie | **85.14** | **81.15** | **83.15** | **66.35** | **88.73** |

In conclusion, the use of RNAErnie not only significantly boosts performance but also enriches the model's ability to represent RNA sequence features, providing strong evidence of the superiority of the RNAErnie pre-trained model in predicting RNA modification sites.

### 3.2. Comparison with other feature extraction methods

In this study, we combine the transformer-based RNA pre-trained model RNAErnie with six traditional feature encoding methods. We focus on two main questions: First, does the use of different feature encoding methods directly impact the model's performance? Second, are combined feature encoding methods more effective? To address these questions, we firstly conducted experiments with individual feature encodings. And then we performed ablation experiments by combining different features to observe their impact on the overall performance of the model [47].

We first encoded each feature individually and evaluated the model's performance. It can be seen from Table 3 that the RNAErnie method is overall superior to traditional feature encoding methods. The superior result underscores the strength of RNAErnie as a large language model, excelling in capturing more comprehensive features and enabling a deeper understanding of RNA sequences compared to traditional methods.

After evaluating the performance of individual methods, we began exploring the impact of feature combinations on model performance. We conducted several combination experiments by progressively adding features. The initial feature encoding combination of RNAErnie+Ksnpf demonstrated strong performance, indicating that the encoding capability of the RNAErnie pre-trained model provided a solid foundation for subsequent feature combinations. In bioinformatics tasks, pre-trained models can capture deep-level features of RNA sequences, offering valuable information for classification tasks. As more features were progressively added, the overall performance of the model showed a steady upward trend in Table 4. This suggests that complementary relationships may exist among the different features, with each capturing unique aspects of the RNA sequences, thus helping the model develop a more comprehensive understanding of the data.

The proposed Voting-ac4C model leverages the RNAErnie pre-trained model along with six traditional feature encoding methods, enabling a more comprehensive feature extraction process. This multi-dimensional feature capture ability allows the model to gain a deeper understanding of RNA sequences, fully utilizing the advantages of each feature type. Consequently, it highlights the superiority of the Voting-ac4C model in terms of feature extraction and classification performance.

**Table 3**
Performance comparison of eleven different individual feature encoding methods. RNAErnie refers to the encoding method using a pre-trained large model, while the others are traditional feature encoding methods, including Ksnpf, ENAC, C2, One-hot, ND, TPCP, Kmer, PCP, PseEIIP and Knfc. Bold values indicate the method that achieves the best performance.

| Methods | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
|---|---|---|---|---|---|
| RNAErnie | **80.73** | 77.69 | **79.21** | **58.45** | **87.55** |
| Ksnpf | 76.08 | 78.26 | 77.17 | 54.36 | 85.42 |
| ENAC | 69.38 | **80.25** | 74.81 | 49.93 | 82.64 |
| C2 | 64.85 | 76.63 | 70.74 | 41.77 | 77.15 |
| One-hot | 67.21 | 74.99 | 71.1 | 42.33 | 78.57 |
| ND | 65.76 | 76.81 | 71.28 | 42.83 | 79.13 |
| TPCP | 69.02 | 78.44 | 73.73 | 47.67 | 82.08 |
| Kmer | 32.42 | 68.47 | 50.45 | 0.97 | 51.59 |
| PCP | 31.15 | 68.11 | 49.63 | −0.77 | 49.88 |
| PseEIIP | 17.21 | **80.25** | 48.73 | −3.26 | 50.94 |
| Knfc | 29.16 | 67.93 | 48.55 | −3.12 | 48.87 |

**Table 4**
Performance comparison of RNAErnie combined with different hybrid feature encoding methods. Bold values indicate the method that achieves the best performance.

| Methods | SN (%) | SP (%) | ACC (%) | MCC (%) | AUC (%) |
|---|---|---|---|---|---|
| RNAErnie+Ksnpf | 81.52 | 77.89 | 79.71 | 59.45 | 86.80 |
| RNAErnie+Ksnpf+ENAC | 81.34 | 79.34 | 80.34 | 60.70 | 86.97 |
| RNAErnie+Ksnpf+ENAC+C2 | 81.81 | 80.61 | 80.71 | 61.46 | 87.00 |
| RNAErnie+Ksnpf+ENAC+C2 + One-hot | 80.25 | **81.70** | 80.97 | 61.96 | 88.09 |
| RNAErnie+Ksnpf+ENAC+C2 + One-hot+ND | 82.06 | 80.79 | 81.43 | 62.86 | 82.27 |
| RNAErnie+Ksnpf+ENAC+C2 + One-hot+ND + TPCP | **85.14** | 81.15 | **83.15** | **66.35** | **88.73** |

### 3.3. Comparison with other feature processing methods

In this study, Deep Neural Network (DNN) are employed as a dimension reduction technique, effectively mapping high-dimensional features to a lower-dimensional space and capturing complex patterns within the data. Most RNA modification site prediction models incorporate feature selection algorithms, which aid in enhancing computational efficiency, eliminating redundant data, and preventing overfitting. If Voting-ac4C does not process the integrated features, it is likely to lead to dimensionality issues, such as overfitting.

To determine the optimal feature processing method for this model, Voting-ac4C evaluated five potential methods: Variance Threshold [53], LASSO [54], ANOVA [55], Random Forest (RF) [56], PCA [57], and RFECV [58] and DNN. Variance Threshold removed features with a variance less than 0.01 to reduce feature redundancy. To maintain consistency, the LASSO, ANOVA, RF, and PCA methods filtered out 50 % of the feature vectors. Recursive Feature Elimination with Cross-Validation (RFECV) method automatically selects the optimal subset of features. A detailed introduction to DNN has been provided in section 2.3.1. All other parameters of the model remained unchanged except for the variation in methods. We utilize five evaluation metrics (Sensitivity, Specificity, Accuracy, MCC, and AUC) to demonstrate the impact of different methods on the prediction results of RNA ac4C modification sites.

From Table 5 and Fig. 2, the Deep Neural Network (DNN) demonstrates superior performance across evaluated metrics, particularly in Sensitivity (SN) and Area Under the Curve (AUC), reaching 85.14 % and 88.73 %, respectively. This effectiveness may stem from the DNN's ability to transform high-dimensional features into a lower-dimensional representation, thereby improving the model's efficiency. Other methods, including Random Forest (RF), ANOVA, Variance Threshold, LASSO, RFECV and PCA, generally exhibit lower predictive performance compared to the DNN method. This may be due to the limitations of these methods in the feature processing process, which results in their failure to retain features that play a crucial role in prediction, thereby

**Table 5**
Performance comparison of seven different feature processing methods. DNN maps high-dimensional features to a lower-dimensional space. The LASSO, ANOVA, RF and PCA methods filtered out 50 % of the feature vectors. RFECV method automatically selects the optimal subset of features. Variance Threshold removed features with a variance less than 0.01. Bold values indicate the method that achieves the best performance.

| Method | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
|---|---|---|---|---|---|
| DNN | **85.14** | 81.15 | **83.15** | **66.35** | **88.73** |
| RF | 79.89 | 80.79 | 80.34 | 60.69 | 87.21 |
| ANOVA | 76.99 | **82.06** | 79.52 | 59.13 | 86.79 |
| Variance Threshold | 77.53 | 80.97 | 79.25 | 58.54 | 87.02 |
| LASSO | 79.16 | 78.8 | 78.98 | 57.97 | 86.98 |
| RFECV | 77.35 | 77.17 | 77.26 | 54.52 | 85.95 |
| PCA | 49.09 | 92.57 | 70.83 | 46.26 | 84.50 |

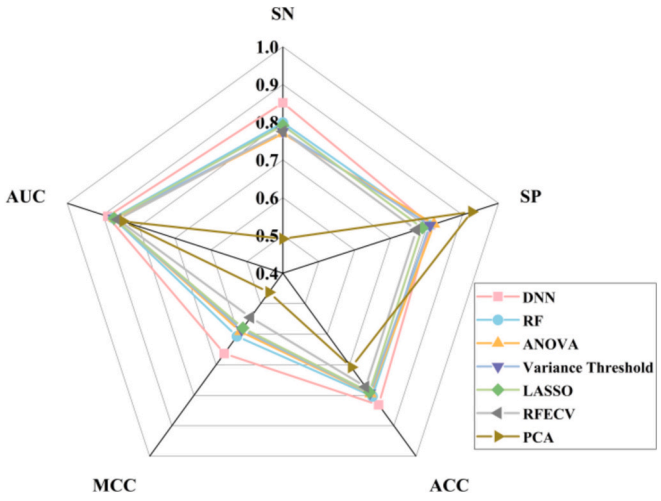



**Fig. 2. Performance comparison of seven different feature processing methods.** The radar chart shows that although the Deep Neural Network (DNN) is slightly lower than PCA in terms of specificity (SP), overall, the performance of DNN is superior to other feature processing methods.

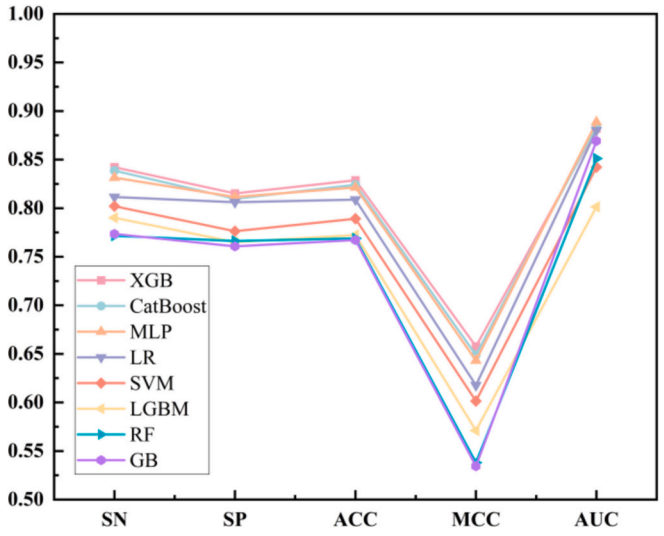



**Fig. 3. The prediction performance of eight different machine learning classifiers.** The line chart indicates that XGB (Extreme Gradient Boosting), CatBoost (a gradient boosting-based algorithm) and MLP (Multilayer Perceptron) outperform other classifiers across five metrics.

affecting the overall predictive capability of the model. Collectively, the DNN demonstrates its effectiveness in mapping high-dimensional features to a lower-dimensional space, facilitating the simplification of feature representation for subsequent analysis.

### 3.4. Comparison with other single machine learning cl-assifiers

In constructing the RNA ac4C modification sites prediction model, we compared various machine learning classifiers, including XGBoost [44], CatBoost, MLP, Logistic Regression (LR), SVM, LightGBM (LGBM), Random Forest (RF) and Gradient Boosting (GB) [59]. To further enhance prediction performance, we selected the classifiers that performed well (with ACC exceeding 82 %)—XGBoost, CatBoost, and MLP—to build a soft voting ensemble model.

Table 6 presents the performance of various classifiers in terms of SN, SP, ACC, MCC, and AUC metrics. Among the compared models, XGBoost, CatBoost, and MLP all demonstrated high ACC and AUC values, indicating their strong capability in capturing RNA features. XGBoost excelled in ACC and MCC due to its effective handling of high-dimensional features. CatBoost slightly outperformed others in AUC, attributed to its automatic handling of categorical features and robust decision tree structure. MLP achieved the best AUC performance by capturing complex relationships in the data through its neural network architecture.

It can be observed that LR, SVM, LGBM, RF and GB performed slightly worse in various aspects compared to the other three classifiers in Fig. 3, and thus were not included in the final ensemble model. Overall, XGBoost, CatBoost, and MLP exhibited strong performance across multiple metrics, making them ideal candidates for constructing the soft voting ensemble model. By combining the prediction probabilities of these classifiers, the soft voting method effectively enhanced the overall predictive performance of the model, achieving an AUC of 88.73 % on the independent testing set, demonstrating excellent generalization ability.

**Table 6**
Performance comparison with eight different machine learning classifiers. Bold values indicate the method that achieves the best performance.

| Method | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
|---|---|---|---|---|---|
| XGB | **84.23** | **81.52** | **82.88** | **65.78** | 88.22 |
| CatBoost | 83.87 | 80.97 | 82.42 | 64.88 | 88.46 |
| MLP | 83.15 | 81.15 | 82.15 | 64.32 | **88.86** |
| LR | 81.15 | 80.61 | 80.88 | 61.77 | 88.02 |
| SVM | 80.21 | 77.63 | 78.92 | 60.15 | 84.22 |
| LGBM | 79.02 | 76.51 | 77.24 | 57.12 | 80.14 |
| RF | 77.17 | 76.63 | 76.9 | 53.8 | 85.11 |
| GB | 77.35 | 76.08 | 76.72 | 53.44 | 86.92 |

### 3.5. Comparison with other ensemble learning methods

To enhance the model's predictive performance, we compared several common ensemble learning methods, including Blending [60], Stacking [61,62], Bagging [63], Hard Voting [64], and Soft Voting [38]. These ensemble methods offer different ways to leverage the strengths of multiple models, and their effectiveness can vary depending on the specific characteristics of the datasets and the base models used.

From the Table 7 and Fig. 4, it can be observed that the performance of other ensemble learning methods, such as Hard Voting, Bagging, Blending and Stacking, is slightly inferior to that of the Soft Voting ensemble method. This may be because Hard Voting only considers the final predicted class, ignoring the confidence levels of different models, potentially leading to information loss. Bagging generates multiple models through random sampling, but the simple averaging might fail to fully leverage the diversity of models. Blending and Stacking rely on a meta-learner, which can increase the complexity of the model and lead to overfitting, especially when the dataset is relatively small.

The superiority of the Soft Voting method in our experiments can be attributed to its approach of aggregating the predicted probabilities for each class from multiple base models. This method leverages the predictive information from each model more comprehensively, effectively combining the strengths of different base models and enhancing overall prediction performance. Additionally, Soft Voting demonstrates better robustness in the face of class imbalance issues by considering the predicted probabilities for each class, rather than simply selecting the most

**Table 7**
Performance comparison with five different ensemble learning methods. Bold values indicate the method that achieves the best performance.

| Methods | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
|---|---|---|---|---|---|
| Blending | 81.70 | 79.89 | 80.79 | 61.60 | 88.05 |
| Stacking | 83.51 | 80.43 | 81.97 | 63.97 | 87.84 |
| Bagging | 84.96 | 80.07 | 82.51 | 65.11 | 88.38 |
| Hard Voting | **86.23** | 79.25 | 82.88 | 65.90 | – |
| Soft Voting | 85.14 | **81.15** | **83.15** | **66.35** | **88.73** |

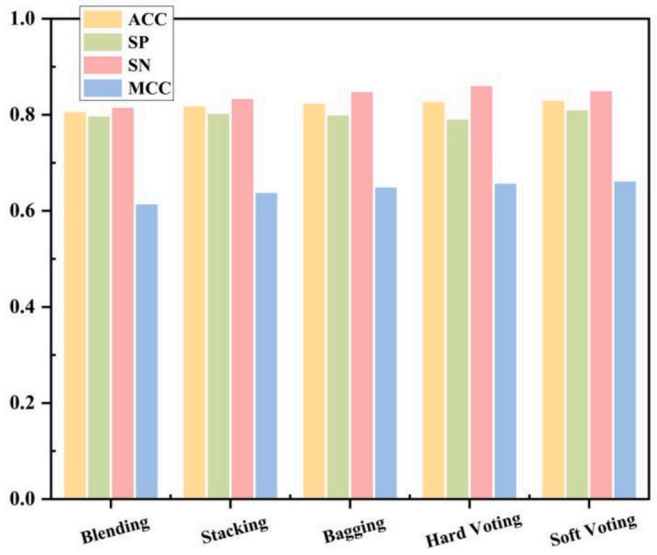



**Fig. 4. Performance Comparison of Four Prediction Metrics for Five Different Ensemble Learning Methods.** The bar chart illustrates the performance metrics of various ensemble learning methods, including Blending, Stacking, Bagging, Hard Voting and Soft Voting. The Soft Voting method demonstrates a slight advantage over the other techniques.

frequent class.

Overall, Soft Voting outperforms other ensemble learning methods across multiple performance metrics. By effectively integrating the prediction probabilities from multiple models, it significantly improves prediction accuracy and stability. Future work could focus on further optimizing the selection and weighting strategies of base models to enhance prediction performance even further.

### 3.6. Comparison with other deep learning methods

In the task of predicting RNA modification sites, the choice of models directly affects their performance, which is a significant challenge in this field. To evaluate the performance of different models on complex features, we conducted comparative experiments involving deep learning models and their combinations, alongside the proposed soft voting ensemble learning model. The soft voting ensemble learning method outperforms all individual deep learning models and their combinations across several evaluation metrics in Table 8, including accuracy (ACC 83.15 %), Matthews Correlation Coefficient (MCC 66.35 %) and Area Under the Curve (AUC 88.88 %).

**Table 8**
Comparison of performance between the soft voting ensemble learning model and various deep learning models. Bold values indicate the method that achieves the best performance.

| Methods | SN (%) | SP (%) | ACC (%) | MCC (%) | AUC (%) |
|---|---|---|---|---|---|
| BiGRU | 79.52 | 76.63 | 78.07 | 56.18 | 82.26 |
| BiLSTM | 75.36 | **81.88** | 78.62 | 57.36 | 82.15 |
| CNN | 82.78 | 74.63 | 78.71 | 57.61 | 84.82 |
| RNN | 81.7 | 78.07 | 79.89 | 59.82 | 86.72 |
| RNN + CNN | 81.7 | 80.25 | 80.97 | 61.96 | 87.78 |
| RNN + BiLSTM | **86.23** | 76.08 | 81.15 | 62.64 | 87.75 |
| RNN + BiGRU | **86.23** | 79.16 | 82.69 | 65.56 | 88.62 |
| RNN + BiLSTM+BiGRU | 85.14 | 79.16 | 82.15 | 64.42 | 88.87 |
| CNN + BiGRU | 84.42 | 78.63 | 81.52 | 63.14 | 88.31 |
| CNN + BiLSTM | 82.6 | 80.79 | 81.70 | 63.41 | 88.52 |
| CNN + BiLSTM+BiGRU | 81.88 | 80.79 | 81.34 | 62.68 | 88.68 |
| RNN + CNN + BiLSTM+BiGRU | **86.23** | 77.71 | 81.97 | 64.18 | 88.73 |
| Soft Voting | 85.14 | 81.15 | **83.15** | **66.35** | **88.88** |

In contrast, while individual deep learning models excel in capturing local dependencies or sequential features, they still exhibit limitations when faced with the complexity and diversity of data features. For instance, CNN is adept at capturing local features but lacks the ability to model global dependencies [65]. RNN and BiGRU, though effective at capturing sequential information, may underperform when handling high-dimensional features [66]. Although BiLSTM can capture temporal information from both forward and backward directions, its strong reliance on temporal dependencies makes it challenging to address non-sequential features, along with longer training times and higher computational complexity in complex high-dimensional data [65].

Even combinations of deep learning models (such as RNN + CNN or RNN + BiLSTM) can enhance performance in certain cases. However, they remain constrained by their respective structures and characteristics, failing to fully integrate diverse features. Consequently, the advantages of individual deep learning models and their combinations do not comprehensively address the complexities of the RNA modification site prediction task. The soft voting method effectively addresses these shortcomings by integrating various types of models. This approach harnesses the complementary strengths of each model, improving predictive accuracy and enhancing the ability to adapt to complex features, thereby increasing the model's generalization performance.

### 3.7. Comparison of existing state-of-the-art methods

To evaluate the performance of the proposed model in RNA ac4C modification sites prediction, we chose five benchmark models for comparison. These models include PACES, XG-ac4C, iRNA-ac4C, Auto-ac4C, and ac4C-AFL. Through this comparison, we aim to demonstrate the advantages of the proposed model across various performance metrics.

The comparison results with existing methods are summarized in Table 9. Our proposed model demonstrates superior performance across all evaluation metrics compared to existing models. Conducting 10-fold cross-validation on the models yielded evaluation metrics of 89.25 %, 84.31 %, 86.78 %, 73.66 % and 93.04 % for SN, SP, ACC, MCC, and AUC, respectively. Furthermore, evaluation on an independent testing set yielded metrics of 85.14 %, 81.15 %, 83.15 %, 66.35 % and 88.73 % for SN, SP, ACC, MCC, and AUC, respectively.

It is evident from Fig. 5 that our model outperforms PACES, XG-ac4C, iRNA-ac4C, Auto-ac4C, and ac4C-AFL in terms of SN, SP, ACC, MCC, and AUC. Regardless of whether employing machine learning or deep learning methodologies, our model consistently showcases superior predictive capabilities for prediction. These findings underscore the stability and superiority of our proposed model, positioning it as an effective and invaluable computational tool for discerning RNA-ac4C modification sites.

### 3.8. Model performance evaluation with an additional independent testing set

To ensure the robustness of model, we selected a new independent testing dataset to validate the performance of Voting-ac4C model. The new independent testing dataset includes 467 positive and 467 negative samples, sourced from the benchmark datasets used in prior research, specifically those curated by Li et al. [9]. This new dataset not only facilitates a more rigorous evaluation of the model's performance but also allows for deeper comparisons with other established ac4C site prediction models.

Table 10 and Fig. 6 present the performance metrics for each model. It is evident that our proposed model, Voting-ac4C, outperforms the other models overall. This is because the Voting-ac4C model integrates the RNAErnie large model with six traditional feature encoding methods, creating a more comprehensive feature extraction framework. This integration allows the model to leverage the strengths of various methods in feature capture, thereby enhancing its ability to understand

**Table 9**
Performance comparison with existing methods. Bold values indicate the method that achieves the best performance.

| Methods | Cross Validation | | | | | Independent testing | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
| PACES [5] | 78.38± 1.86 | 75.75 ± 2.95 | 77.06 ± 1.13 | 54.20 ± 2.19 | 84.84 ± 1.28 | 79.71 | 77.90 | 78.80 | 57.62 | 86.48 |
| XG-ac4C [6] | 93.38± 1.23 | 54.76 ± 2.03 | 74.07 ± 0.87 | 52.22 ± 1.62 | 85.24 ± 1.22 | 92.57 | 59.78 | 76.18 | 55.42 | 87.13 |
| iRNA-ac4C [7] | 77.02 | 83.01 | 80.03 | 60.1 | 87.5 | 76.70 | **82.91** | 79.81 | 59.70 | 88.00 |
| Auto-ac4C [34] | 85.08 ± 4.11 | 77.01 ± 3.61 | 81.05 ± 1.58 | 62.47 ± 3.33 | 87.97 ± 1.48 | 82.61 | 78.80 | 80.71 | 61.46 | 88.94 |
| ac4C-AFL [11] | 85.7 | 81.0 | 83.3 | 66.8 | 90.3 | 84.4 | 80.3 | 82.3 | 64.7 | **89.5** |
| Voting-ac4C (**proposed**) | **89.25** | **84.31** | **86.78** | **73.66** | **93.04** | **85.14** | 81.15 | **83.15** | **66.35** | 88.73 |

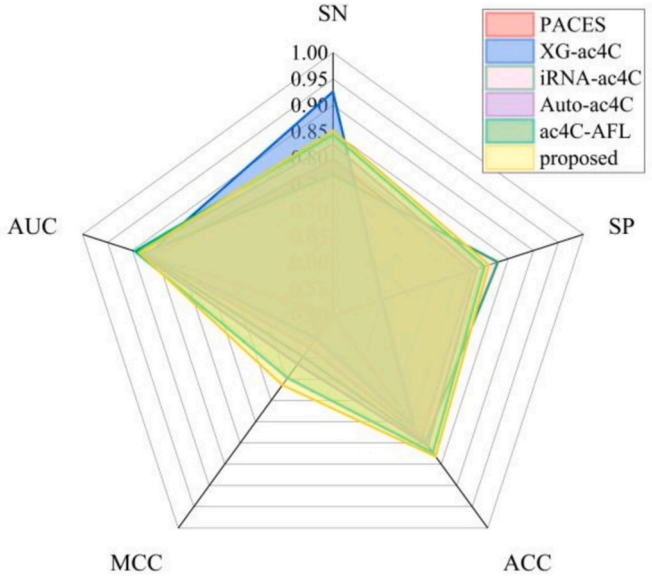



**Fig. 5. Performance Comparison of different models for ac4C modification site prediction on an independent testing set.** The proposed model, Voting-ac4C, demonstrates superior performance with the largest covered area compared to other methods, including PACES, XG-ac4C, iRNA-ac4C, Auto-ac4C, and ac4C-AFL.

**Table 10**
Performance comparison with four different ac4C site prediction models using the additional independent testing set. Bold values indicate the model that achieves the best performance.

| Model | SN(%) | SP(%) | ACC(%) | MCC(%) | AUC(%) |
|---|---|---|---|---|---|
| PACES | 8.12 | **98.29** | 53.21 | 14.84 | 53.22 |
| XG-ac4C | 58.23 | 94.43 | 76.33 | 56.50 | 76.33 |
| DeepAc4C | **82.80** | 75.58 | 79.19 | 58.57 | 86.49 |
| Voting-ac4C | 78.80 | 80.72 | **79.76** | **59.53** | **86.88** |

and analyze RNA sequences. Additionally, the soft voting method employed by Voting-ac4C combines the predictions from different approaches through weighted aggregation, reducing the bias and uncertainty inherent in any single method. This strategy not only improves the model's sensitivity and specificity but also ensures robustness in complex biological contexts.

As a result, Voting-ac4C achieves a favorable balance between sensitivity and specificity, surpassing other models such as PACES, XG-ac4C and DeepAc4C, thereby validating its effectiveness and reliability on the additional independent testing set. Overall, the Voting-ac4C approach demonstrates significant advantages in the prediction of RNA modification sites.

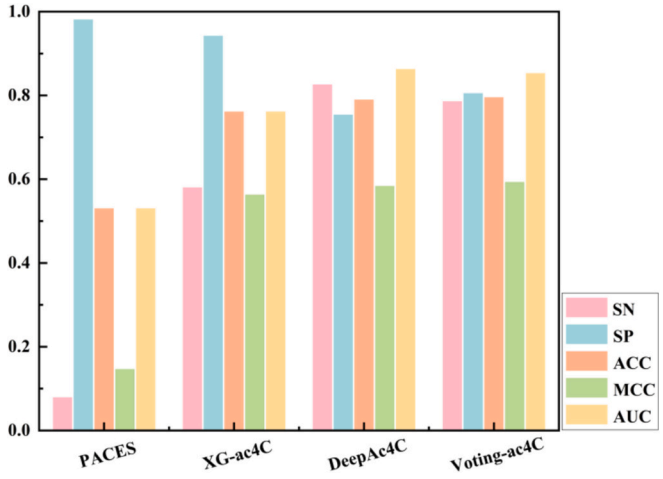



**Fig. 6. Performance comparison with four different ac4C site prediction models using the additional independent testing set.** The bar chart illustrates that while the proposed model, Voting-ac4C, does not achieve optimal results in individual metrics such as SN (Sensitivity) and SP (Specificity), it demonstrates superior overall performance compared to other models, including PACES, XG-ac4C, and DeepAc4C.

### *3.9. Model visualization proves model performance*

In this section, we use Kernel Density Estimation (KDE) plots to assess the model's classification performance and generalization ability [67]. Figs. 6 respectively show the predicted probability density distributions of the model on the training and testing sets. These plots clearly illustrate the model's classification performance for the positive class (1) and the negative class (0).

In Fig. 7A, the predicted probability density distributions for the positive and negative classes are well-separated. The predicted probabilities for the positive class are concentrated near 1.0, while those for the negative class are clustered around 0.0. This indicates that the model demonstrates strong classification ability on the training set and can accurately distinguish between positive and negative samples.

In Fig. 7B, the predicted probabilities for the positive and negative classes remain concentrated in their respective extreme regions, though there is some overlap between the two distributions. This suggests that the model's ability to discriminate between classes has decreased on the testing set. However, most samples in the testing set are still correctly classified, indicating that the model retains good generalization ability.

Overall, these two figures illustrate the model's performance on different datasets, demonstrating that the model exhibits high classification accuracy and robustness during both training and testing phases, though there remains room for improvement in its predictive ability on unseen data.

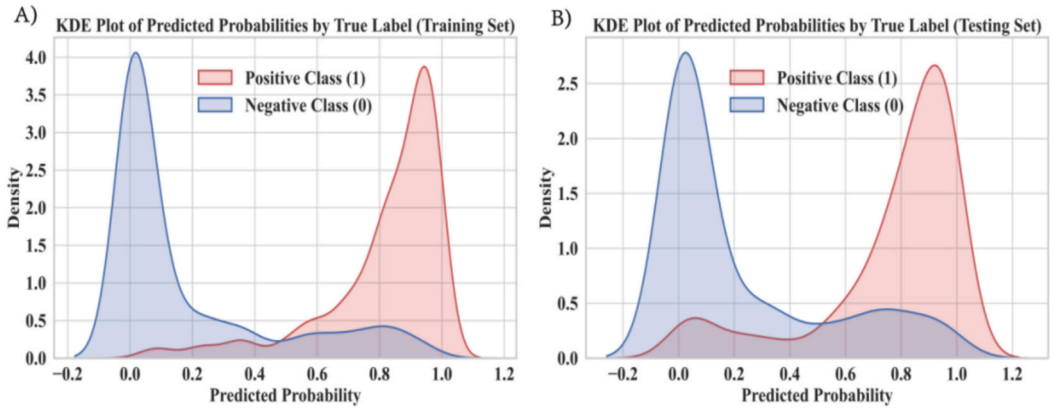



**Fig. 7. Model Visualization. A) KDE Plot on the Training Set.** This plot displays the kernel density estimation (KDE) of predicted probabilities for the positive class (1) and negative class (0). The red area represents the positive class, while the blue area indicates the negative class, with distinct peaks highlighting effective class separation. **B) KDE Plot on the Testing Set.** This plot illustrates the kernel density estimation (KDE) of predicted probabilities for the positive class (1) and negative class (0) in the testing set. The red and blue curves represent the model's predicted distributions for the positive and negative classes, reflecting the model's classification ability on unseen data.

## 4. Conclusion

This study presents an innovative model, Voting-ac4C, for predicting RNA N4-acetylcytidine (ac4C) modification sites. The model encodes data using RNAErnie and six traditional feature encoding methods, followed by dimension reduction through a Deep Neural Network (DNN). The features are then input into a soft voting ensemble learning model. Through ten-fold cross-validation and independent testing, the Voting-ac4C model outperformed existing state-of-the-art models across all metrics, demonstrating its exceptional performance in predicting RNA modification sites.

The innovation of this method lies in its novel integration of the RNAErnie pre-trained model with various traditional feature encoding methods, resulting in a multidimensional feature representation. RNAErnie, designed specifically for RNA sequences, effectively captures contextual information, thereby enhancing the accuracy of ac4C modification site predictions. However, reliance solely on the large model cannot comprehensively reflect all features of RNA sequences. Therefore, this study combines RNAErnie with six traditional feature encoding methods (One-hot, ENAC, C2, ND, TPCP, and Ksnpf), preserving the global feature extraction capabilities while also capturing sequence details through traditional encoding. Additionally, we employed a soft voting method that aggregates the predictions from multiple classifiers, effectively reducing the bias of individual models and achieving more stable predictions when handling complex data. Overall, the Voting-ac4C model demonstrated good robustness and generalization ability in predicting RNA N4-acetylcytidine (ac4C) modification sites and performed excellently in all aspects.

Although the Voting-ac4C model has shown notable success in predicting RNA modification sites, there are still some limitations that need to be addressed. Firstly, the selected feature encoding methods may not fully capture all biological factors influencing RNA modification sites, potentially limiting the model's ability to recognize certain key features. Future research could explore more comprehensive and diverse feature encoding [68,69] approaches to enhance model performance. Additionally, although the model demonstrates strong computational results, the lack of experimental validation of its biological applicability may affect its practical utility. Future work will aim to address this limitation by conducting further experimental validation to strengthen the model's biological relevance.

## CRediT authorship contribution statement

**Zilong Zhang:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Shankai Yan:** Writing – review & editing, Visualization, Supervision, Project administration. **Qingchen Zhang:** Writing – review & editing, Supervision, Resources, Project administration. **Leyi Wei:** Writing – review & editing, Visualization, Validation, Project administration, Investigation. **Feifei Cui:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

## Funding

The work was supported by the National Natural Science Foundation of China (Nos. 62101100, 62262015 , 62102064) and Science and Technology special fund of Hainan Province (ZDYF2024GXJS018) and Hainan Provincial Natural Science Foundation of China 324MS009.

## Declaration of competing interest

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

## Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ijbiomac.2024.136940.

## Data availability

The source code and experimental data is available at http://www.bioai-lab.com/ac4C.

## References

[1] G. Wang, et al., NAT10-mediated mRNA N4-acetylcytidine modification promotes bladder cancer progression, Clin. Transl. Med. 12 (5) (2022) e738.
[2] C. Yang, et al., Prognostic and immunological role of mRNA ac4C regulator NAT10 in Pan-Cancer: new territory for Cancer research? Front. Oncol. 11 (2021) 630417.
[3] B.S. Zhao, I.A. Roundtree, C. He, Post-transcriptional gene regulation by mRNA modifications, Nat. Rev. Mol. Cell Biol. 18 (1) (2017) 31–42.
[4] C. Ao, et al., m5U-SVM: identification of RNA 5-methyluridine modification sites based on multi-view features of physicochemical features and distributed representation, BMC Biol. 21 (1) (2023) 93.
[5] W. Zhao, et al., PACES: prediction of N4-acetylcytidine (ac4C) modification sites in mRNA, Sci. Rep. 9 (1) (2019) 11112.
[6] W. Alam, H. Tayara, K.T. Chong, XG-ac4C: identification of N4-acetylcytidine (ac4C) in mRNA using eXtreme gradient boosting with electron-ion interaction pseudopotentials, Sci. Rep. 10 (1) (2020) 20942.

[7] W. Su, et al., iRNA-ac4C: a novel computational method for effectively detecting N4-acetylcytidine sites in human mRNA, Int. J. Biol. Macromol. 227 (2023) 1174–1181.
[8] L.L. Lou, et al., Stacking-ac4C: an ensemble model using mixed features for identifying n4-acetylcytidine in mRNA, Front. Immunol. 14 (2023) 1267755.
[9] Z. Li, B. Jin, J. Fang, MetaAc4C: a multi-module deep learning framework for accurate prediction of N4-acetylcytidine sites based on pre-trained bidirectional encoder representation and generative adversarial networks, Genomics 116 (1) (2024) 110749.
[10] C. Wang, et al., DeepAc4C: a convolutional neural network model with hybrid features composed of physicochemical patterns and distributed representation information for identification of N4-acetylcytidine in mRNA, Bioinformatics 38 (1) (2021) 52–57.
[11] N.T. Pham, et al., ac4C-AFL: a high-precision identification of human mRNA N4-acetylcytidine sites based on adaptive feature representation learning, Mol. Ther. Nucleic Acids 35 (2) (2024) 102192.
[12] C. Ao, et al., Biological sequence classification: a review on data and general methods, Research 2022 (2022) 0011.
[13] C. Dai, et al., scIMC: a platform for benchmarking comparison and visualization analysis of scRNA-seq data imputation methods, Nucleic Acids Res. 50 (9) (2022) 4877–4899.
[14] J. Qiao, et al., Towards retraining-free RNA modification prediction with incremental learning, Inf. Sci. 660 (2024) 120105.
[15] H. Lin, Artificial intelligence with great potential in medical informatics: a brief review, Medinformatics 1 (1) (2024) 2–9.
[16] Y. Wang, Y. Zhai, Y. Ding, Q. Zou, SBSM-Pro: Support Bio-sequence Machine for Proteins, arXiv 67 (2024) 212106.
[17] L. Manganaro, et al., Non-small cell lung Cancer survival estimation through multi-omic two-layer SVM: a multi-omics and multi-sources integrative model, Curr. Bioinforma. 18 (8) (2023) 658–669.
[18] W. Zhu, et al., A first computational frame for recognizing heparin-binding protein, Diagnostics (Basel) 13 (14) (2023).
[19] L. Zhou, H. Wang, A combined feature screening approach of random Forest and filter-based methods for ultra-high dimensional data, Curr. Bioinforma. 17 (4) (2022) 344–357.
[20] X.Q. Ru, L.H. Li, Q. Zou, Incorporating distance-based top-n-gram and random Forest to identify Electron transport proteins, J. Proteome Res. 18 (7) (2019) 2931–2939.
[21] Y. Li, et al., msBERT-promoter: a multi-scale ensemble predictor based on BERT pre-trained model for the two-stage prediction of DNA promoters and their strengths, BMC Biol. 22 (1) (2024) 126.
[22] Q. Zou, et al., Gene2vec: gene subsequence embedding for prediction of mammalian N-6-methyladenosine sites from mRNA, Rna 25 (2) (2019) 205–218.
[23] X. Zou, et al., Accurately identifying hemagglutinin using sequence information and machine learning methods, Front. Med. (Lausanne) 10 (2023) 1281880.
[24] F. Cui, Z. Zhang, Q. Zou, Sequence representation approaches for sequence-based protein prediction tasks that use deep learning, Brief. Funct. Genomics 20 (1) (2021) 61–73.
[25] N. Wang, et al., Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning, Nat. Mach. Intell. 6 (5) (2024) 548–557.
[26] X. Fu, et al., Hyb_SEnc: an Antituberculosis peptide predictor based on a hybrid feature vector and stacked ensemble learning, IEEE/ACM Trans. Comput. Biol. Bioinform. (2024) 1–17.
[27] R. Liu, et al., AIPPT: Predicts anti-inflammatory peptides using the most characteristic subset of bases and sequences by stacking ensemble learning strategies, in: 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2023.
[28] H. Zulfiqar, et al., Deep-STP: a deep learning-based approach to predict snake toxin proteins by using word embeddings, Front. Med. (2024) 10.
[29] H. Li, B. Liu, BioSeq-Diabolo: biological sequence similarity analysis using Diabolo, PLoS Comput. Biol. 19 (6) (2023) e1011214.
[30] B. Liu, X. Gao, H. Zhang, BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches, Nucleic Acids Res. 47 (20) (2019) e127.
[31] J. Jia, et al., 4mCPred-GSIMP: predicting DNA N4-methylcytosine sites in the mouse genome with multi-scale adaptive features extraction and fusion, Math. Biosci. Eng. 21 (1) (2024) 253–271.
[32] M. He, et al., Multi-head attention-based masked sequence model for mapping functional brain networks, Front. Neurosci. 17 (2023) 1183145.
[33] J. Jia, L. Qin, R. Lei, DGA-5mC: a 5-methylcytosine site prediction model based on an improved DenseNet and bidirectional GRU method, Math. Biosci. Eng. 20 (6) (2023) 9759–9780.
[34] F.L. Lai, F. Gao, LSA-ac4C: a hybrid neural network incorporating double-layer LSTM and self-attention mechanism for the prediction of N4-acetylcytidine sites in human mRNA, Int. J. Biol. Macromol. 253 (Pt 3) (2023) 126837.
[35] M. Harun-Or-Roshid, et al., Meta-2OM: a multi-classifier meta-model for the accurate prediction of RNA 2'-O-methylation sites in human RNA, PLoS One 19 (6) (2024) e0305406.
[36] M.M. Hasan, et al., i4mC-ROSE, a bioinformatics tool for the identification of DNA N4-methylcytosine sites in the Rosaceae genome, Int. J. Biol. Macromol. 157 (2020) 752–758.
[37] Y. Zheng, et al., Machine Learning-Aided Scoring of Synthesis Difficulties for Designer Chromosomes, Science China-Life Sciences, 2023.
[38] Z. Yin, et al., SoftVoting6mA: an improved ensemble-based method for predicting DNA N6-methyladenine sites in cross-species genomes, Math. Biosci. Eng. 21 (3) (2024) 3798–3815.
[39] C.N. Aher, Enhancing Heart Disease Detection Using Political Deer Hunting Optimization-Based Deep Q-Network with High Accuracy and Sensitivity, Medinformatics, 2023.
[40] H. Li, Y. Pang, B. Liu, BioSeq-BLM: a platform for analyzing DNA, RNA, and protein sequences based on biological language models, Nucleic Acids Res. 49 (22) (2021) e129.
[41] G.C. Marino, et al., Deep neural networks compression: a comparative survey and choice recommendations, Neurocomputing 520 (2023) 152–170.
[42] R.A.J. Alhatemi, S. Savaş, A weighted ensemble approach with multiple pre-trained deep learning models for classification of stroke, Medinformatics 1 (1) (2023) 10–19.
[43] J.M. Ahn, J. Kim, K. Kim, Ensemble machine learning of gradient boosting (XGBoost, LightGBM, CatBoost) and attention-based CNN-LSTM for harmful algal blooms forecasting, Toxins (Basel) 15 (10) (2023).
[44] A. Ogunleye, Q.G. Wang, XGBoost model for chronic kidney disease diagnosis, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (6) (2020) 2131–2140.
[45] Y. Yang, et al., Multi-layer perceptron classifier with the proposed combined feature vector of 3D CNN features and lung Radiomics features for COPD stage classification, J. Healthc Eng. 2023 (2023) 3715603.
[46] Z. Teng, et al., i6mA-vote: cross-species identification of DNA N6-Methyladenine sites in plant genomes based on ensemble learning with voting, Front. Plant Sci. 13 (2022) 845835.
[47] H. Wu, et al., StackTADB: a stacking-based ensemble learning model for predicting the boundaries of topologically associating domains (TADs) accurately in fruit flies, Brief. Bioinform. 23 (2) (2022).
[48] S. Jiao, et al., Integrated convolution and self-attention for improving peptide toxicity prediction, Bioinformatics 40 (5) (2024).
[49] C. Ai, et al., MTMol-GPT: De novo multi-target molecular generation with transformer-based generative adversarial imitation learning, PLoS Comput. Biol. 20 (6) (2024).
[50] C. Xiao, et al., PEL-PVP: application of plant vacuolar protein discriminator based on PEFT ESM-2 and bilayer LSTM in an unbalanced dataset, Int. J. Biol. Macromol. 277 (2024) 134317.
[51] Z. Luo, et al., DLm6Am: a deep-learning-based tool for identifying N6,2'-O-Dimethyladenosine sites in RNA sequences, Int. J. Mol. Sci. 23 (19) (2022).
[52] X. Cheng, et al., BiLSTM-5mC: a bidirectional long short-term memory-based approach for predicting 5-Methylcytosine sites in genome-wide DNA promoters, Molecules 26 (24) (2021).
[53] O.A. Kittaneh, The variance entropy multi-level thresholding method, Multimed. Tools Appl. 82 (28) (2023) 43075–43087.
[54] Y. Yao, S. Zhang, T. Xue, Integrating LASSO feature selection and soft voting classifier to identify origins of replication sites, Curr. Genomics 23 (2) (2022) 83–93.
[55] Y.H. Wang, et al., Identification of adaptor proteins using the ANOVA feature selection technique, Methods 208 (2022) 42–47.
[56] Y. Liang, et al., Predicting lncRNA-protein interactions through deep learning framework employing multiple features and random forest algorithm, BMC Bioinform. 25 (1) (2024) 108.
[57] K. Bian, et al., RF-PCA: a new solution for rapid identification of breast Cancer categorical data based on attribute selection and feature extraction, Front. Genet. 11 (2020) 566057.
[58] W. Liu, et al., Prediction of early neurologic deterioration in patients with perforating artery territory infarction using machine learning: a retrospective study, Front. Neurol. 15 (2024) 1368902.
[59] C. Ao, Q. Zou, L. Yu, RFhy-m2G: identification of RNA N2-methylguanosine modification sites based on random forest and hybrid features, Methods 203 (2022) 32–39.
[60] J. Cheng, et al., Hyperspectral technique combined with stacking and blending ensemble learning method for detection of cadmium content in oilseed rape leaves, J. Sci. Food Agric. 103 (5) (2023) 2690–2699.
[61] J. Li, et al., SubLocEP: a novel ensemble predictor of subcellular localization of eukaryotic mRNA based on machine learning, Brief. Bioinform. 22 (5) (2021).
[62] Z. Liu, et al., MulStack: an ensemble learning prediction model of multilabel mRNA subcellular localization, Comput. Biol. Med. 175 (2024) 108289.
[63] E. Lin, C.H. Lin, H.Y. Lane, A bagging ensemble machine learning framework to predict overall cognitive function of schizophrenia patients with cognitive domains and tests, Asian J. Psychiatr. 69 (2022) 103008.
[64] J.A. Morgan-Benita, et al., Hard voting ensemble approach for the detection of type 2 diabetes in Mexican population with non-glucose related features, Healthcare (Basel) 10 (8) (2022).
[65] Y. Yuan, et al., BiLSTM- and CNN-based m6A modification prediction model for circRNAs, Molecules 29 (11) (2024).
[66] S.S. Tng, et al., Improved prediction model of protein lysine Crotonylation sites using bidirectional recurrent neural networks, J. Proteome Res. 21 (1) (2022) 265–273.
[67] Q. Lu, et al., KDE bioscience: platform for bioinformatics analysis workflows, J. Biomed. Inform. 39 (4) (2006) 440–450.
[68] T.T. Ogunjobi, et al., Bioinformatics Applications in Chronic Diseases: A Comprehensive Review of Genomic, Transcriptomics, Proteomic, Metabolomics, and Machine Learning Approaches, Medinformatics, 2024.
[69] X. Ren, et al., HydrogelFinder: a foundation model for efficient self-assembling peptide discovery guided by non-Peptidal small molecules, Adv. Sci. (2024) 2400829.