Mathematical Biosciences and Engineering

MBE, 20(8): 14734–14755.
DOI: 10.3934/mbe.2023659
Received: 02 April 2023
Revised: 08 June 2023
Accepted: 11 June 2023
Published: 07 July 2023

http://www.aimspress.com/journal/MBE

Research article

A normalized differential sequence feature encoding method based on amino acid sequences

Xiaoman Zhao[1,2], Xue Wang[1], Zhou Jin[1] and Rujing Wang[1,2,*]

[1] Institute of Intelligent Machinery, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China
[2] University of Science and Technology of China, Hefei 230026, China

* **Correspondence:** Email: rjwang@iim.ac.cn; Tel: 055165591131; Fax: 055165591131.

Abstract: Protein interactions are the foundation of all metabolic activities of cells, such as apoptosis, the immune response, and metabolic pathways. In order to optimize the performance of protein interaction prediction, a coding method based on normalized difference sequence characteristics (NDSF) of amino acid sequences is proposed. By using the positional relationships between amino acids in the sequences and the correlation characteristics between sequence pairs, NDSF is jointly encoded. Using principal component analysis (PCA) and local linear embedding (LLE) dimensionality reduction methods, the coded 174-dimensional human protein sequence vector is extracted using sequence features. This study compares the classification performance of four ensemble learning methods (AdaBoost, Extra trees, LightGBM, XGBoost) applied to PCA and LLE features. Cross-validation and grid search methods are used to find the best combination of parameters. The results show that the accuracy of NDSF is generally higher than that of the sequence matrix-based coding method (MOS) coding method, and the loss and coding time can be greatly reduced. The bar chart of feature extraction shows that the classification accuracy is significantly higher when using the linear dimensionality reduction method, PCA, compared to the nonlinear dimensionality reduction method, LLE. After classification with XGBoost, the model accuracy reaches 99.2%, which provides the best performance among all models. This study suggests that NDSF combined with PCA and XGBoost may be an effective strategy for classifying different human protein interactions.

Keywords: amino acid sequences; protein interactions; sequence feature extraction; dimensionality reduction methods; integrated learning

1. Introduction

Proteins are the main performers of cellular activities in living organisms and are involved in various aspects of organism growth and reproduction, such as cell signaling, metabolism, apoptosis and necrosis, and the regulation of gene expression [1]. In an organism, proteins do not exist in isolation nor exert their biological properties alone, but rather interact with other proteins in some way to drive or trigger specific biochemical reactions together and synergize their biological properties [2]. The study of protein interactions helps to explore the mechanisms of disease occurrence and to find new drug targets [3], which opens the way for new drug development. Therefore, the interaction of proteins is necessary for classification and prediction. Initially, researchers have often used traditional low-throughput techniques to detect protein interactions, such as nuclear magnetic resonance, chromatographic electrophoresis, and other methods [4]. More mature and well-established experimental techniques for high-throughput detection of proteins, such as yeast two-hybrid screening [5], fluorescence resonance energy transfer [6], phage display [7], and tandem affinity purification [8], are used to detect the interactions between proteins. However, this routine analysis can only identify minimal protein interactions and is not suitable for all proteins of the organism because the accuracy of the identification results is not high [9]. Therefore, a computational protein interaction prediction method that can support highly efficient and high accuracy is needed.

Among the calculation and prediction methods of protein interaction, protein-coding methods, feature extraction methods, and classification algorithms are the three main factors that affect the performance of protein interaction prediction models [10]. Protein sequence data has the characteristics of non-numerical, strong correlations, and varying lengths, which makes it difficult for machine learning methods to process these data directly. Therefore, the coding of the protein sequence data becomes very important. Conjoint triads (CT) [11] used any three consecutive amino acids as a unit and calculated their frequencies and adjacent interactions in amino acid sequences, demonstrating that protein interactions can be predicted by sequences alone. Auto-covariance (AC) [12] considers proximity effects within 1 to 30 amino acids in a protein sequence. However, it is limited to the frequency of occurrence of each amino acid in the sequence. For the local descriptor (LD) [13], the construct feature vectors are constructed by dividing the amino acid sequence into ten local segments, composed, transformed, and distributed, and the codes of the used local segments are linked together to form a complete amino acid sequence code. Recently, in the existing methods, the association characteristics between sequence pairs are not considered in the amino acid sequences. Therefore, the classification process of the traditional methods display obvious classification characteristics, and the classification result is poor. For example, Gui et al. proposed a sequence matrix-based coding method (MOS), which considered amino acid sequences' global features, longevity effects, and coded amino acid sequence data into a vector with consistent dimensionality. The sequential order of the whole sequence was not considered in the CT, AC, and LD coding methods to solve the problem. Although MOS considers the amino acid sequence by constructing the amino acid sequence frequency matrix, it only considers the frequency information of the whole amino acid sequence; it does not consider the anterior-posterior position order relationship of the whole protein sequence.

Although protein-coding methods can represent protein sequences by numbers, digital protein sequences contain correlated noise and redundant feature information to a certain extent. Protein feature extraction methods can accurately and objectively reduce redundancy, shorten training time, and reduce losses, which plays a vital role in protein interaction classification. Linear and nonlinear dimensionality reduction techniques have been widely used in the field of biological proteins. For example, principal component analysis (PCA) transforms raw data into a set of linearly independent representations of various dimensions through linear transformation, and PCA has been widely used in sequence data processing [14]. Shao et al. conducted a principal component analysis on the amino acid content and composition of the woolly bones chicken to obtain an accurate evaluation of the amino acid composition and nutritional value [15]. The local linear embedding (LLE) method represents local linearity as global non-linearity while ensuring that the topology of the original data is maintained after dimensionality reduction. Zhang et al. performed LLE dimensionality reduction to sort out protein sequences from the static protein interaction network (SPIN) and dynamic protein interaction network (DPIN) perspectives [16], thus improving the performance of the classification model.

The encoding methods and feature extraction of sequence data carry protein sequence information that often requires suitable classifiers for the classification prediction of complex compounds, which are formed by molecular interface interactions. Currently, researchers have focused on integrated learning algorithms to construct classification prediction models to address the shortcomings of traditional machine learning algorithms, such as using a single classifier [17]. For example, Liu et al. used AdaBoost to construct a prediction model to predict protein-protein interaction hotspot and non-hotspot residues. The prediction model index recall value reached 53.4%, and the F1 value was 51.2% [18]. Liu et al. proposed the protein crotonylation site prediction model LightGBM-CroSite, and the evaluation metrics on the training set were greatly improved compared with other models [19]. Moreover, Zhang et al. proposed a new StackPDB method for DNA binding protein prediction based on stacking integration, where the optimal feature subset is selected by XGBoost-recursive feature elimination [20]. Extratrees, a variant of RandomForest, uses random sampling of the original training set and random selection of a feature value to divide the decision tree. Han et al. used the extra-trees method to build a forest stock estimation model, and the results showed that the algorithm could effectively reduce the experimental error. Therefore, in the above studies, the four integrated learning algorithms belonging to bagging and boosting used the negative gradient of the loss function as the residual approximation of the current decision tree to fit the new decision tree. Although these integrated learning algorithms are widely cited for plant protein classification prediction problems, the raw sequences are directly used as the input to the model.

In this study, a coding method of a protein sequence based on normalized difference sequence features is proposed. In the protein sequence coding method, the combination of the position relationships between amino acids and the correlation characteristics between sequence pairs is considered, and a normalized difference sequence characteristic coding method is proposed. The position sequence relationships and frequency characteristic relationships of amino acid sequences are analyzed to retain more information about the amino acid sequence. In the feature extraction of protein sequence coding information, PCA and LLE dimensionality reduction methods are used to extract protein sequence features from the coding results. The performance is compared to determine the ideal method for extracting human proteins. In order to make better use of the proposed protein feature descriptor, NDSF is combined with four ensemble learning algorithms to predict and classify: AdaBoost, Etratress, LightGBM, and XGBoost. The technical roadmap is shown in Figure 1.

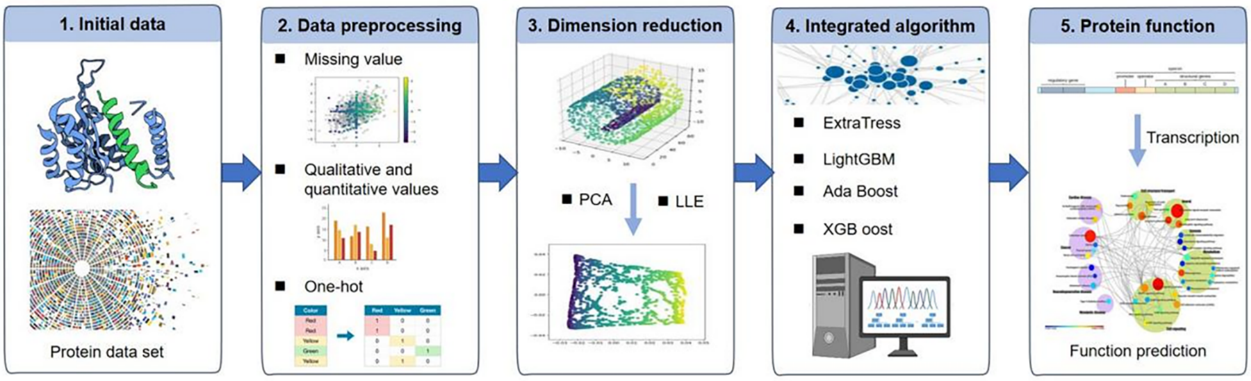

Figure 1. Technology line.

2. Materials and methods

The dataset was provided by Pan et al. [21]. In this dataset, positive samples were obtained from the Human Protein Reference Database (HPRD, 2007 edition) and contained 9476 protein sequences, with a total of 36,630 pairs of protein interaction sequences. Negative samples were collected from Swiss-Prot [22] (http://www.expasy.org/sprot/, version57.3) according to the following requirements: 1) only human proteins were collected; 2) unclear or uncertain subcellular localization terms are excluded (e.g., “potential”, “possible” or “by similarity”); 3) exclude sequences where the annotated has two or more positions; and 4) exclude sequences marked with “fragments” and remove sequences with less than 50 amino acid residues in sequence length. After the above-mentioned process, a total of 2184 human proteins were collected from six different subcellular organelles (cytoplasm, nucleus, endoplasmic reticulum, Golgi apparatus, lysosome, and mitochondria). The data for the negative set were collected using a method that counts proteins specific to different subcellular organelles and constitutes protein interaction pairs for proteins that are in different subcellular organelles. Since these proteins exist in different physical locations, they can be considered as not interacting with each other. It is important to exclude those proteins that are bilocalized, a transcription factor, etc. Although this approach does not necessarily lead to the construction of a precise protein non-interaction library, there is some theoretical basis for using it as a negative set.

The dataset was preprocessed with most protein sequences ranging from 100 to 1000 amino acids in length, de-selecting sequence pairs with less than 50 amino acid residues in sequence length for both positive and negative samples, and protein pairs with uncommon amino acids (B, J, O, Z, U, and X) in the protein sequences in the deleted samples. A total of 36,545 positive and 36,323 negative sample pairs were produced by protein pairs with unusual amino acid deletions. 30,000 positive samples and 30,000 negative samples were selected to form the training data set, and the remaining part was used as the test set to validate the model.

2.1. Normalized differential sequence feature extraction method

To mine the most representative attributes from the samples, protein sequences of different lengths are normalized to carriers of the same size. Efficient feature descriptors can improve the performance of classification models [23]. Therefore, we have proposed the normalized difference sequence feature method, the basic flow of which is shown in Figure 2. It is connected to the sequence pairs to form a new sequence, which encapsulates the global sequence information and local information of the protein. The new sequence pair needs to calculate the relative frequency and position features to establish the connection with a single sequence pair. That is, after the new sequence pair is statistically fused with the merged frequency (MF) and merge position (MP) features, the corresponding values of frequency information MF and position information MP are compared with the original sequence single frequency (SF) and single position (SP). The corresponding value frequency information MF and position information MP are crossed with the original sequence SF and SP, and a crossover feature is calculated, which can be named a normalized difference sequence feature (NDSF).

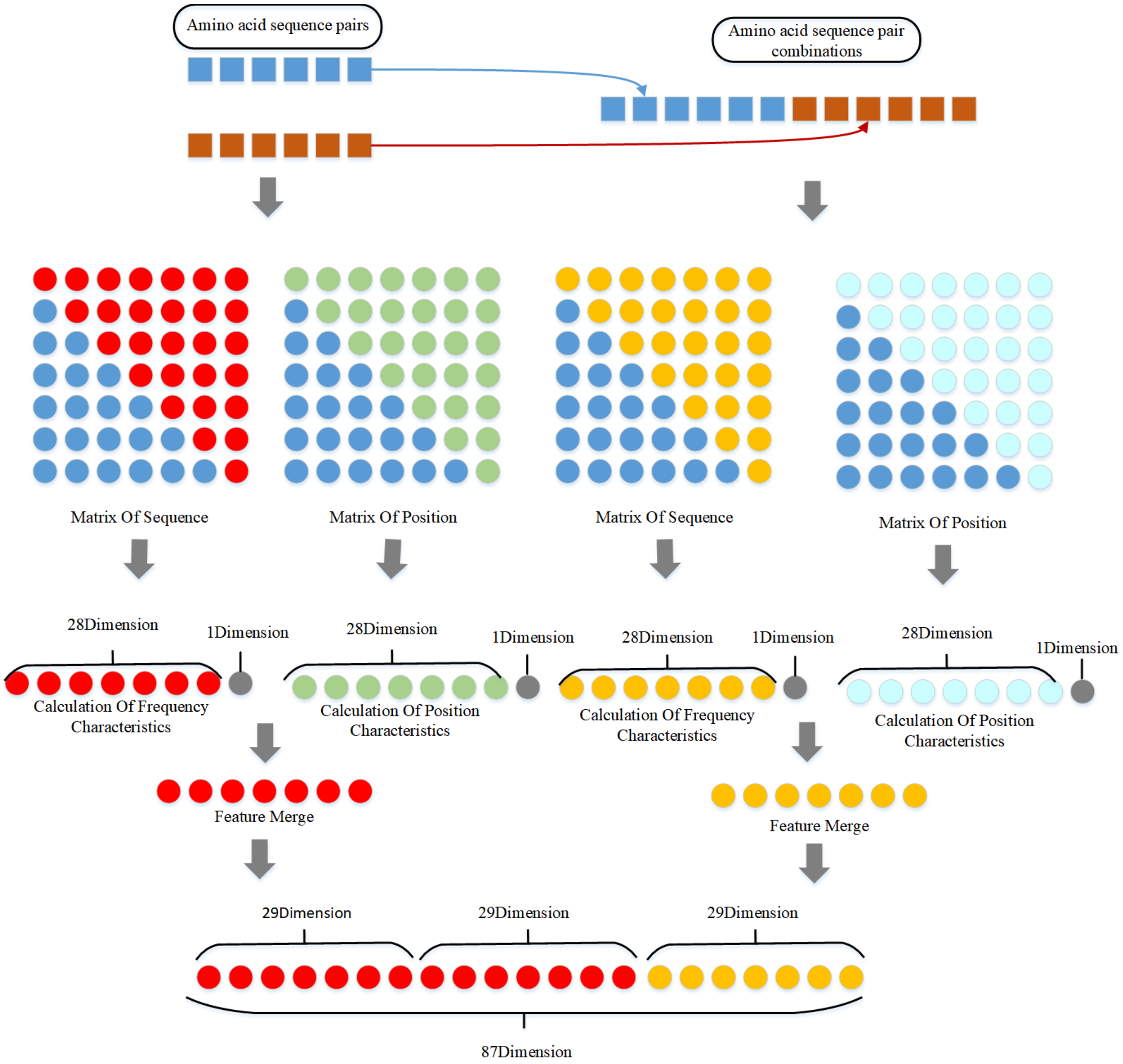

Figure 2. Step-by-step diagram of the NDSF calculation method.

2.1.1. Classification of amino acids

Electrostatic (including hydrogen bonding) and hydrophobic interactions dominate PPIs. These two kinds of interactions may be reflected by the dipoles and volumes of the side chains of amino acids, respectively. To achieve a numerical representation of amino acids, 20 common amino acids were classified into seven categories according to the B3LYP/6-31G in density generalization theory [24] and molecular modeling methods [25], as shown in Table 1.

Table 1. Amino acids grouped based on dipole and side chain volume.

Amino acid type	Grouping
Ala, Gly, Val	1
IIe, Leu, Phe, Pro	2
Tyr, Met, Thr, Ser	3
His, Asn, Gln, Trp	4
Arg, Lys	5
Asp, Glu	6
Cys	7

2.1.2. Position frequency characteristics calculation

The product of the two elements on the diagonal of the sequence matrix is equal to the sum of the symmetrical elements above and below the diagonal. The values on the diagonal and above the sequence matrix are selected to encode the sequences. In addition, the reciprocal of the one-dimensional sequence length (1/L) is added as a component of the sequence matrix encoding, which differentiates the lengths of amino acid sequences. Finally, the sequence matrix is used to encode MOS_CODE for amino acid sequences, and a 29-dimensional vector is obtained as the frequency feature of the sequences.

$$MOS\_CODE = (MOS_{11}, \ldots, MOS_{17}, \ldots, MOS_{77}, \frac{1}{L}) \tag{1}$$

The elements in the upper half of the position matrix cover the positional information of all elements in the sequence. To be consistent with the sequence matrix encoding, the inverse of the amino acid sequence length is chosen as the encoding component of the position matrix. Finally, the position matrix encoding MOP_CODE of the amino acid sequence data is a 29-dimensional vector called the position feature of the amino acid sequence.

$$MOP\_CODE = (MOP_{11}, \ldots, MOP_{17}, \ldots, MOP_{77}, \frac{1}{L}) \tag{2}$$

The overall calculation flow of NDSF is shown in Figure 3.

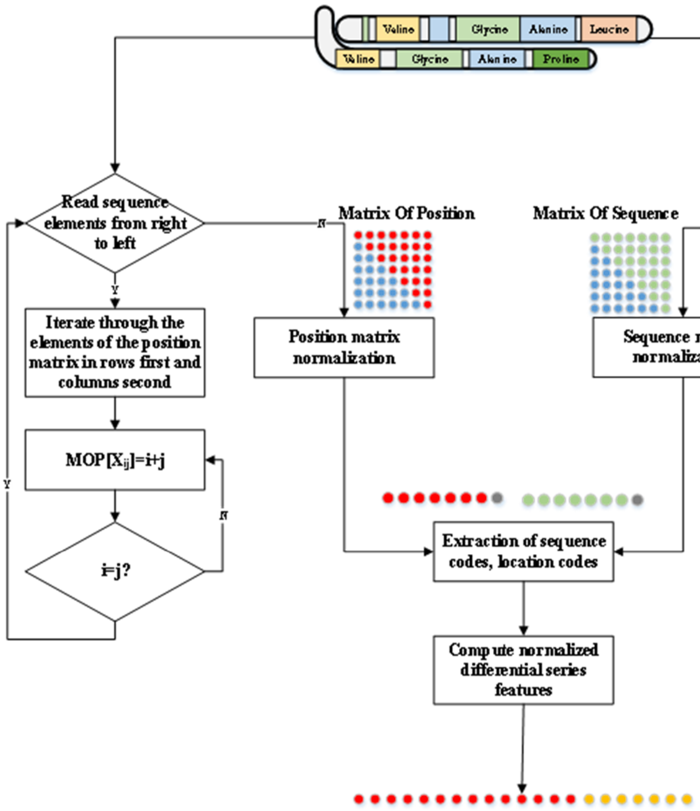
Valine
Glycine
Alanine
Leucine
Valine
Glycine
Alanine
Proline
Read sequence elements from right to left
N
Y
Iterate through the elements of the position matrix in rows first and columns second
MOP[X_ij]=i+j
N
i=j?
Y
Matrix Of Position
Position matrix normalization
Matrix Of Sequence
Sequence matrix normalization
N
Read sequence elements from right to left
Y
Sequence vector = sequence vector + 1
Y
Sequence matrix = sequence vector + sequence matrix (by row)
Extraction of sequence codes, location codes
Compute normalized differential series features

Figure 3. NDSF overall flow chart.

1) Matrix Of sequence calculation

Based on the sequence matrix, the position characteristics of amino acid sequence data are proposed, which allows the amino acid sequences to retain enough original information. It reduces the manual coding algorithm's time and space complexity to the maximum extent. When constructing the location feature matrix, assume a non-empty finite set: $\Omega = \{w_1, \dots, w_N\}$. In sequence S, $S = S_1, S_2, \dots, S_L$, where N represents the eigenvalues of the sequence S, and L represents the length of the sequence S, $S_i \in \Omega, 1 \leq i \leq L$. The position matrix of sequence S: $MOP = [x_{ij}]_{N*N}$, where

$$X_{IJ} = \begin{cases} location\ of\ \dots w_i \dots w_i \dots in\ S, \quad i = j \\ location\ of\ \dots w_i \dots w_j \dots or\ \dots w_j \dots w_i \dots in\ S, i \neq j \end{cases} \tag{3}$$

Finally, since the sum of all the elements in the position matrix is L^3-L^2-L, normalizing all the elements in the position matrix to unify the magnitudes:

$$MOP_i = \frac{MOP_i}{L(L*L-L-1)} \tag{4}$$

Although the amino acid sequence position feature calculation algorithm has a time-delay complexity directly related to the length of the sequence, it contains positional information between amino acids at extra-long distances in the amino acid sequence. However, its effect is negligible.

2) Matrix Of position calculation

In the same way, as the sequence matrix is calculated, the position matrix of the sequence S $MOP = \left[x_{ij}\right]_{N*N}$

$$X_{IJ} = \begin{cases} frequency\ of \ldots w_i \ldots or \ldots w_i \ldots w_i \ldots in\ S, & i = j \\ frequency\ of \ldots w_i \ldots w_j \ldots in\ S, i \neq j \end{cases} \tag{5}$$

Similarly, to normalize the elements in the sequence matrix：

$$MOS_i = \frac{MOS_i}{L(L+1)} \tag{6}$$

2.1.3. Normalization process

The specific normalized differential series feature calculation function is as follows:

$$MF_{new} = Norm\left(\frac{\sqrt{SF_1 \times SF_2}}{MF}\right) \tag{7}$$

$$MP_{new} = Norm\left(\frac{\sqrt{SP_1 \times SP_2}}{MP}\right) \tag{8}$$

Equation (9) shows the linearized transformation of initial data using normalized methods, in which X$_{\text{norm}}$ represents the normalized value, and x is the initial data. The corresponding 29-dimensional feature vectors are obtained, the original 58-dimensional feature vector pair are spliced together as an 87-dimensional vector, and the 174-dimensional vector input is obtained.

$$X_{norm} = \frac{X_i - \underset{1 \le i \le n}{min}(x_i)}{\underset{1 \le i \le n}{max}(x_i) - \underset{1 \le i \le n}{min}(x_i)} \tag{9}$$

2.1.4. Dimensionality reduction processing

Dimension reduction in machine learning is realized by mapping high-dimensional spatial data to a low-dimensional spatial representation, which is divided into linear and nonlinear mappings [26]. PCA is commonly used in linear mapping, and LLE is commonly used in the nonlinear mapping. Therefore, we use the PCA method to transform the data vectors of amino acid sequences that may have a linear correlation into a set of linearly uncorrelated vectors in each dimension by orthogonal transformation [27]. Therefore, these converted data vectors can represent the original information without losing the original data, and this group of converted variables is the main part of the original data [28]. LLE is a data dimension reduction method based on streamlet learning, in which the shape of the stream can be understood as embedding a subspace in a high-dimensional Euclidean space [29]. We used LLE to perform a protein sequence feature extraction on the encoded results to ensure that the topology of the original data is maintained after dimensionality reduction.

In order to reduce the computational complexity, these features were extracted again from the encoded amino acid sequences. Since the NDSF encoded sequence vector has 174 dimensions, we roughly chose the range of vector dimensionality scaling based on the interpretable variance plotted as dimensionality, as shown in Figure 4. The optimal data dimension was precisely found by repeating the experiment.

2.2. Integrated learning classification model

Four classification models (AdaBoost, Extratrees, LightGBM, XGBoost) were selected to distinguish the results of the interactions between different amino acid sequences [30]. AdaBoost has the main advantage of adaptive enhancement. In each iteration round, a new weak classifier is added until some predefined sufficiently small error rate. A pre-specified maximum number of iterations is reached before determining the final strong classifier [31]. In this study, the weights of amino acid sequence samples misclassified by the previous basic classifier are increased. In contrast, the weights of correctly classified amino acid sequence samples are decreased, which is additionally used to train the next basic classifier [32]. The AdaBoost algorithm flow is as follows:

a. Given the training sample set,

$$D = (x_1, y_1), \ldots, (x_m, y_m), \ldots, y \in \{1, -1\} \tag{10}$$

b.

$$w_{IJ} = \begin{cases} \frac{1}{2M}, \; y_i = -1 \\ \frac{1}{2L}, \; y_i = 1 \end{cases} \tag{11}$$

we initialize and normalize the weight coefficients with the formula, where L represents the number of correctly classified samples and M represents the number of incorrectly classified samples.

c. At each time, in the loop t:

1) train the samples according to the probability distribution D1 of the training set and obtain the basic classifier hi, and

2) update the weighting factors according to

$$D_{t+1}(i) = \frac{D_t(i) * e^{-\partial_i y_i h_i(x_i)}}{Z_t} = \frac{e^{-\sum_{j=1}^{t} \partial_i y_i h_i(x_i)}}{L * \prod_{j=1}^{t} Z_t} = \frac{e^{-mrg(x_i, y_i, f_i)}}{L * \prod_{j=1}^{t} Z_t} \tag{12}$$

where Z_t is the normalization factor, hi is the basic classifier, and $mrg(x_i, y_i, f_i)$ is the function boundary of the data points in the following function:

$$Z_t = \sum_{i=1}^{L} D_t(i) * exp(\partial_i y_i h_i(x_i)). \tag{13}$$

3) Obtain the basic classifier hi with the minimum forecast error.

d. Output the final strong classifier H

The second model is Light-GBM, which differs from AdaBoost in that the data sampling and feature sampling are performed in the implementation of Light-GBM to bind mutually exclusive features together, thus reducing the training time of the feature dimensions and model.

The difference of the XGBoost model is that a set of loss functions are customized with the Taylor expansion, which further increases the generalization ability of the model. Its gradient boosting tree-based algorithm adds a regularization term to the objective function, which can reduce the complexity of the model and avoid overfitting:

$$Obj(\phi) = \sum_{i=1}^{n} l(y_i, y_j) + \sum_k \Omega(f_k) \tag{14}$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \omega^2 \tag{15}$$

where y_i is the predicted value, y_j is the true value, $\Omega(f_k)$ is the regular term, f_k is the decision tree, T represents the number of leaf nodes, ω represents the proportion of leaf nodes, ω controls the number of leaf nodes, and λ controls the proportion of leaf nodes.

The XGBoost algorithm performs an iterative operation and a second-order Taylor expansion during the solution of the objective function, as shown in Eq (16):

$$Obj(\phi) = \sum_{i=1}^{n}\left[l(y_i, y_j^{(t-1)}) + g_i f_i(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_k) \tag{16}$$

$$g_i = \alpha_{y_j^{(t-1)}} l(y_i, y_j^{(t-1)}) \tag{17}$$

$$h_i = \alpha_{y_j^{(t-1)}}^2 l(y_i, y_j^{(t-1)}) \tag{18}$$

where Eqs (17) and (18) are the first and second-order derivatives of the loss function, respectively.

As a bagging method, extra trees are different from the random forest in that it randomly selects a characteristic value to divide the decision tree. Compared with random forest, the variance of the model is further reduced, but the deviation is further increased. Finally, these four models use cross-validation and grid search methods to find the best parameters (learning rate and n_estimator) to optimize the parameters. The models are evaluated by comparing their accuracy, recall, and loss.

2.3. Evaluation indicators

To evaluate the performance of the protein interaction prediction model based on amino acid sequences proposed in this paper, three widely used evaluation criteria, including precision, accuracy, recall, loss, Matthews correlation coefficient (MCC) and F-measure (F1) were used in this experiment and calculated as follows:

$$Precision = \frac{TP}{TP+FP} \tag{19}$$

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

$$Recall = \frac{TP}{TP+FN} \tag{21}$$

$$Loss = -(ylog(p) + (1-y)log(1-p)) \tag{22}$$

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(FP+TN)(TP+FN)(TN+FP)(TN+FN)}} \tag{23}$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{24}$$

where TP (true positives) denotes the number of times the initial positive sample is correctly predicted as positive by the model, TN (true negatives) denotes the number of times the model correctly predicted the initial negative sample as negative, FP (false positives) denotes the number of times the initial positive sample is incorrectly predicted as negative by the model, FN (false negatives) denotes the number of times the model incorrectly predicted the initial negative sample as a positive sample, y denotes the actual label of the sample (1 or 0), and p denotes the probability that the model predicts a positive sample [31].

The Gitee website provides access to the data and codes collected as part of the survey (https://gitee.com/mandy1023/ml).

3. Results and discussion

3.1. Comparison of the efficiency of coding methods

The feature vector dimension and coding time corresponding to the amino acid sequence coding method are shown in Table 2. After experimental comparison, the dimensionality of the sequence vector after NDSF encoding is 174, which is only second to MOS. The reduction in the dimensionality of the NDSF encoding vector results in a shorter time to encode an amino acid than that of the conventional encoding method, which is about 80% less than the encoding time of the AC method. NDSF has the complexity of a time delay directly related to the sequence length, so it is suitable for applications with high time requirements.

Table 2. Dimensionality of feature vectors.

Method	Feature vector dimension	Time to encode an amino acid*10^-3
CT	686	2.42
AC	420	2.11
LD	1260	2.52
MOS	58	0.41
NDSF	174	0.64

3.2. Classification prediction results of different amino acid coding methods

After completing data preprocessing, the problem of noisy data and large dimensionality of protein sequence-based numerical features still exists. The inefficiency of traditional coding methods, such as sequence matrices and other methods, to characterize only part of the key features of biological information was addressed. In this study, we put forward a standardized differential sequence feature method for protein-coding and compared it with the results of classification prediction by the protein sequence coding method. Based on a sequence matrix combined with an ensemble learning algorithm, it considers the whole sequence relationship and long-range effect of amino acid sequences, the results of which are shown in Table 3. The accuracy of all four protein interaction prediction models constructed by NDSF combined with AdaBoost, EtraTress, LightGBM, and Xgboost is higher than that of MOS. In the test set, regarding the MOS method, among the four models, the best model was LightGBM, with an accuracy rate of 96.15% and a loss rate of 3.84%. The combination of AdaBoost and MOS had the lowest accuracy rate of 73.85%, a loss rate of 26.14%, and a recall rate of 73.47%. The NDSF optimal model was Light-GBM, with an accuracy of 98.65%, a 2.5% improvement over the MOS method, a loss rate of 1.34%, and a recall rate of 98.43%. The lowest accuracy of AdaBoost combined with NDSF is 81.40%, with a loss rate of 18.59%, and a recall rate of 80.37%.

Table 3. Classification results of MOS and NDSF applied to the integrated learning algorithm.

		AdaBoost		Etratress		LightGBM		XGBoost	
		Train	Test	Train	Test	Train	Test	Train	Test
MOS	loss	24.99%	26.14%	0.08%	5.41%	1.43%	3.84%	2.71%	5.72%
	acc	75.00%	73.85%	99.91%	94.58%	98.56%	96.15%	97.28%	94.27%
	recall	74.47%	73.47%	99.91%	93.42%	98.16%	94.80%	96.42%	92.64%
	MCC	0.4822	0.4617	0.9821	0.9017	0.9697	0.9248	0.9597	0.8978
	F1	0.7422	0.7241	0.9944	0.9214	0.9787	0.9411	0.9571	0.9114
NDSF	loss	18.04%	18.59%	4.63%	1.34%	0.38%	1.23%	1.09%	1.77%
	acc	81.95%	81.40%	99.99%	98.65%	99.61%	98.76%	98.90%	98.22%
	recall	80.80%	80.37%	99.99%	98.23%	99.60%	98.43%	98.66%	97.65%
	MCC	0.6054	0.5874	0.9914	0.9801	0.9784	0.9741	0.9755	0.9612
	F1	0.7818	0.7940	0.9879	0.9783	0.9911	0.9781	0.9799	0.9589

Based on the physical and chemical properties of amino acids and the spatial structure of the protein, the normalized difference sequence feature coding method proposed in this paper takes into account the frequency information of the whole amino acid sequence, as well as the position sequence relationship between the front and back of the whole amino acid sequence. Although the dimension of the sequence vector after NDSF coding is slightly higher, combined with the ensemble learning algorithm and compared with the MOS coding method, although, more satisfactory results are obtained, and the accuracy of the algorithm is verified. Because the frequency and positional features of the sequences are inconsistent in magnitude, the two features need to be spliced. Therefore, it is necessary to normalize the sequences to ensure that the obtained information is consistent. A matrix needs to be constructed by dividing the amino acids into seven categories: one step for every seven amino acids, hence the term "difference". NDFS can retain more information on amino acid sequences by analyzing the position sequence relationship and characteristic frequency relationship of amino acid sequences. On the training set, the accuracy rate of NDSF is 81.95%–99.9%, and MOS is 75.00%–99.91%. Compared with MOS, the average loss rate of the NDSF model decreases by 1.44%, and the average recall rate increases by 2.5%.

3.3. Classification prediction results of different amino acid coding methods followed by dimensionality reduction

To keep most of the information on the original features as much as possible, it avoids the influence of correlation between sequence features on the classification results. Before using principal component analysis for dimension reduction, it is necessary to select the appropriate dimension and draw the explanatory variance as a function of the dimension. It is usually an inflection point on the curve where the interpretable variance rapidly stops increasing. Therefore, 45 dimensions are selected as the termination point, as shown in Figure 4. Figure 4(a) represents the feature selection performed by the model constructed by combining PCA and MOS; Figure 4(b) represents the feature selection performed by the model constructed by combining PCA and NDSF.

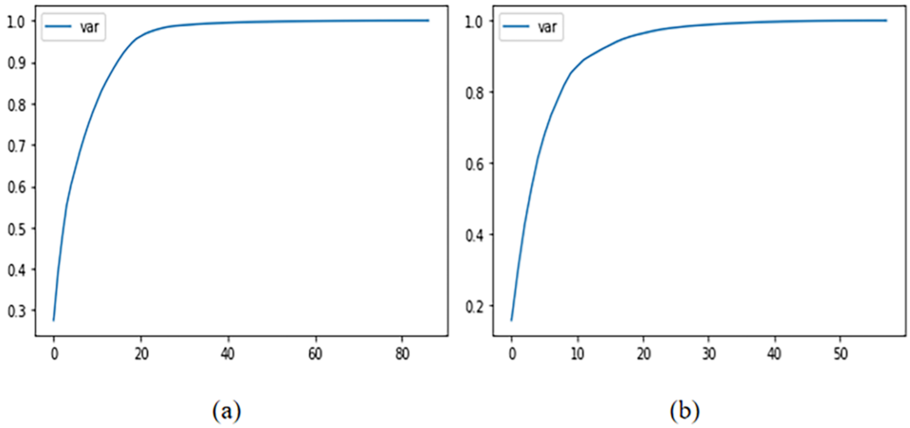

Figure 4. (a) PCA_MOS feature selection, (b) PCA_NDSF feature selection.

Figure 5(a)–(d) show the classification accuracy of the test set constructed from negative-positive samples applied to the two best protein feature extraction methods based on feature dimensionality and four classifiers. The MOS method combined with PCA shows an increasing and then decreasing trend in the accuracy of the model in the range of 0 to 45 dimensions. When it is reduced to 30 dimensions, the performance is superior. For the four classifiers, LightGBM is effective, and the accuracy rate of the best result is 96.16%. In the proposed NDSF method, the accuracy of the PCA-LightGBM model is 99.01% under the conditions of 35 dimensions, and the accuracy is generally higher than that of the MOS coding method. In the NDFS coding method, the difference between the four ensemble learning algorithms is only 4%. Feature extraction of protein sequence data can effectively retain enough information, remove redundant data, and reduce training time. Meanwhile, it obtained a data accuracy rate as high as 99.2%, which is a practical application value.

Using linear mapping and nonlinear mapping to extract features from coding sequences can remove redundant features and reduce training time. Meanwhile, it makes up for the shortage of NDSF coding time, which proves the practicability of the NDSF coding method. As a local linear representation of the global nonlinear coding method of LLE, it has been shown in many studies that it has something to do with keeping the local linear characteristics of the sample when it is reduced (keeping the original topology) [33]. Protein sequence data usually have linear and nonlinear characteristics. Among the NDSF method, the model dimension is 30, the performance is superior, and the accuracy of all models is not less than 75%. Protein interactions are determined by adjacent amino acids and stabilizing the functional interface of specific molecules, which may be the main reason why LLE classification results are lower than PCA dimensionality reduction results. If more features can be found, there will be more room for improvement for the performance of classification prediction. It is believed that an effective combination of computational and feature methods may lead to better prediction results.

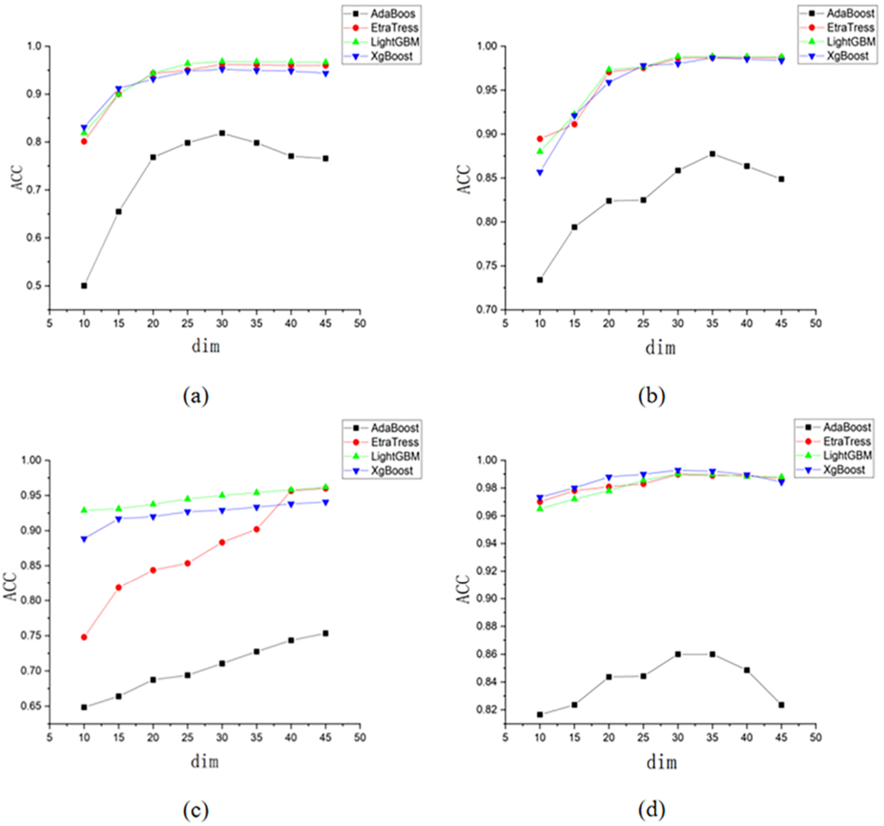

Figure 5. (a) Accuracy of LLE_MOS, (b) Accuracy of LLE_NDSF, (c) Accuracy of PCA_MOS, (d) Accuracy of PCA_NDSF.

3.4. Integrated learning

There are many types of base-based classifiers in machine learning, such as a support vector machine (SVM) algorithm and a K-nearest neighbor (KNN) algorithm. SVM has the following advantages: it can efficiently solve the classification problem of high-dimensional, nonlinear data, ultra-high-dimensional text classification problem, and can perform machine learning for small samples of data [34]. KNN algorithm has the following advantages: easy to understand, high accuracy, insensitive to multiple outliers, and can be used for both regression and classification [35]. A strong classifier is generated by combining base classifiers according to a combination strategy, and the classification performance of the strong classifier is better than the classification performance of each base classifier that combines it. In this paper, to generate a better classification method, we will construct a multi-classifier integration model.

Table 4 summarizes the classification results of different dimensionality reduction methods in which MOS and NDSF are applied to ensemble learning algorithms. The accuracy of the NDSF-based model is improved with different classifiers. The evaluation metrics for each training and test set classification also validate that NDSF outperforms MOS. Similarly, the classification accuracy with either PCA or LLE treatment improves by an average of 4.82%, the recall rate improves by an average of 5.08%, and the loss rate decreases by an average of 4.54%.

When the encoded amino acid sequences contained LLE-processed or PCA-processed features, the accuracy was higher than that of a single encoded result combined directly with the classifier. The accuracy of each classifier was 87.74%, 98.71%, 98.85%, and 98.70%, respectively, when the NDSF feature descriptors processed by the LLE method were used for classification prediction. When the principal component analysis is used instead of the dimension reduction method, the accuracy rates are 86.00%, 98.98%, 99.01%, and 99.29%, respectively. Although the results of the PCA and LLE had advantages and disadvantages under each of the four classifiers, the overall feature extraction results are slightly improved.

Table 4. Comparison of the test set and a training set of integrated learning algorithm.

		AdaBoost		Etratress		LightGBM		XGBoost	
		Train	Test	Train	Test	Train	Test	Train	Test
MOS_LLE	loss	16.45%	17.43%	4.01%	4.77%	0.12%	1.22%	0.20%	4.18%
	acc	82.55%	81.86%	92.21%	96.23%	97.93%	96.81%	95.18%	95.19%
	recall	80.06%	80.00%	93.31%	96.71%	97.21%	96.86%	95.01%	94.92%
	MCC	0.7215	0.5975	0.8954	0.9314	0.9577	0.9365	0.9245	0.9264
	F1	0.7887	0.7899	0.9256	0.9585	0.9702	0.9599	0.9485	0.9315
NDSF_LLE	loss	15.14%	15.26%	11.45%	1.30%	0.12%	2.70%	0.22%	4.58%
	acc	87.86%	87.74%	95.43%	98.71%	98.05%	98.85%	98.74%	98.70%
	recall	86.93%	86.89%	99.92%	98.53%	94.62%	90.20%	98.81%	98.90%
	MCC	0.8479	0.8367	0.9287	0.9734	0.9764	0.9831	0.9733	0.972
	F1	0.8555	0.8513	0.9872	0.9707	0.9277	0.8757	0.9724	0.9713
MOS_PCA	loss	23.61%	24.67%	0.06%	3.91%	1.45%	3.81%	2.85%	5.68%
	acc	76.39%	75.33%	99.93%	96.01%	98.60%	96.17%	97.30%	94.08%
	recall	75.67%	74.66%	99.12%	95.04%	98.22%	95.01%	96.12%	93.99%
	MCC	0.5671	0.5109	0.9824	0.9311	0.9712	0.9285	0.9621	0.9203
	F1	0.7346	0.7398	0.9874	0.9441	0.9753	0.9345	0.9534	0.9212
NDSF_PCA	loss	14.01%	14.78%	11.45%	1.32%	0.30%	0.90%	0.91%	1.19%
	acc	87.00%	86.00%	99.01%	98.98%	99.80%	99.01%	99.32%	99.29%
	recall	87.13%	87.15%	98.77%	98.73%	99.70%	98.97%	99.10%	98.54%
	MCC	0.8244	0.8094	0.9877	0.9746	0.9824	0.9878	0.9801	0.9781
	F1	0.8556	0.8574	0.9742	0.9813	0.9914	0.9809	0.9874	0.9722

Among comprehensive learning methods, this study focuses on the bagging and boosting methods. Bagging uses the same basic classifier, which is very sensitive to training sample data and suitable for parallel learning of multiple basic classifiers. Although it is a simple and effective integrated learning method, its limitation lies in data duplication. In this way, many classifiers also have different errors, which leads to different results for the classifiers. Another boosting focus on learning difficult-to-classify samples can effectively improve prediction accuracy [36]. The bagging (AdaBoost, LightGBM, XGBoost) and boosting methods (Extra trees) are used in the integrated algorithm model of this research. The model has two common parameters-learning rate and N estimator-and cross-validation and grid search methods are used to optimize the parameters to find the best combination of parameters. By comparing the prediction models constructed by AdaBoost, Extratrees, LightGBM, and XGBoost, we further found that LightGBM and XGBoost have better prediction results, and the accuracy rate using these four integrated algorithms is higher than that of traditional machine learning algorithms, which are all above 75%, as shown in Figure 6.

AdaBoost is used for many training data samples and continues to train the model more precisely according to the feedback of the model during the training process and to continuously improve the classification model's prediction accuracy. As shown in Figure 6, Adaboost's classification results are not as ideal as the other three algorithms, and the accuracy rate is within 90%. The reason may be that Adaboost is an algorithm trained by combining multiple weak classifiers and has dependencies among the individual weak classifiers. For samples with unbalanced data, the algorithm will focus on learning a few classes of samples, so the possibility of misclassification of a few classes of samples increases.

Extra trees are classified directly using the random feature and random threshold values on the random feature [20]. The randomness of each sub-model (decision tree) in the extra tree becomes greater, so the variability between each sub-model (decision tree) becomes remarkable. When Extra trees are used as the classification model, the results are significantly improved, with an accuracy of 98.98%, a MCC of 0.9746, a F1 of 0.9813, and a recall rate of 98.73% using the NDSF_PCA model feature descriptors. Because every decision tree is highly random, the over-fitting of the entire model is suppressed.

Compared to the Extra trees model, the best accuracy of the LightGBM model is 0.2% higher than that of the Extra trees model, which shows that the model based on the gradient lifting algorithm improves the fitting degree of the basic classifier [37]. It is pointed out that LightGBM used the histogram algorithm, which takes up less memory and reduces the calculation cost. Therefore, it can improve the calculation speed. It proved that the time efficiency of ensemble learning execution is generally better than that of a single classifier.

XGBoost is mainly used to solve the problem of supervised learning. In order to achieve the best performance of the XGBoost prediction, it is necessary to first obtain the best parameter combination [38]. It is known empirically that the model may be challenging to perform grid search with parameters of high dimensionality. Because of this, this study uses a random search to achieve the best parameter settings, then follows the principle of taking smaller combinations of parameters at a time, and finally sets a reasonable range of parameter values to achieve the training of the model. The hyperparameter adjustment range of the Xgboost algorithm is shown in Table 5. After selecting the best dimensionality reduction method, the results of applying the four integrated learning methods to the test set constructed from the negative-positive samples were compared, as shown in Figure 6. The results of NDSF_PCA combined with the integrated learning algorithm were generally satisfactory.

Table 5. Hyperparameter range of the XGBoost model.

Hyperparameters	Description	Value field
m_d	Depth of tree	1, 2, 3, 4
e_{ta}	Learning rate	0.01~0.5 (at 0.01 intervals)
g_a	Minimum loss function descent value	0~1 (at 0.1 intervals)
c_b	Proportion of features randomly sampled while building the tree	0~1 (at 0.1 intervals)
m_w	Minimum leaf weights	1~10

The NDSF_PCA model based on the XGBoost classifier has the highest accuracy on the training set and the test set, with 99.32% and 99.29%, respectively, a loss rate of 0.91% and 1.19%, a recall rate of 99.10% and 98.54%, a MCC of 0.9801 and 0.9781, and a F1 of 0.9874 and 0.9721, respectively. The classification accuracy of the NDSF PCA model feature descriptors processed by AdaBoost was the lowest, with an accuracy of 87.00% and 86.00% on the training set and the test set, respectively, a MCC of 0.8244 and 0.8094, a F1 of 0.8556 and 0.8574, and a recall rate of 87.13% and 87.15% respectively. The research results were shown that Etratress and LightGBM were 98% to 99%, respectively, and the classification accuracy of the LightGBM classifier even reached 99.80% in the training set.

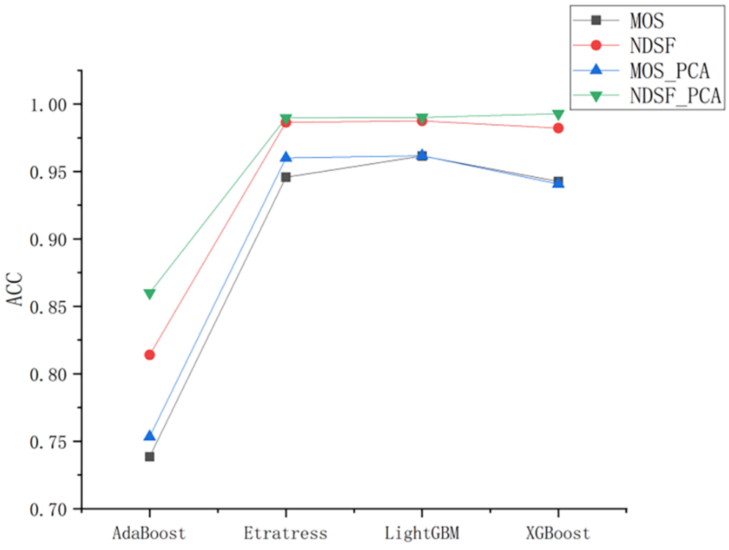

Figure 6. Comparison of methods.

In the NDS_PCA_XGBoost model, an accuracy of 99.2% was achieved with the best performance among all models. The best results were obtained on loss, recall, and acc. The results were shown that the gradient lifting algorithm was based on learning classification. The regression trees (CART) are used to calculate the complexity of each leaf node and minimize the loss in finding the best prediction score. It avoids the over-fitting of the learning model and effectively controls the complexity of the model, which improves the accuracy of the model, as shown in Table 6.

Table 6. Best results of the XGBoost model.

	loss	acc	recall	MCC	F1
NDSF_PCA	1.19%	99.29%	98.54%	0.9781	0.9721
NDSF_LLE	4.58%	98.70%	98.90%	0.9720	0.9713

In summary, we constructed prediction models based on the AdaBoost, Extra trees, LightGBM, and XGBoost algorithms for the human Proteogene dataset. The results showed that the integrated algorithm model with feature extraction significantly improved the predicted protein interactions after coding. In this study, we also found that the integrated learning approach achieves better prediction results for highly unbalanced data and effectively controls the complexity of the model because the algorithm avoids overfitting the learned model.

The prediction models constructed using the ensemble learning methods AdaBoost, Extra trees, LightGBM and XGBoost have better classification results. It can effectively identify positive and negative protein interaction effects, which proves that our modeling methods are effective and available. Protein sequence coding is widely used in chemistry and biology. Peptide-initiated N-substitutedN-carboxyanhydrides (NNCAs) polymerization provides a peptide synthesis mimic to increase its structural diversity and applications [39]. Protein sequence interactions can be used to detect serum histone G levels, which are antibody- and enzyme-independent [20]. Chiral recognition of essential amino acids provides good chiral splitting of essential amino acids for enantiomeric recognition of essential amino acids [40]. With the improvement of various algorithms, the research on coding methods has laid a good foundation for predicting protein interaction.

4. Conclusions

This study verifies the feasibility of a normalized differential sequence feature protein sequence encoding method, which is combined with an integrated learning algorithm to classify protein interactions. In this paper, we compare two dimensionality reduction methods, PCA and LLE, for protein sequence feature extraction of the encoded results. The results are shown that PCA combined with the proposed normalized differential sequence feature protein sequence encoding method can effectively retain sufficient information and remove redundant data, which greatly reduces the reduction loss and decrease the training time. In addition, we combine it with four integrated learning-based algorithms, AdaBoost, Extratrees, LightGBM, and XGBoost. For predictive classification, the XGBoost obtains better classification results to avoid overfitting to the learning model, and effectively controls the complexity of the model.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Availability of data and materials

The Gitee website provides access to the data and codes collected as part of the survey (https://gitee.com/mandy1023/ml).

Authors' contributions

Xiaoman Zhao: Data curation, Software, Investigation, methodology, Validation, Visualization, Writing original draft.Xue Wang: Conceptualization, Data curation, Formal analysis, Writing-review & editing, Funding acquisition, Project administration. Zhou Jin: Investigation, Methodology, Writing review & editing. Rujing Wang: Writing review & editing, Funding acquisition, Project administration.

Acknowledgments

Research on Protein Interaction Prediction Based on Deep Learning (YZJJ2021QN26) by the President of Hefei Research Institute, Chinese Academy of Sciences.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. C. Gustafsson, J. Minshull, S. Govindarajan, J. Ness, A. Villalobos, Engineering genes for predictable protein expression, *Protein Expression Purif.*, **83** (2012), 37–46. https://dx.doi.org/10.1016/j.pep.2012.02.013
2. L. Y. Mei, M. R. Montoya, G. M. Quanrud, M. Tran, A. Villa-Sharma, M. Huang, et al., Bait correlation improves interactor identification by tandem mass tag-affinity Purification-Mass spectrometry, *J. Proteome Res.*, **19** (2020), 1565–1573. https://dx.doi.org/10.1021/acs.jproteome.9b00825
3. I. Paspaltsis, E. Kesidou, O. Touloumi, R. Lagoudaki, M. Boziki, M. Samiotaki, et al., Application of antibody phage display to identify potential antigenic neural precursor cell proteins, *J. Biol. Res. Thessaloniki*, **27** (2020). https://dx.doi.org/10.1186/s40709-020-00123-4
4. A. Rami, M. Behdani, N. Yardehnavi, M. Habibi-Anbouhi, F. Kazemi-Lomedasht, An overview on application of phage display technique in immunological studies, *Asian Pac. J. Trop. Biomed.*, **7** (2017), 599–602. https://dx.doi.org/10.1016/j.apjtb.2017.06.001
5. S. Schuette, B. Piatkowski, A. Corley, D. Lang, M. Geisler, Predicted protein-protein interactions in the moss Physcomitrella patens: a new bioinformatic resource, *BMC Bioinf.*, **16** (2015). https://dx.doi.org/10.1186/s12859-015-0524-1
6. L. L. Song, S. B. Ning, J. X. Hou, Y. Zhao, Performance of protein-ligand docking with CDK4/6 inhibitors: a case study, *Math. Biosci. Eng.*, **18** (2020), 456–470. https://dx.doi.org/10.3934/mbe.2021025
7. Y. C. Wang, J. G. Wang, Z. X. Yang, N. Deng, Sequence-based protein-protein interaction prediction via support vector machine, *J. Syst. Sci. Complexity*, **23** (2010), 1012–1023. https://dx.doi.org/10.1007/s11424-010-0214-z
8. L. Yang, X. D. Zhao, X. L. Tang, Predicting disease-related proteins based on clique backbone in protein-protein interaction network, *Int. J. Biol. Sci.*, **10** (2014), 677–688. https://dx.doi.org/10.7150/ijbs.8430

9. H. P. Zhang, L. B. Liao, K. M. Saravanan, P. Yin, Y. Wei, DeepBindRG: a deep learning based method for estimating effective protein-ligand affinity, *PeerJ*, **7** (2019). https://dx.doi.org/10.7717/peerj.7362

10. X. Y. Zhou, I. Naguro, H. Ichijo, K. Watanabe, Mitogen-activated protein kinases as key players in osmotic stress signaling, *Biochim. Biophys. Acta Gen. Subj.*, **1860** (2016), 2037–2052. https://dx.doi.org/10.1016/j.bbagen.2016.05.032

11. Y. Z. Zhou, Y. Gao, Y. Y. Zheng, Prediction of protein-protein interactions using local description of amino acid sequence, in *Advances in Computer Science and Education Applications*, Springer, 2011. https://doi.org/10.1007/978-3-642-22456-0_37

12. Y. H. Zhu, X. R. Zhang, S. J. Xie, W. Bao, J. Chen, Q. Wu, et al., Oxidative phosphorylation regulates interleukin-10 production in regulatory B cells via the extracellular signal-related kinase pathway, *Immunology*, **167** (2022), 576–589. https://dx.doi.org/10.1111/imm.13554

13. X. Cao, G. X. Yu, W. Ren, M. Guo, J. Wang, DualWMDR: Detecting epistatic interaction with dual screening and multifactor dimensionality reduction, *Hum. Mutat.*, **41** (2020), 719–734. https://dx.doi.org/10.1002/humu.23951

14. P. Malvi, R. Janostiak, S. Chava, P. Manrai, E. Yoon, K. Singh, et al., LIMK2 promotes the metastatic progression of triple-negative breast cancer by activating SRPK1, *Oncogenesis*, **9** (2020). https://dx.doi.org/10.1038/s41389-020-00263-1

15. Y. M. Wu, M. Zhou, K. Chen, S. Chen, X. Xiao, Z. Ji, et al., Alkali-metal hexamethyldisilazide initiated polymerization on alpha-amino acid N-substituted N-carboxyanhydrides for facile polypeptoid synthesis, *Chin. Chem. Lett.*, **32** (2021), 1675–1678. https://dx.doi.org/10.1016/j.cclet.2021.02.039

16. W. Zhang, X. L. Xue, C. W. Xie, Y. Li, J. Liu, H. Chen, et al., CEGSO: Boosting essential proteins prediction by integrating protein complex, gene expression, gene ontology, subcellular localization and Orthology information, *Interdiscip. Sci.-Comput. Life Sci.*, **13** (2021), 349–361. https://dx.doi.org/10.1007/s12539-021-00426-7

17. Y. N. Shen, Y. J. Ding, J. J. Tang, Q. Zou, F. Guo, Critical evaluation of web-based prediction tools for human protein subcellular localization, *Briefings Bioinf.* **21** (2020), 1628–1640. https://dx.doi.org/10.1093/bib/bbz106

18. T. Z. Yu, W. S. Zhang, Semisupervised multilabel learning with joint dimensionality reduction, *IEEE Signal Process Lett.*, **23** (2016), 795–799. https://dx.doi.org/10.1109/lsp.2016.2554361

19. C. Chen, Q. M. Zhang, Q. Ma, B. Yu, LightGBM-PPI: Predicting protein-protein interactions through LightGBM with multi-information fusion, *Chemom. Intell. Lab. Syst.*, **191** (2019), 54–64. https://dx.doi.org/10.1016/j.chemolab.2019.06.003

20. P. P. Hao, H. Li, L. Zhou, H. Sun, J. Han, Z. Zhang, Serum metal ion-induced cross-linking of photoelectrochemical peptides and circulating proteins for evaluating cardiac ischemia/reperfusion, *ACS Sens.*, **7** (2022), 775–783. https://dx.doi.org/10.1021/acssensors.1c02305

21. D. J. W. Tay, Z. Z. R. Lew, J. J. H. Chu, K. S. Tan, Uncovering novel viral innate immune evasion strategies: What has SARS-CoV-2 taught us, *Front. Microbiol.*, **13** (2022). https://dx.doi.org/10.3389/fmicb.2022.844447

22. K. Y. Huang, Q. H. Fang, W. M. Sun, S. He, Q. Yao, J. Xie, et al., Cucurbit[n]uril supramolecular assemblies-regulated charge transfer for luminescence switching of gold nanoclusters, *J. Phys. Chem. Lett.*, **13** (2022), 419–426. https://dx.doi.org/10.1021/acs.jpclett.1c03917

23. Z. Y. Wu, H. Yin, H. He, Y. Li, Dynamic-LSTM hybrid models to improve seasonal drought predictions over China, *J. Hydrol.*, **615** (2022). https://dx.doi.org/10.1016/j.jhydrol.2022.128706
24. C. G. Yan, L. X. Meng, L. Li, J. Zhang, Z. Wang, J. Yin, et al., Age-invariant face recognition by multi-feature fusion and decomposition with self-attention, *ACM Trans. Multimedia Comput. Commun. Appl.*, **18** (2022). https://dx.doi.org/10.1145/3472810
25. W. Wang, D. S. Tekcham, M. Yan, Z. Wang, H. Qi, X. Liu, et al., Biochemical reactions in metabolite-protein interaction, *Chin. Chem. Lett.*, **29** (2018), 645–647. https://dx.doi.org/10.1016/j.cclet.2017.10.002
26. Y. D. Liang, R. F. Sun, L. J. Li, F. Yuan, W. Liang, L. Wang, et al., A functional polymorphism in the promoter of MiR-143/145 is associated with the risk of cervical squamous cell carcinoma in Chinese women a case-control study, *Medicine*, **94** (2015). https://dx.doi.org/10.1097/MD.0000000000001289
27. M. Braaksma, E. S. Martens-Uzunova, P. J. Punt, P. J. Schaap, An inventory of the Aspergillus niger secretome by combining in silico predictions with shotgun proteomics data, *BMC Genomics*, **11** (2010). https://dx.doi.org/10.1186/1471-2164-11-584
28. P. Walther, A. Krauss, S. Naumann, Lewis pair polymerization of epoxides via zwitterionic species as a route to High-Molar-Mass polyethers, *Angew. Chem. Int. Ed.*, **58** (2019), 10737–10741. https://dx.doi.org/10.1002/anie.201904806
29. Y. M. Wu, D. F. Zhang, P. C. Ma, R. Zhou, L. Hua, R. Liu, Lithium hexamethyldisilazide initiated superfast ring opening polymerization of alpha-amino acid N-carboxyanhydrides, *Nat. Commun.*, **9** (2018). https://dx.doi.org/10.1038/s41467-018-07711-y
30. C. H. Xin, X. F. Ban, Z. B. Gu, C. Li, L. Cheng, Y. Hong, et al., Non-classical secretion of 1,4-alpha-glucan branching enzymes without signal peptides in Escherichia coli, *Int. J. Biol. Macromol.*, **132** (2019), 759–765. https://dx.doi.org/10.1016/j.ijbiomac.2019.04.002
31. Y. J. Zhang, S. Yu, R. P. Xie, J. Li, A. Leier, T. Marquez-Lago, et al., PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins, *Bioinformatics*, **36** (2020), 704–712. https://dx.doi.org/10.1093/bioinformatics/btz629
32. C. J. Fee, J. A. Van, Alstine PEG-proteins: Reaction engineering and separation issues, *Chem. Eng. Sci.*, **61** (2006), 924–939. https://dx.doi.org/10.1016/j.ces.2005.04.040
33. C. H. Hung, H. L. Huang, K. T. Hsu, S. J. Ho, S. Y. Ho, Prediction of non-classical secreted proteins using informative physicochemical properties, *Interdiscip. Sci.: Comput. Life Sci.*, **2** (2010), 263–270. https://dx.doi.org/10.1007/s12539-010-0023-z
34. A. X. Wang, S. S. Chukova, B. P. Nguyen, Ensemble k-nearest neighbors based on centroid displacement, *Inf. Sci.*, **629** (2023), 313–323. https://dx.doi.org/10.1016/j.ins.2023.02.004
35. B. P. Nguyen, W. L. Tay, C. K. Chui, Robust biometric recognition from palm depth images for gloved hands, *IEEE Trans. Hum.-Mach. Syst.*, **45** (2015), 799–804. https://dx.doi.org/10.1109/THMS.2015.2453203
36. T. Wang, W. Wang, H. Liu, T. Li, Research on a face real-time tracking algorithm based on particle filter multi-feature fusion, *Sensors*, **19** (2019). https://dx.doi.org/10.3390/s19051245
37. H. J. Tao, X. B. Lu, Smoke vehicle detection based on multi-feature fusion and hidden Markov model, *J. Real-Time Image Process.*, **17** (2020), 745–758. https://dx.doi.org/10.1007/s11554-019-00856-z

38. A. Berg, O. Kukharenko, M. Scheffner, C. Peter, Towards a molecular basis of ubiquitin signaling: A dual-scale simulation study of ubiquitin dimers, *PLOS Comput. Biol.*, **14** (2018). https://dx.doi.org/10.1371/journal.pcbi.1006589
39. V. J. Jameson, T. Luke, Y. T. Yan, A. Hind, M. Evrard, K. Man, et al., Unlocking autofluorescence in the era of full spectrum analysis: Implications for immunophenotype discovery projects, *Cytometry Part A*, **101** (2022), 922–941. https://dx.doi.org/10.1002/cyto.a.24555
40. J. J. Zhang, S. Y. Wang, P. Zhang, S. Fan, H. Dai, Y. Xiao, et al., Engineering a cationic supramolecular charge switch for facile amino acids enantiodiscrimination based on extended-gate field effect transistors, *Chin. Chem. Lett.*, **33** (2022), 3873–3878. https://dx.doi.org/10.1016/j.cclet.2021.11.081