

SCIENTIFIC REPORTS



OPEN

Predicting Protein-Protein Interactions from Matrix-Based Protein Sequence Using Convolution Neural Network and Feature-Selective Rotation Forest

Lei Wang^{1,2}, Hai-Feng Wang¹, San-Rong Liu¹, Xin Yan³ & Ke-Jian Song⁴

Protein is an essential component of the living organism. The prediction of protein-protein interactions (PPIs) has important implications for understanding the behavioral processes of life, preventing diseases, and developing new drugs. Although the development of high-throughput technology makes it possible to identify PPIs in large-scale biological experiments, it restricts the extensive use of experimental methods due to the constraints of time, cost, false positive rate and other conditions. Therefore, there is an urgent need for computational methods as a supplement to experimental methods to predict PPIs rapidly and accurately. In this paper, we propose a novel approach, namely CNN-FSRF, for predicting PPIs based on protein sequence by combining deep learning Convolution Neural Network (CNN) with Feature-Selective Rotation Forest (FSRF). The proposed method firstly converts the protein sequence into the Position-Specific Scoring Matrix (PSSM) containing biological evolution information, then uses CNN to objectively and efficiently extracts the deeply hidden features of the protein, and finally removes the redundant noise information by FSRF and gives the accurate prediction results. When performed on the PPIs datasets *Yeast* and *Helicobacter pylori*, CNN-FSRF achieved a prediction accuracy of 97.75% and 88.96%. To further evaluate the prediction performance, we compared CNN-FSRF with SVM and other existing methods. In addition, we also verified the performance of CNN-FSRF on independent datasets. Excellent experimental results indicate that CNN-FSRF can be used as a useful complement to biological experiments to identify protein interactions.

Protein is the essential component of the living organism, and it participates in various processes of life activities such as metabolism, signal transduction, hormone regulation, DNA transcription and replication^{1,2}. In general, proteins perform their functions in the form of complexes by interacting with other proteins. Studying protein-protein interactions (PPIs) not only help to understand the life process, but also help to explore the pathogenesis of disease and pursue drug targets³. Over the past several decades, the detection methods of protein interaction based on biological experiments, such as tandem affinity purification (TAP)⁴, yeast two-hybrid (Y2H)^{5,6} and mass spectrometric protein complex identification⁷, gradually matured and achieved considerable research results.

However, due to the biological experiment methods are costly and time-consuming, the protein interaction detected by experimental methods can only account for a small part of the complete PPIs networks⁸⁻¹¹. In addition, the detection results are also susceptible to the experimental environment and operational processes, resulting in some false positives and false negatives. Therefore, developing reliable computational methods to predict protein interactions accurately is of great practical significance.

¹College of Information Science and Engineering, Zaozhuang University, Zaozhuang, Shandong, 277100, P.R. China. ²Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi, 830011, P.R. China. ³School of Foreign Languages, Zaozhuang University, Zaozhuang, Shandong, 277100, P.R. China. ⁴School of information engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, 341000, P.R. China. Correspondence and requests for materials should be addressed to L.W. (email: leiwang@ms.xjbc.ac.cn) or X.Y. (email: xinyanuzz@gmail.com)

In fact, there are many computational methods that have been proposed as complementary to experimental methods to predict protein-protein interactions^{12–15}. These methods typically use binary classification model to describe protein-protein pairs with or without interaction, which can be roughly divided into the following categories: protein domains, gene expression, gene neighborhood, protein structure information^{16,17}, literature mining knowledge¹⁸, and phylogenetic profiles^{19,20}. However, if there is no corresponding pre-knowledge, these methods cannot be implemented^{21,22}.

With the rapid development of sequencing technology, protein sequence information is collected and stored in large quantities. There is abundant useful information in the protein sequence, and the experimental results show that using amino acid sequence alone is sufficient to predict the interaction of protein accurately. Therefore, protein interaction prediction methods that directly extract information from amino acid sequences have aroused great interest in recent years^{23–25}. You *et al.* proposed the method of protein interaction prediction based on Support Vector Machine (SVM), considering the sequence order and the dipeptide information of the primary protein sequence. This method has achieved 90.06% accuracy, 94.37% specificity and 85.74% sensitivity in the protein *Yeast* dataset²⁶. Hu *et al.* introduced a novel co-evolutionary feature extraction method, namely CoFex, to predict protein interactions. CoFex can extract the feature vectors that accurately express the protein properties according to the presence or absence of the co-evolutionary features of the two protein sequences, thereby providing the performance of the PPIs prediction²⁷. Pan *et al.* proposed a new hierarchical LDA-RF model to directly predict protein-protein interactions in the primary protein sequences, which can mine hidden internal structures buried into the noisy amino acid sequences in low-dimensional latent semantic space. The experimental results show that this model can effectively predict potential protein interactions⁹. Saha *et al.* constructed an ensemble model for protein interaction prediction based on a majority voting method. The model uses four well-established machine learning methods: support vector machines, random forests, decision trees, and naive Bayes. In the cross-validation experiment, the ensemble learning method achieved over 80% sensitivity and 90% prediction accuracy²⁸. Jeong *et al.* predict protein interactions using algorithms that extract features only from protein sequences and machine learning for computational function prediction. The experimental results show that these features derived from the position-specific scoring matrix are very suitable for protein interaction prediction²⁹.

In this study, we propose a novel sequence-based approach, namely CNN-FSRF, to predict potential protein interactions using deep learning Convolutional Neural Network (CNN) algorithm combined with Feature-Selective Rotation Forest (FSRF) classifier. More specifically, we first use the position-specific scoring matrix to convert each protein alphabet sequence into the numerically matrix-based protein descriptor that contains evolution information. Then we use the convolutional neural network to extract the high-level abstraction features of the protein automatically and objectively. Finally, these features are fed into the feature-selective rotation forest classifier to get the final prediction results. To evaluate the predictive performance of CNN-FSRF, we performed verification in the *Yeast* and *Helicobacter pylori* PPI datasets, respectively. The experimental results show that CNN-FSRF achieves 97.75% and 88.96% accuracy with 99.61% and 91.86% sensitivity at the specificity of 95.89% and 86.11% in the above datasets, respectively. Excellent results indicate that CNN-FSRF can be a useful complement to biological experiments to identify potential protein-protein interactions.

Materials and Methodology

In this section, we outline the main idea behind CNN-FSRF approach. Figure 1 gives a schematic diagram of how CNN-FSRF uses convolution neural network and feature-selective rotation forest classifier to predict protein-protein interactions. As can be seen from the figure, our model can be divided into three steps. The first is matrix-based protein numerical representation. For a given protein, since its sequence is usually represented by the letter symbol of 20 kinds of amino acids, in order to facilitate computer algorithm processing, we use the Position-Specific Scoring Matrix (PSSM) method to convert the letter sequence of the protein into the numerical matrix. The second is feature extraction based on Convolutional Neural Network (CNN). Although the protein sequence contains abundant information, it also mixed with a lot of noise. In order to get a more precise representation, we use the deep learning CNN algorithm to extract its features. CNN can automatically and objectively extract the advanced features of protein information in a layer-by-layer manner, thus effectively avoiding the interference of human factors. The finally is the PPI prediction based on Feature-Selective Rotation Forest (FSRF) classifier. After obtaining the advanced features of the protein, we used FSRF classifier to predict relationship between them. The FSRF classifier has the advantage of greatly improving the classification speed under the premise of guaranteeing the accuracy, so as to quickly and effectively predicts the interaction between proteins.

Golden standard datasets. We evaluate the CNN-FSRF approach through two real PPIs datasets. The *Yeast* dataset collected from the *Saccharomyces cerevisiae* core subset of the Database of Interacting Proteins (DIP) by Guo *et al.*³⁰. The core subset contains a total of 5966 interacting protein pairs. After we remove protein pairs containing less than 50 residues or more than 40% sequence identity protein, the remaining 5594 protein pairs constitute the golden standard positive data set. For the standard negative data set, we constructed based on the assumption of Guo *et al.*³⁰ that there is no interaction between proteins in different subcellular compartments. To avoid the occurrence of imbalanced dataset, we selected the same number of protein pairs as the positive dataset to construct the negative dataset. As a result, there is a total of 11188 protein pairs in the final *Yeast* dataset, with positive and negative samples each accounting for half. For the *Helicobacter pylori* PPIs dataset from Martin *et al.*¹², we use the same method for processing. The final *Helicobacter pylori* dataset consisted of 2916 protein pairs, of which 1458 interacted pairs and 1458 non-interacted pairs.

Evaluation criteria. To evaluate the performance of CNN-FSRF, we use the 5-fold cross-validation and several general evaluation criteria in our experiments. The 5-fold cross-validation randomly divides whole dataset into five independent subsets of the same size. Each time one subset is used as the test set, and the remaining four

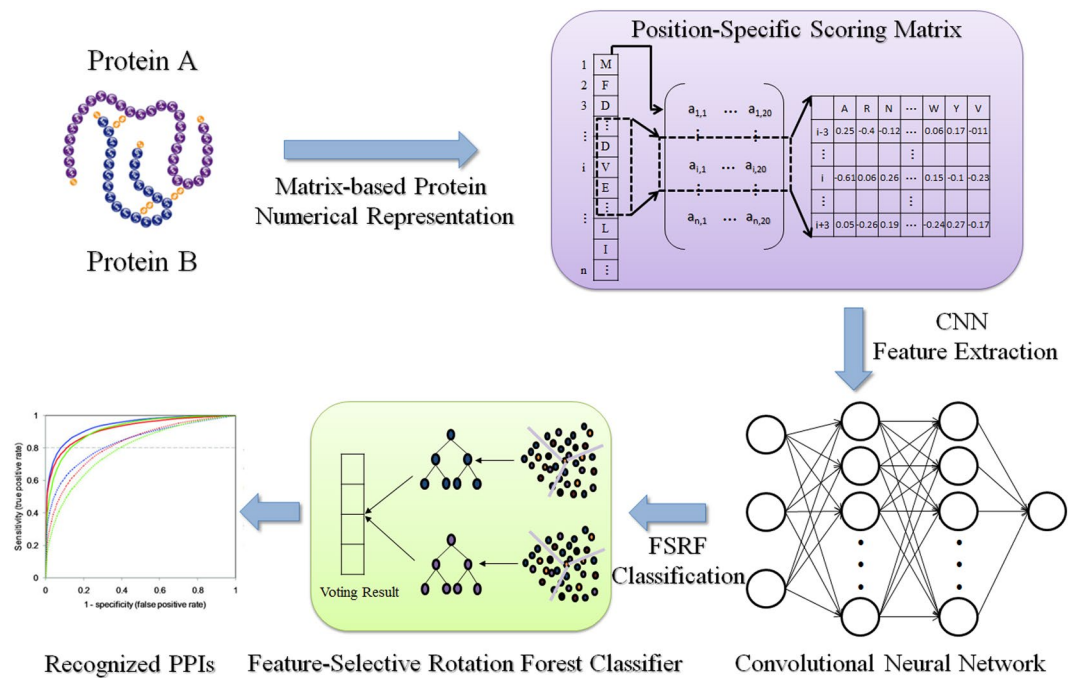


Figure 1. The schematic diagram for predicting protein-protein interactions by integrating convolutional neural network with feature-selective rotation forest model.

subsets are used as the training sets. In the experiment, this process is executed five times to ensure that each subset is used as the test set once. Finally, the average and standard deviation of these five experiments are taken as the final experimental results. We follow the widely used evaluation criteria to evaluate the model, including accuracy (Accu.), sensitivity (Sen.), specificity (Spec.), precision (Prec.), F-Score (F_{score}), and Matthews Correlation Coefficient (MCC). They are defined as:

$$Accu. = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Sen. = \frac{TP}{TP + FN} \quad (2)$$

$$Spec. = \frac{TN}{TN + FP} \quad (3)$$

$$Prec. = \frac{TP}{TP + FP} \quad (4)$$

$$F_{score} = 2 \times \frac{Sen. \times Prec.}{Sen. + Prec.} \quad (5)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (6)$$

where TP indicates the number of positive samples that are correctly identified, TN indicates the number of negative samples that are correctly identified, FP indicates the number of positive samples that are incorrectly identified, and FN indicates the number of negative samples that are incorrectly identified.

In these evaluation criteria, the accuracy reflects the proportion of the correct prediction results of the model. Sensitivity reflects the ability of classification model to identify positive samples. The higher value of sensitivity indicates that the model has a stronger ability to identify positive samples. Precision reflects the ability of classification model to discriminate negative samples. The higher value of precision indicates that the model has a stronger ability to discriminate negative samples. F_{score} is a combination of sensitivity and precision. The higher value of F_{score} indicates that the model is more robust. The Matthew correlation coefficient (MCC) reflects the correlation between the prediction results and the observation results. It is an important indicator of the overall performance of the model. The larger value of MCC indicates that the model has a better performance. In

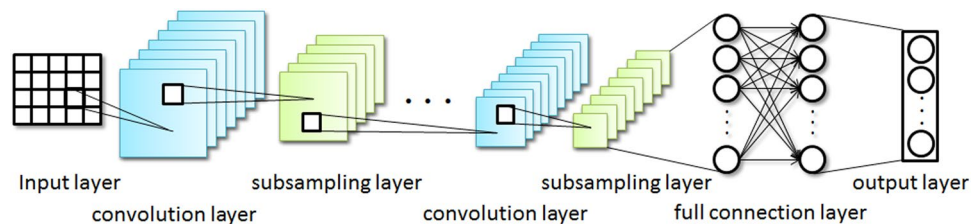


Figure 2. Convolution Neural Network Structure Diagram.

addition, Receiver Operating Characteristic (ROC) curves and Precision-Recall (P-R) curves are also drawn as evaluation criteria. In order to directly measure the quality of the results expressed by the ROC curve, the Area Under a Curve (AUC) is calculated at the same time. Its value ranges from 0 to 1 and the larger the value, the better the performance of the model.

Matrix-based protein numerical representation. Protein sequences are usually stored in the database in the form of letters. In order to facilitate the deep learning algorithm to extract its hidden features, the protein sequence must be encoded into the numerical form. In this study, we use the Position-Specific Scoring Matrix (PSSM) method that can contain biological evolution information to generate matrix-based numeric descriptors^{31,32}. When measuring the matching weights of amino acids, PSSM not only records the importance and relevance of matching, but also records the position of amino acid residues in the sequence. This matrix helps to reveal more evolutionary information of protein sequences and is therefore widely used in many fields of bioinformatics.

PSSM is the matrix of N row of 20 columns, where the row represents the length of the protein sequence and the column represents the 20 naive amino acids. Assume that $P = \{r_{i,j}; i = 1 \cdots N \text{ and } j = 1 \cdots 20\}$, PSSM can be expressed as:

$$P = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,20} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,20} \end{bmatrix} \quad (7)$$

where $r_{i,j}$ in the i row of PSSM mean that the probability of the i th residue being mutated into type j of 20 native amino acids during the procession of evolutionary in the protein from multiple sequence alignments.

In the experiment, we use the sequence comparison tool Position-Specific Iterated BLAST (PSI-BLAST) to obtain the PSSM matrix. BLAST is an effective tool for finding locally similar regions between sequences. It is able to compare nucleotide or protein sequences to sequence databases, and calculate the statistical significance of matches, so as to infer the functional and evolutionary relationships between sequences as well as help identify gene family members. PSI-BLAST is a more sensitive BLAST program that can effectively detect new members of protein families and similar proteins in distantly related species. The feature of PSI-BLAST is that it can use the profile to search the database, re-construct the profile with the results of the search, and then search the database again with the new profile, so repeatedly until no new results are produced. PSI-BLAST naturally extends the BLAST method to find hidden patterns in protein sequences and to find many related proteins with a large sequence difference and a similar structural function. To maximize the effectiveness of the algorithm, we use the non-redundant *SwissProt* as the alignment database. All sequence entries in the *SwissProt* database are searched by experienced protein chemists and molecular biologists for consulting the relevant literature and carefully checking through computer tools. In addition, we also set the expected threshold of the PSI-BLAST algorithm to 0.001, the number of iterations to 3, and the rest of the parameters to the default values.

Convolutional neural network. Deep learning belongs to a branch of machine learning. Its motivation lies in establishing and simulating the neural network of the human brain for learning, and interpreting data in a mechanism that imitates the human brain³³⁻³⁵. Deep learning can form an abstract high-level representation by combining low-level features to discover the rules of data. Therefore, in this paper, we use deep learning convolutional neural network algorithm to extract hidden useful information in protein.

The convolution neural network is a feed-forward neural network. Its neurons can respond to the surrounding units in a part of the coverage and have excellent performance for data feature extraction³⁶. CNN uses forward propagation to calculate the output value and back propagation to adjust weights and biases. CNN is composed of the input layer, the convolution layer, subsampling layer, full connection layer and the output layer. Its structure diagram is shown in Fig. 2.

Assuming that L_i represents the feature map of the i th layer, it can be described as:

$$L_i = h(L_{i-1} \circ W_i + b_i) \quad (8)$$

where W_i means the weight matrix of the convolution kernel of i th layer, b_i means the offset vector, $h(x)$ means the activation function and operator \circ means convolution operations. The subsampling layer usually behind the convolutional layer and the feature map is sampled according to given rules. Assuming that L_i is a subsampling layer, its sampling formula is:

$$L_i = \text{subsampling}(L_{i-1}) \quad (9)$$

Through multiple convolution and sub sampling operations, CNN classifies the extracted features by the fully connected layer, and the probability distribution \mathcal{F} is obtained based on input. The core mathematical idea of CNN is to map the input matrix L_o to a new feature representation \mathcal{F} through multi-layer data transformation.

$$\mathcal{F}(i) = \text{Map}(C = c_i | L_o; (W, b)) \quad (10)$$

where c_i represents the i th label class, L_o denotes the input matrix, and \mathcal{F} denotes the feature expression.

The goal of CNN training is to minimize the network loss function $F(W, b)$. At the same time, to alleviate the over fitting problem, the final loss function $E(W, b)$ is usually controlled by a norm, and the intensity of the over fitting is controlled by the parameter ε .

$$E(W, b) = F(W, b) + \frac{\varepsilon}{2} W^T W \quad (11)$$

When adjusting parameters, CNN usually uses gradient descent method to optimize, update network parameters (W, b) layer by layer from back to front, and use learning rate λ to control the strength of back propagation.

$$W_i = W_i - \lambda \frac{\partial E(W, b)}{\partial W_i} \quad (12)$$

$$b_i = b_i - \lambda \frac{\partial E(W, b)}{\partial b_i} \quad (13)$$

Feature-selective rotation forest. The Rotation Forest (RF) is an ensemble classifier which contains multiple decision trees. It can quickly be applied to many data science problems and can efficiently obtain accurate classification results³⁷. Therefore, it has received high attention and popularity from researchers. The main idea of RF is to randomly divide the data set into multiple subsets and implement the corresponding coordinate transformation, and transform the data from the original space to the new space to increase the difference between the data, so as to improve the diversity and accuracy of the classifier at the same time.

In this study, aiming at the high dimensionality and noise-containing characteristics of the PPIs data, we improved the RF and proposed Feature-Selective Rotation Forest (FSRF) algorithm. The FSRF algorithm can effectively reduce the data dimension and remove the noise information in the data, thus improving the prediction accuracy and speed of the classifier. More specifically, we use the χ^2 method in statistics to calculate the weight of all the features, and rank them according to the weighted values, and delete the small influence on the classification according to the given feature selection rate. The weight of a given feature P can be calculated according to the following formula.

$$\chi^2 = \sum_{i=1}^l \sum_{j=1}^2 \frac{(\rho_{ij} - \sigma_{i,j})^2}{\sigma_{i,j}} \quad (14)$$

where l is the number of values in feature P , ρ_{ij} is the count of the value β_i in feature P belongs to class y_j , defined as:

$$\rho_{ij} = \text{count}(P = \beta_i \text{ and } Y = y_j) \quad (15)$$

$\sigma_{i,j}$ is the expected value of β_i and y_j , defined as:

$$\sigma_{i,j} = \frac{\text{count}(P = \beta_i) \times \text{count}(Y = y_j)}{L} \quad (16)$$

where $\text{count}(P = \beta_i)$ is the number of samples with the value β_i in the feature P , $\text{count}(Y = y_j)$ is the number of samples with the value y_j in the class Y , and L is the total number of samples in the training set.

After calculating the weights of all the features by formula 14, we remove the features with small weight value according to the given weight selection rate ε , and thus obtain a new feature set S . Let $E = (e_1, e_2, \dots, e_n)^T$ be an $n \times L$ matrix which is composed of n observation feature vector for each training sample and $C = (c_1, c_2, \dots, c_n)^T$ denote the corresponding labels. Therefore, the data sample can be represented as $\{e_i, c_i\}$, where $e_i = (e_{i1}, e_{i2}, \dots, e_{iL})$ is an L -dimensional feature vector. According to the number K of given decision trees, the sample set is randomly divided into a subset of the same size and transformed by principal component analysis (PCA) algorithm. Then all coefficients of the principal component are rearranged and stored to form a rotation matrix to change the original training set. Therefore, the decision tree can be represented by T_1, T_2, \dots, T_k , and the training process of one decision tree T_i can be described as follows:

- The sample set S is randomly divided into K (a factor of n) disjoint subsets, and each subset contains the number of features is n/k .
- A corresponding column of features in the subset S_{ij} is selected to form a new matrix E_{ij} from the training dataset E . A new training set $E'_{i,j}$ which is extracted from E_{ij} randomly with $3/4$ of the dataset using

Test set	Accu.(%)	Sen.(%)	Spec.(%)	Prec.(%)	Fscore(%)	MCC(%)	AUC(%)
1	97.36	99.73	95.04	95.18	97.40	94.83	96.97
2	98.17	99.82	96.55	96.59	98.18	96.39	97.92
3	97.45	99.73	95.19	95.36	97.50	95.00	97.17
4	97.27	99.29	95.22	95.48	97.35	94.62	97.13
5	98.48	99.47	97.46	97.58	98.52	96.98	98.52
Average	97.75	99.61	95.89	96.04	97.79	95.57	97.54
Standard Deviation	0.54	0.22	1.07	1.02	0.53	1.05	0.66

Table 1. The 5-fold cross-validation results were generated on the *Yeast* dataset by using the CNN-FSRF method.

- bootstrap algorithm. Loop K times in this way, so that each subset is converted
- Matrix $E'_{i,j}$ is used as the feature transform by PCA technique for producing the coefficient matrix $M_{i,j}$, which j th column coefficient as the characteristic component j th.
 - A sparse rotation matrix R_i is constructed, and its coefficients which obtained from the matrix $M_{i,j}$ expressed as follows:

$$R_i = \begin{bmatrix} \mu_{i,1}^{(1)}, \dots, \mu_{i,1}^{(G_1)} & 0 & \dots & 0 \\ 0 & \mu_{i,2}^{(1)}, \dots, \mu_{i,2}^{(G_2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mu_{i,k}^{(1)}, \dots, \mu_{i,k}^{(G_k)} \end{bmatrix} \quad (17)$$

In the prediction period, provided the test sample e , generated by the classifier T_i of $d_{i,j}(ER_i^t)$ to determine e belongs to class c_i . And then the class of confidence is calculated by means of the average combination, and the formula is as follows:

$$\theta_j(e) = \frac{1}{k} \sum_{i=1}^k d_{i,j}(ER_i^t) \quad (18)$$

Therefore, the test sample e easily assigned to the classes with the greatest possible.

Results and Discussion

In this section, we summarize the experimental results of the CNN-FSRF method on the standard datasets. To comprehensively evaluate the performance of the model, we compare the proposed method with the state-of-the-art Support Vector Machine (SVM) classifier and other excellent methods on the same datasets. In addition, we verified the proposed method on independent datasets. The CNN-FSRF based on protein sequence is implemented by MATLAB platform. For the SVM classifier, we use the LIBSVM implementation designed by Lin *et al.*, which can be downloaded at <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>. The parameters of FSRF and SVM algorithms have been optimized by the grid search method.

Prediction Performance of CNN-FSRF Model. We first performed experiments on *Yeast* dataset, and Table 1 summarizes the results of the 5-fold cross-validation experiment. It can be seen that the accuracy of CNN-FSRF on *Yeast* dataset was as high as 97.75%. In order to better investigate the predictive ability of the model, we also calculate the values of sensitivity, specificity, precision, F_{score} , Matthews correlation coefficient, and AUC. In these evaluation criteria, the F_{score} value that reflects the stability of the model is 97.79% and the MCC and AUC values that reflect the overall performance of the model were 95.57% and 97.54%, and their standard variance were 0.53%, 1.05% and 0.66%, respectively. Figure 3 shows the ROC curves and P-R curves obtained by CNN-FSRF on the *Yeast* dataset respectively. It can be seen from the graph that the curves generated by the five experiments cover most of the coordinate space. The 5-fold cross-validation experimental results demonstrate that CNN-FSRF performs well on the *Yeast* dataset.

We next implement the proposed method on the *Helicobacter pylori* dataset, and its 5-fold cross-validation experimental results are shown in Table 2. We can see from Table 2 that CNN-FSRF achieved an accuracy of 88.96% on the *Helicobacter pylori* dataset. In the F_{score} , MCC, and AUC that comprehensively reflect model performance, the values obtained by CNN-FSRF were 89.26%, 78.09%, and 89.08%, and the standard deviations were 0.67%, 1.16%, and 0.79%, respectively. Figure 4 plots the ROC curves and P-R curves generated on the *Helicobacter pylori* dataset. It can be seen from the figure that although the CNN-FSRF performance on the *Helicobacter pylori* dataset is not excellent on the *Yeast* dataset, it also achieved good performance. This may be due to the fact that the number of samples in the *Helicobacter pylori* dataset (2916) is less than in the *Yeast* dataset (11188). It is well known that the number of samples used to train the classifier in machine learning is closely related to the final test result. The more samples in the training set, the more fully trained the classifier, the higher the model fitting degree learned, and the better the prediction result. Therefore, the results obtained by the proposed model in the *Helicobacter pylori* dataset were not as good as those in the *Yeast* dataset also conform to this rule. In addition, this result can also indicate that the performance of CNN-FSRF will become better as the training set increases.

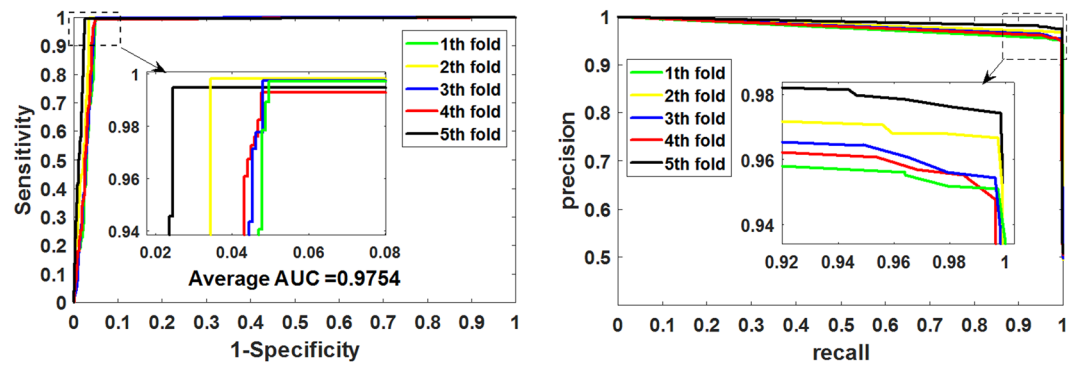


Figure 3. The ROC and P-R curves were generated on the *Yeast* dataset by using the CNN-FSRF method.

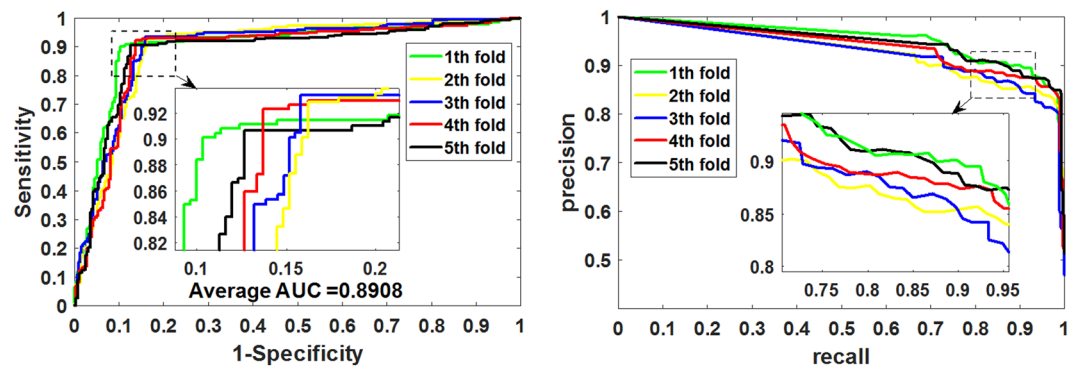


Figure 4. The ROC and P-R curves were generated on the *Helicobacter pylori* dataset by using the CNN-FSRF method.

Test set	Accu.(%)	Sen.(%)	Spec. (%)	Prec.(%)	Fscore(%)	MCC(%)	AUC(%)
1	89.88	90.10	89.66	89.80	89.95	79.76	90.08
2	88.34	92.83	83.79	85.27	88.89	76.97	89.24
3	88.51	93.41	84.19	83.88	88.39	77.52	89.54
4	89.37	92.31	86.27	87.62	89.90	78.81	88.21
5	88.70	90.67	86.62	87.74	89.18	77.40	88.35
Average	88.96	91.86	86.11	86.86	89.26	78.09	89.08
Standard Deviation	0.65	1.42	2.34	2.31	0.67	1.16	0.79

Table 2. The 5-fold cross-validation results were generated on the *Helicobacter pylori* dataset by using the CNN-FSRF method.

Comparison between the proposed model and SVM Model. SVM is a supervised learning model, which is one of the most robust and accurate methods in data mining algorithms³⁸. SVM can map the sample space into the high-dimensional feature space through a non-linear mapping, so that the non-linear separable problem in the original sample space is transformed into a linear separable problem in the feature space. To demonstrate the performance of the proposed method, we compare the CNN-FSRF and SVM model (CNN-SVM) on the same dataset. For fairness, we optimized the parameters of the SVM using the grid search method and used the same protein number descriptors.

The 5-fold cross-validation experimental results by the SVM classifier combined with the CNN extracted feature descriptors were shown in Table 3. It is observed from Table 3 that CNN-SVM achieved the 5-fold cross-validation accuracy of 88.92% and the standard deviation of 1.34% on the *Yeast* dataset. The accuracy is 8.83% lower than that of CNN-FSRF and the standard deviation is 0.80% higher than that of CNN-FSRF. Except that CNN-SVM is 0.11% higher than CNN-FSRF on sensitivity, CNN-SVM is 17.76%, 14.01%, 7.79%, 15.84% and 8.69% lower on specificity, precision, F_{score} , MCC and AUC than CNN-FSRF. However, in the standard deviation, the above evaluation criteria CNN-SVM are 0.03%, 1.64%, 0.95%, 0.59%, 1.17% and 0.63% higher than CNN-FSRF, respectively.

Test set	Accu.(%)	Sen.(%)	Spec. (%)	Prec.(%)	Fscore(%)	MCC(%)	AUC(%)
1	89.27	99.73	78.99	82.35	90.21	80.35	88.34
2	87.89	99.91	76.13	80.36	89.08	78.11	88.42
3	89.05	100.00	78.18	81.97	90.09	80.06	89.11
4	87.48	99.56	75.20	80.33	88.92	77.21	87.48
5	90.89	99.38	82.14	85.15	91.72	82.94	90.91
CNN-SVM Average	88.92	99.72	78.13	82.03	90.00	79.73	88.85
CNN-SVM Standard Deviation	1.34	0.25	2.71	1.97	1.12	2.22	1.29
CNN-FSRF Average	97.75	99.61	95.89	96.04	97.79	95.57	97.54
CNN-FSRF Standard Deviation	0.54	0.22	1.07	1.02	0.53	1.05	0.66

Table 3. Comparison of 5-fold cross-validation results of CNN-FSRF and CNN-SVM on *Yeast* dataset.

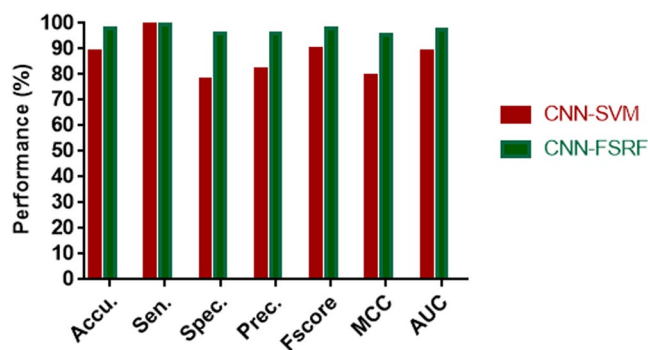


Figure 5. Comparison of performance between CNN-FSRF and CNN-SVM on the *Yeast* dataset.

To facilitate observation, we present these evaluation criteria in the form of histogram. At the same time, we also plotted ROC curves and P-R curves of CNN-FSRF and CNN-SVM on the same coordinate axis. It can be clearly seen from Fig. 5 that CNN-FSRF performed better than CNN-SVM on accuracy and F_{score} , which reflects the prediction accuracy and the stability of the model. In addition, it can be clearly seen from Fig. 6 that the proposed CNN-FSRF also outperforms CNN-SVM on comprehensive evaluation criteria AUC reflecting the overall performance of the model. This indicates that the overall performance of CNN-FSRF is superior to that of CNN-SVM. Therefore, we have reason to believe that the proposed CNN-FSRF method can effectively predict the interaction between proteins.

Comparison with existing methods. To further evaluate the performance of CNN-FSRF, we collected the work of other researchers on the same *Yeast* and *Helicobacter pylori* datasets and used 5-fold cross-validation method to predict PPI. Since some works do not provide more evaluation criteria, we only list the common evaluation criteria of these works, including accuracy, sensitivity, precision and MCC.

Table 4 lists the performance of several previous works and our model on the *Yeast* dataset. From the table we can see that the proposed method achieves the best results in accuracy, sensitivity and MCC, but only the third result in precision. Specifically, the proposed model achieved 97.75% on the accuracy, which is 1.15% higher than the second highest Wangs' work. The model has a great advantage in sensitivity, and achieves 99.61% of the results, which is 4.49% higher than the second highest Zhangs' work. The results obtained from the proposed model on precision generally achieved only the third highest 95.89% result, which was 3.47% lower than the first high Wangs' work. The proposed model on the MCC also has a large advantage, achieving 96.04% of the results, which is 2.63% higher than the second highest Wangs' work. Generally speaking, the comprehensive performance of the proposed method is superior to other methods in the table, and has highly competitive in predicting PPI. In addition, we can also see that Wangs' work, Dus' work, Zhangs' work, Patels' work and the proposed model all use deep learning-based algorithms, and the results obtained by these methods are significantly better than those of other methods in the table that do not use deep learning. This demonstrates that the use of deep learning algorithm can effectively improve the performance of the model.

We collected previous work on the *Helicobacter pylori* dataset and summarized the results in Table 5. We can see from the table that our model achieved the best results in terms of accuracy, sensitivity, and precision, and achieved the second best result on the MCC. Specifically, CNN-FSRF is 1.46% higher in accuracy than the second Ensemble ELM model, 2.91% higher in sensitivity than the second Ensemble ELM model, 0.71% higher in precision than the second Ensemble ELM model, and 0.04% lower in MCC than the first Ensemble ELM model. Generally, our model achieved the highest prediction accuracy on the *Helicobacter pylori* dataset, and the performance of the model ranked second, but it is only 0.04% less than the first one.

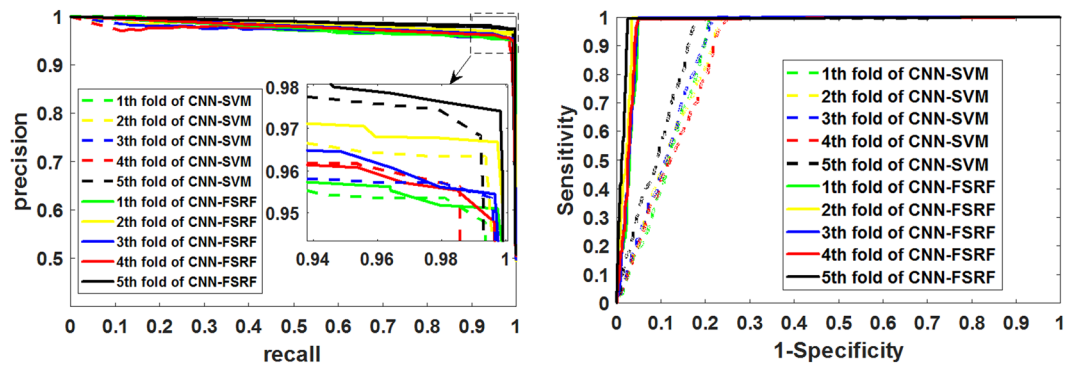


Figure 6. Comparison of ROC curves and P-R curves of CNN-FSRF and CNN-SVM on the same coordinate axis.

Author	Model	Accu.(%)	Sen.(%)	Prec.(%)	MCC(%)
Yangs' work ⁴⁰	Cod1	75.08 ± 1.13	75.81 ± 1.20	74.75 ± 1.23	N/A
	Cod2	80.04 ± 1.06	76.77 ± 0.69	82.17 ± 1.35	N/A
	Cod3	80.41 ± 0.47	78.14 ± 0.90	81.86 ± 0.99	N/A
	Cod4	86.15 ± 1.17	81.03 ± 1.74	90.24 ± 0.45	N/A
Zhous' work ⁴¹	SVM + LD	88.56 ± 0.33	87.37 ± 0.22	89.50 ± 0.60	77.15 ± 0.68
Yous' work ⁴²	PCA-EELM	87.00 ± 0.29	86.15 ± 0.43	87.59 ± 0.32	77.36 ± 0.44
Guos' work ³⁰	ACC	89.33 ± 2.67	89.93 ± 3.68	88.87 ± 6.16	N/A
	AC	87.36 ± 1.38	87.30 ± 4.68	87.82 ± 4.33	N/A
Wangs' work ⁴³	SAE	96.60 ± 0.22	93.73 ± 0.46	99.36 ± 0.41	93.41 ± 0.41
Dus' work ⁴⁴	DeepPPI	94.43 ± 0.30	N/A	96.65 ± 0.59	88.97 ± 0.62
Zhangs' work ⁴⁵	EnsDNN	95.29 ± 0.43	95.12 ± 0.45	95.45 ± 0.89	90.59 ± 0.86
Patels' work ⁴⁶	DeepInteract	92.67	86.85	98.31	85.96
Our model	CNN-FSRF	97.75 ± 0.54	99.61 ± 0.22	95.89 ± 1.02	96.04 ± 1.05

Table 4. The performance comparison between different methods on the *Yeast* dataset.

Model	Accu.(%)	Sen.(%)	Prec.(%)	MCC(%)
HKNN	84.00	86.00	84.00	N/A
Boosting ⁴⁷	79.52	80.37	81.69	70.64
Signature products ¹²	83.40	79.90	85.70	N/A
Ensemble of HKNN ⁴⁸	86.60	86.70	85.00	N/A
Ensemble ELM ¹²	87.50	88.95	86.15	78.13
Phylogentic bootstrap ⁴⁹	75.80	69.80	80.20	N/A
Our model	88.96	91.86	86.86	78.09

Table 5. The performance comparison of different methods on the *Helicobacter pylori* dataset.

We can also see from Tables 4 and 5 that the performance of these methods we collected on the *Helicobacter pylori* dataset is generally not as good as that on *Yeast* dataset, which is likely to be related to the number of dataset samples, and also in accordance with the conclusions of our previous section. In addition, it can be seen from the horizontal comparison that the results obtained by our model on the *Helicobacter pylori* dataset are only slightly better than the other methods, but the results obtained on the *Yeast* dataset are much better than the other methods. This indicates that with the increase of data sets, our approach can quickly improve overall performance and is well-suited for large datasets.

Performance on independent datasets. Although CNN-FSRF achieved high light performance on the *Yeast* and *Helicobacter pylori* datasets, we further verify its performance on independent datasets. Specifically, we first train the CNN-FSRF using the entire *Yeast* dataset, and then use the trained model to predict the interaction among the proteins in the *C. elegans*, *E. coli*, *H. sapiens* and *M. musculus* datasets. This in biological experiments means using protein interactions identified in one organism to predict interactions in other organisms. This approach is based on the assumption that homologous proteins have the ability to maintain their interactions.

Species	Test pairs	Accu.(%)	Sen.(%)	F _{score} (%)
<i>C. elegans</i>	4013	96.41	96.41	98.17
<i>E. coli</i>	6954	95.47	95.47	97.68
<i>H. sapiens</i>	1412	98.65	98.65	99.32
<i>M. musculus</i>	313	93.27	93.27	96.52

Table 6. Prediction results of four species based on the proposed method.

The hypothesis is based on the assumption that homologous species have similar functional behaviors, so that they maintain the same PPIs³⁹.

The *C. elegans*, *E. coli*, *H. sapiens* and *M. musculus* datasets contain only pairs of interacting proteins, the numbers of which are 4013, 6954, 1412, and 313, respectively. Therefore, in the experiment we only calculated meaningful accuracy, sensitivity and F_{score}. Table 6 lists the experimental results on the independent datasets. As can be seen from the table, CNN-FSRF achieved good results in these four datasets, with average accuracy, sensitivity, and F_{score} of 95.95%, 95.95% and 97.92%, respectively. Excellent experimental results show that our model can also achieve good results in independent datasets. This fully demonstrates that our method not only has good performance, but also has good generalization and can be applied to different protein interaction prediction problems.

Conclusions

In this study, we develop a novel sequence-based approach to accurately predict potential protein-protein interactions by combining deep learning convolutional neural network with feature-selective rotation forest. It is well known that extracting effective feature descriptors is the key to predicting PPIs, so the main advantage of this paper is that it can extract the feature information of protein objectively and profoundly by the convolution neural network. Then use FSRF to remove noise information and give accurate prediction results. The experimental results show that CNN-FSRF performs significantly well in predicting PPIs. CNN-FSRF obtained 97.75% and 88.96% prediction accuracy using the 5-fold cross-validation in the real PPIs datasets *Yeast* and *Helicobacter pylori*. In the experiment, we compared the CNN-FSRF with the SVM model and other existing methods. In addition, we validated our approach on the independent datasets. Excellent experimental results demonstrate that our approach can be an effective tool to accurately predict potential protein interactions. In future research, we will continue to study the use of deep learning to extract effective protein features in the hope of achieving better results.

References

- Zhang, Q. C. *et al.* Structure-based prediction of protein-protein interactions on a genome-wide scale. *Nature* **490**, 556–+, <https://doi.org/10.1038/nature11503> (2012).
- Wang, L. *et al.* Advancing the prediction accuracy of protein-protein interactions by utilizing evolutionary information from position-specific scoring matrix and ensemble classifier. *Journal Of Theoretical Biology* **418**, 105–110, <https://doi.org/10.1016/j.jtbi.2017.01.003> (2017).
- You, Z. H., Lei, Y. K., Gui, J., Huang, D. S. & Zhou, X. B. Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data. *Bioinformatics* **26**, 2744–2751, <https://doi.org/10.1093/bioinformatics/btq510> (2010).
- Gavin, A. C. *et al.* Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* **415**, 141–147, <https://doi.org/10.1038/415141a> (2002).
- Ito, T. *et al.* A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America* **98**, 4569–4574, <https://doi.org/10.1073/pnas.061034498> (2001).
- Krogan, N. J. *et al.* Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature* **440**, 637–643, <https://doi.org/10.1038/nature04670> (2006).
- Ho, Y. *et al.* Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* **415**, 180–183, <https://doi.org/10.1038/415180a> (2002).
- Yang, Y. D. & Zhou, Y. Q. Specific interactions for ab initio folding of protein terminal regions with secondary structures. *Proteins-Structure Function and Bioinformatics* **72**, 793–803, <https://doi.org/10.1002/prot.21968> (2008).
- Pan, X.-Y., Zhang, Y.-N. & Shen, H.-B. Large-Scale Prediction of Human Protein-Protein Interactions from Amino Acid Sequence Based on Latent Topic Features. *Journal of Proteome Research* **9**, 4992–5001, <https://doi.org/10.1021/pr100618t> (2010).
- Katona, G. *et al.* Fast two-photon *in vivo* imaging with three-dimensional random-access scanning in large tissue volumes. *Nature Methods* **9**, 201–208 (2012).
- Katona, G., Garcia-Bonete, M. J. & Lundholm, I. V. Estimating the difference between structure-factor amplitudes using multivariate Bayesian inference. *Acta Crystallographica* **72**, 406–411 (2016).
- Martin, S., Roe, D. & Faulon, J. L. Predicting protein-protein interactions using signature products. *Bioinformatics* **21**, 218–226, <https://doi.org/10.1093/bioinformatics/bth483> (2005).
- Jiao, Q. J., Zhang, Y. K., Li, L. N. & Shen, H. B. BinTree seeking: a novel approach to mine both bi-sparse and cohesive modules in protein interaction networks. *Plos One* **6**, e27646 (2011).
- Luo, X. *et al.* A Highly Efficient Approach to Protein Interactome Mapping Based on Collaborative Filtering Framework. *Scientific Reports* **5**, <https://doi.org/10.1038/srep07702> (2015).
- Urquiza, J. M. *et al.* Method for prediction of protein-protein interactions in yeast using genomics/proteomics information and feature selection. *Neurocomputing* **74**, 2683–2690 (2011).
- Zhang, Q. C. *et al.* Structure-based prediction of protein-protein interactions on a genome-wide scale (vol 490, pg 556, 2012). *Nature* **495**, 127–127, <https://doi.org/10.1038/nature11977> (2013).
- Zhang, Q. C., Petrey, D., Norel, R. & Honig, B. H. Protein interface conservation across structure space. *Proc Natl Acad Sci USA* **107**, 10896–10901 (2010).
- Kafkas, Ş., Varoğlu, E., Rebholz-Schuhmann, D. & Taneri, B. Functional variation of alternative splice forms in their protein interaction networks: a literature mining approach. *Bmc Bioinformatics* **11**, P1 (2010).
- Xu, J. *et al.* Refined phylogenetic profiles method for predicting protein-protein interactions. *Bioinformatics* **21**, 3409 (2005).

20. Sun, J., Li, Y. & Zhao, Z. Phylogenetic profiles for the prediction of protein-protein interactions: how to select reference organisms? *Biochem Biophys Res Commun* **353**, 985–991 (2007).
21. Autore, F. *et al.* Large-scale modelling of the divergent spectrin repeats in nesprins: giant modular proteins. *Plos One* **8**, e63633 (2013).
22. Zhang, J., Yang, J., Huang, T., Shu, Y. & Chen, L. Identification of novel proliferative diabetic retinopathy related genes on protein-protein interaction network. *Neurocomputing* **217**, 63–72 (2016).
23. Zhang, Y.-N., Pan, X.-Y., Huang, Y. & Shen, H.-B. Adaptive compressive learning for prediction of protein-protein interactions from primary sequence. *Journal of Theoretical Biology* **283**, 44–52, <https://doi.org/10.1016/j.jtbi.2011.05.023> (2011).
24. Wang, D. D., Wang, R. & Yan, H. Fast prediction of protein-protein interaction sites based on Extreme Learning Machines. *Neurocomputing* **128**, 258–266 (2014).
25. Zhu, L., You, Z. H. & Huang, D. S. Increasing the reliability of protein-protein interaction networks via non-convex semantic embedding. *Neurocomputing* **121**, 99–107 (2013).
26. You, Z. H. *et al.* Detecting Protein-Protein Interactions with a Novel Matrix-Based Protein Sequence Representation and Support Vector Machines. *Biomed Research International* **2015**, 1–9 (2015).
27. Hu, L. & Chan, K. C. Extracting Coevolutionary Features from Protein Sequences for Predicting Protein-Protein Interactions. *IEEE/ACM Trans Comput Biol Bioinform* **14**, 155–166 (2017).
28. Saha, I. *et al.* Ensemble learning prediction of protein-protein interactions using proteins functional annotations. *Molecular Biosystems* **10**, 820–830, <https://doi.org/10.1039/c3mb70486f> (2014).
29. Jeong, J. C., Lin, X. & Chen, X.-W. On Position-Specific Scoring Matrix for Protein Function Prediction. *Ieee-Acm Transactions on Computational Biology and Bioinformatics* **8**, 308–315, <https://doi.org/10.1109/tcbb.2010.93> (2011).
30. Guo, Y., Yu, L., Wen, Z. & Li, M. Using support vector machine combined with auto covariance to predict proteinprotein interactions from protein sequences. *Nucleic Acids Research* **36**, 3025–3030, <https://doi.org/10.1093/nar/gkn159> (2008).
31. Gao, Z. G. *et al.* Ens-PPI: A Novel Ensemble Classifier for Predicting the Interactions of Proteins Using Autocovariance Transformation from PSSM. *Biomed Research International*, **8**, <https://doi.org/10.1155/2016/4563524> (2016).
32. Wang, L. *et al.* A Computational-Based Method for Predicting Drug-Target Interactions by Using Stacked Autoencoder Deep Neural Network. *Journal Of Computational Biology* **25**, 361–373, <https://doi.org/10.1089/cmb.2017.0135> (2018).
33. Ngiam, J. *et al.* In International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July. 689–696.
34. Zhou, S., Chen, Q. & Wang, X. Active deep learning method for semi-supervised sentiment classification. *Neurocomputing* **120**, 536–546 (2013).
35. Wang, L. *et al.* RFDT: A Rotation Forest-based Predictor for Predicting Drug-Target Interactions Using Drug Structure and Protein Sequence Information. *Current Protein & Peptide Science* **19**, 445–454, <https://doi.org/10.2174/138920371866616114111656> (2018).
36. Guo, X., Chen, L. & Shen, C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. *Measurement* **93**, 490–502 (2016).
37. Rodriguez, J. J. & Kuncheva, L. I. Rotation forest: A new classifier ensemble method. *Ieee Transactions on Pattern Analysis and Machine Intelligence* **28**, 1619–1630, <https://doi.org/10.1109/tpami.2006.211> (2006).
38. Pal, M. & Foody, G. M. Feature Selection for Classification of Hyperspectral Data by SVM. *IEEE Transactions on Geoscience & Remote Sensing* **48**, 2297–2307 (2010).
39. Shi, M.-G., Xia, J.-F., Li, X.-L. & Huang, D.-S. Predicting protein-protein interactions from sequence using correlation coefficient and high-quality interaction dataset. *Amino Acids* **38**, 891–899, <https://doi.org/10.1007/s00726-009-0295-y> (2010).
40. Yang, L., Xia, J.-F. & Gui, J. Prediction of Protein-Protein Interactions from Protein Sequence Using Local Descriptors. *Protein and Peptide Letters* **17**, 1085–1090 (2010).
41. Zhou, Y. Z., Gao, Y. & Zheng, Y. Y. Prediction of Protein-Protein Interactions Using Local Description of Amino Acid Sequence. *Advances in Computer Science and Education Applications, Pt II* **202**, 254–262 (2011).
42. You, Z.-H., Lei, Y.-K., Zhu, L., Xia, J. & Wang, B. Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. *Bmc Bioinformatics* **14**, <https://doi.org/10.1186/1471-2105-14-s8-s10> (2013).
43. Wang, Y. B. *et al.* Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network. *Molecular Biosystems* **13**, 1336–1344 (2017).
44. Du, X. *et al.* DeepPPI: Boosting Prediction of Protein-Protein Interactions with Deep Neural Networks. *Journal of Chemical Information & Modeling* **57**, 1499 (2017).
45. Long, Z., Yu, G., Xia, D. & Wang, J. Protein-Protein Interactions Prediction based on Ensemble Deep Neural Networks. *Neurocomputing*, S0925231218306337- (2018).
46. Tripathi, R. DeepInteract: Deep Neural Network based Protein-Protein Interaction prediction tool. *Current Bioinformatics* **11** (2017).
47. Liu, B. *et al.* QChIPat: a quantitative method to identify distinct binding patterns for two biological ChIP-seq samples in different experimental conditions. *Bmc Genomics* **14**, <https://doi.org/10.1186/1471-2164-14-s8-s3> (2013).
48. Nanni, L. & Lumini, A. An ensemble of K-local hyperplanes for predicting protein-protein interactions. *Bioinformatics* **22**, 1207–1210, <https://doi.org/10.1093/bioinformatics/btl055> (2006).
49. Bock, J. R. & Gough, D. A. Whole-proteome interaction mining. *Bioinformatics* **19**, 125–134, <https://doi.org/10.1093/bioinformatics/19.1.125> (2003).

Acknowledgements

This work is supported by the National Natural Science Foundation of China, under Grants 61702444, in part by the West Light Foundation of The Chinese Academy of Sciences, under Grant 2018-XBQNXZ-B-008, in part by the Zaozhuang Science and Technology Development Plan, under Grant 2018GX07. The authors would like to thank all anonymous reviewers for their constructive advices.

Author Contributions

L.W., H.W. and X.Y. conceived the algorithm, carried out the analyses, prepared the data sets, carried out experiments, and wrote the manuscript. S.L., K.S. and L.W. designed, performed and analyzed experiments and wrote the manuscript. All authors read and approved the final manuscript.

Additional Information

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019