Hindawi
Scientific Programming
Volume 2020, Article ID 8866557, 8 pages
https://doi.org/10.1155/2020/8866557

Research Article

ACT-SVM: Prediction of Protein-Protein Interactions Based on Support Vector Basis Model

Wenzheng Ma,[1] Yi Cao,[1] Wenzheng Bao,[2] Bin Yang,[3] and Yuehui Chen[1]

[1]*School of Information Science, University of Jinan, Jinan 250022, China*
[2]*School of Information and Electrical Engineering, Xuzhou University of Technology, Xuzhou 221018, China*
[3]*School of Information Science and Engineering, Zaozhuang University, Zaozhuang 277100, China*

Correspondence should be addressed to Wenzheng Bao; baowz55555@126.com

Received 7 April 2020; Revised 13 May 2020; Accepted 22 May 2020; Published 20 July 2020

Academic Editor: Chenxi Huang

The interactions between proteins play important roles in several organisms, and such issue can be involved in almost all activities in the cell. The research of protein-protein interactions (PPIs) can make a huge contribution to the prevention and treatment of diseases. Currently, many prediction methods based on machine learning have been proposed to predict PPIs. In this article, we propose a novel method ACT-SVM that can effectively predict PPIs. The ACT-SVM model maps protein sequences to digital features, performs feature extraction twice on the protein sequence to obtain vector A and descriptor CT, and combines them into a vector. Then, the feature vectors of the protein pair are merged as the input of the support vector machine (SVM) classifier. We utilize nonredundant *H. pylori* and human dataset to verify the prediction performance of our method. Finally, the proposed method has a prediction accuracy of 0.727897 for *H. pylori* data and a prediction accuracy of 0.838799 for human dataset. The results demonstrate that this method can be called a stable and reliable prediction model of PPIs.

1. Introduction

Proteins are the material basis of all life composed of 20 types of amino acids in the level of biology [1]. There are several kinds of proteins with different properties and functions, which play a pivotal role in the cells and tissues of various biological species. Not only is it an important part of the living organism, but also it participates in and carries all important life activities in the life process. However, most proteins often do not perform their functions alone. Instead, it is more common that two or more proteins work together by forming a protein complex, and a large protein-protein interaction network is finally built [2–6]. Obviously, PPIs play a key role in cellular processes and are involved in many important biological processes such as immune response, material transport, and gene expression regulation. Therefore, exploring the interactions between proteins has become one of the most important links in researching the function and mechanism of proteins [7–9]. In addition, PPIs are a major molecular mechanism of virus pathogenic, which makes them one of the important research objects for disease discovery and treatment.

The importance of researching PPIs has advanced the methods for predicting and identifying PPIs [10–13]. In recent years, some high-throughput laboratory biotechnology has been widely utilized in PPIs, such as yeast two-hybrid (Sato et al.; Schwikowski et al.; Coates Hall) [14–16] and coimmunoprecipitation (Free et al.) [17]. However, they all have some defects in common or personality. For example, some methods fail to overcome higher proportion of false negatives and false positives, and some methods require more sample material to extract proteins, which is surprisingly expensive. At the same time, methods such as protein phylogenetic profile (Kim et al.) [18, 19], natural language processing (Daraselia et al.) [20], and protein tertiary structure (Aloy and Russell) [21] have also been favored by researchers. However, if there is no known protein-related biological knowledge, these kinds of methods are difficult to implement, and some of them cannot fully predict PPIs [22, 23].

In addition, with the tireless efforts of researchers, it was found that PPIs can be predicted based on the amino acid sequence of the protein [24–27]. At the same time, machine learning has been utilized by researchers far and wide. Then, a large number of prediction methods based on protein sequences and machine learning algorithms have appeared [13, 28–32]. For example, Cui et al. [33] utilized support vector machine classifier to predict human proteins that interact with viral proteins [34–37]. The L1-logreg classifier proposed by Dhole et al. can effectively predict PPIs and advance related research such as drug design. Xia et al. [38] proposed a sequence-based multiclassifier system called Spinning Forest to infer PPIs [39]. The performance of their method on the *Saccharomyces cerevisiae* and *H. pylori* datasets is better than previously published literature methods. And as an effective machine learning method, deep learning is also utilized in the prediction of PPIs (Du et al.) [40].

In this paper, we propose a novel prediction model which is based on support vector machine to predict PPIs named ACT-SVM. Two different methods were utilized to extract features from protein sequences, and finally we reconstruct them into a feature vector. First, we extract an A vector for each protein sequence in the dataset. Hereafter, we construct composition (C) and transformation (T) descriptors to describe protein sequences. Last, we utilize their combination as the input of the classifier. In general, the area under curve (AUC), accuracy (Acc), specificity (Sp), and Matthew correlation coefficient (Mcc) are utilized to evaluate the performance of our prediction method.

We have additionally constructed 5 different classifiers for comparing the predictive performance, including *k*-nearest neighbor (KNN), artificial neural network (ANN), random forest (RF), naive Bayes (NB), and logistic regression (LR). We utilized H. pylori and human datasets to evaluate our novel predictor. Experimental results demonstrate that the novel model based on support vector machine which is proposed by us is performs best.

2. Methods and Materials

In scientific research, it is extremely important to first define the workflow. Our working flow is demonstrated in Figure 1. First, we obtained nonredundant *H. pylori* and human datasets. Then, we map each protein sequence to digital features by constructing A vector, composition, and transformation (CT) and combine them into one feature vector as the input of the classifier. The following process is to input the extracted digital feature into different classifiers to train different classification models and evaluate them by 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation, respectively. Finally, on the independent test datasets, we sequentially verified the 6 trained models. In addition, we utilize AUC, Acc, Sp, Sn, and MCC indicators to evaluate the performance of our novel predictive silver and five models utilized as a comparison.

2.1. Dataset. As people pay an increasing attention to PPIs, the number of databases utilized to research PPIs is increasing, such as BioGRID, GeneMANIA, and DIP. However, there is inevitable redundancy in the data in these existing databases. To make our prediction tool more effective, we derived nonredundant *H. pylori* and human PPIs dataset utilized by Kong et al. [41]. They downloaded the *H. pylori* and human PPIs dataset from the DIP database and utilized the cd-hit tool to construct nonredundant sequences for these two datasets. After removing redundancy, the *H. pylori* dataset contains 1458 interacting protein pairs and 1457 noninteracting protein pairs, while the human dataset has 3899 interacting protein pairs and 4262 noninteracting protein pairs.

2.2. Sequence Feature Vectors

2.2.1. Construct a Vector. When constructing the A vector, we refer to the physical and chemical properties of the protein. The 20 amino acids that make up the protein sequence are divided into 6 classes, as demonstrated in Table 1.

In this way, according to the category, we replace each amino acid in the sequence with the corresponding C_1, C_2, . . ., C_6. Then, we can obtain a simplified sequence. We utilize f_i to describe the frequency of occurrence of each element in the simplified sequence ($i = 1, 2, ..., 6$) and finally get the A vector. The detailed definitions of f_i and A vector are illustrated by equations (1) and (2).

$$f_i = \frac{m_i}{l}, \tag{1}$$

where l is the length of the protein sequence, m_i is the number of type i amino acids in the protein sequence, $i = 1, 2, ..., 6$. For example, if there is a sequence "MGPDDSKRYE," it can be replaced with C_1, C_6, C_6, C_5, C_5, C_3, C_4, C_4, C_2, and C_5. We can see that there are one C_1, one C_2, one C_3, two C_4, three C_5 and two C_6 in the simplified sequence. Thus, $f_1 = 1 * 100\%/10 = 10\%$, $f_2 = 1 * 100\%/10 = 10\%$, $f_3 = 1 * 100\%/10 = 10\%$, $f_4 = 2 * 100\%/10 = 20\%$, $f_5 = 3 * 100\%/10 = 30\%$, and $f_6 = 2 * 100\%/10 = 20\%$.

A vector can be constructed as

$$A = (f_1, f_2, \ldots, f_i, \ldots, f_6). \tag{2}$$

Then, we got a 6-dimensional A vector to describe the feature of the protein.

2.2.2. Sparse Matrix and Descriptor. First, we construct a $20 \times n$ sparse matrix B, where n is the number of amino acids in the protein sequence. We assume that there is a protein sequence $S = S_1, S_2, \ldots, S_n$. At the same time, we put 20 amino acids in E, $E = \{A, V, L, I, M, C, F, W, Y, H, S, T, N, Q, K, R, D, E, G, P\}$. When the i-th amino acid in E is the same as the j-th amino acid in S, the corresponding element b_{ij} in the sparse matrix takes 1; otherwise, it takes 0. The sparse matrix of this protein sequence is demonstrated in the following:

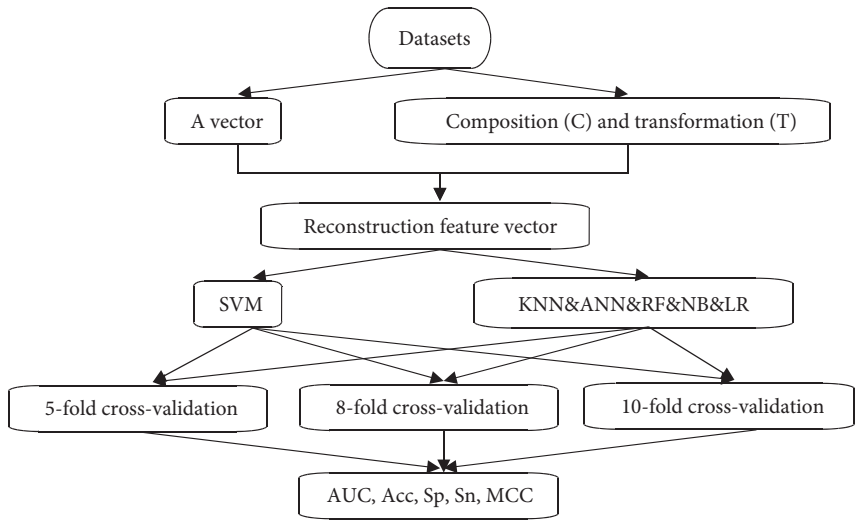

Figure 1: The working flow.

Table 1: Classification of proteins.

Category	Property	Amino acid
C_1	Aliphatic	A, C, I, L, M, V
C_2	Aromatic	F, H, W, Y
C_3	Polar	N, Q, S, T
C_4	Positive	K, R
C_5	Negative	D, E
C_6	Special conformations	G, P

$$B_{20\times n} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ b_{31} & b_{32} & b_{33} & \cdots & b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{20,1} & b_{20,2} & b_{20,3} & \cdots & b_{20,n} \end{pmatrix}, \quad (3)$$

$$b_{ij} = \begin{cases} 0, & E(i) \neq S(j) \\ 1, & E(i) = S(j) \end{cases}$$

Next, we divide each of the 20 row vectors in the sparse matrix into P subvectors. The descriptor consists of composition (C) and transformation (T), and they are extracted from each subvector. Among them, the composition (C) is composed of two parts, including the frequency of 0 and 1 in the subsequence. The transition (T) consists of three parts: the sum of the number of 01 and 10 in the subvector, the number of "11" and the number of "111." Suppose $P = 4$, and the first subsequence of a protein sequence is "MYAHQAAA." Then, the first subvector of the first row vector in the sparse matrix is {0, 0, 1, 0, 0, 1, 1, 1}. Obviously, there are four "0," four "1," two "01," one "10," two "11," and one "111." Therefore, the five parts of the composition and transformation (CT) are $4 * 100\%/8 = 50\%$, $4 * 100\%/8 = 50\%$, 3 $(2 + 1 = 3)$, 2, and 1. Thus, a protein sequence is mapped into a 400-dimensional $(4 * 20 * 5 = 400)$ vector.

2.2.3. Reconstruction of Feature Vectors. For each protein sequence, we extracted two feature vectors, including a 6-dimensional vector A and a 400-dimensional descriptor. Then, we combined them into a 406-dimensional vector as the feature vector of a protein. Finally, the feature vectors of two proteins are connected as a 812-dimensional feature vector, describing the PPIs between them.

2.3. Classifier Construction. Our model is based on SVM. As a linear classifier, SVM is widely utilized in classification problems. Its learning strategy is to maximize the interval. Finally, it can find a geometric hyperplane with the largest distance in the feature space to segment the sample. SVM is extremely stable and sparse. The partitioning hyperplane in the sample space can be described as

$$\boldsymbol{\omega}^T \mathbf{x} + b = 0. \quad (4)$$

Among them, the direction of the hyperplane is determined by $\boldsymbol{\omega}$, and b represents the distance from the origin to the hyperplane. If the hyperplane can correctly classify the samples, one side of the hyperplane is positive samples and the other side is negative samples. Assume that the samples in the sample space are $(\mathbf{x}_i, y_i)$, $y_i \in \{+1, -1\}$, which can be expressed as

$$\begin{cases} \boldsymbol{\omega}^T + b \geq +1, & y_i = +1 \\ \boldsymbol{\omega}^T + b \leq -1, & y_i = -1. \end{cases} \quad (5)$$

The distance from any point in the sample space to the hyperplane can be described by equation (6):

$$d = \frac{\left|\boldsymbol{\omega}^T \mathbf{x} + b\right|}{\|\boldsymbol{\omega}\|}. \quad (6)$$

The closest sample point to the hyperplane is called the support vector. The sum of the distance from the positive sample support vector to the hyperplane and the distance from the negative sample support vector to the hyperplane is called the interval, which can be expressed as

$$\gamma = \frac{2}{\|\boldsymbol{\omega}\|}. \tag{7}$$

The ultimate goal of support vector machine is to find a hyperplane that maximizes the interval, so the support vector machine can be described as

$$\max_{\omega,b} \frac{2}{\|\omega\|}, \tag{8}$$

$$s.t.\ y_i\left(\boldsymbol{\omega}^T \mathbf{x}_i + b\right) \geq 1, \quad i = 1, 2, \ldots, n, \tag{9}$$

where m is the number of samples. Formulas (8) and (9) can also be rewritten as

$$\min_{\omega,b} \frac{1}{2}\|\omega^2\|, \quad s.t.\ y_i\left(\boldsymbol{\omega}^T \mathbf{x}_i + b\right) \geq 1, \quad i = 1, 2, \ldots, n. \tag{10}$$

Through continuous experimentation, we finally set the kernel function of the SVM classifier to a linear kernel function. And combined with our proposed feature extraction method ACT, it showed superior prediction performance on *H. pylori* and human dataset.

2.4. Evaluation of the Predictor. In order to verify the reliability and stability of our proposed predictor, we trained 6 models using *H. pylori* and human dataset and performed 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation [42]. In actual training, the model usually fits the training data better, but it is not particularly ideal for novel data outside the training data. k-fold cross-validation can be utilized to evaluate the generalization ability of models, so as to choose a better model and prevent the model from being too complex and causing overfitting. The basic idea of k-fold cross-validation is to divide the dataset into k parts in equal proportions. Then each part of the data is utilized in turn as the test dataset, and the other $k-1$ parts of the data are utilized as training data. k-fold cross-validation is performed for k trainings to ensure that the k parts of the data have been the test data; the remaining $k-1$ parts have been utilized as training data. The obtained K experimental results are equally divided as the final score of the model ultimately. For k-fold cross-validation, we set k to 5, 8, and 10, respectively, to verify the performance of our model.

In this paper, we employ four evaluation indicators to evaluate the predictive performance of our proposed method, including accuracy (Acc), sensitivity (Sn), specificity (Sp), and Matthew correlation coefficient (Mcc). Among them, Acc reflects the model's ability to classify positive samples correctly; Sn measures the classifier's ability to recognize positive samples; Sp reflects the model's ability to recognize negative samples; Mcc returns a value between −1 and +1, which is an indicator often utilized to measure the performance of binary classification models. Their definitions are as follows:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

$$\text{Sn} = \frac{\text{TP}}{\text{FN} + \text{TP}},$$

$$\text{Sp} = \frac{\text{TN}}{\text{FP} + \text{TN}},$$

$$\text{Mcc} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TN} + \text{FN})}}, \tag{11}$$

where TP is the number correctly divided into positive samples, FP is the number incorrectly divided into positive samples, FN is the number incorrectly divided into negative samples, and TN is correctly divided into negative samples. In addition, we still utilize the AUC value to evaluate the performance of our proposed model. AUC is defined as the area under the ROC curve. In many cases, the ROC curve does not clearly indicate which classifier works better. As a numerical value, the larger the corresponding AUC value, the better the classifier. Thus, we utilize the AUC value as one of the evaluation criteria of the model.

3. Result and Discussion

3.1. Model Stability Analysis. K-fold cross-validation is widely utilized to compare the performance of different machine learning models on a specific dataset. The principle of k-fold cross-validation is to divide the dataset into equal k shares for k trainings and finally take the average of the K results. However, there may be outliers in the k-time results, which means that this classifier may not have good stability for the prediction of all samples. We utilized *H. pylori* and human dataset to train 6 models and performed 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation to evaluate their performance. We draw boxplots to reflect the stability of 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation of the two datasets in 6 classifiers. Six boxplots were drawn to describe the results of 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation of two datasets in 6 classifiers. Among them, the ordinate of the boxplot is accuracy (Acc), and the abscissa is 6 classifiers. That is to say, each boxplot has 6 boxes, and each box stores the Acc value in the k times of k-fold cross-validation in the classifier. The boxplots of the *H. pylori* dataset on 6 classifiers for 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation are demonstrated in Figure 2(a), and the boxplots for the human dataset are demonstrated in Figure 2(b).

The hollow dots appearing in the boxplots are outliers, the size of the boxes reflects the degree of dispersion of the data, and the height of the boxes represents the accuracy value. From the 5-fold cross-validation box diagram in Figure 2(a), we can see that there are outliers in the 5 Acc values obtained by KNN, NB, and SVM in 5 trainings, while the box of the RF classifier is too large that the data is more discrete. The box size of the ANN and LR classifiers is

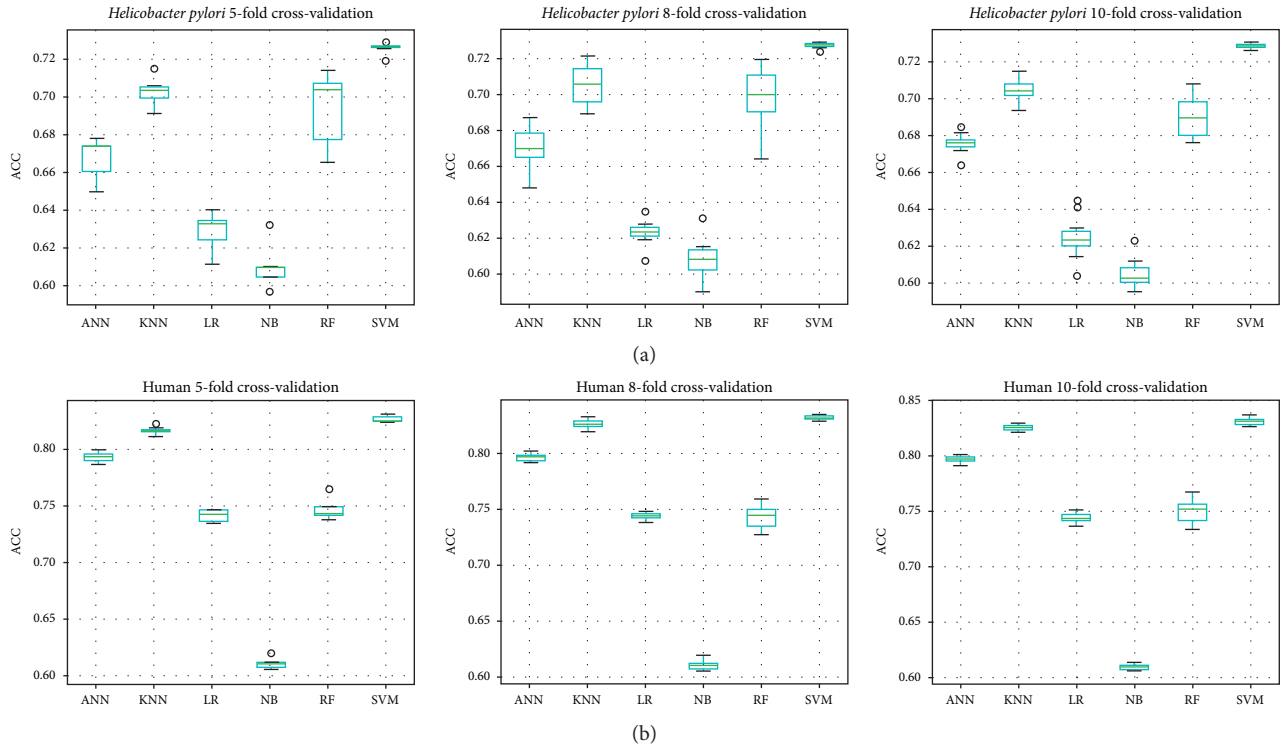

Figure 2: Cross-validated boxplots of *H. pylori* and human dataset.

similar, but from the box height, it can be seen that the accuracy of the ANN is higher. Therefore, on the *H. pylori* dataset, the best performing model using 5-fold cross-validation is ANN. Although the SVM classifier has an outlier in the 8-fold cross-validation, the impact is not significant. Since the outlier has a very small offset and the cabinet is small in size and high in position, SVM still performs best. In this way, we can see from Figure 2 in turn that, on the *H. pylori* dataset, the best performing model using 10-fold cross-validation is SVM. On human dataset, the most stable classifiers with 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation are still SVM. This can prove that the predictor which is based in SVM we proposed performs the most stability in k-fold cross-validation.

Table 2: Performance comparison in classifiers.

Dataset	Classifier	AUC	Acc	MCC	Sn	Sp
H. pylori	ANN	0.7412	0.6738	0.3544	0.5780	0.7698
	SVM	**0.8010**	**0.7279**	**0.4558**	0.7238	0.7320
	KNN	0.7746	0.7055	0.4177	**0.7942**	0.6168
	RF	0.7815	0.7030	0.4104	0.6312	**0.7749**
	LR	0.6969	0.6266	0.2602	0.5129	0.7406
	NB	0.6378	0.6043	0.2089	0.5780	0.6306
Human	ANN	0.8819	0.8008	0.6007	**0.7883**	0.8122
	SVM	0.8938	**0.8388**	**0.6791**	0.7774	0.8950
	KNN	**0.9165**	0.8118	0.6444	0.6575	**0.9531**
	RF	0.8875	0.7849	0.5812	0.6511	0.9073
	LR	0.8177	0.7505	0.5001	0.6915	0.8046
	NB	0.6089	0.6105	0.2642	0.2527	0.9378

3.2. Model Performances. To verify the reliability of our proposed method, we constructed 5 traditional classifiers for comparison, including KNN, RF, ANN, LR, and NB. We utilized *H. pylori* and human datasets and chose 8-fold cross-validation to evaluate the classifiers we constructed. Finally, we utilize 5 evaluation indicators (AUC, Acc, Sn, Sp, and Mcc) to evaluate the predictive performance of each classifier. The experimental results demonstrate that the SVM classifier performs best, as demonstrated in Table 2.

In Table 2, the AUC, Acc, and MCC values of the SVM classifier are the highest of the six classifiers, reaching 0.800963, 0.727897, and 0.455814, respectively, in the *H. pylori* dataset. The KNN classifier has the highest Sn value 0.794168, while the RF classifier has the highest Sp value 0.953052. Although the Sn and Sp values of the SVM classifier are not the highest values, they are not much lower than the highest value, which are 0.723842 and 0.731959, respectively. More importantly, the Sn and Sp values of the SVM classifier are the closest, which means that its ability to correctly predict positive and negative samples is similar. In human dataset, the Acc value of the SVM classifier reached 0.838799, and the MCC value was also the highest among the six classifiers. Although AUC, Sn, and Sp are not the highest values, they are close to the highest values. As in the *H. pylori* dataset, the SVM classifier has the smallest difference in its ability to identify positive and negative samples. From these data, it is clear that the SVM classifier has higher accuracy, pretty good stability, and higher reliability compared to the other five classifiers. Thus, the superior performance of our proposed method has been further verified.

Table 3: Comparison of features.

Dataset	Classifier	Method	AUC	Acc	MCC	Sn	Sp
H. pylori	ANN	FCTP	0.6772	0.6137	0.2337	0.5009	0.7268
		ACT	0.7412	0.6738	0.3544	0.5780	0.7698
	SVM	FCTP	0.7038	0.6549	0.3099	0.6535	0.6564
		ACT	0.8010	0.7279	0.4558	0.7238	0.7320
	KNN	FCTP	0.5747	0.5554	0.1148	0.6913	0.4192
		ACT	0.7746	0.7056	0.4176	0.7942	0.6168
	RF	FCTP	0.7553	0.6601	0.3372	0.5043	0.8162
		ACT	0.7815	0.7030	0.4104	0.6312	0.7749
	LR	FCTP	0.6866	0.62578	0.2547	0.5489	0.7027
		ACT	0.6969	0.6266	0.2602	0.5129	0.7406
	NB	FCTP	0.5186	0.5013	0.0024	0.6072	0.3952
		ACT	0.6378	0.6043	0.2089	0.5780	0.6306
Human	ANN	FCTP	0.8980	0.8275	0.6541	0.8024	0.8504
		ACT	0.8819	0.8008	0.6007	0.7883	0.8122
	SVM	FCTP	0.8320	0.7582	0.5150	0.7274	0.7864
		ACT	0.8938	0.8388	0.6791	0.7774	0.8950
	KNN	FCTP	0.9373	0.8557	0.7181	0.7652	0.9384
		ACT	0.9165	0.8118	0.6444	0.6575	0.9531
	RF	FCTP	0.8950	0.8112	0.6247	0.7357	0.8803
		ACT	0.8875	0.7849	0.5812	0.6511	0.9073
	LR	FCTP	0.8180	0.7444	0.4873	0.7146	0.7717
		ACT	0.8177	0.7505	0.5001	0.6915	0.8046
	NB	FCTP	0.6472	0.6414	0.3525	0.2822	0.9701
		ACT	0.6089	0.6105	0.2642	0.2527	0.9378

3.3. Comparison of Features. For classification issues, the performance of a model is determined by many aspects. One of the very important factors is the choice of feature extraction methods. Feature extraction can transform our original data into features that can better represent the data, improve the prediction accuracy of unknown data, and directly affect the prediction results of the model. Nowadays, researchers have proposed many feature extraction methods, which are dedicated to abstracting the most effective features for classification and recognition from the data. In this paper, we will utilize 6 prebuilt classifiers to compare our feature extraction method ACT with the feature extraction method FCTP proposed by Kong et al. Comparative experimental results are demonstrated in Table 3.

The experimental results demonstrate that, in the *H. pylori* dataset, the five evaluation indexes of the six classifier models using our proposed feature extraction method are better than those using FCTP. In the human dataset, the performance of the model constructed by our method combining SVM and LR is better than that of Kongs' method. In particular, our proposed model ACT-SVM has an Acc value which is 0.08 higher than that of the model using FCTP. Although on the human dataset FCTP performs better on ANN, KNN, RF, and NB, our method also demonstrates good performance with a small gap in indicators in all aspects. Overall, FCTP performed well on the human dataset but performed poorly on the *H. pylori* dataset. Our feature extraction method demonstrates good prediction performance on both datasets and is relatively stable. Therefore, the method we proposed is further proved to be a reliable and stable prediction model for PPIs.

4. Conclusions

In recent years, the problem of identifying PPIs has been valued by researchers and in-depth research. Several efforts to solve this problem have appeared one after another. Although machine learning methods are widely utilized in the prediction of PPIs, there is still a lack of predictors that can accurately and efficiently make predictions. Our proposed model ACT-SVM can effectively predict PPIs. We utilize a combination of A vector, composition, and transition (CT) descriptors as the digital features of the amino acid sequence and utilize them as input to train the SVM model. We evaluate the performance of our proposed method by constructing multiple classifiers using 5-fold cross-validation, 8-fold cross-validation, and 10-fold cross-validation. With these evaluations, we can easily get the conclusion that the model we proposed has the better performance in the majority of situations. The prediction accuracy of our method for *H. pylori* data reaches 0.727897, and the prediction accuracy for human dataset reaches 0.838799. The experimental results demonstrate that our proposed model based on SVM can efficiently predict PPIs. It has good performance on *H. pylori* and human dataset and can be utilized as a research tool to support biomedical and other fields.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the grants of the National Science Foundation of China (nos. 61902337 and 61702445), the grant from the Ph.D. Programs Foundation of Ministry of Education of China (no. 20120072110040), the grant of Shandong Provincial Natural Science Foundation, China (no. ZR2018LF005), Natural Science Fund for Colleges and Universities in Jiangsu Province (no. 19KJB520016), and Jiangsu Provincial Natural Science Foundation (no. SBK2019040953).

References

[1] S. Brohee and J. Van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC Bioinformatics*, vol. 7, no. 1, p. 488, 2006.

[2] N. Sugaya and K. Ikeda, "Assessing the druggability of protein-protein interactions by a supervised machine-learning method," *BMC Bioinformatics*, vol. 10, no. 1, p. 263, 2009.

[3] J. Shen, J. Zhang, X. Luo et al., "Predicting protein-protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4337–4341, 2007.

[4] Q. C. Zhang, D. Petrey, L. Deng et al., "Structure-based prediction of protein-protein interactions on a genome-wide scale," *Nature*, vol. 490, no. 7421, pp. 556–560, 2012.

[5] J. Wu, T. Vallenius, K. Ovaska, J. Westermarck, T. P. Mäkelä, and S. Hautaniemi, "Integrated network analysis platform for protein-protein interactions," *Nature Methods*, vol. 6, no. 1, pp. 75–77, 2009.

[6] J. De Las Rivas and C. Fontanillo, "Protein–protein interactions essentials: key concepts to building and analyzing interactome networks," *PLoS Computational Biology*, vol. 6, no. 6, Article ID e1000807, 2010.

[7] R. K. Barman, S. Saha, and S. Das, "Prediction of interactions between viral and host proteins using supervised machine learning methods," *PLoS one*, vol. 9, no. 11, Article ID e112034, 2014.

[8] Z.-H. You, J.-Z. Yu, L. Zhu, S. Li, and Z.-K. Wen, "A MapReduce based parallel SVM for large-scale predicting protein-protein interactions," *Neurocomputing*, vol. 145, pp. 37–43, 2014.

[9] S. Patel, "DeepInteract: deep neural network based protein-protein interaction prediction tool," *Current Bioinformatics*, vol. 12, pp. 551–557, 2017.

[10] G.-H. Liu, H.-B. Shen, and D.-J. Yu, "Prediction of protein–protein interaction sites with machine-learning-based data-cleaning and post-filtering procedures," *The Journal of Membrane Biology*, vol. 249, pp. 141–153, 2016.

[11] P. Chatterjee, "PPI_SVM: prediction of protein-protein interactions using machine learning, domain-domain affinities and frequency tables," *Cellular and Molecular Biology Letters*, vol. 16, no. 2, pp. 264–278, 2011.

[12] Z.-H. You, "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis," *BMC Bioinformatics*, vol. 14, 2013.

[13] L. Wei, P. Xing, J. Zeng, J. Chen, R. Su, and F. Guo, "Improved prediction of protein-protein interactions using novel negative samples, features, and an ensemble classifier," *Artificial Intelligence in Medicine*, vol. 83, pp. 67–74, 2017.

[14] T. Sato, M. Hanada, S. Bodrug et al., "Interactions among members of the Bcl-2 protein family analyzed with a yeast two-hybrid system," *Proceedings of the National Academy of Sciences*, vol. 91, no. 20, pp. 9238–9242, 1994.

[15] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein-protein interactions in yeast," *Nature Biotechnology*, vol. 18, no. 12, pp. 1257–1261, 2000.

[16] P. Coates and P. Hall, "The yeast two-hybrid system for identifying protein-protein interactions," *The Journal of Pathology*, vol. 199, no. 1, pp. 4–7, 2003.

[17] R. B. Free, L. A. Hazelwood, and D. R. Sibley, "Identifying novel protein-protein interactions using co-immunoprecipitation and mass spectroscopy," *Current Protocols in Neuroscience*, vol. 46, no. 1, pp. 5–28, 2009.

[18] Y. Kim and S. Subramaniam, "Locally defined protein phylogenetic profiles reveal previously missed protein interactions and functional relationships," *Proteins: Structure, Function, and Bioinformatics*, vol. 62, no. 4, pp. 1115–1124, 2006.

[19] V. S. Rao et al., "Protein-protein interaction detection: methods and analysis," *International Journal of Proteomics*, vol. 2014, Article ID 147648, 12 pages, 2014.

[20] N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo, "Extracting human protein interactions from MEDLINE using a full-sentence parser," *Bioinformatics*, vol. 20, no. 5, pp. 604–611, 2004.

[21] P. Aloy and R. B. Russell, "InterPreTS: protein interaction prediction through tertiary structure," *Bioinformatics*, vol. 19, no. 1, pp. 161-162, 2003.

[22] Y.-A Huang et al., "Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding," *BMC Bioinformatics17*, vol. 1, p. 184, 2016.

[23] S.-W. Zhang, L.-Y. Hao, and T.-H. Zhang, "Prediction of protein-protein interaction with pairwise kernel support vector machine," *International Journal of Molecular Sciences*, vol. 15, no. 2, pp. 3220–3233, 2014.

[24] L. Liu, Y. Cai, W. Lu, K. Feng, C. Peng, and B. Niu, "Prediction of protein-protein interactions based on PseAA composition and hybrid feature selection," *Biochemical and Biophysical Research Communications*, vol. 380, no. 2, pp. 318–322, 2009.

[25] X. Li, B. Liao, Y. Shu, Q. Zeng, and J. Luo, "Protein functional class prediction using global encoding of amino acid sequence," *Journal of Theoretical Biology*, vol. 261, no. 2, pp. 290–293, 2009.

[26] S. Martin, D. Roe, and J.-L. Faulon, "Predicting protein-protein interactions using signature products," *Bioinformatics*, vol. 21, no. 2, pp. 218–226, 2005.

[27] L. Nanni, "Hyperplanes for predicting protein-protein interactions," *Neurocomputing*, vol. 69, no. 1-3, pp. 257–263, 2005.

[28] L. Burger and E. Van Nimwegen, "Accurate prediction of protein–protein interactions from sequence alignments using a Bayesian method," *Molecular Systems Biology*, vol. 4, no. 1, p. 165, 2008.

[29] L. Nanni and A. Lumini, "An ensemble of K-local hyperplanes for predicting protein-protein interactions," *Bioinformatics*, vol. 22, no. 10, pp. 1207–1210, 2006.

[30] X.-Y. Pan, Y.-N. Zhang, and H.-B. Shen, "Large-Scale prediction of human Protein–Protein interactions from amino acid sequence based on latent topic features," *Journal of Proteome Research*, vol. 9, no. 10, pp. 4992–5001, 2010.

[31] G. Singh et al., "Springs: prediction of protein-protein interaction sites using artificial neural networks," *PeerJ Pre-Prints*, 2014.

[32] S. Dohkan, A. Koike, and T. Takagi, "Prediction of protein-protein interactions using support vector machines," in *Proceedingsof the Fourth IEEE Symposium on Bioinformatics and Bioengineering*, May 2004.

[33] G. Cui, C. Fang, and K. Han, "Prediction of protein-protein interactions between viruses and human by an SVM model," *BMC bioinformatics*, vol. 13, no. Suppl 7, 2012.

[34] J. R. Bradford and D. R. Westhead, "Improved prediction of protein-protein binding sites using a support vector machines approach," *Bioinformatics*, vol. 21, no. 8, pp. 1487–1494, 2005.

[35] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences," *Nucleic Acids Research*, vol. 36, no. 9, pp. 3025–3030, 2008.

[36] A. Koike and T. Takagi, "Prediction of protein-protein interaction sites using support vector machines," *Protein Engineering Design and Selection*, vol. 17, no. 2, pp. 165–173, 2004.

[37] Z.-H. You, "Detecting protein-protein interactions with a novel matrix-based protein sequence representation and support vector machines," *BioMed Research International*, vol. 2015, Article ID 867516, 9 pages, 2015.

[38] J.-F. Xia, K. Han, and D.-S. Huang, "Sequence-based prediction of protein-protein interactions by means of rotation forest and autocorrelation descriptor," *Protein & Peptide Letters*, vol. 17, no. 1, pp. 137–145, 2010.

[39] L. Wong et al., "Detection of protein-protein interactions from amino acid sequences using a rotation forest model with a novel PR-LPQ descriptor," in *Proceedings of the International Conference on Intelligent Computing*, Springer, Cham, Fuzhou, China, August 2015.

[40] X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, and Y. Zhang, "DeepPPI: boosting prediction of protein-protein interactions with deep neural networks," *Journal of Chemical Information and Modeling*, vol. 57, no. 6, pp. 1499–1510, 2017.

[41] M. Kong, "FCTP-WSRC: protein–protein interactions prediction via weighted sparse representation based classification," *Frontiers in Genetics*, vol. 11, p. 18, 2020.

[42] Z. Lu, S. Lu, G. Liu, Y. Zhang, J. Yang, and P. Phillips, "A pathological brain detection system based on radial basis function neural network," *Journal of Medical Imaging and Health Informatics*, vol. 6, no. 5, pp. 1218–1222, 2016.