Journal of Molecular Graphics and Modelling 107 (2021) 107962

Contents lists available at ScienceDirect

Journal of Molecular Graphics and Modelling

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

Prediction of protein ubiquitination sites via multi-view features based on eXtreme gradient boosting classifier

Yushuang Liu [a,b,1], Shuping Jin [a,b,1], Lili Song [a,b,1], Yu Han [a,b], Bin Yu [a,b,c,*]

[a] College of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao, 266061, China
[b] Artificial Intelligence and Biomedical Big Data Research Center, Qingdao University of Science and Technology, Qingdao, 266061, China
[c] Key Laboratory of Computational Science and Application of Hainan Province, Haikou, 571158, China

ARTICLE INFO

Keywords:
Ubiquitination sites
Multi-view features
LASSO
SMOTE
XGBoost

ABSTRACT

Ubiquitination is a common and reversible post-translational protein modification that regulates apoptosis and plays an important role in protein degradation and cell diseases. However, experimental identification of protein ubiquitination sites is usually time-consuming and labor-intensive, so it is necessary to establish effective predictors. In this study, we propose a ubiquitination sites prediction method based on multi-view features, namely UbiSite-XGBoost. Firstly, we use seven single-view features encoding methods to convert protein sequence fragments into digital information. Secondly, the least absolute shrinkage and selection operator (LASSO) is applied to remove the redundant information and get the optimal feature subsets. Finally, these features are inputted into the eXtreme gradient boosting (XGBoost) classifier to predict ubiquitination sites. Five-fold cross-validation shows that the AUC values of Set1-Set6 datasets are 0.8258, 0.7592, 0.7853, 0.8345, 0.8979 and 0.8901, respectively. The synthetic minority oversampling technique (SMOTE) is employed in Set4-Set6 unbalanced datasets, and the AUC values are 0.9777, 0.9782 and 0.9860, respectively. In addition, we have constructed three independent test datasets which the AUC values are 0.8007, 0.6897 and 0.7280, respectively. The results show that the proposed method UbiSite-XGBoost is superior to other ubiquitination prediction methods and it provides new guidance for the identification of ubiquitination sites. The source code and all datasets are available at https://github.com/QUST-AIBBDRC/UbiSite-XGBoost/.

1. Introduction

Post-translational modification (PTM) plays an essential role in protein function and cell physiology [1–3]. After the coding gene DNA sequence is transcribed into mRNA and translated into protein sequence, post translational modification usually is occurred. However, abnormal post-translational modification may cause certain diseases [4]. For instance, human cancer, neurodegenerative diseases, immune diseases, metabolic syndrome, and other physiological mechanisms of disease [5]. Protein ubiquitination is a common post-translational modification. It is a process that attaches ubiquitin protein to the substrate. The increase of ubiquitin-protein can affect the function of the protein in many ways. For example, it can send signals through the proteasome to degrade proteins [6–8], change cell positioning, promote and prevent protein interactions [9,10]. Meanwhile, ubiquitin-protein is a small regulatory protein in ubiquitination modification, which can be found in almost all eukaryotic tissues. Ubiquitination is performed in three steps, that are activation, binding, and connection. The execution is carried out by ubiquitin activating enzyme E1s, E2s, E3s, and proteasomes, which together constitute ubiquitin Proteasome system [11–13]. Therefore, ubiquitination is a major field of bioinformatics research in recent years and becomes a new target for research and development of drugs [14].

The identification of ubiquitination sites is a key step in understanding the mechanism of ubiquitination [15]. However, ubiquitination is rapid and reversible. Traditional experimental methods include high-throughput mass spectrometry (MS) technology [16–19], ubiquitin antibodies and ubiquitin-binding protein [19,20], combined with liquid chromatography and mass spectrometry [21]. There methods are time-consuming and expensive. Currently, machine learning is used to study protein ubiquitination sites, which greatly reduced the cost of experimental identification. The first study on ubiquitination sites was developed by Tung et al. [22] in 2008, who proposed a prediction model

* Corresponding author. College of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao, 266061, China.
E-mail address: yubin@qust.edu.cn (B. Yu).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.jmgm.2021.107962
Received 4 December 2020; Received in revised form 3 May 2021; Accepted 2 June 2021
Available online 15 June 2021

called UbiPed and used the IPMA algorithm to select 31 physicochemical attributes of 531 features in AAindex data. Finally, this feature information was inputted into classifier to predict ubiquitination sites. Chen et al. [23] proposed a prediction model called hCKSAAP_UbSite which using support vector machine (SVM) classifier incorporated with *k*-spaced acid pairs, the binary amino acid coding, amino acid property encoding, and the protein aggregation propensity to identify ubiquitination sites. The AUC values of the prediction model was 0.7700 on training datasets and the AUC of the independent test set was 0.7570. Qiu et al. [24] proposed a prediction model called IUBIq-LYS, using the gray system model from extract sequence evolution information and SVM classifier to identify ubiquitination sites. Wang et al. [25] constructed a prediction model EAS-Ubisite based on SVM, applying the physical and chemical properties to predict ubiquitination sites of the negative samples. Finally, the accuracy of prediction model was 0.9200 and the Markov correlation coefficient reached 0.4800. Cai et al. [26] based on the physical and chemical properties of proteins used four Bayesian networks including Naïve Bayes (NB), feature selection NB, model averaged NB, and efficient bayesian multivariate classifier. Three regression methods including SVM, Logistic Regression and LASSO were used to predict ubiquitination sites. Finally, the AUC values of the six datasets reached 0.7244, 0.6467, 0.6667, 0.7200, 0.7235 and 0.6001, respectively. Nguyen et al. [27] based on SVM incorporated amino acid composition, amino acid pair composition (AAPC) and evolutionary information which fused two or more features to extract protein information. Cai et al. [28] based on the K nearest neighbor (KNN) classifier developed a tool for predicting ubiquitination sites. Also they used maximum relevance and minimum redundancy (mRMR) and incremental feature selection (IFS) methods to optimize features, and consequently to get the optimal feature subset. Cui et al. [29] proposed a protein ubiquitination sites method called UbiSitePred. Proposed pseudo amino acid composition, binary encoding, composition of k-spaced acid pairs and position specific score matrix methods were used to extract feature information. Also, LASSO was used to reduce the dimension of fusion information. The prediction accuracy rate of ubiquitination sites was improved effectively.

The imbalance of the data set has a certain impact on prediction model, which will cause the results to be mostly determined by the majority of samples. He et al. [30] used the self-sampling bootstrapping method to deal with the imbalance data. Finally, the accuracy of test set reached 66.43%. Wang et al. [31] developed a S-sulfenylation sites predictor named SulSite-GTB, which used SMOTE algorithm to deal with unbalanced samples and LASSO was selected to remove redundant information. SulSite-GTB achieved accuracy of 88.53%, AUC of 0.9425, and MCC of 0.7740 on the independent test dataset. Compared with the other tools of predicting S-sulfenylation sites, this predictor significantly improved the prediction performance of the model.

Although the identification of ubiquitination sites achieved remarkable results, there are still many great room for improvement. First, the importance of multi-view features in the study of ubiquitination sites is ignored. Second, feature selection contains redundant information, which interferes with the prediction of ubiquitination sites. Finally, the imbalance between positive samples and negative samples may lead to overfitting phenomenon, thus ignoring the influence of minority samples.

For the above three issues, this paper proposes a novel method of protein post-translational modification called UbiSite-XGBoost. First, the single-view features including proposed pseudo amino acid composition (PseAAC), composition of *k*-spaced acid pairs (CKSAAP), adapted normal distribution bi-profle Bayes (ANBPB), amino acid property encoding (AAindex), encoding based on grouped weight (EBGW), BLOSUM62, and pseudo position-specific score matrix (PsePSSM) are transformed into digital information. Secondly, we employ LASSO to remove redundancy information and select the optimal feature subset. Finally, the XGBoost classifier is used to predict the ubiquitination sites. The protein ubiquitination prediction model is evaluated by five-fold cross-validation. The AUC values on Set1-Set6 datasets reach more than 0.7500. Besides, the SMOTE method is employed in Set4-Set6 unbalanced datasets, and the AUC values are 0.9777, 0.9782 and 0.9860, respectively. Meanwhile, we have constructed three independent test datasets which the AUC values are 0.8007, 0.6897 and 0.7280, respectively. The results show that the UbiSite-XGBoost method can effectively identify ubiquitination sites and improve the prediction effect of post-translational modifications.

2. Materials and methods

2.1. Datasets

In this paper, the six datasets were collected from different databases and literatures [21–23,26,32]. The data Set1 was selected from the UniProt database [22]. 157 lysine ubiquitination sites were obtained from 105 proteins, which containing ubiquitination sites were positive samples. Meanwhile, the negative samples were 3676 lysine protein sequences not labeled with ubiquitination sites. The sequence length is 13. The data Set2 and Set3 were extracted 9537 ubiquitination sites from 3852 proteins, which were obtained independent data set and training datasets of literature [23], respectively. Blastclust was used to remove redundancy and make homology less than 30%. The data Set4-Set6 were three unbalanced datasets obtained from different literatures, respectively. The initial datasets were filtered with a 40% sequence identification threshold [21,32]. The sequence length of the data Set2-Set6 is 27. For the convenience of writing the following articles, the six training datasets are denoted as Set1, Set2, Set3, Set4, Set5 and Set6, respectively. In addition, form the database were built by Xu et al. [33], three independent test datasets were constructed. First, three independent test datasets were derived from 11, 34 and 13 proteins, respectively, which positive samples were experimentally identified and the rest were negative samples. Secondly, CD-HIT [34] was used to eliminate 30% of sequence redundancy. Finally, the three datasets were 393, 651 and 762 equally-long protein sequences, named Independent test1, Independent test2, Independent test3, respectively. To ensure the sequence length the same, we used 'X' instead of a meaningless amino acid. The specific datasets are shown in Table 1.

2.2. Feature encoding

2.2.1. Sequence-based features

PseAAC: In order to use effective numerical expressions to form protein or polypeptide samples [35], Chou et al. [36] proposed pseudo amino acid composition (PseAAC), which was applied to reflect the correlation between proteins and predictive properties. This method replaced the traditional amino acid composition (AAC) method and was widely used in bioinformatics protein and protein-related predictions. For instance, predicting protein subcellular localization [37] and membrane protein type, etc. [38–41]. The general form of PseAAC is:

$$V = (V_1, V_2, \cdots, V_{20}, V_{20+1}, \cdots, V_{20+m})^T \quad (1)$$

The components are as follows:

Table 1
The number of datasets on Training datasets and Independent test datasets.

Datasets	Name	All	Positive	Negative
Train datasets	Set1	300	150	150
	Set2	6838	3419	3419
	Set3	12,236	6118	6118
	Set4	4608	263	4345
	Set5	3651	131	3520
	Set6	676	37	639
Independent test sets	Independent test1	393	92	301
	Independent test2	651	176	475
	Independent test3	762	96	666

$$v_i = \begin{cases} \dfrac{f_i}{\sum_{j=1}^{20} f_i + \omega \sum_{k=1}^{m} C_k} & 1 \le i \le 20 \\ \dfrac{\omega C_{i-20}}{\sum_{j=1}^{20} f_i + \omega \sum_{k=1}^{m} C_k} & 20+1 \le i \le 20+m \end{cases} \quad (2)$$

where f_i is the frequency of the i-th amino acid appearing in the protein, and the ω is the weighted factor, set at 0.05. C_k is k-th sequence correlation factor, and m is the characteristic parameters. 20+ m dimensional vector represents 20 kinds of amino acid composition and m-dimensional sequence related factors [29,42–44]. Finally, 20+ m dimensional feature vectors are obtained.

CKSAAP: The composition of k-spaced acid pairs (CKSAAP) is a encoding method that converts protein feature coding to digital information. It is widely used in bioinformatics [23,45–47]. The 21 amino acids form 441 amino acid pairs, e.g. $AA, AC, \cdots, XX$ and n is the sequence length. the feature vector is:

$$\left(\frac{N_{AA}}{N_{total}}, \frac{N_{AC}}{N_{total}}, \cdots, \frac{N_{XX}}{N_{total}}\right) \quad (3)$$

where N_{ij} is the total number of ij amino acid pairs included at the k interval, N_{total} is the number of all amino acid pairs with k as the interval, and $N_{total} = n - k - 1$ can be obtained by the composition of feature vectors [29].

ANBPB: adapted normal distribution bi-profle Bayes (ANBPB) theorem is proposed by Jia et al. [48]. This method is improvedbased on BPB algoruthm [49]. The probability vector of a peptide is encoded by considering the information both positive and negative samples. Suppose that we have an unlabeled sample $P = \{p_1, p_2, \cdots, p_n\}$, where each $p_j (j = 1, 2, \cdots, n)$ is an amino acid and n represents the sequence window size. The ANBPB feature vector $\overrightarrow{P}$ is:

$$\overrightarrow{P} = (p_1, p_2, \cdots, p_n, p_{n+1}, \cdots, p_{2n}) \quad (4)$$

where $p_j = (1, 2, \cdots, n)$ and $p_j = (n+1, n+2, \cdots, 2n)$ represent the posterior probability of each amino acid in the positive and negative sample datasets, respectively. The posterior probability $p_j (j = 1, 2, 3 \cdots, 2n)$ is coded by the adapted normal distribution as follows:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt \quad (5)$$

where, $\phi(x)$ is the standard normal distribution function. Finally, adds the positive and negative of these two samples to obtain a vector of 2* n dimension.

2.2.2. Physicochemical property-based features

AAindex: Amino acid property encoding (AAindex) is a database [50], which can reflect the unique properties of amino acids. However, some uncertain information entries make the number of amino acid properties smaller than the total number of amino acid properties. 544 features are collected from the AAindex database [51,52]. In this paper, according to the research of Xiang et al. [53], we select 16 amino acid physicochemical properties from the AAindex database as the objects for analysis. Therefore, the length of the protein sequence is L, and each amino acid residue gets 16 physical and chemical property scores. Finally, the 16*L dimensional characteristics are obtained.

EBGW: Encoding based on grouped weight (EBGW) is a feature encoding method for protein sequences proposed by Zhang et al. [54]. According to the hydrophobicity and electric charge of amino acids, the amino acids are regarded as four categories, as shown in Table 2.

$< W1, W2 > vs < W3, W4 >$, $< W1, W3 > vs < W2, W4 >$,$< W1, W4 > vs < W2, W3 >$ are the three groups that do not overlap. the protein sequence P, the following formulas are used to change the sequence P into three binary sequences.

Table 2
Grouping of the 20 Standard acids to four basic groups (W1–W4).

Group	Amino acids
Neutral and hydrophobic amino acids (W1)	A, F, G, I, L, M, P, V, W
Neutral and polar amino acids (W2)	C, N, Q, S, T, Y
Acidic amino acid (W3)	K, H, R
Basic amino acids (W4)	D, E

$$f_1(p_i) = \begin{cases} 1, p_i \in \{W1, W2\} \\ 0, p_i \in \{W3, W4\} \end{cases} \quad (6)$$

$$f_2(p_i) = \begin{cases} 1, p_i \in \{W1, W3\} \\ 0, p_i \in \{W2, W4\} \end{cases} \quad (7)$$

$$f_3(p_i) = \begin{cases} 1, p_i \in \{W1, W4\} \\ 0, p_i \in \{W2, W3\} \end{cases} \quad (8)$$

The three binary sequences are divided into L sliver sequences, and a fixed parameter L is set. For f_1, the j sequence is expressed as $R(j) = Sum(j)/Q(j)$, where $Sum(j)$ is the number 1 of the j-th subsequence in the entire sequence, $Q(j) = \text{int}(j * l / L)$ is the length of the j-th subsequence, and l represents the length of the protein sequence. In summary, the 3*L dimension vector can be acquired.

2.2.3. Evolutionary-derived features

BLOSUM62: This feature extraction method mainly reflects the *evolutionary* information of proteins and is suitable for describing those with the same coding length [52]. The matrix was established by considering the differences in physicochemical properties between two amino acids. The similarity between the two peptides did not exceed 62%, and each line represented one of the 20 common amino acids [15]. For each amino acid substitution score, protein sequence of length L is represented by a 20*L dimensional feature vector.

PsePSSM: As one of the important indicators for the assessment of biological performance, position specific score matrix (PSSM) was applied in numerous biological researches. This method was firstly proposed by Jones et al. [55]. It used 20 primitive amino acids and mainly found the evolutionary information of protein sequences. PSI-BLAST program [56] compared all protein sequences with SwissProt non-redundant database [31]. In this paper, the protein sequence length is L. When feature extraction is carried out, the positional specificity score matrix obtained is a L*20 matrix. The algorithm is:

$$P_{PSSM} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,20} \\ \cdots & \cdots & \cdots & \cdots \\ p_{i,1} & p_{i,2} & \cdots & p_{i,20} \\ \cdots & \cdots & \cdots & \cdots \\ p_{L,1} & p_{L,2} & \cdots & p_{L,20} \end{pmatrix} \quad (9)$$

where p_{ij} denotes the position-specific score obtained by replacing the i amino acid with the j amino acid during the evolutionary process [57–59]. To consider the sequence information of the residues amino acids, pseudo position-specific score matrix (PsePSSM) was proposed [60]. Before using PsePSSM, standardization should be carried out first, and then formula should be used:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

The elements in the original standardized PsePSSM matrix are reduced to 0 or 1, and the standardized matrix is denoted as $\overline{P_{PSSM}} = (\overline{P_1}, \overline{P_2}, \cdots, \overline{P_{20}})$. However, this method is not perfect, and it has no real meaning for some residues amino acids without mutations. Its algorithm is as follows:

$$\theta_j^{\lambda} = \frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}\left(P_{i,j} - P_{(i+\lambda),j}\right)^2, \quad (j = 1, 2, \cdots, 20;\ \lambda < L,\ \lambda \neq 0) \tag{11}$$

where θ_j^{λ} is the order correlation factor λ of amino acid type j, thus a protein sequence can be expressed as a formula by PsePSSM:

$$P_{PsePSSM} = \left(\overline{P_1}, \overline{P_2}, \cdots, \overline{P_{20}}, \theta_1^1, \cdots, \theta_{20}^1, \cdots, \theta_1^{\lambda}, \cdots, \theta_{20}^{\lambda}\right)^T \tag{12}$$

It can be seen from the formula that using the algorithm will generate a $20*\lambda + 20$ matrix.

2.3. LASSO

The least absolute shrinkage and selection operator (LASSO) algorithm was first put forward by Tibshirani [61]. It is mainly used to simplify the model by constructing a penalty function. The dimension reduction method is as follows: First, given a sample dataset $X = [x_1, x_2, \cdots, x_N]^T \in R^{N\times d}$, x_i represents the feature vector of the i sample, N is the number of samples, d is the feature vector. $Y = [y_1, y_2, \cdots, y_N] \in R^N$represents the feature vector that corresponds to these samples. Secondly, the optimized objective function after dimension reduction using LASSO is:

$$\min\left\|Y - X\omega^T\right\|_2^2 + \lambda\|\omega\|_1 \tag{13}$$

where ω is the estimation vector of regression coefficient, $\lambda > 0$ is the data fitting degree of regularization parameter in the model, and its complexity of equilibrium model. If the regular term ω_1 adopts ℓ_1-normal form, the characteristic space will be sparse. Finally, the irrelevant and redundant feature coefficient can be set as 0, and characteristics of non-zero coefficients are retained for subsequent classification.

2.4. SMOTE

For unbalanced data, the results of classifier algorithm are usually greatly influenced by the majority of samples. When there is a large gap between positive and negative samples, overfitting may occur, and then the influence of a small number of samples is ignored. The synthetic minority oversampling technique (SMOTE) is an effective method put forward by Chawla et al. [62] in 2002 that used stochastic undersampling for the majority of class samples and stochastic oversampling for the minority class samples to reach the balance of data. The algorithm can be described as: for each sample x in a small class of dataset, using the euclidean distance to calculate the k nearest neighbor of each sample, and then the sampling ratio of M is determined according to the unbalanced ratio column of the few samples. M samples are randomly selected from k nearest neighbors of each sample [63]. If $x_n (n = 1, 2, 3, \cdots, M)$ is selected as the nearest neighbor and random linear interpolation is conducted between a small number of samples and x_n, then the new sample x_{new} is:

$$x_{new} = x + rand\ (0,\ 1) \times (x_n - x) \tag{14}$$

where $rand(0, 1)$ represents the generation of random numbers between 0 and 1. A new low-class sample set is added to the original low-class sample set. Finally, a new dataset is formed, that is, a dataset balanced with the number of multi-class samples.

2.5. XGBoost

The eXtreme gradient boosting (XGBoost) was improved by Chen et al. [64] and widely applied in bioinformatics [65–68]. In this algorithm, loss function was deduced by second-order Taylor extension and regularization terms controlling the complexity of the model were added to the objective function, effectively avoiding over-fitting [69]. In the classifier, model performance is predicted by adding new trees through continuous feature splitting. Each additional tree will learn a new function $f(x)$ and then fit the residua predicted last time. Finally, the corresponding score of each tree is added up to the predicted value of the sample. The objective function during training divide into two parts: the gradient boosting algorithm and the regular term. The objective function is defined as:

$$L(\varnothing) = \sum_{i=1}^{n} l(y^{'}_i, y_i) + \sum_k \Omega(f_k) \tag{15}$$

where n is the number of samples, $\Omega(f_k)$ represents the regular term, f_k represents the decision tree, and if it is a convex function, then l represents the loss of a single sample. Regularization item defines the complexity of the model:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \tag{16}$$

where ω is a vector of leaf node values in the decision tree, and T is the number of leaf nodes. But for the loss function, the model can not be optimized in Euclidean space when the function is used as the parameter in the formula to integrate the tree [69].

Therefore, f_t is chosen to be added to the loss function to train the model, and $\hat{y}_i^{(t)}$ represents the prediction that becomes the i instance in the next iteration. The second-order Taylor expansion of the objective function is carried out by using Taylor expansion. The regularization target can be expressed as:

$$L(\varphi) = \sum_{i=1}^{n}\left[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \tfrac{1}{2} b_i f_t^2(x_i)\right] + \Omega(f_t) \tag{17}$$

where $f_t(x)$ is an increment, the first-order and second-order gradient statistics of the loss function are represent by $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $b_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$. The model in XGBoost stops training when the stop criteria are met or the number of accelerated iterations t is met.

2.6. Performance evaluation

To better assess the performance of this method, this article selects four indicators [70–73], accuracy (ACC), specificity (Sp), sensitivity (Sn), F-score with Matthew's correlation coefficient (MCC). Where Sn is the correct ratio of postive samples for prediction and Sp is the correct ratio of prediction for negative samples, the ACC is positive samples and negative samples prediction correct ratio, the value range of these three indicators is 0–1; The MCC reflects the degree of correlation between the predicted situation and the real target value. If the value is positive, the predicted value is related to the target, otherwise it is not related and its value can be from −1 to 1. When the value is the largest, the best predictive performance is achieved. Each indicator is defined as follows:

$$Sn = \frac{TP}{TP + FN} \tag{18}$$

$$Sp = \frac{TN}{TN + FP} \tag{19}$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{20}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{21}$$

In the above formulas, TP, TN, FP, and FN denotes true positive, true negative, false positive, and false negative, respectively. In addition,

AUC value and ROC curve [74] are used as the comprehensive index of the robustness of the evaluation model. With the higher AUC value, the greater the area under the ROC curve, and the better the robustness of the model.

The prediction method in this paper is called UbiSite-XGBoost, and Fig. 1 shows the specific process. The operating environment is: Windows Server 2012R2 Intel (R) Xeon (TM) CPU e5-2650@2.30ghze5-2650@2.30ghz 2.30ghz with 32.0GB of RAM, MATLAB2016a and Python3.6.

The steps of the UbiSite-XGBoost model can be described in the following ways:

(1) The training datasets and Independent test datasets are obtained from UniProt and literatures, respectively, and the dataset is divided into positive samples and negative samples.
(2) Feature coding. Protein sequences are converted to digital information based on amino acid residues sequence information, physical-chemical properties and evolutionary-derived features. Feature extraction methods with parameters, such as PseAAC, CKSAAP, EBGW, PsePSSM are adjusted and the optimal parameters are selected.
(3) Feature selection and balance data. The redundant information is removed by LASSO and the optimal feature subset is obtained. Besides, the optimal characteristic subset of Set4, Set5, Set6 and three independent test sets treated by SMOTE method in balance.
(4) Selection and evaluation of models. According to the step 2) and 3), the selected optimal feature subset is inputted into the XGBoost classifier, the independent test sets and five cross-validation are used to validate the model predicted performance. The ACC, AUC, Sn, Sp, MCC as evaluation indicators were used to evaluate the predictive performance of the model.

3. Results and discussion

3.1. Sequence analysis of protein ubiquitination sites

In this paper, we used Two-Sample-Logo [75] to draw the double-sequence identification diagram of the sample fragment of the ubiquitination sites and the non-ubiquitination sites in the datasets, which can more clearly compare the residue information near the ubiquitination sites and the nonubiquitination sites. Due to different sequence lengths, the positions of flanking amino acids in the Set1 dataset are described within the range of −6 to +6, and the five Set2-Set6 datasets are described within the range of −13 to 13. The sequence analysis diagram of the six data sets is shown in Fig. 2.

As can be seen from Fig. 2, there are obvious differences in amino acids near of ubiquitination sites and non-ubiquitination sites. For example, for the Set1 dataset, the positively charged residues of lysine (K), negatively charged valine (V), and non-polar residues phenylalanine (F) are significantly more important in the downstream. Glutamic acid (E) and aspartic acid (D) are enriched in the positive sample. For the

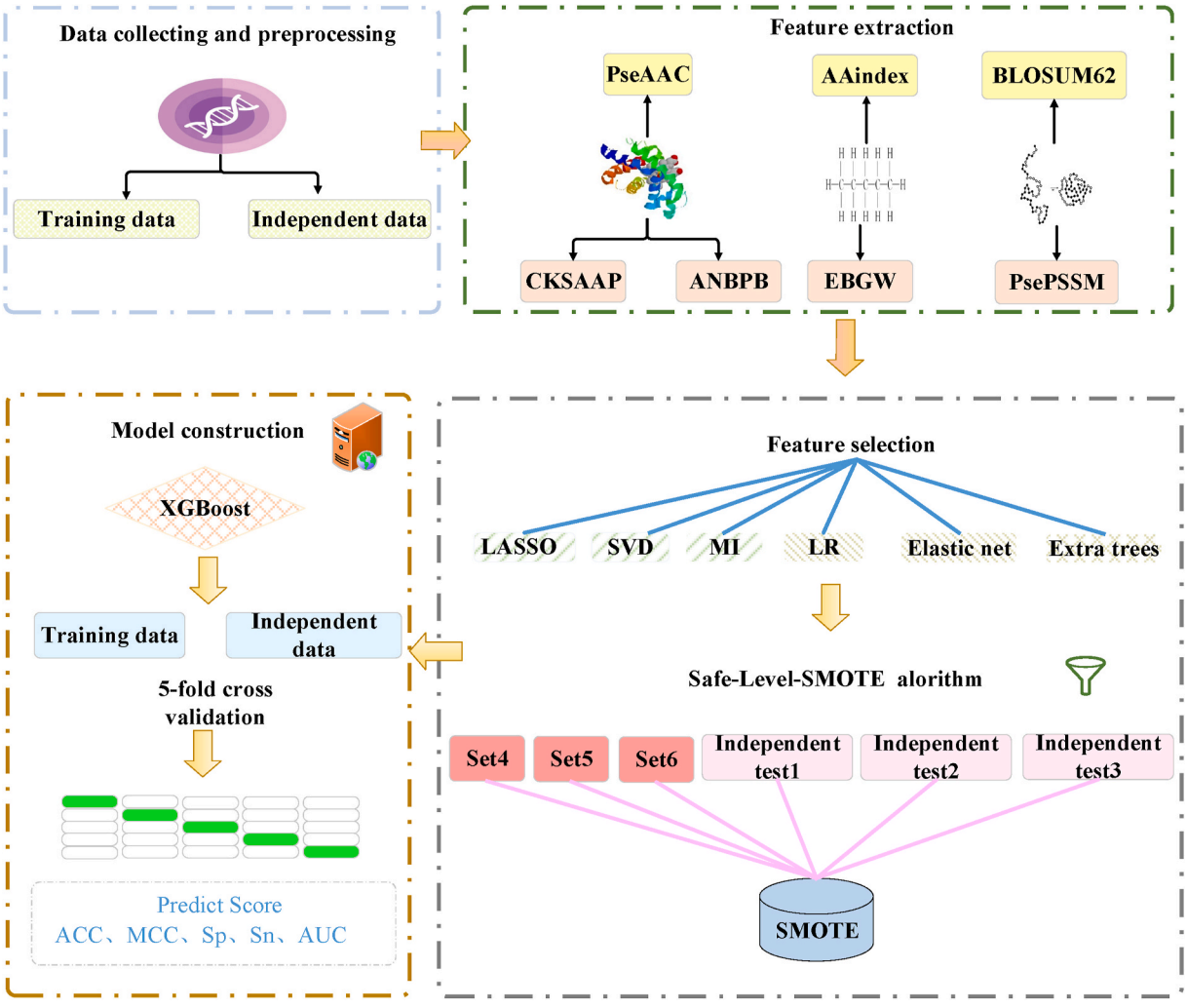

Fig. 1. Schematic diagram of the UbiSite-XGBoost prediction method.

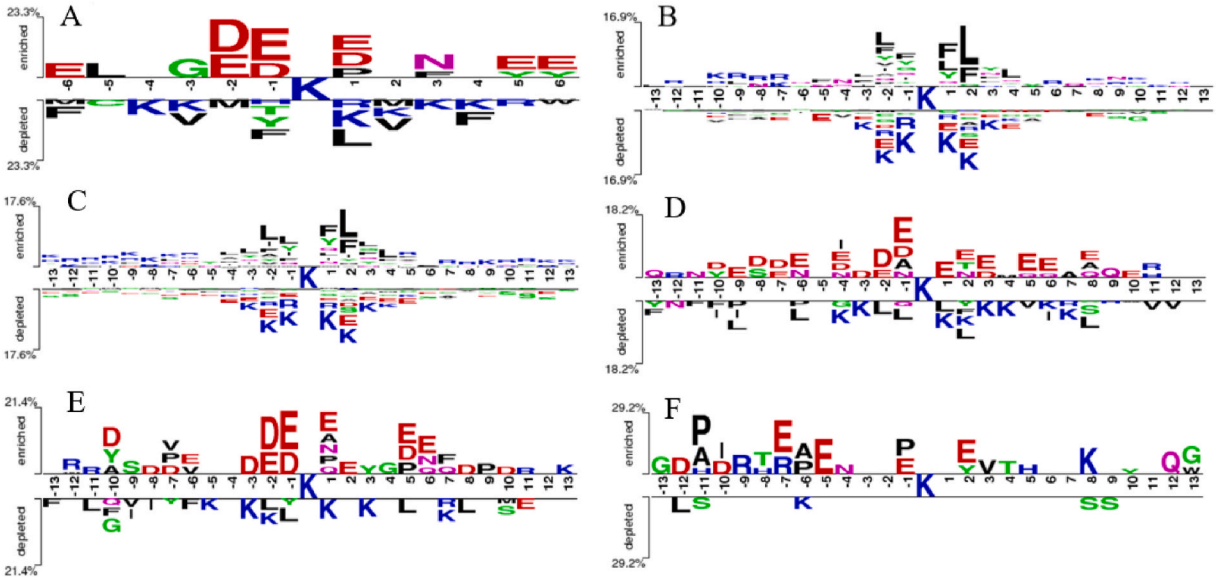

Fig. 2. Sequence characteristics positive and negative samples across the Set1-Set6 datasets.

Set6 dataset, the amino acid residues of near the ubiquitination sites (K) are relatively rare. Glutamic acid (E) is significantly enriched at −7, −5, −1, +2 positions, and lysine (K) also occurs very infrequently, only at −6 and +8. Therefore, it can be inferred that the distance between amino acid positions in the sequence plays an important role in distinguishing ubiquitination sites and the non-ubiquitination sites.

3.2. Analysis of feature parameters results

To make the model achieve better prediction results, the parameters of PseAAC, CKSAAP, EBGW, and PsePSSM were adjusted, respectively. We input different parameters into the XGBoost classifier and use five-fold cross-validation to obtain different AUC values of corresponding parameters, then the optimal parameters are selected according to the AUC values. The sequence length of Set1 is 13, so the interval parameters of in the CKSAAP are selected to 0,1,⋯,11. Since the datasets contain virtual amino acids X, the values of m in the PseAAC is set 1,2,⋯,7. Also, the number of parameters in the EBGW is set to 1,2,⋯,13, and the parameters of PsePSSM is set to 0,1,2,⋯,13. The selection process is shown in Fig. 3. The detailed optimization results are shown in Table S1–S4.

It can be seen from Fig. 3 (A) that the AUC reaches its maximum value for all six data sets $k = 7$. Therefore, it is selected $k = 7$ as the optimal parameter of CKSAAP. The dimension of each protein sequence is 441*8 = 3528 dimensional vectors. In Fig. 3 (B), when the optimal parameter is $m = 4$, an eigenvector with dimension 20 + 4 = 24 is obtained for each data set. In Fig. 3 (C), it can be seen that Set1-Set6 parameters are all reaching the optimal value. Therefore, the model is finally selected $L = 10$ as the optimal parameter and the extracted dimension of EBGW is 3*10 = 30. It can be seen from Fig. 3 (D), the optimal parameter $\lambda = 12$ value is selected in the PsePSSM algorithm. Finally, $20 + 20*\lambda = 260$ dimensional vectors are obtained for each protein sequence.

3.3. Analysis of feature extraction results

In bioinformatics, the use of effective features to extract sequence information has a significant impact for the prediction results of the model. However, the single-view features encoding method cannot explain the feature information of the ubiquitination sites. We employ seven feature extraction methods, PseAAC, CKSAAP, ANBPB, AAindex, EBGW, and BLOSUM62, respectively. These methods are inputted the classifier XGBoost to obtain the new predicted values. The AUC results are shown in Table 3.

From Table 3, for Set1 dataset, the AUC of evolutionary-derived information ANBPB reaches 0.7438. The AUC of the optimal single feature BLOSUM62 in Set2 dataset reaches 0.6930. For Set3 dataset, the AUC physicochemical properties feature AAindex reaches 0.7420. For Set4 and Set5 datasets, the AUC value of CKSAAP is superior to other single features, reaching 0.7282 and 0.7857, and the highest value of Set6 single feature is 0.7567. After the fusion of the six datasets, the AUC values are 0.7327, 0.7447, 0.7732, 0.7593, 0.8112 and 0.6954, respectively. Therefore, the feature encoding after the fusion of seven features are more comprehensive, and better prediction results are obtained, which further illustrates the importance of multi-view features.

3.4. Analysis of feature selection results

After seven features are extracted, the extracted features are fused to form a multi-dimensional feature space. It contains noise and redundant information, which has certain interference to the prediction of ubiquitination sites. In order to remove some uncorrelated features and get a better training model, this article compares six dimension reduction methods including LASSO, mutual information (MI) [76,77], Elastic net [78–80], logistic regression (LR) [81], extra-trees (ET) [82], and singular value decomposition (SVD) [83]. Among them, the LASSO parameter is 0.001–0.006, and XGBoost is selected for classification and AUC is used as the evaluation index. The results show that the LASSO dimension reduction effect is the best when the parameter is 0.005. The LR parameter is 0.01–0.1, XGBoost is selected to classify and AUC is used as the evaluation index. The results show that the LR dimension reduction effect is the best when the parameter is 0.09. The ET parameters are selected from 10 to 100, XGBoost is selected for classification and AUC is used as the evaluation index. The results show that the ET dimension reduction effect is the best when the parameter is 50. In addition, the corresponding dimensions of MI and SVD are consistent with LASSO. The corresponding dimensions and detailed values of the six datasets in different dimension reduction methods are shown in Table S5. To compare these dimensional reduction methods more conveniently and intuitively, the ROC curve is drawn, as shown in Fig. 4.

As shown in Fig. 4, the ROC curve of LASSO in six different datasets contains MI, Elastic net, LR, ET, and SVD respectively. The maximum AUC values of the seven dimensional reduction methods are 0.8258, 0.7592, 0.7853, 0.8345, 0.8979, and 0.8901 respectively. For Set2, Set3,

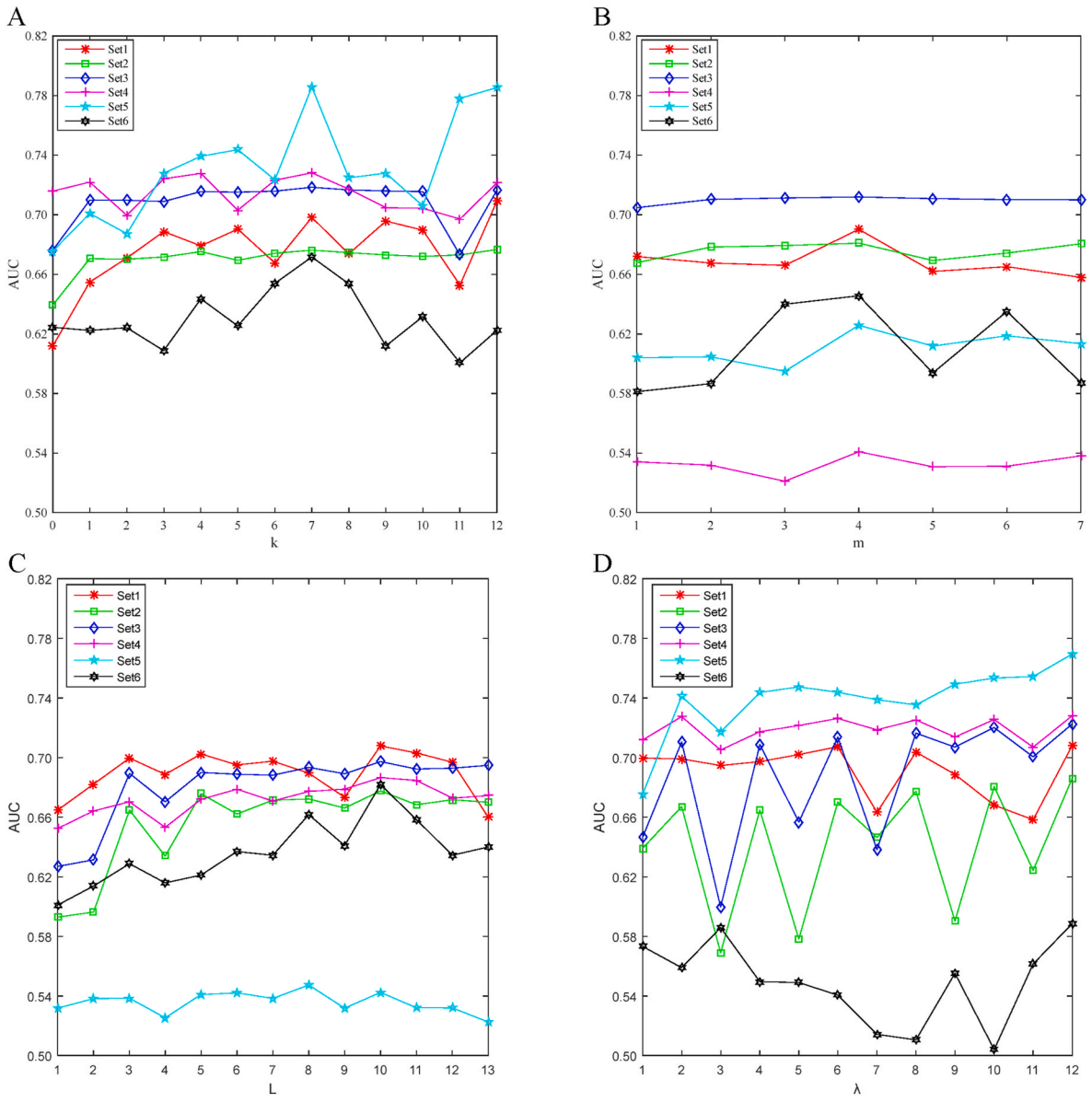

Fig. 3. Based on the Set1-Set6 data set, the comparison results of different parameters. (A) The parameter optimization of k in CKSAAP. (B) The parameter optimization of m in PseAAC. (C) The parameter optimization of L in EBGW. (D) The parameter optimization of λ in PsePSSM.

Table 3
On the basis of the Set1-Set6, the comparison of the AUC value of different feature codes.

AUC	Set1	Set2	Set3	Set4	Set5	Set6
PseAAC	0.6902	0.6809	0.7120	0.5409	0.6258	0.6456
CKSAAP	0.6882	0.6763	0.7184	0.7282	0.7857	0.6717
ANBPB	0.7438	0.6896	0.7085	0.7003	0.7098	0.7567
AAindex	0.6811	0.6862	0.7420	0.6552	0.4996	0.5994
EBGW	0.7078	0.6777	0.6973	0.6865	0.5423	0.6819
BLOSUM62	0.7131	0.6930	0.7416	0.6752	0.5115	0.5728
PsePSSM	0.7080	0.6859	0.7225	0.7280	0.7694	0.5884
ALL	**0.7327**	**0.7447**	**0.7732**	**0.7593**	**0.8112**	**0.6954**

and Set6 datasets, the AUC value of SVD dimension reduction has the worst performance. The AUC value of LASSO is 7.96%, 5.89%, and 21.12% are better than the corresponding value of SVD. For the Set1 datasets, ET method predicts that the ubiquitination sites reach the lowest value, and the -corresponding AUC value is 13.69% lower than LASSO. Therefore, through the above comparison, it can be concluded that the feature selection algorithm LASSO can effectively improve the robustness of the model, provide excellent feature information for the classifier, and more accurately predict the ubiquitination sites.

3.5. *The effectiveness of SMOTE*

In this paper, a total of six datasets have been studied, among which Set1, Set2, and Set3 are balanced datasets, while Set4, Set5, and Set6 are seriously unbalanced datasets. When there is a large gap between positive and negative samples, overfitting may occur, thus ignoring the influence of a small number of samples. In order to avoid overfitting and improve the generalization ability of the model, this paper reasonably deletes or adds some samples to achieve data balance [84]. We use the SMOTE algorithm on the basis of LASSO processing data to randomly under-sample the negative samples in the 3 datasets, and randomly over-sample the positive samples. In this paper, we select the over-sample parameter of 500 and under-sample parameters of 120 to obtain three balanced datasets of 3156, 1572, and 444 respectively. The comparison of AUC values in the classification algorithm XGBoost

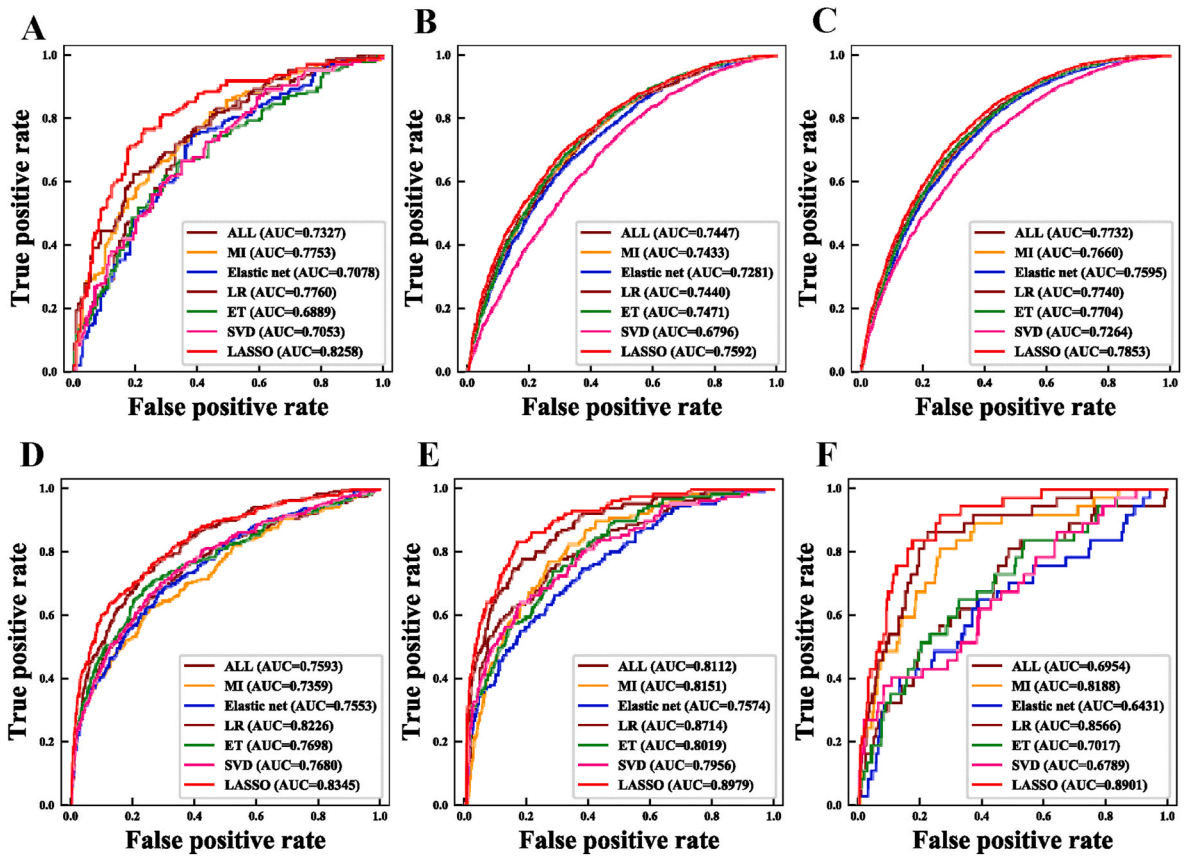

Fig. 4. The ROC curves of ALL, MI, Elastic net, LR, ET, SVD, LASSO feature selection. (A–F) on Set1-Set6. ALL: PseAAC + CKSAAP + ANBPB + EBGW + AAindex + BLOSUM62+PsePSSM.

between the balanced and unbalanced datasets is shown in Fig. 5.

It can be seen from Fig. 5 that the predictive performance of the built model has a significant advantage while using the three data sets after SMOTE to build the model. Taking AUC as the evaluation index, the AUC values of Set4, Set5, and Set6 in the balanced dataset are 0.9777, 0.9782, and 0.9860 respectively. For Set4, the AUC value of the balanced dataset is 14.32% higher than that of the unbalanced dataset. The AUC value of the balanced dataset in Set5 is 8.03% higher than that of the unprocessed dataset. The AUC value of the balanced dataset in Set6 is 9.59% higher than that of the unbalanced dataset. Therefore, the AUC value of Set4-Set6 after SMOTE algorithm is significantly better than the corresponding value of three original data which can effectively improve the accuracy of predicting the ubiquitination sites.

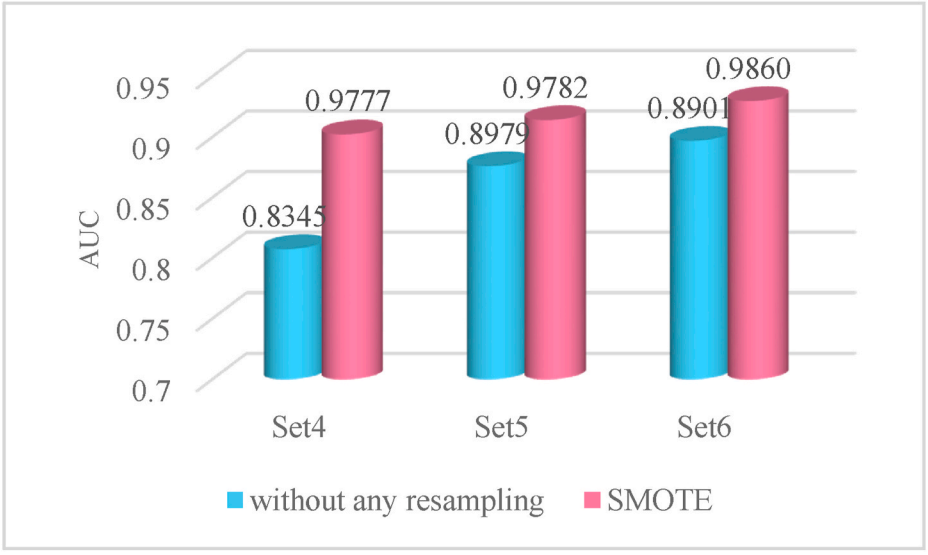

Fig. 5. Comparison of the AUC values before and after SMOTE method on Set4-Set6 datasets.

3.6. Selection of classification algorithm

To verify the effectiveness of XGBoost method, we select eight classification algorithms for comparison. They include XGBoost [64], gradient boosting tree (GTB) [85,86], random forest (RF) [60,85,87–89], decision tree (DT), K-nearest neighbor algorithm (KNN) [90–92], LightGBM [93–95], Extra Trees [96], AdaBoost [97], and Bagging [98,99], where XGBoost, GTB, DT, ExtraTrees, AdaBoost, and Bagging algorithms all adopt default parameters. The iteration times of LightGBM is 500 and the learning rate is 0.01. The Euclidean distance is used in the KNN algorithm, and the number of neighboring points is 5. The number of RF decision trees is set to 1000, and the node segmentation criterion is Gini coefficient.

The purpose of better explain the prediction results, ROC curves of six datasets under nine different classifiers are analyzed. The prediction performance of nine classifiers is mainly determined by the ROC curve, namely the AUC value. The detailed results are displayed in Table S6. The ROC curves of different classifiers are shown in Fig. 6.

As shown in Fig. 6, the characteristic subsets of the six datasets are placed into different classifiers, and the AUC values of XGBoost are 0.8258, 0.7592, 0.7853, 0.8345, 0.8979, and 0.8901, all of which are higher than other classifiers. For the Set1, Set3-Set6 datasets, the worst performance is DT, whose AUC values are 0.6041, 0.5551, 0.5718 and 0.5561, respectively, which is lower than the AUC indexes of XGBoost (18.12%, 27.94%, 32.61% and 33.4%). Therefore, through performing performance evaluations on six different datasets among nine classification algorithms through five-fold cross-validation, the XGBoost classification algorithm can better classify the ubiquitination sites.

3.7. Comparison with other methods

In order to verify the effect of the prediction model UbiSite-XGBoost and determine whether the model has good robustness, we compare the AUC values of the eight methods including EBMC, NB, FSNB, MANB, SVM, LR, LASSO and DeepUbi [100], the results as shown in Table 5.

As shown in Table 5, the AUC values of the six UbiSite-XGBoost datasets reach 0.8258, 0.7592, 0.7853, 0.8345, 0.8979, and 0.8901, respectively. As for other comparison methods, EBMC, LR, and LASSO methods are used to obtain the highest AUC value of Set1. NB and FSNB are more accurate in predicting Set4 dataset. SVM and MANB can accurately predict Set5 dataset. However, this paper predicted that the evaluation index of UbiSite-XGBoost is clearly higher than the AUC value compared with the paper of Cai et al. [26], and the variation range is between 0.7600 and 0.9000, all reaching over 0.7600, the highest reached 89.79%. Meanwhile, the AUC values of UbiSite-XGBoost are all higher than those of DeepUbi model. This shows that our prediction model have achieved satisfactory results. It has good robustness and

Table 5
Comparison of AUC values of UbiSite-XGBoost with other methods on Set1-Set6.

Methods	Set1	Set2	Set3	Set4	Set5	Set6
UbiSite-XGBoost	**0.8258**	**0.7592**	**0.7853**	**0.8345**	**0.8979**	**0.8901**
EBMC [26]	0.6714	0.6467	0.6667	0.6646	0.6373	0.6001
NB [26]	0.5289	0.5330	0.5141	0.6036	0.5505	0.5134
FSNB [26]	0.5613	0.5582	0.5633	0.6193	0.5637	0.4838
MANB [26]	0.5545	0.5502	0.5192	0.6108	0.5804	0.5690
SVM [26]	0.6597	0.6035	0.6102	0.6670	0.6763	0.5758
LR [26]	0.7244	0.6410	0.6476	0.7200	0.7235	0.5546
LASSO [26]	0.6933	0.6041	0.6129	0.5000	0.5000	0.5000
DeepUbi [100]	0.7738	0.6898	0.7533	0.7576	0.8887	0.6684

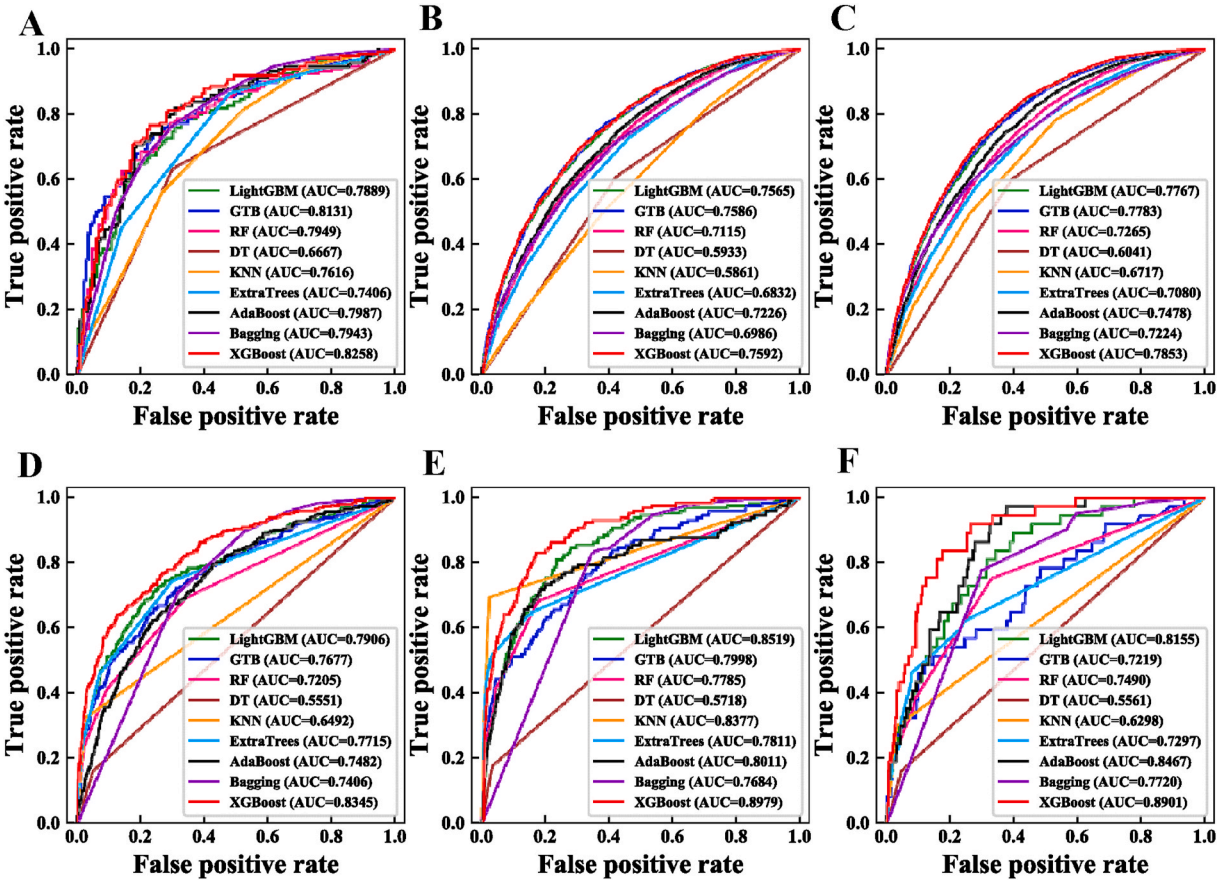

Fig. 6. The ROC curves of LightGBM, GTB, RF, DT, KNN, ExtraTrees, Adaboost Bagging, XGBoost classifiers (A–F) on Set1-Set6.

Table 6
The different prediction results of UbiSite-XGBoost based on Independent test datasets.

Datasets	Resampling method	AUC	ACC (%)	Sn (%)	Sp (%)	MCC
Independent test1	Without any resampling	0.8007	78.09	19.30	96.00	0.2510
	SMOTE	0.9939	94.11	99.27	88.95	0.8871
Independent test2	Without any resampling	0.6897	74.19	19.32	94.53	0.2119
	SMOTE	0.9904	92.94	98.86	87.03	0.8651
Independent test3	Without any resampling	0.7280	87.13	5.053	98.95	0.0756
	SMOTE	0.9812	92.28	97.40	87.16	0.8504

stability for different datasets, which greatly improves the prediction performance of ubiquitination sites.

3.8. The evaluation of ubiquitination sites prediction model on independent testsets

In order to better verify the predictive performance of the model, we built three new independent test datasets. First, the single-view features including PseAAC, CKSAAP, ANBPB, AAindex, EBGW, BLOSUM62, and PsePSSM are transformed into digital information. The optimal parameter is selected of PseAAC, CKSAAP, EBGW, PsePSSM, and the seven methods is fused. Secondly, we employ LASSO to remove redundancy information and select the optimal feature subset. Finally, the XGBoost classifier is used to predict the ubiquitination sites. The specific indicators of the three independent test sets as shown in Table 6.

As can be seen from Table 6, for Independent test1, the ACC, MCC, AUC reach 78.09%, 0.2510 and 0.8007, respectively. For Independent test2, the ACC, MCC, AUC reach 74.19%, 0.2119 and 0.6897, respectively. For Independent test3, the ACC, MCC, AUC values reach 87.13%, 0.0756 and 0.7280, respectively. Also, after we used SMOTE method to deal with the imbalance, different indicators have been greatly improved. The results show that the prediction model constructed in this paper is an effective prediction tool for protein ubiquitination sites.

4. Conclusion

With the development of biomedical mega data, post-translational modification becomes the hotspot of bioinformatics research. Protein ubiquitination can regulate a lot of physiological processes and plays an essential role in the life organisms, but the study of ubiquitination sites still faces great difficulties and challenges. Building predictive models through machine learning can greatly reduce the work of identifying ubiquitination sites in experiments. In this study, a new ubiquitination sites prediction model UbiSite-XGBoost is proposed. Firstly, seven single-view features extraction methods are fused in order to transform protein sequence into digital information. Secondly, LASSO is employed to obtain the optimal feature subsets. By analyzing the contribution rates of single features after LASSO feature selection, CKSAAP has the greatest impact on the model performance, while BLOSUM62 and AAindex also have a large impact on the model performance. The importance of multi-view features for the prediction of ubiquitination sites is fully explained. The LASSO algorithm is able to generate sparsity to compress the variable space, remove redundant information, and preserve the best subset of features for the classifier through the ℓ_1 regularization parameter. To address the serious imbalance of Set4, Set5 and Set6 datasets, the SMOTE algorithm uses most negative samples without prejudice, thereby avoiding over-fitting caused by ignoring a few samples, and further improving the robustness of the prediction results. Finally, we first proposed XGBoost classifier to predict ubiquitination sites. XGBoost takes a second order Taylor expansion of the loss function and introduces a second derivative to improve the accuracy of the model. At the same time, it can also customize the loss function and use ensemble learning to support parallel operation, so it not only reduces overfitting, but also has good model prediction and generalization ability. At last, the AUC values reached 0.8258, 0.7592, 0.7853, 0.8345, 0.8979, and 0.8901, respectively. All of the six datasets achieved superior predictive performance. In addition, we have constructed three independent test datasets which the AUC values are 0.8007, 0.6897 and 0.7280, respectively. Thus, the model Ubisite-XGBoost can effectively predict ubiquitination sites. Although our model can effectively improve the prediction accuracy of model, there is still certain space for improvement. For the next step, we need to construct a larger dataset of ubiquitination sites and further study the ubiquitination sites using deep learning methods to predict the different sites.

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.

Acknowledgments

We thank anonymous reviewers for valuable suggestions and comments. This work was supported by the National Natural Science Foundation of China (No. 61863010), the Key Research and Development Program of Shandong Province of China (No. 2019GGX101001), and the Key Laboratory Open Foundation of Hainan Province (No. JSKX202001), and Science and Technology Project of Colleges in Shandong Province (No. J17KB122).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jmgm.2021.107962.

References

[1] G. Grotenbreg, H. Ploegh, Dressed-up proteins, Nature 446 (2007) 993–995.
[2] R. Geiss-Friedlander, F. Melchior, Concepts in sumoylation: a decade on, Nat. Rev. Mol. Cell Biol. 8 (2007) 947–956.
[3] T.M. Filtz, W.K. Vogel, M. Leid, Regulation of transcription factor activity by interconnected post-translational modifications, Trends Pharmacol. Sci. 35 (2014) 76–85.
[4] L. Prabhu, A. V Hartley, M. Martin, F. Warsame, E. Sun, T. Lu, Role of post-translational modification of the Y box binding protein 1 in human cancers, Genes Dis 2 (2015) 240–246.
[5] D. Hoeller, C.M. Hecker, I. Dikic, Ubiquitin and ubiquitin-like proteins in cancer pathogenesis, Nat. Rev. Canc. 6 (2006) 776–788.
[6] L. Hicke, Protein regulation by monoubiquitin, Nat. Rev. Mol. Cell Biol. 2 (2001) 195–201.
[7] C.M. Pickart, Ubiquitin enters the new millennium, Mol. Cell. 8 (2001) 499–504.
[8] C.M. Pickart, Mechanisms underlying ubiquitination, Annu. Rev. Biochem. 70 (2001) 503–533.
[9] J.D. Schnell, L. Hicke, Non-traditional functions of ubiquitin and ubiquitin-binding proteins, J. Biol. Chem. 278 (2003) 35857–35860.
[10] D. Mukhopadhyay, H. Riezman, Proteasome-independent functions of ubiquitin in endocytosis and signaling, Science 315 (2007) 201–205.
[11] B.T. Dye, B.A. Schulman, Structural mechanisms underlying post-translational modification by ubiquitin-like proteins, Annu. Rev. Biophys. Biomol. Struct. 36 (2007) 131–150.
[12] Y. Ye, M. Rape, Building ubiquitin chains: E2 enzymes at work, Nat. Rev. Mol. Cell Biol. 10 (2009) 755–764.
[13] M. Neutzner, A. Neutzner, Enzymes of ubiquitination and deubiquitination, Essays Biochem. 52 (2012) 37–50.
[14] N. Amanda, S.B. Mcmahon, Rise of the rival, Science 327 (2010) 964–965.

[15] B. Yu, Z. Yu, C. Chen, A. Ma, B. Liu, B. Tian, Q. Ma, DNNAce: prediction of prokaryote lysine acetylation sites through deep neural networks with multi-information fusion, Chemomet. Intell. Lab. 200 (2020) 103999.
[16] D.S. Kirkpatrick, C. Denison, S.P. Gygi, Weighing in on ubiquitin: the expanding role of mass-spectrometry-based proteomics, Nat. Rev. Mol. Cell Biol. 7 (2005) 750–757.
[17] J. Peng, D. Schwartz, J.E. Elias, C.C. Thoreen, D. Cheng, G. Marsischky, J. Roelofs, D. Finley, S.P. Gygi, A proteomics approach to understanding protein ubiquitination, Nat. Biotechnol. 21 (2003) 921–926.
[18] S.A. Wagner, P. Beli, B.T. Weinert, M.L. Nielsen, J. Cox, M. Mann, C. Choudhary, A proteome-wide, quantitative survey of in vivo ubiquitylation sites reveals widespread regulatory roles, Mol. Cell. Proteomics 10 (2011), 013284.
[19] G. Xu, J.S. Paige, S. R Jaffrey, Global analysis of lysine ubiquitination by ubiquitin remnant immunoaffinity profiling, Nat. Biotechnol. 28 (2010) 868–873.
[20] W. Kim, E.J. Bennett, E.L. Huttlin, A. Guo, J. Li, A. Possemato, M.E. Sowa, R. Rad, J. Rush, M.J. Comb, J.W. Harper, S.P. Gygi, Systematic and quantitative assessment of the ubiquitin-modified proteome, Mol. Cell. 44 (2011) 325–340.
[21] P. Radivojac, V. Vacic, C. Haynes, R.R. Cocklin, A. Mohan, J.W. Heyen, M. G. Goebl, L.M. Iakoucheva, Identification, analysis, and prediction of protein ubiquitination sites, Proteins 78 (2010) 365–380.
[22] C.W. Tung, S.Y. Ho, Computational identification of ubiquitylation sites from protein sequences, BMC Bioinf. 9 (2008) 310.
[23] Z. Chen, Y. Zhou, J. Song, Z. Zhang, hCKSAAP_UbSite: improved prediction of human ubiquitination sites by exploiting amino acid pattern and properties, Proteomics 1834 (2013) 1461–1467.
[24] W.R. Qiu, X. Xiao, W.Z. Lin, K.C. Chou, iUbiq-Lys: prediction of lysine ubiquitination sites in proteins by extracting sequence evolution information via a gray system model, J. Biomol. Struct. Dyn. 33 (2015) 1731–1742.
[25] J.R. Wang, W.L. Huang, M.J. Tsai, K.T. Hsu, H.L. Huang, S.Y. Ho, ESA-UbiSite: accurate prediction of human ubiquitination sites by identifying a set of effective negatives, Bioinformatics 33 (2017) 661–668.
[26] B. Cai, X. Jiang, Computational methods for ubiquitination site prediction using physicochemical properties of protein sequences, BMC Bioinf. 17 (2016) 116.
[27] V.N. Nguyen, K.Y. Huang, C.H. Huang, K.R. Lai, T.Y. Lee, A new scheme to characterize and identify protein ubiquitination sites, BMC Bioinf. 14 (2017) 393–403.
[28] Y. Cai, T. Huang, L. Hu, X. Shi, Y. Li, Prediction of lysine ubiquitination with mRMR feature selection and analysis, Amino Acids 42 (2012) 1387–1395.
[29] X. Cui, Z. Yu, B. Yu, M. Wang, B. Tian, Q. Ma, UbiSitePred: a novel method for improving the accuracy of ubiquitination sites prediction by using LASSO to select the optimal Chou's pseudo components, Chemomet. Intell. Lab. 184 (2019) 28–43.
[30] F. He, R. Wang, J. Li, L. Bao, D. Xu, X. Zhao, Large-scale prediction of protein ubiquitination sites using a multimodal deep architecture, BMC Syst. Biol. 12 (2018) 109.
[31] M. Wang, X. Cui, B. Yu, C. Chen, Q. Ma, H. Zhou, SulSite-GTB: Identification of protein S-sulfenylation sites by fusing multiple feature information and gradient tree boosting, Neural Comput, Appl 32 (2020) 13843–13862.
[32] Z. Chen, Y.Z. Chen, X.F. Wang, C. Wang, R.X. Yan, Z. Zhang, Prediction of ubiquitination sites by using the composition of k-spaced amino acid pairs, PloS One 6 (2016), e22930.
[33] H. Xu, J. Zhou, S. Lin, W. Deng, Y. Zhang, Y. Xue, PLMD: an updated data resource of protein lysine modifications, J. Genet Genomics. 44 (2017) 243–250.
[34] W. Li, A. Godzik, Cd-hit, A fast program for clustering and comparing large sets of protein or nucleotide sequences, Bioinformatics 22 (2006) 1658.
[35] H. Xie, L. Fu, X. Nie, Using ensemble SVM to identify human GPCRs N-linked glycosylation sites based on the general form of Chou's PseAAC, Protein, Eng. Des. Sel. 26 (2013) 735–742.
[36] K.C. Chou, Prediction of protein cellular attributes using pseudo amino acid composition, Proteins 43 (2001) 246–255.
[37] F. Li, Q. Li, Predicting protein subcellular location using Chou's pseudo amino acid composition and improved hybrid approach, Protein, Pep. Lett. 15 (2008) 612–616.
[38] M. Wang, J. Yang, G. Liu, Z. Xu, K. Chou, Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition, Protein Eng. Des. Sel. 17 (2004) 509–516.
[39] Y. Chen, K. Li, Predicting membrane protein types by incorporating protein topology, domains, signal peptides, and physicochemical properties into the general form of Chou's pseudo amino acid composition, J. Theor. Biol. 318 (2013) 1–12.
[40] M. Hayat, A. Khan, Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition, J. Theor. Biol. 271 (2011) 10–17.
[41] Q. Zhang, S. Li, Q. Zhang, Y. Zhang, Y. Han, R. Chen, B. Yu, MpsLDA-ProSVM: predicting multi-label protein subcellular localization by wMLDAe dimensionality reduction and ProSVM classifier, Chemometr. Intell. Lab. Syst. 208 (2021) 104216.
[42] B. Yu, S. Li, C. Chen, J. Xu, W. Qiu, X. Wu, R. Chen, Prediction subcellular localization of Gram-negative bacterial proteins by support vector machine using wavelet denoising and Chou's pseudo amino acid composition, Chemomet. Intell. Lab. 167 (2017) 102–112.
[43] B. Tian, X. Wu, C. Chen, W. Qiu, Q. Ma, B. Yu, Predicting protein-protein interactions by fusing various Chou's pseudo components and using wavelet denoising approach, J. Theor. Biol. 462 (2019) 329–346.
[44] B. Yu, L. Lou, S. Li, Y. Zhang, W. Qiu, X. Wu, M. Wang, B. Tian, Prediction of protein structural class for low-similarity sequences using Chou's pseudo amino acid composition and wavelet denoising, J. Mol. Graph. Model. 76 (2017) 260–273.
[45] X. Zhao, W. Zhang, X. Xin, Z. Ma, M. Yin, Prediction of protein phosphorylation sites by using the composition of k-spaced amino acid pairs, PloS One 7 (2012), e46302.
[46] Z. Ju, J. Cao, Prediction of protein N-formylation using the composition of k spaced amino acid pairs, Anal. Biochem. 534 (2017) 40–45.
[47] M.P. Mosharaf, M.M. Hassan, F.F. Ahmed, M.S. Khatun, M.A. Moni, M.N. H. Mollah, Computational prediction of protein ubiquitination sites mapping on arabidopsis thaliana, Comput. Biol. Chem. 85 (2020) 107238.
[48] C. Jia, W. He, Y. Yao, OH-PRED: prediction of protein hydroxylation sites by incorporating adapted normal distribution bi-profile Bayes feature extraction and physicochemical properties of amino acids, J. Biomol. Struct. Dyn. 35 (2016) 829–835.
[49] J. Shao, D. Xu, S. Tsai, Y. Wang, S. Ngai, Computational identification of protein methylation sites through bi-profile bayes feature extraction, PloS One 4 (2009), e4920.
[50] S. Kawashima, M. Kanehisa, AAindex: amino acid index database, Nucleic Acids Res. 28 (2000), 374-374.
[51] M.M. Hasan, H. Kurata, GPSuc: global prediction of generic and species-specific succinylation sites by aggregating multiple sequence features, PloS One 13 (2018), e0200283.
[52] M.M. Hasan, D.J. Guo, H. Kurata, Computational identification of protein S-sulfenylation sites by incorporating the multiple sequence features information, Mol. Biosyst. 13 (2017) 2545–2550.
[53] Q. Xiang, F. Kaiyan, L. Bo, Y. Liu, G. Huang, Prediction of lysine malonylation sites based on pseudo amino acid, Comb. Chem. High Throughput Screen. 20 (2017) 622–628.
[54] Z. Zhang, Z. Wang, Z. Zhang, Y. Wang, A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine, FEBS Lett. 580 (2006) 6169–6174.
[55] D.T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, J. Mol. Biol. 292 (1999) 195–202.
[56] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.
[57] Q. Zhang, S. Li, B. Yu, Q.M. Zhang, Y. Han, Y. Zhang, Q. Ma, DMLDA-LocLIFT, Identification of multi-label protein subcellular localization using DMLDA dimensionality reduction and LIFT classifier, Chemomet. Intell. Lab. 206 (2020) 104148.
[58] Q. Zhang, Y. Zhang, S. Li, Y. Han, S. Jin, H. Gu, B. Yu, Accurate prediction of multi-label protein subcellular localization through multi-view feature learning with RBRL classifier, Briefings Bioinf. (2021), https://doi.org/10.1093/bib/bbab012.
[59] Q. Zhang, P. Liu, Y. Han, Y. Zhang, X. Wang, B. Yu, StackPDB: predicting DNA-binding proteins based on XGB-RFE feature optimization and stacking ensemble classifier, Appl. Soft Comput. 99 (2021) 106921.
[60] H. Shen, K. Chou, Nuc-PLoc: a new web-server for predicting protein subnuclear localization by fusing PseAA composition and PsePSSM, Protein, Eng. Des. Sel. 20 (2007) 561–567.
[61] R. Tibshirani, Regression shrinkage and selection via the LASSO, J. Roy. Stat. Soc. B 58 (1996) 267–288.
[62] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[63] Y. Liu, Z. Yu, C. Cheng, Y. Han, B. Yu, Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net, Anal. Biochem. 609 (2020) 113903.
[64] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[65] H. Zhou, C. Chen, M. Wang, Q. Ma, B. Yu, Predicting Golgi-resident protein types using conditional covariance minimization with XGBoost based on multiple features fusion, IEEE Access 7 (2019) 144154–144164.
[66] B. Yu, W. Qiu, C. Chen, A.J. Ma, J. Jiang, H. Zhou, Q. Ma, SubMito-XGBoost: predicting protein submitochondrial localization by fusing multiple feature information and eXtreme gradient boosting, Bioinformatics 36 (2020) 1074–1081.
[67] C. Chen, Q. Zhang, B. Yu, Z. Yu, Q. Ma, Y. Zhang, Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier, J. Comput. Biol. Med. 123 (2020) 103899.
[68] M.M. Hasan, S. Basith, M.S. Khatun, G. Lee, B. Manavalan, H. Kurata, Meta-i6mA: an interspecies predictor for identifying DNA N6-methyladenine sites of plant genomes by exploiting informative features in an integrative machine-learning framework, Briefings Bioinf. (2020) bbaa202.
[69] J. Yu, S. Shi, F. Zhang, G. D Chen, M. Cao, PredGly: predicting lysine glycation sites for homo sapiens based on XGboost feature optimization, Bioinformatics 35 (2019) 2749–2756.
[70] K. Chou, Prediction of protein signal sequences and their cleavage sites, Proteins 42 (2001) 136–139.
[71] W. Chen, P. Feng, H. Lin, K. Chou, iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition, Nucleic Acids Res. 41 (2013) e68.
[72] W. Qiu, S. Li, X. Cui, Z. Yu, M. Wang, J. Du, Y. Peng, B. Yu, Predicting protein submitochondrial locations by incorporating the pseudo-position specific scoring matrix into the general Chou's pseudo-amino acid composition, J. Theor. Biol. 45 (2018) 86–103.

[73] Y. Huo, L. Xin, C. Kang, M. Wang, Q. Ma, B. Yu, SGL-SVM: a novel method for tumor classification via support vector machine with sparse group Lasso, J. Theor. Biol. 486 (2019) 110098.
[74] X. Wang, B. Yu, A.J. Ma, C. Chen, B. Liu, Q. Ma, Protein-protein interaction sites prediction by data using relaxed Lasso with synthetic minority oversampling technique, Bioinformatics 35 (2019) 2395–2402.
[75] V. Vacic, L.M. Iakoucheva, P. Radivojac, Two Sample Logo: a graphical representation of the differences between two sets of sequence alignments, Bioinformatics 22 (2006) 1536–1537.
[76] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E 69 (2004), 066138.
[77] B.C. Ross, Mutual information between discrete and continuous data sets, PloS One 9 (2014), e87357.
[78] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. Roy. Stat. Soc. B 67 (2005) 301–320.
[79] B. Yu, C. Chen, X. Wang, Z. Yu, A. Ma, B. Liu, Prediction of protein-protein interactions based on elastic net and deep forest, Expert Syst. Appl. 176 (2021) 114876.
[80] M. Wang, L. Yue, X. Cu, C. Chen, H. Zhou, Q. Ma, B. Yu, Prediction of extracellular matrix proteins by fusing multiple feature information, Elastic Net, and Random Forest algorithm, Mathematics 8 (2020) 169.
[81] D. Ichikawa, T. Saito, W. Ujita, H. Oyama, How can machine-learning methods assist in virtual screening for hyperuricemia? A healthcare machine-learning approach, J. Biomed. Inf. 64 (2016) 20–24.
[82] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (2006) 3–42.
[83] C. Kang, Y. Huo, L. Xin, B. Tian, B. Yu, Feature selection and tumor classification for microarray data using relaxed Lasso and generalized multi-class support vector machine, J. Theor. Biol. 463 (2019) 77–91.
[84] H. Shi, S. Liu, J. Chen, X. Li, Q. Ma, Predicting drug-target interactions using lasso with random forest based on evolutionary information and chemical structure, Genomics 111 (2019) 1839–1852.
[85] H. Ai, R. Wu, L. Zhang, X. Wu, J. Ma, H. Hu, L. Huang, W. Chen, J. Zhao, H. Liu, pSuc-PseRat: predicting lysine succinylation in proteins by exploiting the ratios of sequence coupling and properties, J. Comput. Biol. 24 (2017) 1050–1059.
[86] B. Yu, C. Chen, H. Zhou, B. Liu, Q. Ma, Prediction of protein-protein interactions based on L1-regularized logistic regression and gradient tree boosting, Dev. Reprod. Biol. (2021), https://doi.org/10.1016/j.gpb.2021.01.001.
[87] J. Jia, Z. Liu, X. Xiao, B. Liu, K. Chou, pSuc-Lys: predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach, J. Theor. Biol. 394 (2016) 223–230.
[88] L. Breiman, Random forest, Mach. Learn. 45 (2001) 5–32.
[89] X. Sun, T. Jin, C. Chen, X. Cui, Q. Ma, B. Yu, -R.F. RBPro, Use chou's 5-steps rule to predict RNA-binding proteins via random forest with elastic net, Chemometr. Intell. Lab. Syst. 197 (2020) 103919.
[90] G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer, KNN model-based approach in classification, Lect. Notes Comput. Sci. 2888 (2003) 986–996.
[91] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theor. 13 (2002) 21–27.
[92] A. Li, L. Wang, Y. Shi, M. Wang, Z. Jiang, H. Feng, Phosphorylation site prediction with a modified k-nearest neighbor algorithm and BLOSUM62 matrix, IEEE Engineering in Medicine and Biology 27th Annual Conference 6 (2005) 6075–6078.
[93] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, in: 31st Conference Neural Information Processing Systems NIPS, Curran Associates Inc, 57 Morehouse Lane, Red Hook, NY, United States, 2017, pp. 3146–3154.
[94] C. Chen, Q. Zhang, Q. Ma, B. Yu, LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion, Chemometr. Intell. Lab. 191 (2019) 54–64.
[95] L. Yue, M. Wang, X. Yang, Y. Han, L. Song, B. Yu, Fertility-LightGBM: a fertility-related protein prediction model by multi information fusion and light gradient boosting machine, Biomed. Signal Process Contr. (2020), https://doi.org/10.1101/2020.08.24.264325.
[96] H.D. Ismail, R.H. Newman, D.B. Kc, RF-Hydroxysite: a random forest based predictor for hydroxylation sites, Mol. Biosyst. 12 (2016) 2427–2435.
[97] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to Boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.
[98] L. Breiman, Bagging predictors, J. Mach. Learn. 24 (1996) 123–140.
[99] Q. Zhang, Y. Zhang, S. Li, Y. Han, S. Jin, H. Gu, B. Yu, Accurate prediction of multilabelprotein subcellular localization through multi-view feature learning with RBRL classifier, Briefings Bioinf. (2021), https://doi.org/10.1093/bib/bbab012.
[100] H. Fu, Y. Yang, X. Wang, H. Wang, Y. Xu, DeepUbi: a deep learning framework for prediction of ubiquitination sites in proteins, BMC Bioinf. 20 (2019) 86.