Prediction of Plant Ubiquitylation Proteins and Sites by Fusing Multiple Features

Meng-Yue Guan
Jingdezhen Ceramic Institute
Qian-Kun Wang
Jingdezhen Ceramic Institute
Peng Wu
Jingdezhen Ceramic Institute
Wang-Ren Qiu
qiuone@163.com

Jingdezhen Ceramic Institute
Wang-Ke Yu
Jingdezhen Ceramic Institute
Xuan Xiao
Jingdezhen Ceramic Institute

Research Article

Keywords: LGBM, Deep Learning, Multiple Features, Post-Translational Modification, Ubiquitylation.

Posted Date: September 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2032518/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Current Bioinformatics on November 20th, 2023. See the published version at https://doi.org/10.2174/1574893618666230908092847.

Abstract

Protein ubiquitylation is an important post-translational modification (PTM), which is considered to be one of the most important processes regulating cell function and various diseases. Therefore, accurate prediction of ubiquitylation proteins and their PTM sites is of great significance for the study of basic biological processes and the development of related drugs. Researchers have developed some large-scale computational methods to predict ubiquitylation sites, but there is still much room for improvement. Much of the research related to ubiquitylation is cross-species while the life pattern is diversified, and the prediction method always shows its specificity in practical application. This study just aims to the issue of plants, and has constructed computational methods for identifying ubiquitylation protein and ubiquitylation sites. To better reflect the protein sequence information and obtain better prediction, the KNN scoring matrix model based on functional domain GO annotation and word embedding model (CBOW and Skip-Gram) are used to extract the features, and the light gradient boosting machine (LGBM) is selected as the ubiquitylation proteins prediction engine. As results, accuracy (ACC), precision (precision), recall (recall), F1_score and AUC are respectively 85.12%, 80.96%, 72.80%, 0.7637 and 0.9193 in the 10-fold cross-validations on independent data set. In the ubiquitylation sites prediction model, Skip-Gram, CBOW and EAAC feature extraction codes were used to extract protein sequence fragment features, and the predicted results on training and independent test data have also achieved good performance. In a word, the comparison results demonstrate that our models have a decided advantage in predicting ubiquitylation proteins and sites, and it may provide useful insights for studying the mechanisms and modulation of ubiquitination pathways. The datasets and source codes used in this study are available at: https://github.com/gmywqk/Ub-PS-Fuse.

1 Introduction

As one of the most important post-translational modifications (PTMs), ubiquitylation is widely involved in some key processes of life activities[4, 5]. Ubiquitylation has important regulatory functions and plays an important role in inflammation, cell division, signal transduction, hypersensitivity, proteasome degradation, down regulation, transcription and deoxyribonucleic acid repair[3, 6–11]. It is also related to a variety of diseases, such as periodontal disease[12]], cancer[13], Alzheimer's disease[14], liver disease[15], and so on. More and more evidences show that ubiquitylation is almost involved in the whole life cycle from germination to flowering of plant seeds, aging and pathogen response[16, 17]. Therefore, we are facing a challenging and interesting problem for exploring new ubiquitylation proteins and their modification sites to gain new insights into their mechanisms and functions. There are two main reasons for its difficult to study this issue: Ubiquitylation is second only to glycosylation[18], which has the most complex modification due to its different target binding modes; Similar to the phosphorylation[19] pathway, the ubiquitylation modification pathway is reversible, that is, the modification of ubiquitylation protein can be removed by deubiquitinase. Over the past few decades, due to technological advances in ubiquitylation protein (or peptide) affinity analysis, mass spectrometry, ubiquitin antibodies[20] and ubiquitin-binding proteins[20, 21], a wide range of analysis of ubiquitylation proteins. However, these large-

scale experimental screening techniques for identifying ubiquitylation sites are time-consuming, expensive, and laborious. In contrast, machine learning-based computational methods provide another alternative strategy for predicting ubiquitylation sites in a low-cost and efficient manner.

Regarding the identification of ubiquitylation proteins, Qiu[21] firstly proposed a computational method by using evolutionary profiles and functional domain annotation to predict ubiquitylation proteins in multiple species in 2019, and achieved good results. Moreover, Qiu also identified phosphorylation[22], acetylation[23], S-nitrosylation[24] and pupylation[25] in proteins with KNN scoring matrix based on GO annotation information. For predictive analysis of ubiquitylation sites, although many models have been proposed to identify ubiquitylation sites in different species, only a few have been designed for plants. To predict Arabidopsis ubiquitylation sites, Chen[26] designed an online support vector machine-based predictor, named "AraUbiSite", combined with K-spaced amino acid pairs and amino acid composition (AAC) feature extraction methods. Mosharaf[27] proposed a random forests model by using a feature extraction method encoded with binary amino acid. Mosharaf[28] used a random forest model by combining the feature extraction methods of K-spaced amino acid pairs and binary amino acid encoding. In recent years, deep learning has been widely used in the field of bioinformatics. Wang[29] firstly used word embedding to extract features of plant ubiquitylation sites in 2020, and then combined with a multi-layer convolutional neural network model to predict them. Siraj[30] developed a deep learning-based predictor UbiComb in 2021, which uses two modules to extract features, namely: an embedding module and physicochemical properties module. Recently, Yin[31] proposed a prediction model UPFPSR based on random forest classifier, which was developed using multiple physicochemical properties of amino acids and sequence-based statistics. The aforementioned predictive models are helpful to scientists, however they also have certain limitations, such as training on small datasets, using shallow machine learning models, and using limited deep neural networks. Therefore, there is still a lot of room for improving prediction performance.

In this study, we develop a prediction framwork, named as Ub-PS-Fuse and shown in Fig. 1, for exploring ubiquitylation proteins and their modification sites of plants. In the prediction model of ubiquitinated proteins, the positive samples were collected from the PLMD[32] database, and the negative samples were collected from the UniProKB[33] database. The fundamental step in building a predictive model is feature extraction, and the KNN scoring matrix[23] and word embeddings[34, 35] (Skip-Gram and CBOW) models based on functional domain GO annotations are used in this work. The 10-fold cross-validation is applied to evaluate and enhance the performance of classifier.

In the ubiquitylation sites prediction model, the dataset was constructed by Wang[29]. Sequence information is encoded into numerical feature vectors by using multiple feature encoding schemes of EAAC, Skip-Gram[35] and CBOW[34], LGBM was selected as the classifier and validated by 10-fold cross-validation. Comparison results show that our proposed model is better than other existing predictors in the terms of performance and stability.

2. Materials

2.1 Datasets for Prediction Ubiquitylation proteins

To obtain scientifically rigorous prediction results, a rigorous benchmark dataset is necessary. In this study, the positive samples were collected from the Protein Lysine Modification Database (PLMD)[32], with a total of 2139 ubiquitylation proteins. A total of 117,065 negative samples were collected from the UniProKB[36] database. After 30% de-redundancy of negative samples, 4,278 non-ubiquitylation proteins were randomly selected. In the datasets for predicting ubiquitylation proteins, denoted as Ub-P, a positive sample is a protein with at least one ubiquitination site, and the negative sample dataset is the set of proteins without any ubiquitylation site. A ubiquitylation protein sequence can be represented as:

$$P = R_1 R_2 R_3 \cdots R_i \cdots R_L$$

1

where R_i represents the ith amino acid residue (20 common amino acids and a pseudo-amino acid "X"), and$L$ represents the length of the protein sequence.

2.2 Datasets for Prediction Ubiquitylation sites

This paper used the same training and independent test dataset as Wang[29], which were collected from the PLMD[32] database and experimentally validated lysine ubiquitylation sites. The dataset consists of Oryza sativa subsp indica, O. sativa subsp japonica, and Arabidopsis thaliana, and the dataset includes a total of 2139 ubiquitylation proteins. In the datasets for prediction ubiquitylation sites, denoted as Ub-S, a potential ubiquitylation site-contained sample can be expressed by a fragment with 31 amino acids. If the number of upstream or downstream amino acids is less than 15, it would be supplemented with pseudo-amino acid "X" to ensure that each ubiquitylation protein fragment has the same amino acids. After a series of processing, a total of 7000 protein sequence fragments were obtained, including 3500 positive samples (ubiquitylation sites) and 3500 negative samples (non-ubiquitylation sites). Among these positive samples and negative samples, 2750 positive samples and 2750 negative samples were randomly selected to form the training set, and the remaining 750 positive samples and 750 negative samples were used to form an independent test set. Table 1 summarizes the datasets for predicting ubiquitylation proteins and ubiquitylation sites.

Table 1
Datasets of ubiquitylation proteins and ubiquitylation sites for this work.

Datasets	Positive	Negative	Total	Ratio
Ub-P Training	1711	3422	5133	1:2
Ub-P Testing	428	856	1284	1:2
Ub-S Training	2750	2750	2500	1:1
Ub-S Testing	750	750	1500	1:1

For a more detailed and comprehensive formulation of ubiquitylation site sequences, the fragments of potential ubiquitylation sites can be expressed with Eq. (2).

$$\theta_\delta\left(K\right) = R_1 R_2 \cdots R_{\delta-1} R_\delta K R_{\delta+1} R_{\delta+2} \cdots R_{2\delta-1} R_{2\delta}$$

2

Where the center "K" represents "Lysine"[37], R_δ represents the δ-th amino acid residue of the upstream, and $\mathrm{R}_{\delta+1}$ represents the δ-th amino acid residue of the downstream, δ is an integer, and so forth. In addition, the peptide sequence $\theta_\delta\left(\mathrm{K}\right)$can be divided into two types: $\theta_\delta^+\left(\mathrm{K}\right)$ and $\theta_\delta^-\left(\mathrm{K}\right)$, where $\theta_\delta^+\left(\mathrm{K}\right)$ represents a true ubiquitylation segment with "K" at the center point, $\theta_\delta^-\left(\mathrm{K}\right)$ indicates a non-ubiquitylation segment with "*K*" at its center. The sliding window method was used to segment ubiquitylation protein sequences with different window sizes. According to the analysis of the preference of ubiquitylation protein sequences by Wang[29], it can be seen that when the window size is 31 (i.e. $\delta = 15$), the best prediction is achieved.

In order to equalize the site sequence lengths, the missing amino acids are filled with "X" residues when the sequence fragments are divided. The ubiquitylation site dataset takes the form of Eq. (3).

$$D_\delta\left(K\right) = D_\delta^+\left(K\right) \cup \mathrm{D}_\delta^-\left(K\right)$$

3

Among them, the subset of positive samples $D_\delta^+\left(K\right)$ represents the ubiquitylation site fragment samples centered on "K", and the subset of negative samples $D_\delta^-\left(K\right)$ represents the non-ubiquitylation site fragment samples centered on "K". $D_\delta\left(K\right)$ represents the benchmark dataset.

3 Feature Extraction And Methods

3.1 GO-KNN

GO-KNN[23] extracts features based on a KNN score annotated by functional domain GO. In this study, we need to find out the GO information of all proteins, if the GO information cannot be found for a given protein, it would be replaced with its homologous protein GO information. We need to calculate the distance between each protein sequence and other protein sequences. Take proteins P^1 and P^2 as example, their GO annotations can be expressed as $P_{GO}^1 = \left\{GO_1^1, GO_2^1, \cdots, GO_m^1\right\}$ and $P_{GO}^2 = \{GO_1^2, GO_2^2, \cdots, GO_n^2\}$. Among them, GO_i^1 and GO_i^2 represent the *i*-th GO of proteinsP^1 and P^2, respectively, and m and n represent the numbers of GO, respectively. The extraction steps are as follows:

Step 1: Calculating the distance between the two proteins, such as formula (4). Where ∪ and ∩ represent the intersection and union of sets, respectively, and ⌊ ⌋ represents the number of elements in the set.

$$Distance\left(R_1, R_2\right) = 1 - \frac{\left\lfloor P_{GO}^1 \cap P_{GO}^2 \right\rfloor}{\left\lfloor P_{GO}^1 \cup P_{GO}^2 \right\rfloor}$$

4

Step 2: Sorting all calculated distances from small to large.

Step 3: Selecting the k nearest neighbors of the protein sequence and calculating the percentage of positive samples of the k nearest neighbors to the whole samples.

In this study, the selection of the k value is based on the number of samples. For example, the number of positive and negative samples in the ubiquitylation protein training set in this study is 5133, so we choose the k value as 2, 4, and 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096. Finally, a 12-dimensional feature vector is formed, so the digital feature vector of protein R_1 can be expressed as: $\left(x_1, x_2, \cdots, x_{12}\right)$. Since the total number of positive and negative samples in the independent test set of ubiquitylation proteins in this study is 1284, a 10-dimensional feature vector is formed.

3.2 Word Embedding

Word Embedding[34, 35, 38] is a method of converting words in text into numeric vectors. The word embedding process is to embed a high-dimensional space containing the number of all words into a low-dimensional continuous vector space, each word or phrase is mapped to a vector on the real number domain, and the result of word embedding generates a word vector. Here is a brief description of how word embedding was applied in this study:

Step 1: The ubiquitylation protein sequence is firstly split into fragments and a word book is created. In this study we used five different word embedding models, that is, cutting ubiquitylation protein sequences into different fragment lengths, which can be set to 1, 2, 3, 4, and 5, respectively, moving the window with a step size of 1 and removing duplicates, which generates the five wordbooks with a number of *v*. Let's

take the splitting fragment length of 3 as an example, and the detailed splitting process is shown in Fig. 2.

Step 2: Use the Skip-Gram and CBOW models to train the data separately. There are two mainstream word embedding methods today, namely word2vec and Glove. The word2vec algorithm has two training modes: predict current word by context (CBOW) and predict context by current word (Skip-Gram). However, the training speed of Skip-Gram is slower than that of the CBOW model, because the Skip-Gram model does not ignore low-frequency words, but the accuracy of Skip-Gram is generally higher than that of the CBOW model (this is what Mikolov[34] said). In order to speed up the training speed of word vectors, this research adopts the negative sampling technique and the back propagation algorithm. The dimension size of the word vectors we choose for each wordbook is 500 dimensions. The specific training process for CBOW models and Skip-gram models is shown in Fig. 3.

Step 3: Extract features using Skip-Gram and CBOW models. By training the Skip-Gram and CBOW models respectively, we obtained five word-vector matrices M respectively. Then, we combined the features of each ubiquitylation protein sequence of the five word-vector matrices, as shown in formula (5), and finally obtained a 2500-dimensional Skip-Gram and CBOW feature vector respectively.

$$W = \mathrm{M}(1)\mathrm{M}(2)\mathrm{M}(3)\mathrm{M}(4)\mathrm{M}(5)$$

5

Where M(1) represents a 500-dimensional word vector with a word segment length of 1, M(2) represents a 500-dimensional word vector with a word segment length of 2, and so on, M(5) represents a 500-dimensional word vector with a word segment length of 5. "" means to merge (vertically) two word-vectors.

3.3 EAAC

EAAC is called "enhanced amino acid composition" and is widely used in bioinformatics research, such as predicting protein malonylation sites[39], lysine glutarylation sites[40]. The EAAC coding process is briefly described as follows: In a fixed-length protein sequence fragment, the frequency of occurrence of 20 amino acids and a pseudo-amino acid "X" is calculated, as shown in formula (6).

$$f(\chi, K) = N(\chi, Q)/N\left(O\right), \chi \in (A, C, D, \ldots, Y, X)$$

6

Among them, $N\left(\chi, Q\right)$ represents the number of amino acids χ when the sliding window size is O, $N\left(Q\right)$ represents the size Q of the sliding window, generally defaulting to 5. Considering that the fixed length of the protein sequence fragments of the ubiquitylation site and the non-ubiquitylation site is 31, a feature vector of (31 − 5 + 1)·20 = 567 dimensions is finally formed.

4. Model Evaluation Metrics And Operation Engine

4.1 Model Evaluation Metrics

Four metrics were used to evaluate the performance of the model when predicting ubiquitylation proteins and their corresponding modification sites. That is, accuracy (ACC), precision (Precision), recall rate (Recall), and F1-score[41–43], respectively, formula (7) is defined in detail.

$$\begin{cases} ACC = \frac{TP+TN}{TP+TN+FP+FN} \\ Precision = \frac{TP}{TP+FP} \\ Recall = \frac{TP}{TP+FN} \\ F1-score = \frac{2\times Precision\times Recall}{Precision+Recall} \end{cases}$$

7

When predicting ubiquitylation proteins, where TP, FP, TN and FN represent the number of true positives (predicted ubiquitylation proteins to be actual ubiquitylation proteins), false positives (predicted ubiquitylation proteins to be actual non-ubiquitylation proteins), true negatives (predicting non-ubiquitylation proteins to be actual non-ubiquitylation proteins) and false negatives (predicting non-ubiquitylation proteins to be actual ubiquitylation proteins), respectively.

When predicting ubiquitylation sites, where TP, FP, TN and FN represent the number of true positives (predicted ubiquitylation sites are actual ubiquitylation sites), false positives (predicted ubiquitylation sites are actual non-ubiquitylation sites), true negatives (predicted non-ubiquitylation sites are actual non-ubiquitylation sites) and false negatives (predicted non-ubiquitylation sites are actual ubiquitylation sites), respectively.

Additionally, we plotted the receiver operating characteristic curve (ROC) and calculated the area under the ROC (AUROC) as the evaluation measure of this work for completeness to further evaluate our model performance. To make the results more convincing and stable, we used the average of ten 10-fold cross-validations to represent the overall performance of the model.

4.2 Operation Engine

4.2.1 LGBM

LGBM (Light Gradient Boosting Machine) is an efficient gradient boosting decision tree using gradient-based one-sided sampling (Goss)[44] and exclusive feature binding (EFB)[45] proposed by Ke. It supports efficient parallel training, and has the advantages of faster training speed, lower memory consumption, better accuracy, and support for distributed and fast processing of massive data. It is an ensemble model of decision trees based on basic classifiers, which can be trained sequentially by fitting the negative

gradient of the loss function. In this study, the selected parameters are: learning_rate = 0.1, n_estimators = 800, max_depth = 4. LightGBM code is available at https://github.com/Microsoft/LightGBM. This algorithm is widely used in classification[46], ranking[47] and many other machine learning[48] tasks.

4.2.2 RF

The Random Forest (RF)[49, 50] algorithm is based on the Classification and Regression Tree (CART)[51] technique. Due to its flexibility and generalization ability, the algorithm has been applied in fields such as bioinformatics[52], data mining, and energy architecture[53]. It is an algorithm formed by integrating multiple decision trees through the idea of ensemble learning. In the random forest, each decision tree is a classifier. For a given sample, each tree will get a classification result, integrate all the voting results, and the final output is the category with the most votes.

4.2.3 SVM

Support Vector Machine (SVM)[54], is a supervised learning model. It has been widely used in various fields such as marketing management[55], bioinformatics[49] and image retrieval[56]. Compared with other machine learning algorithms, the advantage of the SVM algorithm is that the dimension of the SVM parameters is equal to the number of training samples[50]. Its main idea is to find a hyperplane that distinguishes the two classes, maximize the margin, and some points in the sample that are closest to the hyperplane, these points are called support vectors.

4.2.4 DNN

Due to the strong learning ability of deep learning algorithms and as a cutting-edge technology, deep learning is also widely used in the field of bioinformatics to predict protein modification sites[57], RNA modification sites[58], etc. In this study, the deep learning classifier consists of three layers: Input layer. Hidden layer, the hidden layer has four layers, three LeakyRelu activation functions and two batch normalization functions (BatchNormalization, which makes training more stable and speeds up learning). The output layer performs binary classification through the sigmoid activation function. We choose the Adam algorithm as the optimizer and the cross-entropy loss formula as the loss function.

4.2.5 BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM)[59] network is a combination of forward LSTM[60] and backward LSTM, which is commonly used in natural language processing (NLP) tasks to model contextual information and effectively capture contextual information. Bidirectional semantic dependencies. This network has been widely used to predict protein sites[61], identify antimicrobial peptides and their functional types[62], and drug-target interactions[63]. In this study, the BiLSTM classifier consists of five layers: BiLSTM layer, LSTM cell unit is 128. A dropout layer with dropout of 0.4 and 64 hidden nodes A fully connected layer with 32 hidden nodes and ReLU activation. A dropout layer, the parameter of dropout is 0.4 and the number of hidden nodes is 16. A fully connected layer with 1 hidden

node and binary classification via sigmoid activation. We choose the Adam algorithm as the optimizer and the cross-entropy loss formula as the loss function.

5. Results And Discussion

5.1 Results and discussion for Ubiquitylation Proteins prediction

5.1.1 Effect of the Different Features on Training Dataset

In this study, three feature encodings, namely GO-KNN, CBOW and Skip-Gram models, were used. The feature dimensions of the three feature codes are 12 dimensions, 2500 dimensions, and 2500 dimensions, respectively, so the feature set, noted as Ub-P-Fuse, has 5012 dimensions. Test with LGBM classifier training and 10-fold cross-validation, the prediction results of different features are shown in Table 2.

From Table 2, we can see that the prediction results after random combination of three feature codes are much better than the prediction results of single feature code. Among them, when the Skip-Gram and CBOW models are combined with the GO-KNN model in pairs, the effect is better than the combination of their own models, with an average increase of four points, which further verifies the importance of the GO-KNN model. However, when the three feature codes are fused (Ub-P-Fuse), the prediction effect is better than that of pairwise combination coding. Therefore, the results show that the multi-feature fusion Ub-P-Fuse (GO-KNN + CBOW + Skip-Gram) improves the prediction effect.

Table 2
A comparison of different features for predicting ubiquitylation proteins.

Method	ACC(%)	Precision(%)	Recall(%)	F1-score
GO-KNN	82.04	74.60	70.66	0.7254
CBOW	82.21	74.53	70.88	0.7262
Skip-Gram	82.87	76.50	70.32	0.7323
GO-KNN + CBOW	86.79	80.73	79.33	0.7993
GO-KNN + Skip-Gram	86.75	80.95	78.77	0.7982
CBOW + Skip-Gram	83.17	76.24	71.98	0.7400
Ub-P-Fuse	**87.22**	**81.51**	**79.77**	**0.8058**

5.1.2 Effect of the Different Classifiers on Training Dataset

In this work, we used the five classifiers described above to identify ubiquitylation proteins. After 10-fold cross-validation on the training set with Ub-P-Fuse, the evaluation index results of each classifier with a ratio of positive and negative samples of 1:2 are shown in Table 3.

Table 3
A comparison of different classifiers on the training set for predicting ubiquitylation proteins.

Method	ACC(%)	Precision(%)	Recall(%)	F1-score
LGBM	**87.22**	**81.51**	79.77	**0.8058**
RF	85.45	81.24	69.14	0.7465
SVM	86.33	80.34	78.08	0.7915
DNN	83.24	75.49	75.74	0.7483
BiLSTM	85.43	77.61	**80.23**	0.7849

From Table 3, we can see that the BiLSTM classifier has the best performance on the evaluation index of Recall, but from the comprehensive index, the LGBM classifier obtained the best results. In order to better compare the effects of different classifiers, the prediction results of the five classifiers are shown in Fig. 4.

The area under the ROC curve can evaluate the predictive performance of the model. It can be seen from Fig. 4 that LGBM classifier has a higher ROC curve accuracy than others in 10-fold cross-validation. The area under the curve of LGBM, RF, SVM, DNN, BiLSTM is 0.9397, 0.9108, 0.9284, 0.9086 and 0.9286 respectively. Therefore, compared with the other four classifiers, the LGBM classifier is the best choice for the proposed model.

5.1.3 Effect of the Different Classifiers on Independent Dataset

To validate the effect of the Ub-P-Fuse, 428 ubiquitylation and 856 non-ubiquitylation proteins were independently tested in this study, as shown in Table 4. Experimental results show that the Ub-P-Fuse based on the LGBM algorithm outperforms other popular algorithms on all these predefined features.

Table 4
A comparison of different classifiers on the independent test sets for predicting ubiquitylation proteins.

Method	ACC(%)	Precision(%)	Recall(%)	F1-score	AUC
LGBM	**85.12**	**80.96**	72.80	**0.7637**	**0.9193**
RF	83.33	80.89	66.10	0.7237	0.8945
SVM	84.11	79.45	70.60	0.7460	0.9065
DNN	80.06	70.24	70.89	0.6988	0.8716
BiLSTM	82.78	75.02	**78.09**	0.7527	0.9096

5.2 Results and discussion for Ubiquitylation Sites prediction

5.2.1 Effect of the Different Features on Training Dataset

In this study, we have also tried three single feature encodings, Skip-Gram, CBOW and EAAC, and the numbers of features are 2500, 2500, and 567 dimensions, respectively. After several single coding combinations, the model Ub-S-Fuse is finally obtained, which is fused by Skip-Gram and EAAC feature coding, and the dimension is 3057. Through 10-fold cross-validation, we select the LGBM classifier for training, and the prediction results of different feature extraction with a ratio of positive and negative samples of 1:1 are shown in Table 5.

From Table 5, we can see that the indicators of the Skip-Gram model are higher than other single codes, so the importance of this code is self-evident. After the fusion of the two models, the Ub-S-Fuse (Skip-Gram + EAAC) model performs much better than other single feature and fusion models in various indicators. This also reflects that the EAAC coding model also contributed a lot to this model, which probably improved the prediction effect by three points.

Table 5
A comparison of different feature extraction methods on the training set for predicting ubiquitylation sites.

Method	ACC(%)	Precision(%)	Recall(%)	F1-score
Skip-Gram	83.85	83.80	83.31	0.8354
EAAC	78.82	79.36	77.83	0.7858
CBOW	75.11	75.29	74.77	0.7499
Skip-Gram + EAAC	**86.53**	**87.04**	**85.83**	**0.8641**
Skip-Gram + CBOW	83.07	83.65	82.22	0.8291
Skip-Gram + EAAC + CBOW	86.11	86.73	85.17	0.8593

5.2.2 Effect of the Different Classifier on Training Dataset

Choosing the right classifier (machine learning and deep learning) is also a crucial step in predicting the outcome. When predicting ubiquitylation sites, we chose LGBM, RF, SVM, DNN and BiLSTM algorithms. In order to verify the effectiveness and superiority of the LGBM algorithm used to predict ubiquitylation sites, we compared these algorithms through 10-fold cross-validation on the same training set, and the prediction results are shown in Table 6.

Table 6
A comparison of different classifiers on the training set for predicting ubiquitylation sites.

Method	ACC(%)	Precision(%)	Recall(%)	F1-score	AUC
LGBM	**86.53**	**87.04**	**85.83**	**0.8641**	**0.9342**
RF	79.90	81.62	77.18	0.7931	0.8783
SVM	83.18	83.90	82.02	0.8294	0.9049
DNN	81.30	81.15	81.78	0.8134	0.8953
BiLSTM	81.19	81.17	81.86	0.8123	0.8959

As can be seen from Table 6, the prediction effect of the LGBM classifier is much better than other traditional machine learning and deep learning. The classifier performance of LGBM is 4–14% higher than other traditional machine learning metrics, and nearly 4–6% higher than that of deep learning. This further validates that deep learning tends to perform well on large-scale data, and our dataset is only built on plants, which is not large enough. In order to better and more comprehensively evaluate the performance of the classifier, the ROC curves of different classifiers are shown in Fig. 5.

5.2.3 Effect of the Different Classifier on Independent Dataset

To further demonstrate the generalization ability of the Ub-S-Fuse model and LGBM, we also further test the prediction results of different classifiers on an independent test set, as shown in Fig. 6. The results show that the performance of the LGBM classifier is still better than other classifiers in various indicators.

5.2.4 Comparison with other methods on Independent Dataset

To demonstrate the effectiveness of our Ub-S-Fuse model, we performed 10-fold cross-validation on the same independent test set to objectively compare with two existing methods (CNN + word2vec and UPFPSR). The dataset contains 750 positive samples and 750 negative samples, and the specific performance comparison is shown in Table 7. The results show that between these three predictors, Ub-S-Fuse outperforms the first two models on every measure. The ACC, Precision, Recall, F1 score and AUC of Ub-S-Fuse are about 9%-11%, 9%-11%, 7%-12%, 8%-11% and 10%-12% higher than those of the first two models respectively. Taken together, all the results show that the model has high confidence in the prediction of plant ubiquitylation sites and is more suitable for identifying plant ubiquitylation sites.

Table 7
Comparison with other methods on independent test set when predicting ubiquitylation sites.

Method	ACC(%)	Precision(%)	Recall(%)	F1-score	AUC
CNN + word2vec[29]	75.6	73.3	76.7	0.7493	0.82
UPFPSR[31]	77.3	75	81.7	0.7824	0.84
Ub-S-Fuse	**86.13**	**84.42**	**88.70**	**0.8638**	**0.9378**

6. Conclusion

Ub-PS-Fuse was developed for better prediction of ubiquitylation proteins and sites. To predict ubiquitylation proteins, we used three feature extraction methods, GO-KNN, Skip-Gram and CBOW. GO-KNN extracts features based on a KNN scoring matrix annotated by functional domain GO, and Word Embedding is a method to convert words in text into numeric feature vectors. Finally, we selected the LGBM classifier as the prediction engine. We then evaluate the performance of Ub-P-Fuse using an independent test dataset to demonstrate the generalization ability of the Ub-P-Fuse. To predict ubiquitylation sites, three feature extraction codes and one fusion feature extraction code were used, namely Word Embedding (Skip-Gram and CBOW), EAAC and Ub-S-Fuse. Validation was performed by using 10-fold cross-over, and the extracted feature vectors were input into the LGBM classifier for classification. The performance of Ub-S-Fuse is evaluated on an independent test dataset and compared with other existing methods, and it is concluded that the prediction performance of Ub-S-Fuse is better

than other existing methods. The above process can be summarized as shown in Fig. 1. These processes only require computational models and do not require any physical and chemical experiments, which saves experimental costs and improves work efficiency. We hope this work contributes to computationally solving biological problems.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

Competing interests

The authors have declared that no competing interest exists.

Funding

This work was supported by grants from the National Natural Science Foundation of China (No. 62162032, 31760315), and the Natural Science Foundation of Jiangxi Province, China (No. 20202BAB202007). Key Program for S&T Cooperation Projects of Jiangxi Province (No.20212BDH80021).

Author’s contributions

WQ conceived and designed the experiments. MG performed the extraction of features, model construction, model training, and evaluation. QW,PW and WY analyzed the data and implemented the classifiers.MG and WQ drafted the manuscript. XX supervised this project and revised the manuscript. WK prepared the figures. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

References

1. WELCHMAN R L, GORDON C, MAYER R J. Ubiquitin and ubiquitin-like proteins as multifunctional signals [J]. Nature reviews Molecular cell biology, 2005, 6(8): 599-609.
2. HERRMANN J, LERMAN L O, LERMAN A. Ubiquitin and ubiquitin-like proteins in protein regulation [J]. Circ Res, 2007, 100(9): 1276-91.
3. TUNG C W, HO S Y. Computational identification of ubiquitylation sites from protein sequences [J]. BMC Bioinformatics, 2008, 9(1): 310.
4. HE D, LI M, DAMARIS R N, et al. Quantitative ubiquitylomics approach for characterizing the dynamic change and extensive modulation of ubiquitylation in rice seed germination [J]. The Plant Journal, 2020, 101(6): 1430-47.
5. OH E, AKOPIAN D, RAPE M. Principles of ubiquitin-dependent signaling [J]. Annual Review of Cell and Developmental Biology, 2018, 34: 137-62.
6. XU G, JAFFREY S R. The new landscape of protein ubiquitination [J]. Nat Biotechnol, 2011, 29(12): 1098-100.
7. STARITA L M, PARVIN J D. The multiple nuclear functions of BRCA1: transcription, ubiquitination and DNA repair [J]. Curr Opin Cell Biol, 2003, 15(3): 345-50.
8. PARK H-B, KIM J-W, BAEK K-H. Regulation of Wnt signaling through ubiquitination and deubiquitination in cancers [J]. International Journal of Molecular Sciences, 2020, 21(11): 3904.
9. PORRO A, BERTI M, PIZZOLATO J, et al. FAN1 interaction with ubiquitylated PCNA alleviates replication stress and preserves genomic integrity independently of BRCA2 [J]. Nature communications, 2017, 8(1): 1-14.
10. STANKOVIC-VALENTIN N, MELCHIOR F. Control of SUMO and ubiquitin by ROS: signaling and disease implications [J]. Molecular aspects of medicine, 2018, 63: 3-17.
11. CORN J E, VUCIC D. Ubiquitin in inflammation: the right linkage makes all the difference [J]. Nature structural & molecular biology, 2014, 21(4): 297-300.
12. TSUCHIDA S, SATOH M, TAKIWAKI M, et al. Ubiquitination in periodontal disease: A review [J]. International journal of molecular sciences, 2017, 18(7): 1476.
13. CHAN C-H, JO U, KOHRMAN A, et al. Posttranslational regulation of Akt in human cancer [J]. Cell & bioscience, 2014, 4(1): 1-9.
14. SCHMIDT M F, GAN Z Y, KOMANDER D, et al. Ubiquitin signalling in neurodegeneration: mechanisms and therapeutic opportunities [J]. Cell Death & Differentiation, 2021, 28(2): 570-90.
15. YAMADA T, MURATA D, ADACHI Y, et al. Mitochondrial stasis reveals p62-mediated ubiquitination in Parkin-independent mitophagy and mitigates nonalcoholic fatty liver disease [J]. Cell metabolism, 2018, 28(4): 588-604. e5.
16. LU D, LIN W, GAO X, et al. Direct ubiquitination of pattern recognition receptor FLS2 attenuates plant innate immunity [J]. Science, 2011, 332(6036): 1439-42.
17. MARINO D, PEETERS N, RIVAS S. Ubiquitination during plant immune signaling [J]. Plant physiology, 2012, 160(1): 15-27.

18. LI F, ZHANG Y, PURCELL A W, et al. Positive-unlabelled learning of glycosylation sites in the human proteome [J]. BMC bioinformatics, 2019, 20(1): 1-17.
19. LUO F, WANG M, LIU Y, et al. DeepPhos: prediction of protein phosphorylation sites with deep learning [J]. Bioinformatics, 2019, 35(16): 2766-73.
20. CHEN X, QIU J-D, SHI S-P, et al. Incorporating key position and amino acid residue features to identify general and species-specific Ubiquitin conjugation sites [J]. Bioinformatics, 2013, 29(13): 1614-22.
21. QIU W, XU C, XIAO X, et al. Computational prediction of ubiquitination proteins using evolutionary profiles and functional domain annotation [J]. Current genomics, 2019, 20(5): 389-99.
22. QIU W R, SUN B Q, XIAO X, et al. iPhos-PseEvo: Identifying Human Phosphorylated Proteins by Incorporating Evolutionary Information into General PseAAC via Grey System Theory [J]. Mol Inform, 2017, 36(5-6): 1600010.
23. QIU W R, XU A, XU Z C, et al. Identifying Acetylation Protein by Fusing Its PseAAC and Functional Domain Annotation [J]. Front Bioeng Biotechnol, 2019, 7: 311.
24. QIU W-R, WANG Q-K, GUAN M-Y, et al. Predicting S-nitrosylation proteins and sites by fusing multiple features [J]. Mathematical Biosciences and Engineering, 2021, 18(6): 9132-47.
25. QIU W-R, GUAN M-Y, WANG Q-K, et al. Identifying Pupylation Proteins and Sites by Incorporating Multiple Methods [J]. Frontiers in Endocrinology, 2022, 13: 1-11.
26. CHEN J, ZHAO J, YANG S, et al. Prediction of protein ubiquitination sites in Arabidopsis thaliana [J]. Current Bioinformatics, 2019, 14(7): 614-20.
27. MOSHARAF M, AHMED F, HASSAN M, et al. In Silico Prediction of Protein Ubiquitination Sites by Using Binary Encoding on Arabidopsis thaliana [J]. Int J Statist Sci, 2019, 18: 65-76.
28. MOSHARAF M P, HASSAN M M, AHMED F F, et al. Computational prediction of protein ubiquitination sites mapping on Arabidopsis thaliana [J]. Computational Biology and Chemistry, 2020, 85: 107238.
29. WANG H, WANG Z, LI Z, et al. Incorporating deep learning with word embedding to identify plant ubiquitylation sites [J]. Frontiers in Cell and Developmental Biology, 2020, 8: 1-13.
30. SIRAJ A, LIM D Y, TAYARA H, et al. Ubicomb: A hybrid deep learning model for predicting plant-specific protein ubiquitylation sites [J]. Genes, 2021, 12(5): 717.
31. YIN S, ZHENG J, JIA C, et al. UPFPSR: a ubiquitylation predictor for plant through combining sequence information and random forest [J]. Mathematical Biosciences and Engineering, 2022, 19(1): 775-91.
32. XU H, ZHOU J, LIN S, et al. PLMD: an updated data resource of protein lysine modifications [J]. Journal of Genetics and Genomics, 2017, 44(5): 243-50.
33. BOUTET E, LIEBERHERR D, TOGNOLLI M, et al. UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view [M]. Plant Bioinformatics. Springer. 2016: 23-54.
34. MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint arXiv:13013781, 2013,

35. YANG K K, WU Z, BEDBROOK C N, et al. Learned protein embeddings for machine learning [J]. Bioinformatics, 2018, 34(15): 2642-8.
36. UNIPROT CONSORTIUM T. UniProt: the universal protein knowledgebase [J]. Nucleic acids research, 2017, 45(D1): D158-D69.
37. HASAN M A M, AHMAD S. mLysPTMpred: Multiple Lysine PTM Site Prediction Using Combination of SVM with Resolving Data Imbalance Issue [J]. Natural Science, 2018, 10(09): 370-84.
38. LEVY O, GOLDBERG Y. Neural word embedding as implicit matrix factorization [J]. Advances in neural information processing systems, 2014, 27: 2177-85.
39. WANG M, CUI X, LI S, et al. DeepMal: Accurate prediction of protein malonylation sites by deep neural networks [J]. Chemometrics and Intelligent Laboratory Systems, 2020, 207: 104175.
40. DOU L, LI X, ZHANG L, et al. iGlu_AdaBoost: identification of lysine glutarylation using the AdaBoost classifier [J]. Journal of Proteome Research, 2020, 20(1): 191-201.
41. MANAVALAN B, SHIN T H, KIM M O, et al. PIP-EL: A New Ensemble Learning Method for Improved Proinflammatory Peptide Predictions [J]. Front Immunol, 2018, 9: 1783.
42. LI F, CHEN J, GE Z, et al. Computational prediction and interpretation of both general and specific types of promoters in Escherichia coli by exploiting a stacked ensemble-learning framework [J]. Briefings in bioinformatics, 2021, 22(2): 2126-40.
43. XIE R, LI J, WANG J, et al. DeepVF: a deep learning-based hybrid framework for identifying virulence factors using the stacking strategy [J]. Briefings in bioinformatics, 2021, 22(3): bbaa125.
44. KE G, MENG Q, FINLEY T, et al. Lightgbm: A highly efficient gradient boosting decision tree [J]. Advances in neural information processing systems, 2017, 30: 3146–54.
45. LIU Y, YU Z, CHEN C, et al. Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net [J]. Analytical Biochemistry, 2020, 609: 113903.
46. ZHOU K, HU Y, PAN H, et al. Fast prediction of reservoir permeability based on embedded feature selection and LightGBM using direct logging data [J]. Measurement Science and Technology, 2020, 31(4): 045101.
47. CHEN C, ZHANG Q, MA Q, et al. LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion [J]. Chemometrics and Intelligent Laboratory Systems, 2019, 191: 54-64.
48. LIANG W, LUO S, ZHAO G, et al. Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms [J]. Mathematics, 2020, 8(5): 765.
49. CAI C Z, HAN L Y, JI Z L, et al. SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence [J]. Nucleic Acids Res, 2003, 31(13): 3692-7.
50. ZAVALJEVSKI N, STEVENS F J, REIFMAN J. Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions [J]. Bioinformatics, 2002, 18(5): 689-96.

51. GORDON A D, BREIMAN L, FRIEDMAN J H, et al. Classification and Regression Trees [J]. Biometrics, 1984, 40(3): 358.
52. BOULESTEIX A L, JANITZA S, KRUPPA J, et al. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics [J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2012, 2(6): 493-507.
53. AHMAD M W, MOURSHED M, REZGUI Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption [J]. Energy and Buildings, 2017, 147: 77-89.
54. NOBLE W S. What is a support vector machine? [J]. Nat Biotechnol, 2006, 24(12): 1565-7.
55. CUI D, CURRY D. Prediction in Marketing Using the Support Vector Machine [J]. Marketing Science, 2005, 24(4): 595-615.
56. TONG S, CHANG E. Support vector machine active learning for image retrieval [J]. Proceedings of the ninth ACM international conference on Multimedia, 2001, 107-18.
57. WANG D, LIANG Y, XU D. Capsule network for protein post-translational modification site prediction [J]. Bioinformatics, 2019, 35(14): 2386-94.
58. XU H, JIA P, ZHAO Z. Deep4mC: systematic assessment and computational prediction for DNA N4-methylcytosine sites by deep learning [J]. Briefings in bioinformatics, 2021, 22(3): bbaa099.
59. ZHANG S, ZHENG D, HU X, et al. Bidirectional long short-term memory networks for relation classification, proceedings of the Proceedings of the 29th Pacific Asia conference on language, information and computation, F, 2015 [C].
60. GRAVES A. Long short-term memory [M]. Supervised sequence labelling with recurrent neural networks. Springer. 2012: 37-45.
61. QIAO Y, ZHU X, GONG H. BERT-Kcr: prediction of lysine crotonylation sites by a transfer learning method with pre-trained BERT models [J]. Bioinformatics, 2022, 38(3): 648-54.
62. XIAO X, SHAO Y-T, CHENG X, et al. iAMP-CA2L: a new CNN-BiLSTM-SVM classifier based on cellular automata image for identifying antimicrobial peptides and their functional types [J]. Briefings in bioinformatics, 2021, 22(6): bbab209.
63. CHEN W, CHEN G, ZHAO L, et al. Predicting drug–target interactions with deep-embedding learning of graphs and sequences [J]. The Journal of Physical Chemistry A, 2021, 125(25): 5633-42.

Figures

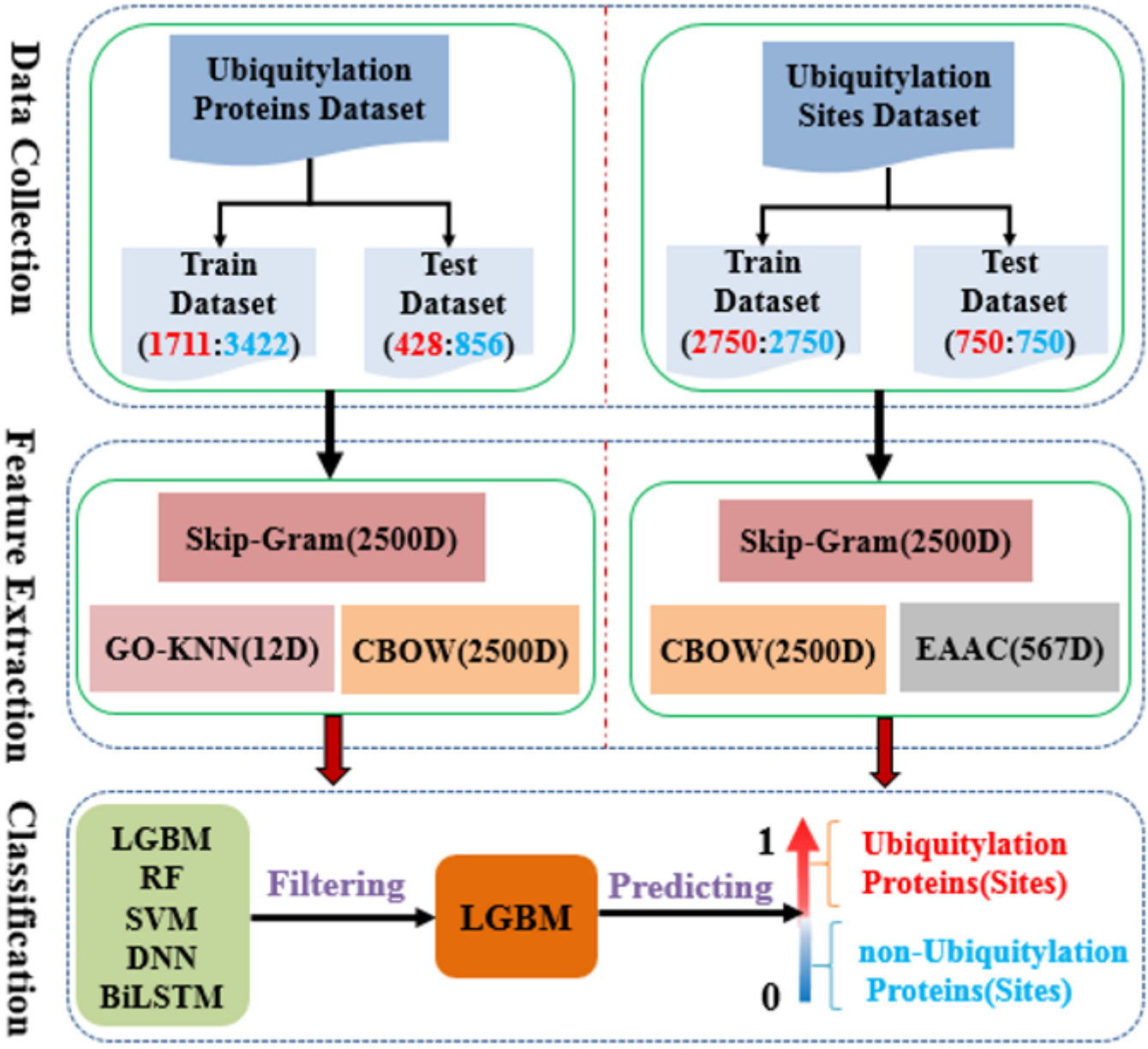

Figure 1

The framework of Ub-PS-Fuse

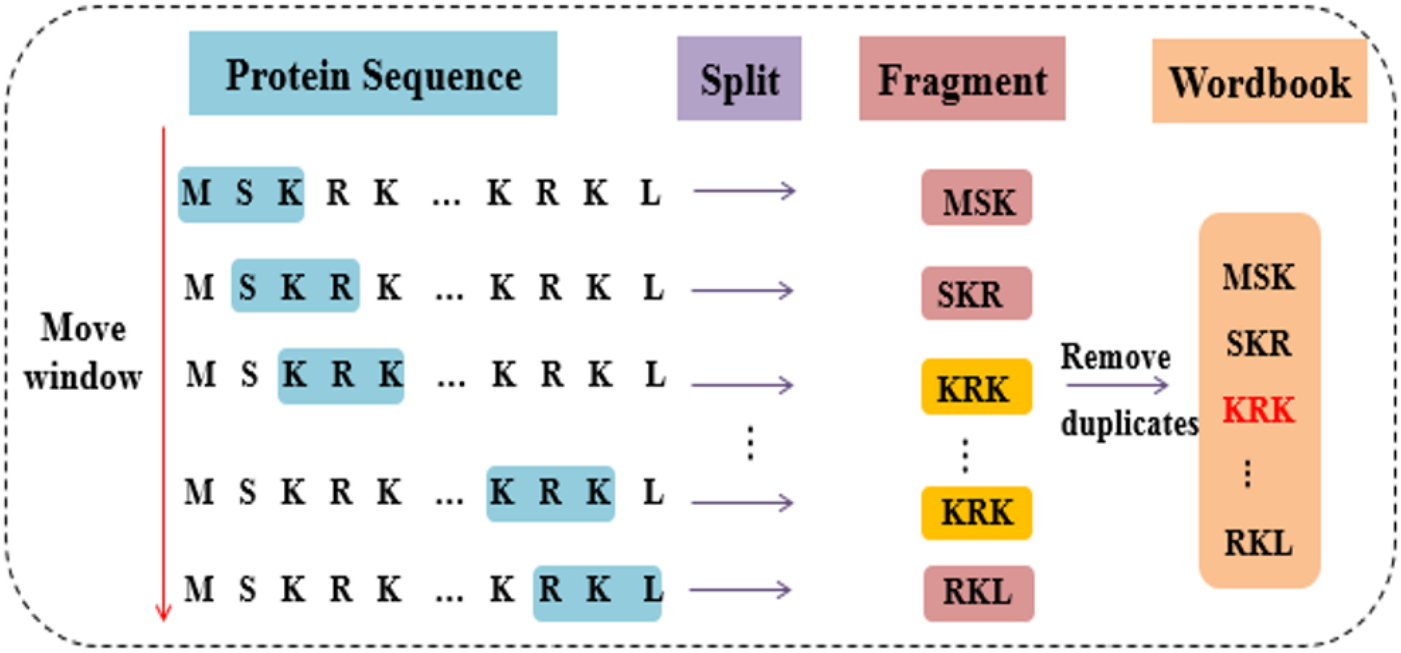

Figure 2

Detailed process of splitting ubiquitylation protein sequences

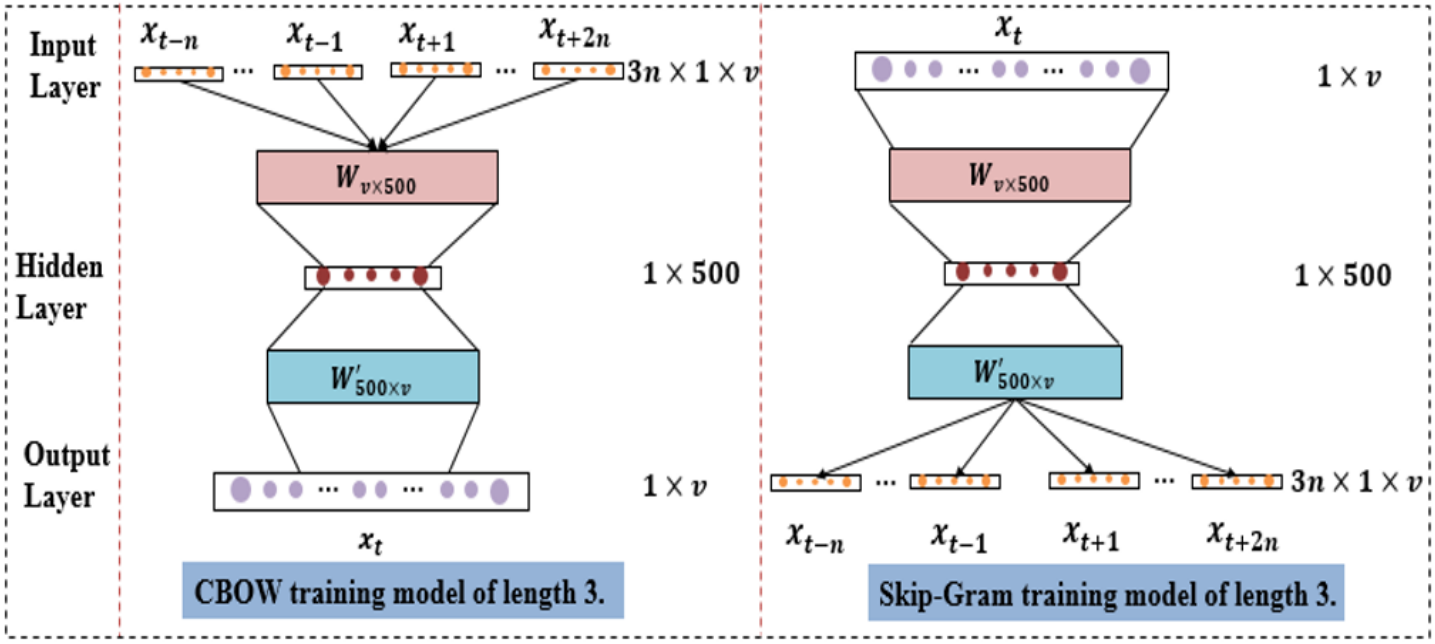

Figure 3

The specific training process for CBOW models and Skip-gram models.

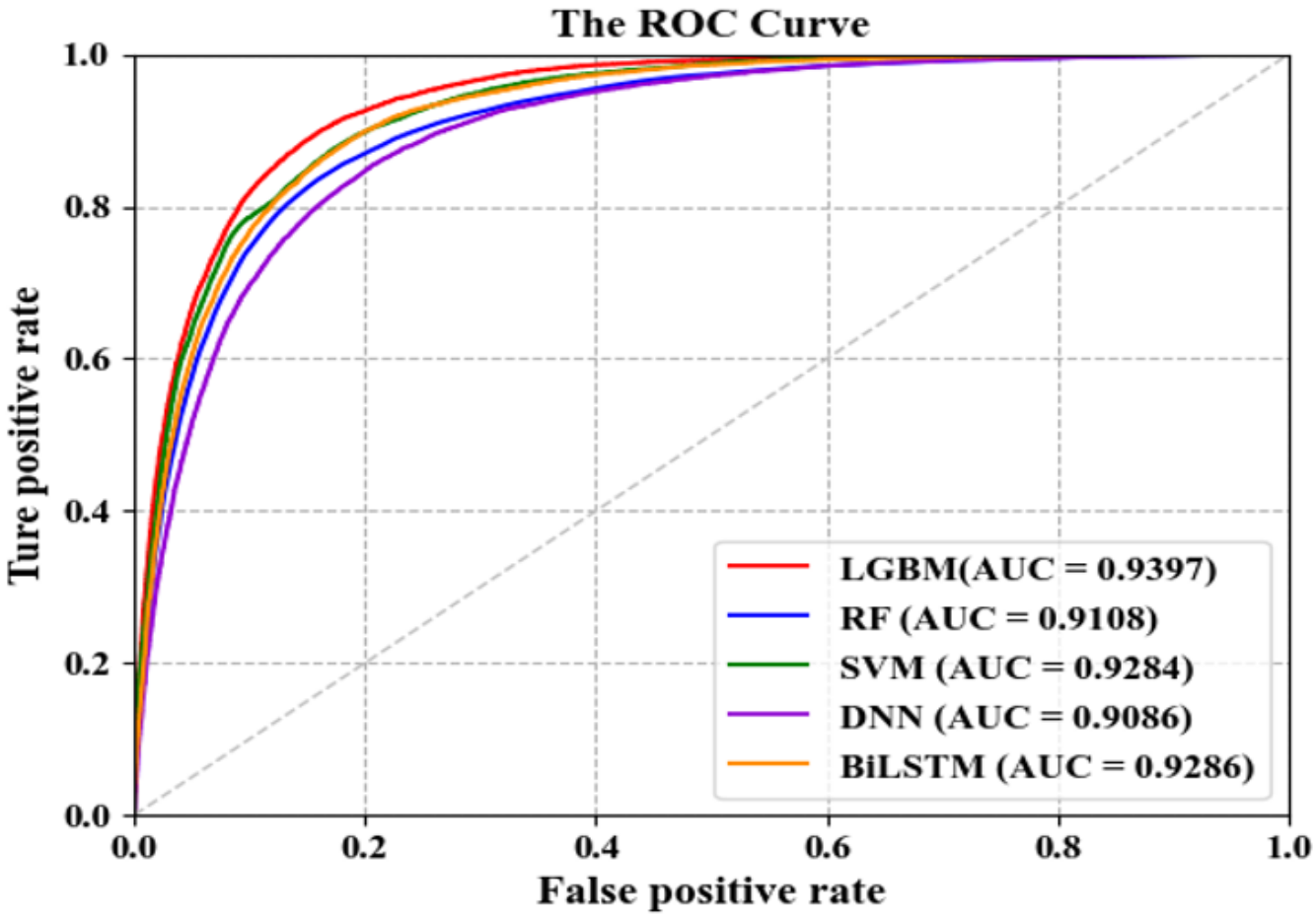

Figure 4

ROC curves of different classifiers on the training set when predicting ubiquitylation proteins.

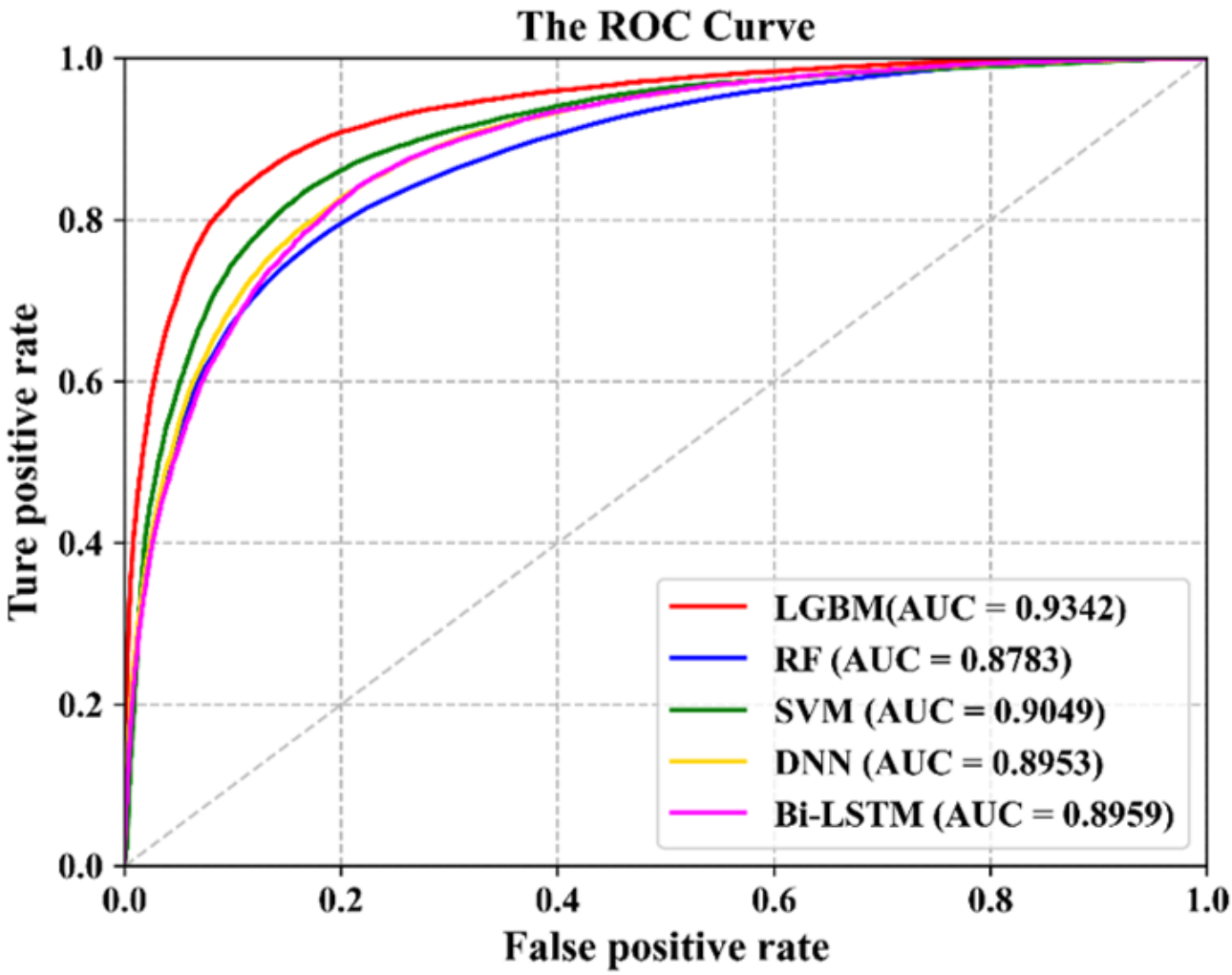

Figure 5

ROC curves of different classifiers on the training set when predicting ubiquitylation sites.

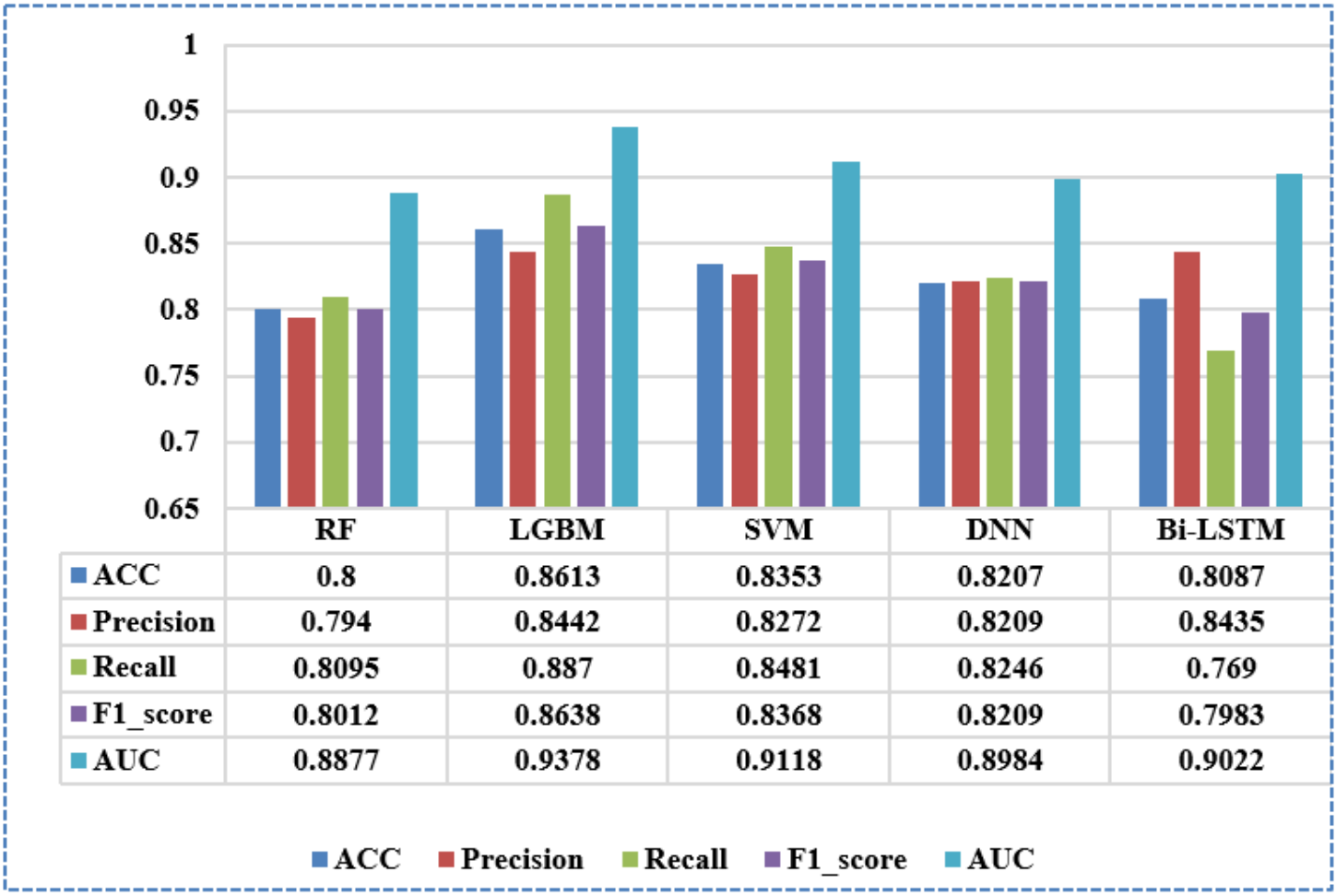

Figure 6

Prediction results of different classifiers on independent test sets when predicting ubiquitylation sites