Mul-SNO: A Novel Prediction Tool for S-Nitrosylation Sites Based on Deep Learning Methods

Qian Zhao, Jiaqi Ma, Yu Wang, Fang Xie, Zhibin Lv, Yaoqun Xu, Hua Shi, and Ke Han

***Abstract*—Protein s-nitrosylation (SNO) is one of the most important post-translational modifications and is formed by the covalent modification of nitric oxide and cysteine residues. Extensive studies have shown that SNO plays a pivotal role in the plant immune response and treating various major human diseases. In recent years, SNO sites have become a hot research topic. Traditional biochemical methods for SNO site identification are time-consuming and costly. In this study, we developed an economical and efficient SNO site prediction tool named Mul-SNO. Mul-SNO ensembled current popular and powerful deep learning model bidirectional long short-term memory (BiLSTM) and bidirectional encoder representations from Transformers (BERT). Compared with existing state-of-the-art methods, Mul-SNO obtained better ACC of 0.911 and 0.796 based on 10-fold cross-validation and independent data sets, respectively.**

***Index Terms*—Deep learning, machine learning, post-translational modification (PTM), s-nitrosylation (SNO).**

I. Introduction

PROTEIN post-translational modifications (PTMs) refer to covalent modification and general enzymatic processes that occur outside the polypeptide chain [1]. S-nitrosylation is one of the most important PTMs and involves the oxidative modification process of nitric oxide (NO) and cysteine residues (Cys) [2], as shown in Fig. 1. It has been confirmed that SNO plays a key role in plant immune regulation [3], cell senescence [4], diabetes and other physiological and pathological aspects [5]. The efficient and accurate prediction of SNO sites can greatly promote the research of biological function mechanisms and the development of drugs for related diseases.

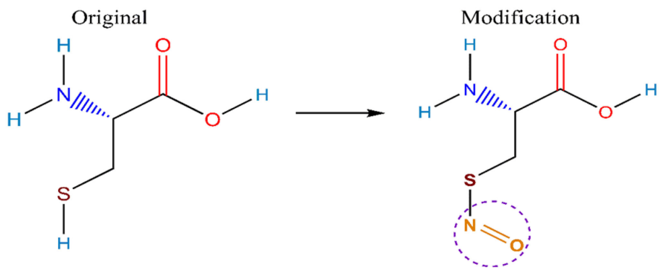

Fig. 1. Schematic diagram of S-nitrosylation site modification.

The study of SNO site prediction has become a hot spot in recent years. As Zhao *et al.* [6] recently introduced in a review on SNO site prediction, the research on SNO site prediction can be roughly divided into two categories: the first category uses traditional biochemical methods, but these methods have the problem of being time-consuming and expensive; the second category is the use of computational biology methods, such as machine learning or deep learning, which is also the topic of the current study. Over the years, many researchers have performed in-depth research and proposed many excellent SNO site predictors. Unfortunately, they still have some problems. For example, Hao *et al.* [7] pioneered computational methods to predict SNO sites for the first time in 2006, but the experimental results were not satisfactory. In the following years, many researchers followed up with related works. From 2010 to 2014, many predictors were successively proposed based on computing methods to predict SNO sites, such as GPS-SNO [8], CPR-SNO [9], SNOSite [10], iSNO-PseAAC [11], iSNO-AAPair [12], iSNO-ANBPB [13] and PSNO [14]. The relevant research progress has been introduced in detail in our previous work, so there is no need to repeat it here [6]. However, although the number of data sets used by these predictors is increasing based on previous studies, the growth rate is still too slow, and the total amount is insufficient. Therefore, these predictors cannot fully express sequence characteristics on new large-scale data sets, and it is currently difficult to achieve excellent performance. In 2018, Xie *et al.* [15] used deep learning to predict SNO sites for the first time. In this work, they constructed a new type of large data set and designed an eight-layer neural network to predict SNO sites. In 2019, Li [16] and Hasan [17] *et al.* proposed a multi-feature hybrid ensemble algorithm. Li used 9 feature extraction methods to improve the prediction performance and max-relevance-max-distance (MRMD) for final feature selection. Hasan used 4 different coding schemes for the sequence and ensemble support vector machine (SVM) and

Manuscript received July 2, 2021; revised October 5, 2021; accepted October 23, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62001090 and in part by the Natural Science Foundation of Heilongjiang Province under Grant LH2020F009. *(Corresponding author: Ke Han.)*

Qian Zhao, Jiaqi Ma, Yu Wang, Fang Xie, Yaoqun Xu, and Ke Han are with the School of Computer and Information Engineering, Harbin University of Commerce, Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin 150028, China (e-mail: zhaoqian@hrbeu.edu.cn; machiachi@163.com; wangyu.cn@hotmail.com; xiefang.cn@outlook.com; xuyq@hrbcu.edu.cn; thruster@163.com).

Zhibin Lv is with the Department of medical instruments and information, College of Biomedical Engineering, Sichuan University, Chengdu 610065, China (e-mail: lvzhibin@pku.edu.cn).

Hua Shi is with the School of Opto-electronic and Communication Engineering, Xiamen University of Technology, Xiamen 361024, China (e-mail: shihua@xmut.edu.cn).

Digital Object Identifier 10.1109/JBHI.2021.3123503

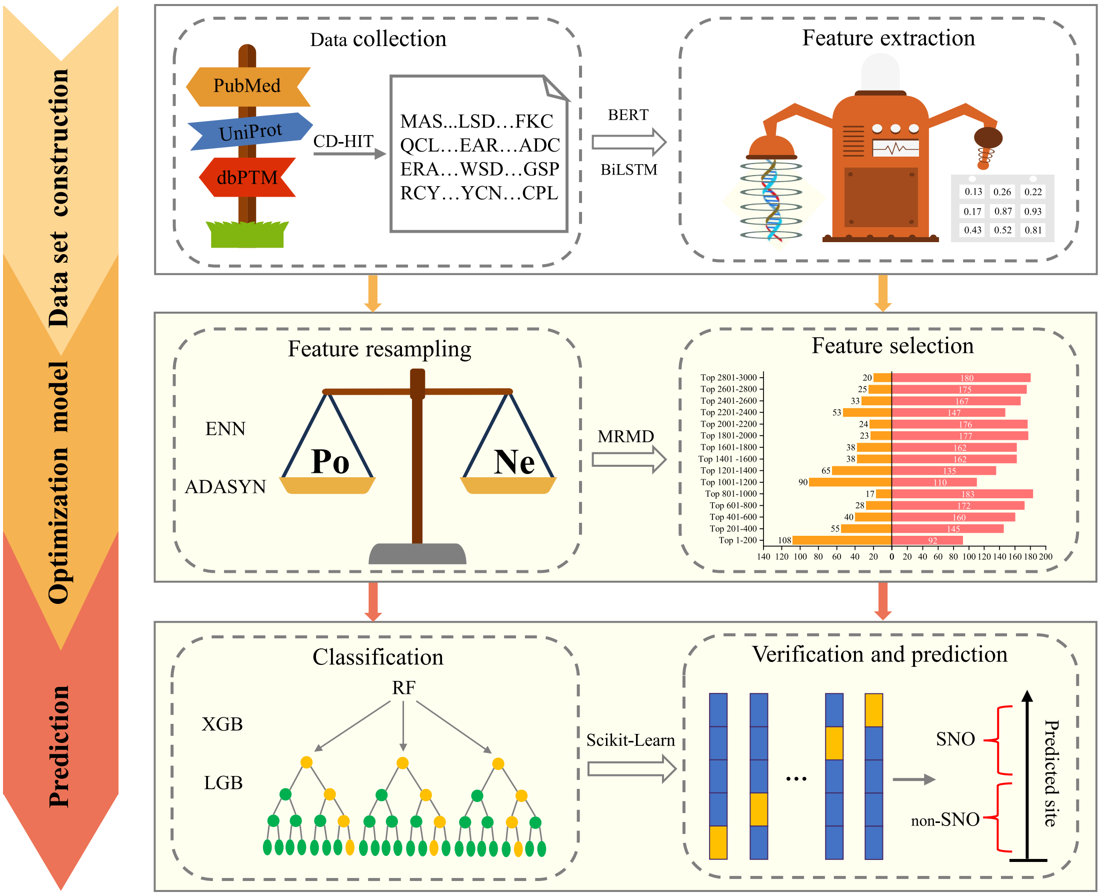

Fig. 2. Overall framework of Mul-SNO. In the data preprocessing stage, experimental data is obtained by de-redundancy (CD-HIT) and feature extraction; in the model optimization stage, ENN and ADASYN algorithms are used for feature resampling, and then MRMD is used for feature selection. Finally, random forest (RF) was used to cross-validate and predict SNO sites.

random forest (RF) to improve the prediction performance. In 2021, Siraj *et al.* [18] used an embedded layer and BiLSTM recurrent neural network to predict SNO sites, and the accuracy was improved by 3% compared with existing methods. Almost at the same time, Li *et al.* [19] constructed a new large-scale data set including s-nitrosylation, s-palmitoylation, s-sulfenylation, s-sulfhydration and s-sulfinylation through related databases and the literature. In this work, the neural network they designed based on these data sets achieved the best results thus far. The average AUCs of s-nitrosylation, s-palmitoylation, s-sulfenylation, s-sulfhydration, and s-sulfinylation were 0.793, 0.807, 0.796 and 0.876, respectively.

However, the current performance of various predictors is still insufficient and cannot fully meet the demand. In this work, we developed Mul-SNO ensembled sequence embedding features by BiLSTM and BERT for cysteine sequence prediction [20], [21], [22]. As shown in the flowchart of Fig. 2, Mul-SNO used ENN, ADASYN and MRMD to determine the best feature vector space and to solve the data imbalance for model training [23], [24]. Three classifiers, xgboost [25]–[28], lightgbm [29] and random forest [30], [31] were used for model development, and the best model was selected based on the evaluation score. The 10-fold cross-validation score of Mul-SNO was 0.914, 0.829 and 0.914 for ACC, MCC and ROC, respectively. We believe that Mul-SNO will greatly promote the development of related drugs and facilitate further research on S-nitrosylation. The prediction server can be obtained for free at http://lab.malab.cn/~mjq/Mul-SNO/.

II. Materials and Methods

A. Data Sets

High-quality data sets are an important cornerstone to build an effective model. Many data in previous data sets have been experimentally confirmed to be incorrectly annotated, so it is an indispensable choice to use the latest data set. Recently, Li etal. [19] constructed a high-quality dataset based on the extensively published literature and several authoritative databases, where the modified sequence is considered positive data; otherwise, it is negative.

In this study, our training set is from Li *et al.* [19] and the independent test set is from DeepNitro. Finally, we obtained 23041 s-nitrosylation sequences from 10671 proteins. Cysteine residue C is in the center of the peptide fragment sequence and can be formulated as:

$$P \;= R_{-\xi}\; R_{-(\xi-1)} \ldots R_{-2}R_{-1}CR_{+1}R_{+2} \ldots R_{+(\xi-1)}R_{+\xi} \tag{1}$$

where the subscript ξ is an integer, and $R_{-\xi}$ and $R_{+\xi}$ represent the ξ-th upstream and downstream of cysteine C, respectively. P denotes a peptide fragment of the entire representative category. If the SNO site was modified, it was defined as SNO; otherwise, it was defined as non-SNO. It can be formulated as ($\cup$ means union):

$$P = SNO \cup non{-}SNO \tag{2}$$

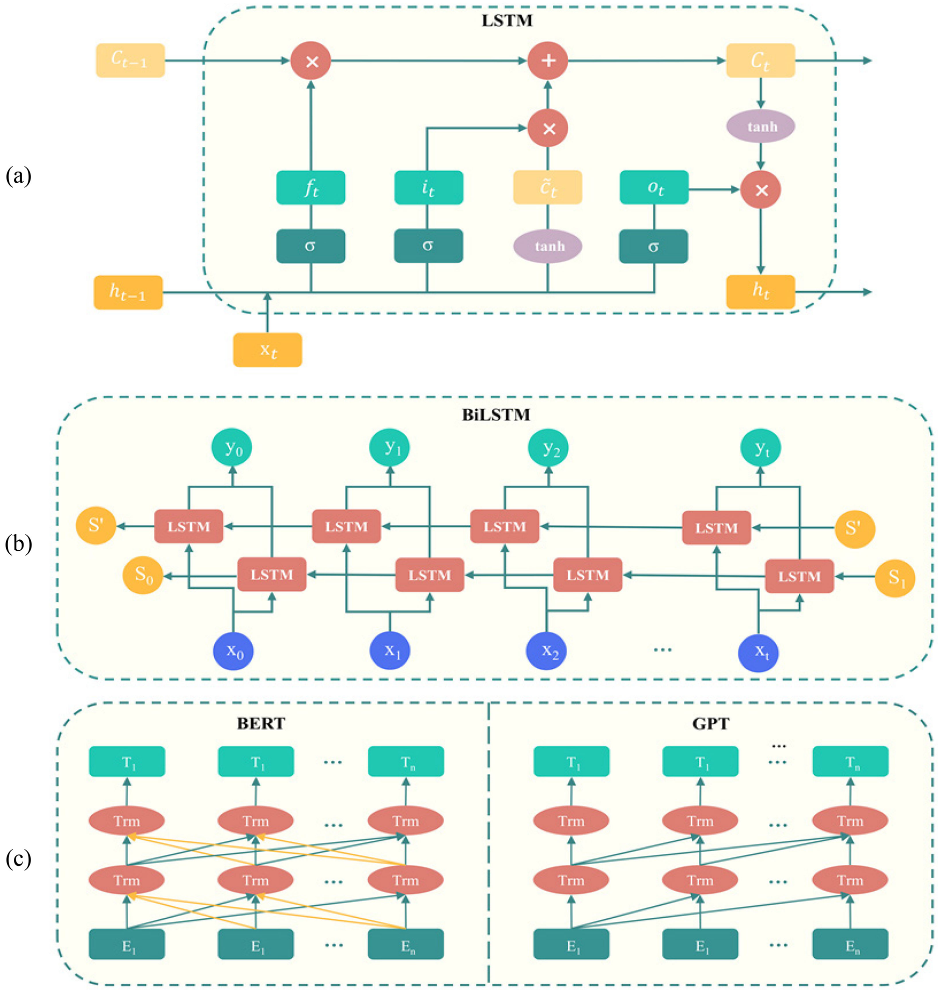

Fig. 3. (a) Loop unit structure of LSTM network. (b) Bi-directional Long Short-Term Memory structure diagram. (c) Schematic diagram of the training model of BERT and GPT.

For a fair comparison with pCysMod, we also set ξ to 15 when constructing the data set. If the peptide sequence was less than 15 in length, "$*$" was used to fill in so that each peptide had the same length.

In machine learning problems, the amount, complexity, and imbalance ratio of data are the main factors that affect the performance of the classifier [32], [33]. To solve this problem, we randomly select SNO and non-SNO sequences to make them 1:1. In addition, the redundancy of the sequence plays a vital role in the noise interference of data, storage space and calculation time. To effectively solve this problem, CD-HIT has been widely used in sequence analysis research [34]–[37]. In this study, we set the threshold of CD-HIT to 30%, which can more convincingly prove the reliability of Mul-SNO performance. Finally, we obtained the training set of 13894 SNO sites and 11374 non-SNO sites. Using the same strategy as the training set, we obtained an independent test set containing 332 SNO sites and 1632 non-SNO sites.

B. *Feature Extraction*

Traditional biological sequences (DNA or protein, etc.) cannot be directly recognized by the computer. If these sequences are correctly predicted and analyzed by the classifier, they need to be converted into equal-length for input through feature extraction [38]. In this section, we will introduce the feature extraction algorithms used by Mul-SNO: BiLSTM and BERT, which are widely used in sequence coding research and have achieved excellent results [39], [40].

1) Features Embedded With the Pertained BiLSTM Model: BiLSTM is a modified form of LSTM based on the RNN architecture [41], which consists of two layers of recurrent neural networks, forward LSTM and backward LSTM. LSTM (Fig. 3(a)) has a unique gate architecture, namely, forget gate f_t, input gate i_t and output gate o_t. The functions of these three gates are as follows: (1) Forget gate f_t controls how much information needs to be forgotten in the internal state c_{t-1} at the previous moment; (2) input gate i_t controls how much information needs to be saved in the candidate state $\tilde{c}_t$ at the current moment; and (3) the output gate o_t controls how much information of the internal state c_t at the current moment needs to be output to the external state h_t. Their calculation formulas are as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f \quad (3)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o \quad (5)$$

where σ represents the logistic function and its output interval is (01), x_t is the input at the current moment, and h_{t-1} is the external state at the previous moment. These gates only convey useful information for subsequent calculations by means of information forgetting and memorizing new information, thus effectively solving the problem of gradient explosion or disappearance.

LSTM can establish a long-distance timing dependency, but there is a problem when processing traditional natural language processing (NLP) tasks, which cannot be encoded from back to front. BiLSTM (Fig. 3(b)) solves this problem, and this feature can be effectively combined with global structural similarity between proteins and pairwise residue contact maps for individual proteins so that the vector matrix mapped from the amino acid sequence can be fully characterized [39].

2) Feature Embedded With the Pertained BERT Model: BERT is a framework for processing NPL tasks launched by the Google team in 2018 [22]. Its wide applicability and excellent performance have been praised by academia and industry since it appeared, and it has also been widely used in computational methods to explore protein sequences [42], [43], [44]. BERT uses the same structure as generative pretraining (GPT) [45], that is, the language model is first used for pretraining, and then the fine-tuning mode is used to solve downstream tasks. The difference is that in the explosion model stage, based on unidirectional prediction [46], BERT proposes the use of a context-based omnidirectional prediction method to predict [MASK], that is, deep bidirectional prediction (Fig. 3(c)).

The traditional encoder-decoder framework cannot fully understand the meaning of the entire sentence, so many details will be lost when processing long text. In BERT, the author uses the Transformer attention mechanism[47], which can enable multiple focus points for the same sentence simultaneously, instead of being limited to front-to-back or back-to-front sequential processing. In this step, BERT first randomly occludes 15% of the vocabulary of the predicted text and allows the model to make predictions based on the context to initially obtain the parameters for training the Transformer model. Then, these parameters are used to continue training the model by identifying the continuity of the selected sentence. These two steps are called pretraining.

3) Availability and Implementation: Feature extraction was performed by a developing toolkit named eFeature, and the source code and tutorial can be found at http://lab.malab.cn/soft/eFeature.

C. Resampling Strategies

In traditional classification tasks, data imbalance is a widespread and challenging problem that can easily cause the classifier to make incorrect judgments. With the advancement of related research, researchers have proposed three resampling strategies based on the characteristics of data distribution: oversampling, under-sampling and hybrid methods (over- and under-sampling). These methods have also been widely recognized and used in the field of bioinformatics [48]–[52].

ENN (edited data set using nearest neighbors) is a classic under-sampling algorithm that was proposed by Wilson *et al.* in 1972[23]. To optimize the decision boundary, ENN will find k nearest neighbors (usually k = 3) for each data to undergo under-sampling and delete noise neighbors with different categories. ADASYN (adaptive synthetic sampling) is an oversampling algorithm improved from SMOTE [24], [53]. The difference from SMOTE is that ADASYN does not simply generate the same number of minority samples but automatically assigns a weight to each minority sample. More emphasis is placed on creating more samples around low-density sample points (such as boundary points). However, if there are outliers in the low-density area of the data distribution, the direct use of ADASYN will greatly increase the outliers, which is not good for the fitting of the model. Thus, we use ENN to eliminate these outliers and then combine them with ADASYN to strengthen key data points, thereby reducing learning bias and making the decision boundary smoother.

D. Feature Selection

With the development of feature engineering, the disaster of dimensionality has become a problem that cannot be ignored [54]. To reduce noise interference, save computing resources, and ultimately improve prediction accuracy, many corresponding dimensionality reduction algorithms have been proposed, for example, minimum redundancy maximum relevance (mRMR) [55], analysis of variance (ANOVA) [56], principal component analysis (PCA) and max-relevance-max-distance (MRMD) [57], [58]. After testing, we choose MRMD, which is more friendly to experimental data, for feature selection.

Unlike traditional feature selection algorithms, MRMD not only focuses on features that are highly correlated with the target but also proposes a comprehensive measure to measure the independence of each feature based on a distance function and to measure the correlation of sub-features through the Pearson correlation coefficient. MRMD can be defined as:

$$max\ MR_i = \left| PCC\left(\overrightarrow{F_i}, \overrightarrow{C_i}\right)\right|\ (1 \leq i \leq M) \tag{6}$$

$$mean\left(maxMD_i\right)\ = \frac{1}{3}\ \left(ED_i + COS_i + TC_i\right)\left(1 \leq i \leq M\right) \tag{7}$$

$$MRMD_{score} =\ \max\left(w_r * MR_i + w_d * MD_i\right) \tag{8}$$

The formula mainly includes three parts: (1) $maxMR_i$ represents the maximum MR (max-relevance) value among i features. Furthermore, F is the M-D vector composed of the i-th feature of each instance, and *C* also represents the MD vector of each element derived from target class c. (2) MD_i represents the maximum distance of i features. Moreover, *ED*, *COS,* and *TC* are the Euclidean distance, cosine similarity and Tanimoto coefficient, respectively. (3) w_r $(0 < w_r < 1)$ and w_d $(0 < w_d < 1)$ represent the weight values of MR and MD, respectively. See the detailed derivation in Zou et al [58].

E. Classifier

In this study, we use RF (random forest) as a classifier. RF is one of the most classic ML algorithms in bioinformatics research and is used to construct predictors [59]–[61]. RF is an ensemble learning algorithm belonging to the bagging (bootstrap aggregating) type. It can train the classifier by reselecting k new data sets through sampling with replacement of the original data set. These classifiers classify new samples and then use a majority vote or average the output to count the classification results of all classifiers. The category with the highest result is the final label. In addition, to prove the robustness of our model, LGB (lightgbm) and XGB (xgboost) are also used as reference classifiers. During the experiment, these classifiers all use default parameters. RF can be implemented using the friendly machine learning library sklearn (version 0.24), and the LGB and XGB codes can be found on GitHub (https://github.com/Microsoft/LightGBM) and the website https://xgboost.readthedocs.io/, respectively.

F. Performance Evaluation

How to objectively and correctly evaluate predictor performance is an essential part. Common evaluation methods include subsampling tests, independent data set tests, k-fold cross-validation, and jackknife tests. To make a fair and reasonable evaluation with previous experiments, we also use jackknife as the evaluation standard because it is considered the most

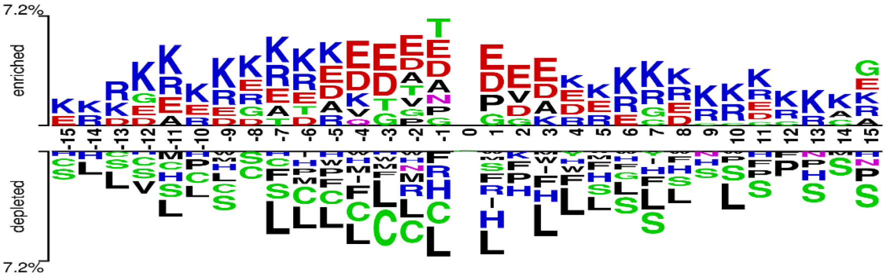

Fig. 4. The position-specific preference of SNO and non-SNO.

objective cross-validation method [62], [63], [63], [64], [65],. In this study, we use sensitivity (Sn), specificity (Sp), accuracy (Acc) and Matthews' correlation coefficient (MCC) to quantify the performance of our model [66], [67], [68], which can be formulated as:

$$S_n = \frac{TP}{TP + FN} \tag{9}$$

$$S_p = \frac{TN}{TN + FP} \tag{10}$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{12}$$

FP, TN, and FN represent true positives (i.e., accurately predicted as SNO sites), false positives (i.e., incorrectly predicted as SNO sites), true negatives (i.e., accurately predicted as non-SNO sites), and false negatives (incorrectly predicted as non-SNO sites), respectively. MCC is a correlation coefficient describing the actual classification and the predicted classification, and its value range is $[-1, 1]$. 1 indicates perfect prediction, 0 indicates that the predicted result is not as good as random prediction, and -1 means that the predicted classification is completely inconsistent with the actual classification.

III. Results and Discussion

A. Position-Specific Differences Analysis

According to the SNO site modification process, we hypothesize that SNO sites and non-SNO sites have different position-specific differences. In this study, we used the graphical network tool two sample logo to analyze statistically based distinct patterns or conserved sequence motifs between the SNO site and non-SNO site [69]. Fig. 4 shows the distribution of different residues in the cysteine sequence (t-test, P <0.05), where the SNO site is located above the x-axis and vice versa for the non-SNO site.

We observed the following points: (1) Overall, non-SNO sites have more leucine (L) and serine (S) than SNO sites. In addition, unmodified cysteine (C) was significantly enriched in the negative data but was basically not reflected in the positive data. We assume that the process of SNO site modification will lead to changes in the surrounding residue fragments. Therefore, predicting the SNO site may not limit the modification site itself, and the distribution of leucine (L) and serine (S) can also be used as an important basis for exclusion. (2) Glutamic acid (E) and lysine (K) were significantly enriched upstream and downstream of the SNO site, and they were approximately symmetrically distributed. In addition, compared with the upstream concentration of glutamic acid (E), the downstream quantity was significantly reduced. This indicates that the physicochemical properties of specific amino acids near the SNO site can be used as an effective information feature for predicting the SNO site. (3) Sugar-generating amino acids such as glutamic acid (E), aspartic acid (D) and glycine (G) have a higher frequency near the SNO site. This indicates that sugar-generating amino acids show a better preference for modified cysteine. The above information is based on our statistics and observations, and further research is needed to prove this hypothesis. The above information shows that the sequence coding scheme based on position and frequency is reliable and effective in predicting SNO sites, which is fully reflected in our algorithm coding scheme.

TABLE I
Statistical Significance of the Difference in Predictive Performance (ACC) of the Three Classifiers

Methods	Classifier	P value	Accepted (h_1/h_2)
BERT	RF with XGB	0.00037	h_2
	RF with LGB	0.00017	h_2
	XGB with LGB	0.00018	h_2
BiLSTM	RF with XGB	0.00017	h_2
	RF with LGB	0.00017	h_2
	XGB with LGB	0.68	$\boldsymbol{h_1}$

B. Performance of Single Features

We used the two feature extraction methods BiLSTM and BERT, to map the protein sequence into a vector matrix. In this section, each of them is individually tested for classification, and the results are shown in Fig. 5(a) and Fig. 5(b).

According to Fig. 5(a) and Fig. 5(b), for both BERT and BiLSTM feature encoding methods, compared with XGB and LGB, RF achieved the highest accuracy rates of 0.899 and 0.882, respectively. Besides, we also use two-sample t-test to evaluate the significant difference in ACC performance of different classifiers.

The null hypothesis (h_1) is that the mean of the two samples (the ACC of the two classifiers) is the same. If h_1 is true, it means that the performance between the two classifiers is not significantly improved. Another hypothesis (h_2) is that the mean of the two samples is different. If h_2 is true, it means that the performance between the two classifiers has been significantly improved. Besides, the significance level (alpha) threshold is 0.05. The results of the t-test are shown in Table I.

When using BERT as the feature extraction method, the P values of RF compared with XGB and LGB are 0.00037

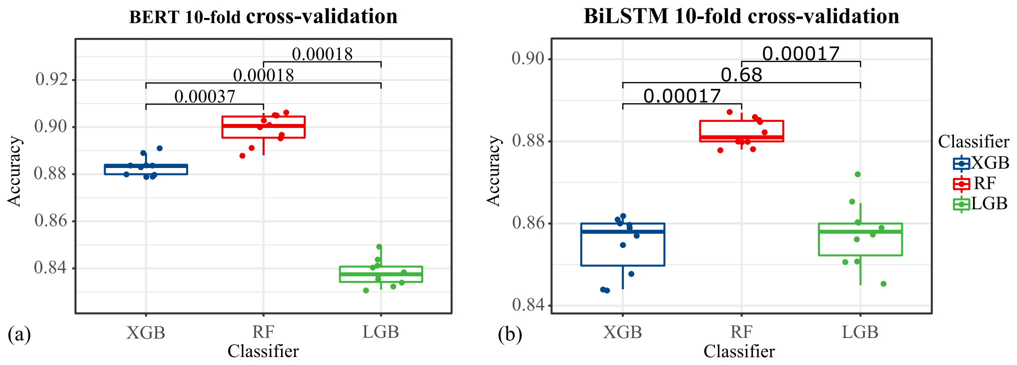

Fig. 5 (a) BERT 10-fold cross-validation accuracy and P value between different classifiers. (b) BiLSTM 10-fold cross-validation accuracy and P value between different classifiers.

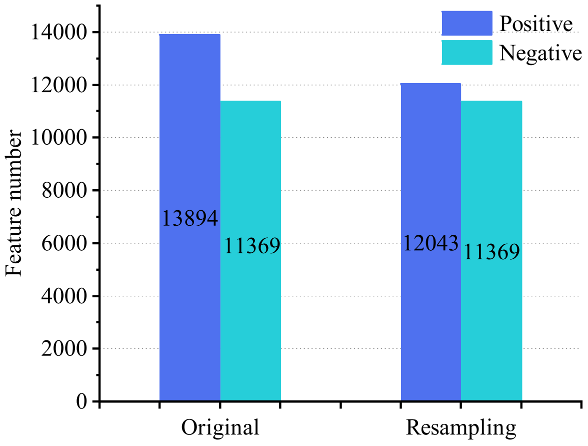

Fig. 6. Distribution of the number of positive and negative samples.

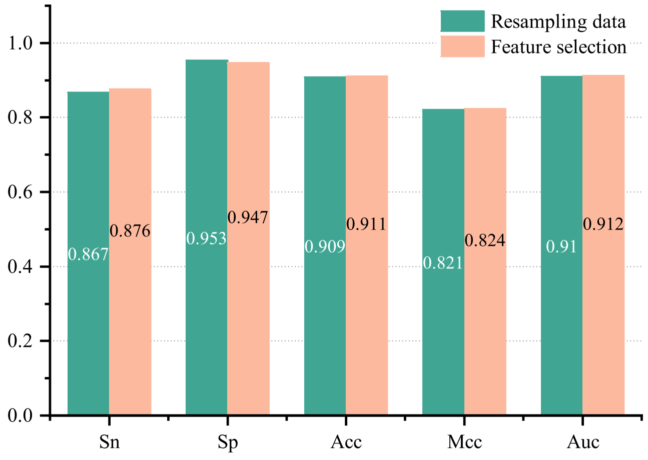

Fig. 7. Feature selection performance comparison chart.

and 0.00018, respectively. The P value of XGB and LGB is also 0.00018. This means that h_2 is established, that is, the significant difference between the three classifiers is not significantly improved. When using BiLSTM as the feature extraction method, the P values of RF compared with XGB and LGB are both 0.00017. This means that h_2 is established, and the performance of RF is significantly improved compared to XGB and LGB. In addition, we also noticed that the P value of XGB and LGB is 0.68. This shows that h_1 is established, and the performance between the two is not significantly improved.

In general, this experiment also confirms our conjecture: 1) When features (such as BERT) have a better performance on data representation, the performance of different classifiers has little difference; 2) Compared with XGB and LGB, RF has a stronger ability to distinguish S-nitrosylation data. Therefore, in the following experiments, we will use RF as the main classifier and XGB and LGB as control experiments.

C. Performance of Mixed Features

In this section, two feature extraction methods, BERT and BiLSTM, are used to obtain new 4374D feature data through feature mixing. Then, we set up a set of comparative experiments using mixed features. First, we used three classifiers to test the 4374D (dimension) feature data to judge the effect. Second, we used MRMD to calculate feature importance, eliminate feature redundancy and noise, and remove 1374D features with lower scores. In this way, the optimized feature data of 3000D are finally obtained. In order to prove the effectiveness of our model, we performed 10-fold cross-validation on the original mixed data, the resample data, and the final feature selection data. For a fair comparison, the latter's data processing methods all include the former. In addition, the data distribution before and after resampling is shown in Fig. 6.

In various performance evaluations of machine learning, the accuracy rate does not accurately measure the effectiveness of the imbalance data model, so we choose the F1 value (F1-score) as a reference value. Here, we compare the original mixed data and resample data through 10-fold cross-validation, and the F1 value has also changed from 0.644 to 0.894. This can prove that our resampling strategy is effective. In addition, we also compared the resampled data with the feature selected data by 10-fold cross-validation (Fig. 7). After grid search, we have set the n_estimators of the random forest classifier to 300 in the experiment.

We can see that after feature selection, the prediction accuracy has increased from 0.909 to 0.911. In addition to Sp (specificity), all other indicators have also been slightly improved. Although the performance improvement is limited, MRMD is still very helpful in eliminating noise features, reducing calculation pressure, and saving calculation time. This can save almost a third of computing resources.

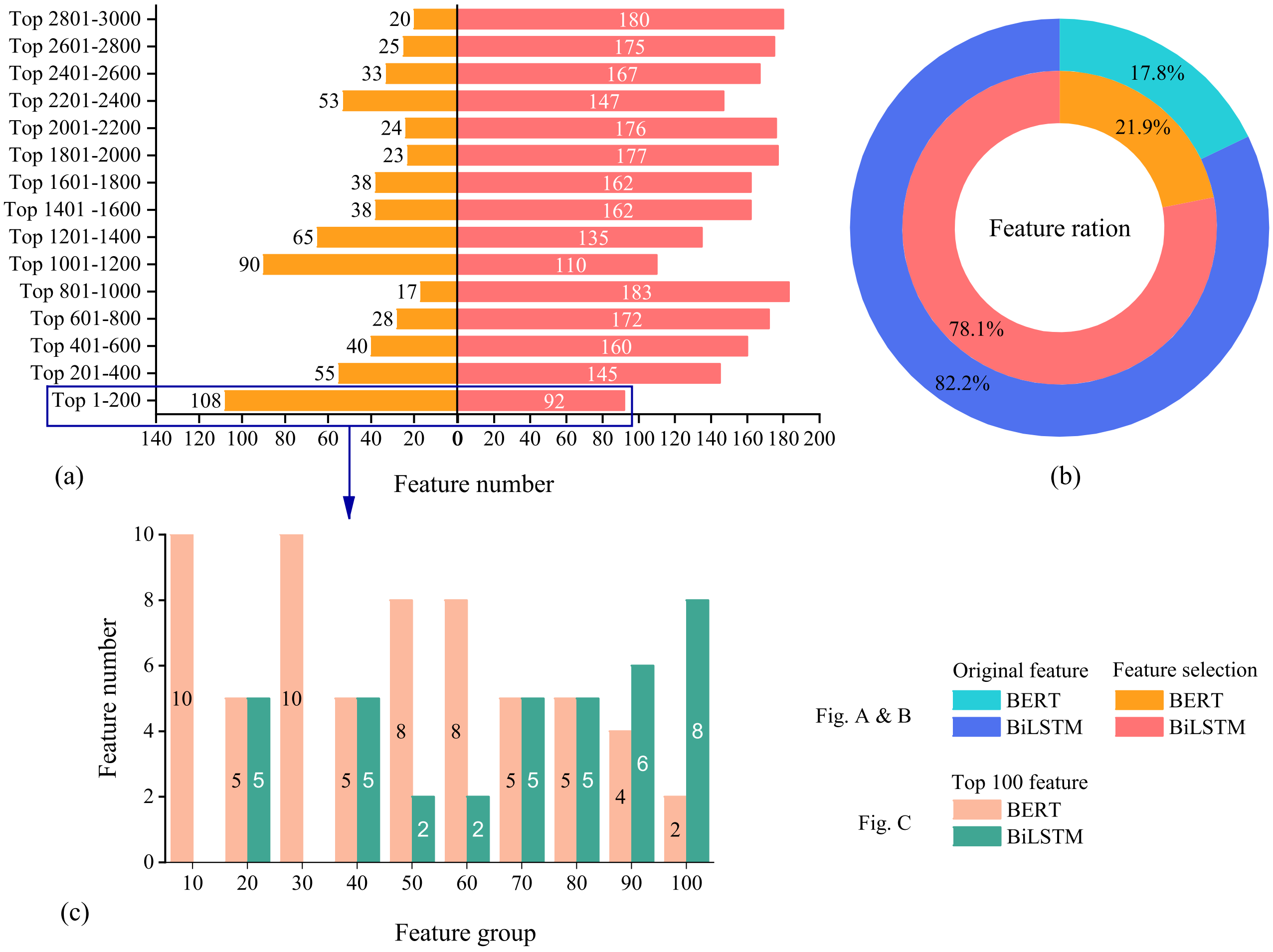

Fig. 8. (a) Comparison of the number of features between BiLSTM and BERT with different importance ranking groups. (b) BiLSTM and BERT feature number ratio before and after feature selection. (c) The ranking of the top 100 feature importance.

TABLE II
COMPARISON OF MUL-SNO WITH EXISTING METHODS

Data	Method	Sn	Sp	Acc	Mcc	Auc	References
Training set	PcysMod	0.400	0.850	0.777	0.236	0.743	[19]
	Mul-SNO	**0.876**	**0.947**	**0.911**	**0.824**	**0.912**	-
Independent test set	DeepNitro	0.58	0.76	0.73	0.22	0.73	[15]
	PreSNO	0.60	0.71	0.711	0.30	**0.80**	[17]
	RecSNO	**0.77**	0.71	0.71	0.30	**0.80**	[18]
	Mul-SNO	0.76	**0.83**	**0.80**	**0.59**	0.80	-

PcysMod: https://www.frontiersin.org/articles/10.3389/fcell.2021.617366/full
Mul-SNO: http://lab.malab.cn/~mjq/Mul-SNO/
DeepNitro: https://www.sciencedirect.com/science/article/pii/S1672022918303474
PreSNO: https://pubs.rsc.org/en/content/articlelanding/2019/mo/c9mo00098d/unauth
RecSNO: https://ieeexplore.ieee.org/abstract/document/9313999

D. Comparison With State-of-the-Art Methods

To make a fair and rigorous comparison, we used the pcysMod dataset as the training set and the DeepNitro dataset as the independent test set. We compared them with existing models, such as DeepNitro [15], PreSNO [17], pCysMod [19], RecSNO [18], and the predictor of Li *et al.* [16]. The comparison between Mul-SNO and existing methods are shown in Table II.

It can be seen from the comparison that Mul-SNO far exceeds the existing predictors in all comparisons. In addition, in the classification task, MCC considers true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). It is generally considered that this indicator is a relatively balanced indicator. Especially in the case of imbalance, MCC can comprehensively measure the quality of a classifier. In this study, the MCC of Mul-SNO reached a high value of 0.824 and 0.59 on the training set and independent test set, respectively. This fully shows that our model far exceeds the existing predictors and has an excellent performance in predicting SNO sites.

E. Feature Importance Analysis

In this section, to enhance the feature expression ability, we continue to explore two questions: (1) Which feature extraction method is more important for BiLSTM and BERT? (2) Are their respective feature focuses different? Thus, we first studied the comparison of the number of features of BiLSTM and BERT before and after feature selection. Among the original features (Fig. 8(b)), BiLSTM has 3604D features, BERT has 768D features, and their ratio is 4.7:1. In addition, we observe that after MRMD feature selection and removing noise features with lower scores, their proportions do not change much. In this case, the feature dimension of BiLSTM is 2355D, that of BERT is 645D, and their ratio is 3.6:1. Therefore, we believe that both BiLSTM and BERT have excellent feature expression capabilities, which is also consistent with the previous single feature classification test results.

Then, we sort the features of BiLSTM and BERT according to the feature importance score from high to bottom. As shown in Fig. 8(c) each 200D feature is a group. On the whole, the ratio of different groups of BiLSTM and BERT is stable. However, we also noticed that among all the TOP 100 features, BERT occupies 15 places, and the feature score is high. In addition, in the Top 50 features, BERT occupies 38 places. This shows that in some key features, BERT can learn and understand the properties of cysteine peptides more effectively. Thus, if more key features of BERT can be further mined and some inefficient features of BiLSTM can be eliminated, the recognition rate of SNO sites can be further improved.

IV. Conclusion

S-nitrosylation is an important post-translational modification (PTM). The correct and efficient identification of SNO sites can greatly promote the further development of related research. In this study, we proposed a novel and robust SNO site predictor, i.e., Mul-SNO. Mul-SNO ensembles two deep feature representation algorithms, BiLSTM and BERT, and after effective feature selection, it can fully represent the feature information of cysteine peptides. Rigorous and comprehensive verification experiments also meticulously proved that our classifier is more robust and accurate than existing methods. We believe that Mul-SNO will be of great assistance to the research of related physiological mechanisms and the development of drugs.

References

[1] D. Knorre, N. Kudryashova, and T. Godovikova, “Chemical and functional aspects of posttranslational modification of proteins,” *Acta Naturae*, vol. 1, no. 3, pp. 29–51, Dec. 2009.

[2] C. T. Stomberski, D. T. Hess, and J. S. Stamler, “Protein S-nitrosylation: Determinants of specificity and enzymatic regulation of S-nitrosothiol-based signaling,” *Antioxid Redox Signal*, vol. 30, no. 10, pp. 1331–1351, Apr. 2019.

[3] R. Di Blasi *et al.*, “Non-histone protein methylation: Biological significance and bioengineering potential,” *ACS Chem. Biol.*, vol. 16, no. 2, pp. 238–250, Feb. 19, 2021.

[4] S. Rizza *et al.*, “S-nitrosylation drives cell senescence and aging in mammals by controlling mitochondrial dynamics and mitophagy,” *Proc. Nat. Acad. Sci.*, vol. 115, no. 15, pp. E3388–E3397, 2018.

[5] G. Chen *et al.*, “bFGF alleviates diabetes-associated endothelial impairment by downregulating inflammation via S-nitrosylation pathway,” *Redox Biol.*, vol. 41, 2021, Art. no. 101904.

[6] Q. Zhao *et al.*, “Recent advances in predicting protein S-nitrosylation sites,” *BioMed Res. Int.*, vol. 2021, pp. 2314–6133, 2021.

[7] G. Hao *et al.*, “SNOSID, a proteomic method for identification of cysteine S-nitrosylation sites in complex protein mixtures,” *Proc. Nat. Acad. Sci.*, vol. 103, no. 4, pp. 1012–1017, Jan. 2006.

[8] Y. Xue *et al.*, “GPS-SNO: Computational prediction of protein S-nitrosylation sites with a modified GPS algorithm,” *PloS One*, vol. 5, no. 6, Jun. 2010, Art. no. e11290.

[9] Y.-X. Li *et al.*, “An efficient support vector machine approach for identifying protein S-nitrosylation sites,” *Protein Peptide Letters*, vol. 18, no. 6, pp. 573–587, Jun. 2011.

[10] T.-Y. Lee *et al.*, “SNOSite: Exploiting maximal dependence decomposition to identify cysteine S-nitrosylation with substrate site specificity,” *PloS One*, vol. 6, no. 7, Jul. 2011, Art. no. e21849.

[11] Y. Xu *et al.*, “iSNO-PseAAC: Predict cysteine S-nitrosylation sites in proteins by incorporating position specific amino acid propensity into pseudo amino acid composition,” *PLoS One*, vol. 8, no. 2, 2013, Art. no. e55844.

[12] Y. Xu, “iSNO-AAPair: Incorporating amino acid pairwise coupling into PseAAC for predicting cysteine S-nitrosylation sites in proteins,” *PeerJ*, vol. 1, 2013, Art. no. e171.

[13] C. Jia, X. Lin, and Z. Wang, “Prediction of protein S-nitrosylation sites based on adapted normal distribution bi-profile Bayes and Chou’s pseudo amino acid composition,” *Int. J. Mol. Sci.*, vol. 15, no. 6, pp. 10410–10423, 2014.

[14] J. Zhang *et al.*, “PSNO: Predicting cysteine S-nitrosylation sites by incorporating various sequence-derived features into the general form of Chou’s PseAAC,” *Int. J. Mol. Sci.*, vol. 15, no. 7, pp. 11204–11219, Jun. 2014.

[15] Y. Xie *et al.*, “DeepNitro: Prediction of protein nitration and nitrosylation sites by deep learning,” *Genomic. Proteomic. Bioinf.*, vol. 16, no. 4, pp. 294–306, Aug. 2018.

[16] T. Li *et al.*, “Identification of S-nitrosylation sites based on multiple features combination,” *Sci. Rep.*, vol. 9, no. 1, Feb. 2019, Art. no. 3098.

[17] M. M. Hasan *et al.*, “Prediction of S-nitrosylation sites by integrating support vector machines and random forest,” *Mol. Omics*, vol. 15, no. 6, pp. 451–458, Dec. 2019.

[18] A. Siraj, T. Chantsalnyam, H. Tayara, and K. T. Chong, “RecSNO: Prediction of protein S-nitrosylation sites using a recurrent neural network,” *IEEE Access*, vol. 9, pp. 6674–6682, 2021, doi: 10.1109/ACCESS.2021.3049142.

[19] S. Li *et al.*, “pCysMod: Prediction of multiple cysteine modifications based on deep learning framework,” *Front. Cell Devlop. Biol.*, vol. 9, no. 117, Feb. 2021, Art. no. 617366.

[20] T. Chen *et al.*, “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN,” *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.

[21] J. Chen, Q. Zou, and J. Li, “DeepM6ASeq-EL: Prediction of human N6-methyladenosine (m6A) sites with LSTM and ensemble learning,” *Front. Comput. Sci.*, vol. 16, no. 2, pp. 162302, Sep. 2021.

[22] J. Devlin *et al.*, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:04805*.

[23] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *IEEE Trans. Syst., Man Cybern.*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.

[24] H. He *et al.*, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[25] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, California, USA, 2016, pp. 785–794.

[26] X. Yu *et al.*, “Exploiting XG boost for predicting enhancer-promoter interactions,” *Curr. Bioinf.*, vol. 15, no. 9, pp. 1036–1045, Feb. 2020.

[27] Z. Lv, D. Wang, H. Ding, B. Zhong, and L. Xu, "Escherichia coli DNA N-4-methycytosine site prediction accuracy improved by light gradient boosting machine feature selection technology," *IEEE Access*, vol. 8, pp. 14851–14859, Aug. 2020, doi: 10.1109/ACCESS.2020.2966576.

[28] D. Mrozek, P. Daniłowicz, and B. J. I. S. Małysiak-Mrozek, "HDInsight4PSi: Boosting performance of 3D protein structure similarity searching with HDInsight clusters in Microsoft Azure cloud," *Inf. Sci.*, vol. 349, pp. 77–101, Jul. 2016.

[29] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 3146–3154, 2017.

[30] M. Pal, "Random forest classifier for Remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, 2005.

[31] X. Ru, L. Li, and Q. Zou, "Incorporating distance-based top-n-gram and random forest to identify electron transport proteins," *J. Proteome Res.*, vol. 18, no. 7, pp. 2931–2939, Jul. 2019.

[32] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Nov. 2002.

[33] Q. Zou *et al.*, "Sequence clustering in bioinformatics: An empirical study," *Brief. Bioinf.*, vol. 21, no. 1, pp. 1–10, 2018.

[34] L. Fu *et al.*, "CD-HIT: Accelerated for clustering the next-generation sequencing data," *Bioinf.*, vol. 28, no. 23, pp. 3150–3152, Dec. 2012.

[35] J. J. A. Armenteros *et al.*, "SignalP 5.0 improves signal peptide predictions using deep neural networks," *Nature Biotechnol.*, vol. 37, no. 4, pp. 420–423, Apr. 2019.

[36] J. R. Cole *et al.*, "Ribosomal database project: Data and tools for high throughput rRNA analysis," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D633–D642, Jan. 2014.

[37] D. Wang *et al.*, "DM3Loc: Multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism," *Nucleic Acids Res.*, vol. 49, no. 8, May 2021, Art. no. e46.

[38] B. Manavalan *et al.*, "4mCpred-EL: An ensemble learning framework for identification of DNA N(4)-methylcytosine sites in the mouse genome," *Cells*, vol. 8, no. 11, Oct. 2019, Art. no. 1332.

[39] T. Bepler and B. Berger, "Learning protein sequence embeddings using information from structure," 2019, *arXiv:08661*.

[40] R. Rao *et al.*, "Evaluating protein transfer learning with TAPE," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 9689–9701, 2019.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] R. Rao *et al.*, "Evaluating protein transfer learning with tape," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, Art. no. 9689.

[43] A. Nambiar *et al.*, "Transforming the language of life: Transformer neural networks for protein prediction tasks," in *Proc. 11th ACM Int. Conf. Bioinf., Comput. Biol. Health Informat.*, 2020, Art. no. 5.

[44] M. Heinzinger *et al.*, "Modeling the language of life–deep learning protein sequences," *bioRxiv*, 2019, Art. no. 614313.

[45] A. Radford *et al.*, "Improving language understanding by generative pre-training," 2018.

[46] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:04805*.

[47] A. Vaswani *et al.*, "Attention is all you need," 2017, *arXiv:03762*.

[48] S. u. J. J. o. C. I. Korkmaz, and Modeling, "Deep learning-based imbalanced data classification for drug discovery," *J. Chem. Inf. Model.*, vol. 60, no. 9, pp. 4180–4190, Sep. 2020.

[49] Y. Zhu *et al.*, "Inspector: A lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling," *Anal. Biochem.*, vol. 593, Art. no. 113592, Mar. 2020.

[50] L. Dou *et al.*, "Accurate identification of RNA D modification using multiple features," in *Proc. RNA Biol.*, 2021, pp. 1–11.

[51] J. Zhang, Z. Zhang, L. Pu, J. Tang, and F. Guo, "AIEpred: An ensemble predictive model of classifier chain to identify anti-inflammatory peptides," in *Proc. IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2020, pp. 1–1.

[52] Y. Ding *et al.*, "Identification of protein–ligand binding sites by sequence information and ensemble classifier," *J. Chem. Inf.*, vol. 57, no. 12, pp. 3149–3161, 2017.

[53] N. V. Chawla *et al.*, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[54] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[55] N. De Jay *et al.*, "mRMRe: An R package for parallelized mRMR ensemble feature selection," *Bioinf.*, vol. 29, no. 18, pp. 2365–2368, Sep. 2013.

[56] D. C. Hoaglin and R. E. Welsch, "The hat matrix in regression and ANOVA," *Amer. Statistician*, vol. 32, no. 1, pp. 17–22, 1978.

[57] A. M. Martinez and A. C. Kak, "Pca versus lda," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[58] Q. Zou *et al.*, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, no. P2, pp. 346–354, 2016.

[59] N. Q. K. Le *et al.*, "iMotor-CNN: Identifying molecular functions of cytoskeleton motor proteins using 2D convolutional neural network via Chou's 5-step rule," *Anal. Biochem.*, vol. 575, pp. 17–26, Jun. 2019.

[60] 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. Macromolecules*, vol. 157, pp. 752–758, Aug. 2020.

[61] M. M. Hasan *et al.*, "IRC-Fuse: Improved and robust prediction of redox-sensitive cysteine by fusing of multiple feature representations," *J. Comput. Aided Mol. Des.*, vol. 35, no. 3, pp. 315–323, Mar. 2021.

[62] L. Wei *et al.*, "Improved and promising identification of human microRNAs by incorporating a high-quality negative set," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 1, pp. 192–201, Jan. 2014.

[63] L. Wei *et al.*, "Integration of deep feature representations and handcrafted features to improve the prediction of N6-methyladenosine sites," *Neurocomputing*, vol. 324, pp. 3–9, Jan. 2019.

[64] L. Wei *et al.*, "Improved prediction of protein–protein interactions using novel negative samples, features, and an ensemble classifier," *Artif. Intell. Med.*, vol. 83, pp. 67–74, Nov. 2017.

[65] L. Wei, J. Tang, and Q. Zou, "Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information," *Inf. Sci.*, vol. 384, pp. 135–144, Apr. 2017.

[66] P. Baldi *et al.*, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinf.*, vol. 16, no. 5, pp. 412–424, May 2000.

[67] N. Q. K. Le and T.-T. Huynh, "Identifying SNAREs by incorporating deep learning architecture and amino acid embedding representation," vol. 10, no. 1501, Dec. 2019.

[68] D. T. Do and N. Q. K. Le, "Using extreme gradient boosting to identify origin of replication in saccharomyces cerevisiae via hybrid features," *Genomic.*, vol. 112, no. 3, pp. 2445–2451, May 2020.

[69] V. Vacic, L. M. Iakoucheva, and P. Radivojac, "Two sample logo: A graphical representation of the differences between two sets of sequence alignments," *Bioinf.*, vol. 22, no. 12, pp. 1536–1537, Jun. 2006.