Received May 31, 2020, accepted June 15, 2020, date of publication June 23, 2020, date of current version July 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004446

Identification of SNARE Proteins Through a Novel Hybrid Model

GUILIN LI

Department of Software Engineering, School of Informatics, Xiamen University, Xiamen 361005, China

e-mail: glli@xmu.edu.cn

ABSTRACT SNARE proteins are a large family of membrane fusion proteins. As a lot of human diseases are related to the SNARE proteins, they have attracted people to study them. Traditionally, the SNARE proteins can be identified through bioinformatics techniques, which are expensive and time-consuming. Some researchers attempt to identify the SNARE proteins by the machine learning algorithms. A deep learning model called SNARE-CNN is proposed to predict SNARE proteins. A 2D convolutional neural network is constructed and the Position-Specific Scoring Matrix (PSSM) profile is used to distinguish the SNARE proteins from the other kinds of proteins. Although the SNARE-CNN can achieve high accuracy, the performance of the model still has room to improve. In this paper, a novel hybrid model, that combines the random forest algorithm with the oversampling filter and 188D feature extraction method, is proposed. By trying different combinations of feature extraction methods, filtering methods and classification algorithms, the hybrid model, we proposed, can achieve the best performance among all combinations. Experiments show that the performance of our hybrid model is better than that of the SNARE-CNN model.

INDEX TERMS SNARE protein identification, feature representation, feature filtering.

I. INTRODUCTION

SNARE proteins are a large family of membrane fusion proteins [1]. Although they differ greatly in structure and size, they all have a sequence of 60 to 70 amino acids, called the SNARE motif. The integration of cell membranes in eukaryotes can be catalyzed by SNARE proteins. The SNARE proteins are critical for various types of cellular activities, such as synaptic transmission, cytokinesis, and cell growth. There are mainly two classification methods for SNARE proteins. The first one is based on the location of SNARE protein distribution, which divides the SNARE proteins into vesicle membrane SNARE (v-SNARE, mainly VAMP and related proteins) and target membrane SNARE (t-SNARE, mainly including Syntaxin and SNAP-25). The second one is based on the type of amino acid residues in the SNARE protein domain, which divides the SNARE proteins into arginine SNARE protein (R-SNARE) and glutamine SNARE protein (Q-SNARE).

Nowadays, researchers have identified a variety of SNARE proteins and some studies show that a lot of diseases are associated with the loss of function of the SNARE protein, such as neurodegenerative diseases, mental diseases, cancer, etc. [2]–[5]. Therefore, SNARE proteins are very important for human health and it is necessary to develop some techniques to find them. Because SNARE proteins play a vital role in human diseases, they have attracted many researchers to study it. The conserved domains in SNARE proteins were analyzed by using the bioinformatics techniques [6]. Phylogenetic characteristics of SNARE-dependent membrane transport and sequence motifs were extracted by [7]. A framework was proposed by [8] to predict the functions of SNAREs. A database was built by [9], [10] to save and classify SNAREs. In addition, [11] found the convergent evolution of Legionella effectors, by SNARE mimics the membrane fusion. Reference [12] analyzed the damages botulinum neurotoxin caused on SNARE proteins and proposed that the truncated SNAP-25 mutant will destroy SNARE core complex assembly, thereby inhibiting the synaptic membrane fusion.

The identification of SNARE proteins achieved high accuracy, but most of the methods are by means of bioinformatics techniques, which are expensive and time-consuming. While the machine learning techniques are seldom adopted by researchers to the identification of SNARE proteins, which has been widely used to identify other types of

The associate editor coordinating the review of this manuscript and approving it for publication was Quan Zou.

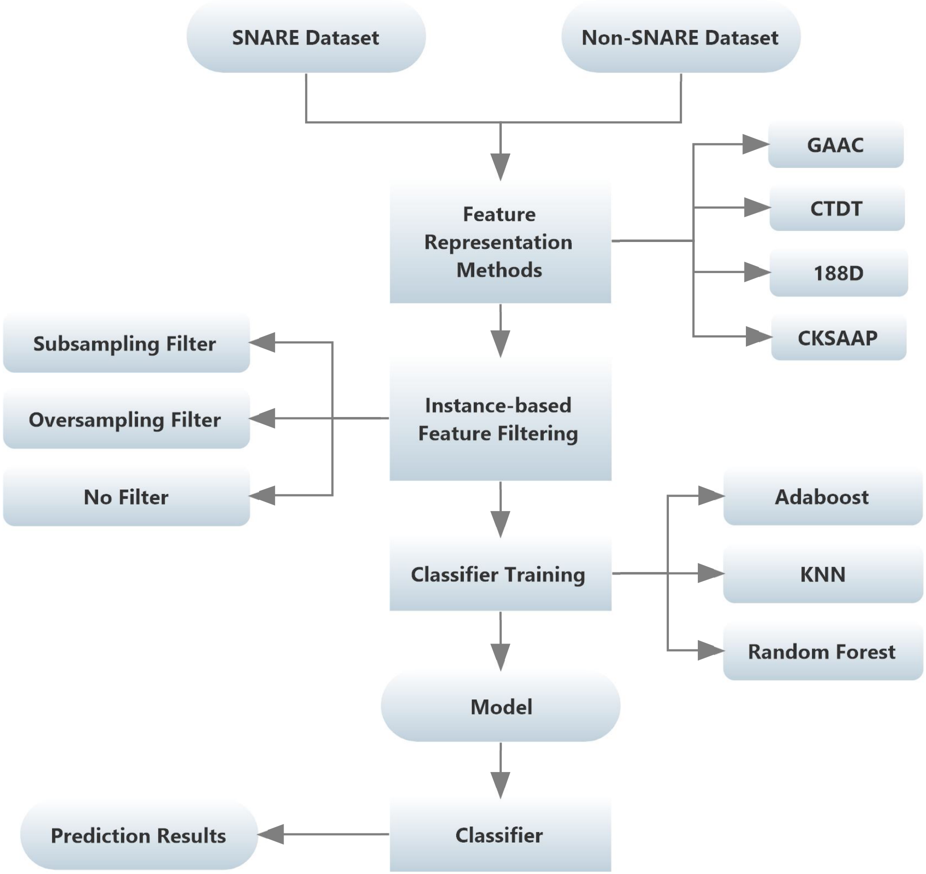


FIGURE 1. Framework of the SNARE proteins identification procedure.

proteins [13]–[32]. Reference [33] proposed a deep learning model to predict SNARE proteins. They constructed a 2D convolutional neural network (CNN), and the Position-Specific Scoring Matrix (PSSM) profile was used to extract features from SNARE proteins to train the CNN model. The experimental results showed that the SNARE-CNN model achieved high performance in terms of sensitivity, specificity, accuracy and MCC, which can be further improved.

In this paper, we develop several machine learning models to identify the SNARE proteins. Firstly, four types of features, which are the Grouped Amino Acid Composition (GAAC) [34], the Composition/Transition/Distribution (CTDT) [35], [36], the 188D [37] and CKSAAP [38]–[40] methods, are used to extract features from the SNARE proteins. As the number of positive instances and negative instances in the feature set is imbalanced, which will affect the performance of the classifier greatly, we need to process the extracted feature sets before classification [41]–[66]. Two types of filtering methods are applied to process the extracted feature sets, which are subsampling filter and oversampling filter. After filtering, the number of positive and negative instances is balanced. Finally, three types of machine learning algorithms are used to identify the SNARE proteins, which are the Adaboost, KNN and Random Forest algorithms. The experimental results show that the performance of the Random Forest algorithms, based on the 188D feature extraction method and oversampling filter, is the best among all combinitions of models. Furthermore, the comparison results show that the performance of the random forest algorithm with oversampling filter and 188D feature extraction method can improve the performance of the SNARE-CNN model.

The contributions of this work include (1) Different combinations of feature extraction methods, filtering methods and classification algorithms are used to identify the SNARE proteins. (2) Experimental results show that the performance of the random forest algorithm with oversample filtering and 188D feature extraction method is the best among all combinations. (3)Extensive experiments are done to show that our hybrid model can improve the performance of the SNARE-CNN model.

The rest of the paper is organized as follows. The data sets used for the experiments and the methods for identifying SNARE proteins are introduced in section 2. The experimental results are given in Section 3. Finally, we conclude our work in Section 4.

II. METHODS

The framework for the SNARE protein classification is shown in figure 1. First, four kinds of feature extraction methods, named GAAC, CTDT, 188D and CKSAAP, are used to extract the features from the SNARE and non-SNARE sequences. As the number of positive and negative instances in the dataset is imbalanced, we apply two kinds of filtering algorithms, which are the subsampling filter and oversampling filter to balance the instances in the dataset. Then, the balanced dataset is used to train the models by three kinds of classification algorithms, which are the Adaboost, KNN and Random Forest. Finally, the 10-fold cross-validation is used to evaluate the performance of the classification results.

Traditionally, the 10 fold cross-validation method divides the whole data set into 10 folds, every 9 folds of the data set are used to train the model and the 1 fold left is used to test the model. 10 classification results can be obtained. The final evaluation result is calculated from the weighted average accuracy of the 10 results. It should be noted that in this paper, if the oversampling or subsampling filters are applied to the whole data set, the test data will be ‘‘polluted’’, which makes inaccurate evaluation results. So in our experiments, only the 9 folds of data are oversampled or subsampled to train the machine learning model. After the model is constructed, the 1 fold left is used to evaluation the model.

A. DATASET

The SNARE dataset was downloaded from the UniProt database [67]–[69]. The problem of identification of SNARE proteins can be seen as a classification problem distinguishing SNARE proteins from general proteins, so we collect some general proteins as negative instances. To build a precise model, the negative instances collected need to have a similar structure and function with the positive instances. The vesicular transport proteins are chosen, which are counted as negative instances to perform the classification problem. Finally, a SNARE dataset with positive and negative instances is formed and used to construct the hybrid models.

B. FEATURE EXTRACTION METHODS

1) GROUPED AMINO ACID COMPOSITION (GAAC)

In the GAAC encoding, the 20 kinds of AAs are divided into five classes based on their physicochemical properties, such as molecular size, hydrophobicity and charge. The five classes include the aromatic group (g1: FYW), aliphatic group (g2: GAVLMI), negative charged group (g3: DE), positive charge group (g4: KRH) and uncharged group (g5: STCPNQ). GAAC encoding is the frequency of each amino acid group contained in a protein sequence, which is defined as:

$$f(g) = \frac{N(g)}{N}, \quad g \in \{g1, g2, g3, g4, g5\}$$

where $N(g)$ is the number of AA in group g, N is the length of the protein sequence.

2) COMPOSITION/TRANSITION/DISTRIBUTION (CTDT)

The Composition, Transition and Distribution (CTD) features represent the ACC distribution patterns of a specific physicochemical property or structural in a protein sequence. The final ’T’ in CTDT represents three transition patterns. which are transitions from the polar group to the neutral group, transitions between the neutral group and the hydrophobic group and those between the hydrophobic group and the polar group. The CTDT encoding is the percentage frequency with which the three kinds of transitions happen. The transition descriptor can then be calculated as follows:

$$T(r, s) = \frac{N(r, s) + N(s, r)}{N - 1}$$

where $N(r, s)$ and $N(s, r)$ are the numbers of transitions from ’r’ to ’s’ or vice versa in the sequence, while N is the length of the protein.

3) 188D

As the Amino Acids possess a variety of properties, 188 features are extracted for the cytokine prediction, which is denoted as a 188D Feature Vector (FV).

The first 20 features (1–20) are denoted as $FV_1, \ldots, FV_{20}$:

$$FV_i = \frac{n_i}{L} \quad (i = 1, \ldots, 20)$$

where n_i is the number of the 20 AAs appeared in the sequence and L is the length of the sequence.

Eight kinds of properties are used to extract the 168 features left from a sequence, including the polarity, hydrophobicity, secondary structure, polarizability, surface tension, charge, normalized Van der Waals volume and solvent accessibility. 21 features are extracted according to each kind of physicochemical property, all of which consist of the left 168 features in the 188D.

4) CKSAAP

CKSAAP features are sequence-based features. It computes a pair of ACCs’ frequency separated by k other ACCs ($k = 0, 1, \ldots, 5$). For example, let AB ($k = 0$) represent the combination of two consecutive AACs, f(AB) denotes the frequency of the combination of AB. As there are 20 AACs used to represent the protein, there are $20 = 400$ possible combinations of each two amino acids including the combination of itself. CKSAAP features are encoded by all of the 400 combination frequencies. Thus, the value on each dimension of CKSAAP is the occurrence frequency of each two consecutive AACs in the protein sequence, which is given in the following formula.

$$\left(\frac{N_{AA}}{N_{total}}, \frac{N_{AC}}{N_{total}}, \frac{N_{AD}}{N_{total}}, \ldots, \frac{N_{YY}}{N_{total}} \right)_{400}$$

where numerator denotes the combinations of the consecutive AACs in the protein sequence, N is the length of the protein sequence.

Suppose k equal to 5, the total number of CKSAAP features will be $400 \times 6 = 2400$.

C. SUBSAMPLE AND OVERSAMPLE FILTERING METHODS

Subsampling achieves the balance of positive and negative instances by reducing the number of instances in the majority class. It randomly removes some instances from the majority class to reduce its size. The disadvantage of the subsampling is some information in the majority class, which may be important for classification, can be lost. While oversampling achieves balance of positive and negative instances by increasing the number of instances of the class with fewer instances. Some instances in the minority class will be copied, which will increase the size of minority class. The disadvantage is that the over-fitting problems can be caused.

D. CLASSIFICATION ALGORITHMS

1) ADABOOST

Adaboost is actually a weak classification algorithm promotion process, which improves the classification accuracy through a series of training. A stronger final classifier is composed of many weak classifiers trained by the same training set.

The adaboost algorithm works as follows: the first weak classifier is trained by the N instances in the training set. A new training set is formed by the error-separated instances of the first weak classifier and other instances left in the whole training set. Then the second weak classifier is trained by the new training set. In the same way, a new training set is formed and a third weak classifier is trained. Finally, the final strong classifier is composed of all the weak classifiers, which means the classification of an instance is determined by the weight of each weak classifier.

2) KNN

KNN is an instance-based classification algorithm. The training instances are multi-dimensional feature vectors, where each one carries a class label. During the training phase, the KNN algorithm stores the feature vectors and their labels in an efficient way, which can be easily found. During the classification phase, a vector without a label will be classified according to the class labels of its k nearest neighbors in the multi-dimensional space, where k is a user-defined constant.

3) RANDOM FOREST

The random forest algorithm is an ensemble machine learning algorithm, which can be used to solve regression classification, and other kinds of problems. During the training phase, the random forest algorithm constructs m decision trees with m training sets, which are produced by sampling with replacement from the same training set. The prediction of the random forest is the class predicted by a majority of the decision trees in the forest. Random forest algorithm can overcome the problem of overfitting to the training set of a single decision tree.

III. EXPERIMENTS

In this section, five groups of experiments are done to verify the performance for different combinations of the feature extraction methods, filtering methods and classification algorithms. Four kinds of feature extraction methods used are the GAAC, CTDT, 188D and CKSAAP. For each feature extraction method, three kinds of filtering methods are used, which are the subsampling filter, oversampling filter and no filter methods. Three types of machine learning algorithms are used to classify the filtered data, which are the adaboost, KNN and random forest algorithms.

Furthermore, four types of metrics, which are Sensitivity (SN), Specificity (SP), Accuracy (ACC), and Matthew's correlation coefficient (MCC), are used to evaluate the performance of different combinations.

The sensitivity (or the true positive rate) is defined by formula (1), which is used to measure the probability of actual positives correctly classified. The specificity (or the true negative rate) is defined by formula (2), which is used to measure the probability of actual negatives correctly classified. The accuracy (ACC) is defined by formula (3), which is the proportion of correct predictions to the total number of predictions. The Matthews correlation coefficient (MCC) defined by formula (4) is used to measure the quality of two-class classifications in machine learning algorithms. MCC is a correlation coefficient between the observed and predicted classifications. The range of MCC is between -1 and $+1$. When the value of MCC equals to $+1$, it indicates an exact match between the observation and prediction. 0 represents the pridiction is no better than the random prediction. While the value of MCC equals to -1, it means a total disagreement.

$$SN = \frac{TP}{TP + FN} \tag{1}$$

$$SP = \frac{TN}{TN + FP} \tag{2}$$

$$ACC = \frac{TN + TP}{TN + FP + TP + FN} \tag{3}$$

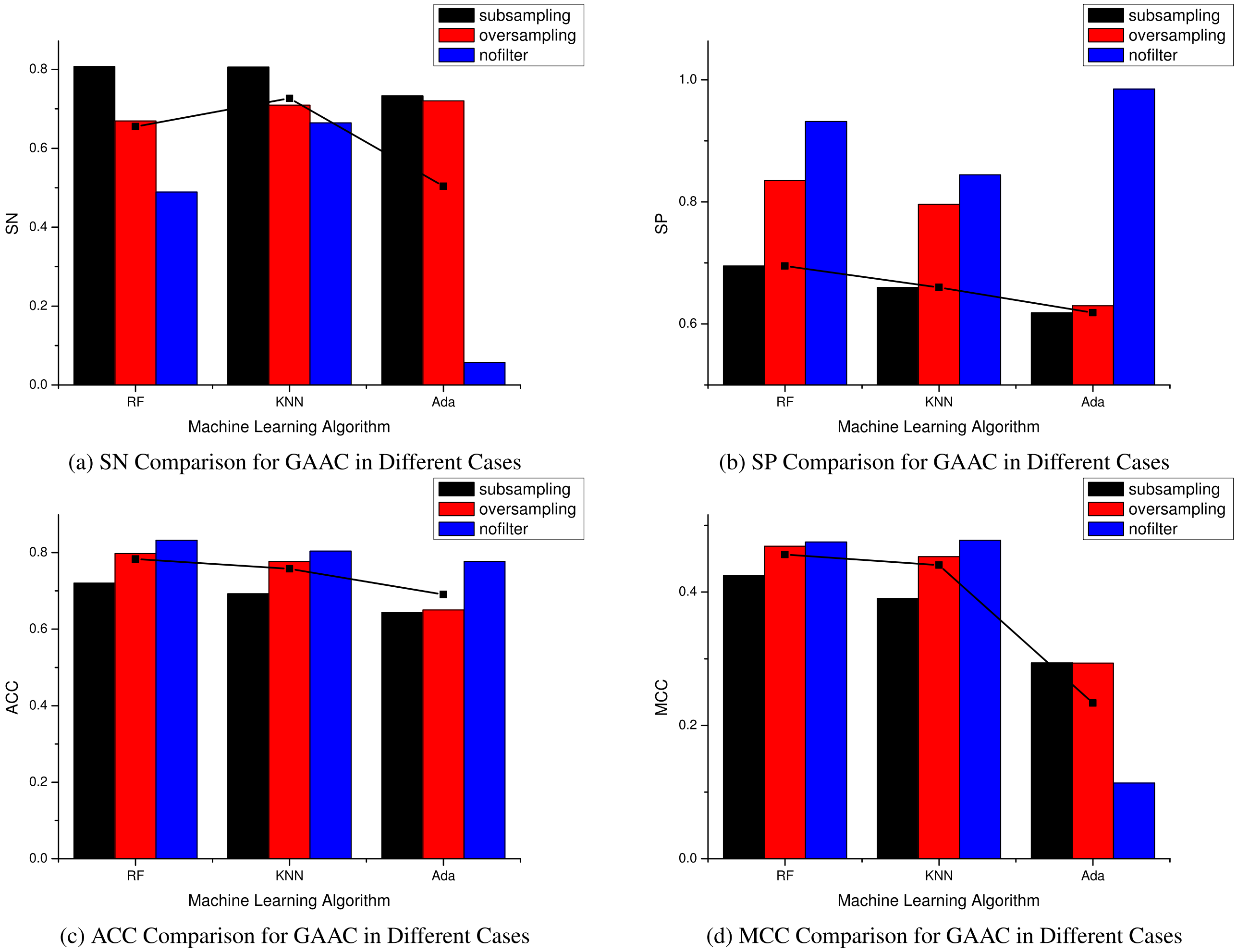
$$MCC = \frac{1 - (\frac{FN}{TP+FN} + \frac{FP}{TN+FP})}{\sqrt{(1 + \frac{FP-FN}{TP+FN})(1 + \frac{FN-FP}{TN+FP})}} \tag{4}$$

where TP (True Positive) is the number of correctly classified SNAREs in the positive data set, FP (False Positive) is the number of instances that are misclassified in the negative data set; TN (True Negative) is the number of correctly classified non-SNAREs in the negative data set; and FN (False Negative) is the number of instances misclassified in the positive data set. The 10-fold cross-validation is used to evaluate the performace of the classification results.

Weka is used to do the experiments. Details of the parameters used in the experiments are shown in table 1.

A. PERFORMANCE OF THE GAAC

In this section, the GAAC method is used to extract the features from the SNAREs data set. After extracting the GAAC features from the positive and negative sequences, the subsampling and oversampling filters are applied to the extracted features. Together with the data set with no filter, we get three feature sets, which are the feature set with subsampling filter, with oversampling filter and with no filter. Three classification algorithms are used to classify the three



(a) SN Comparison for GAAC in Different Cases

(b) SP Comparison for GAAC in Different Cases

(c) ACC Comparison for GAAC in Different Cases

(d) MCC Comparison for GAAC in Different Cases

FIGURE 2. Performance comparison for GAAC in different cases.

TABLE 1. Parameters set for the experiments.

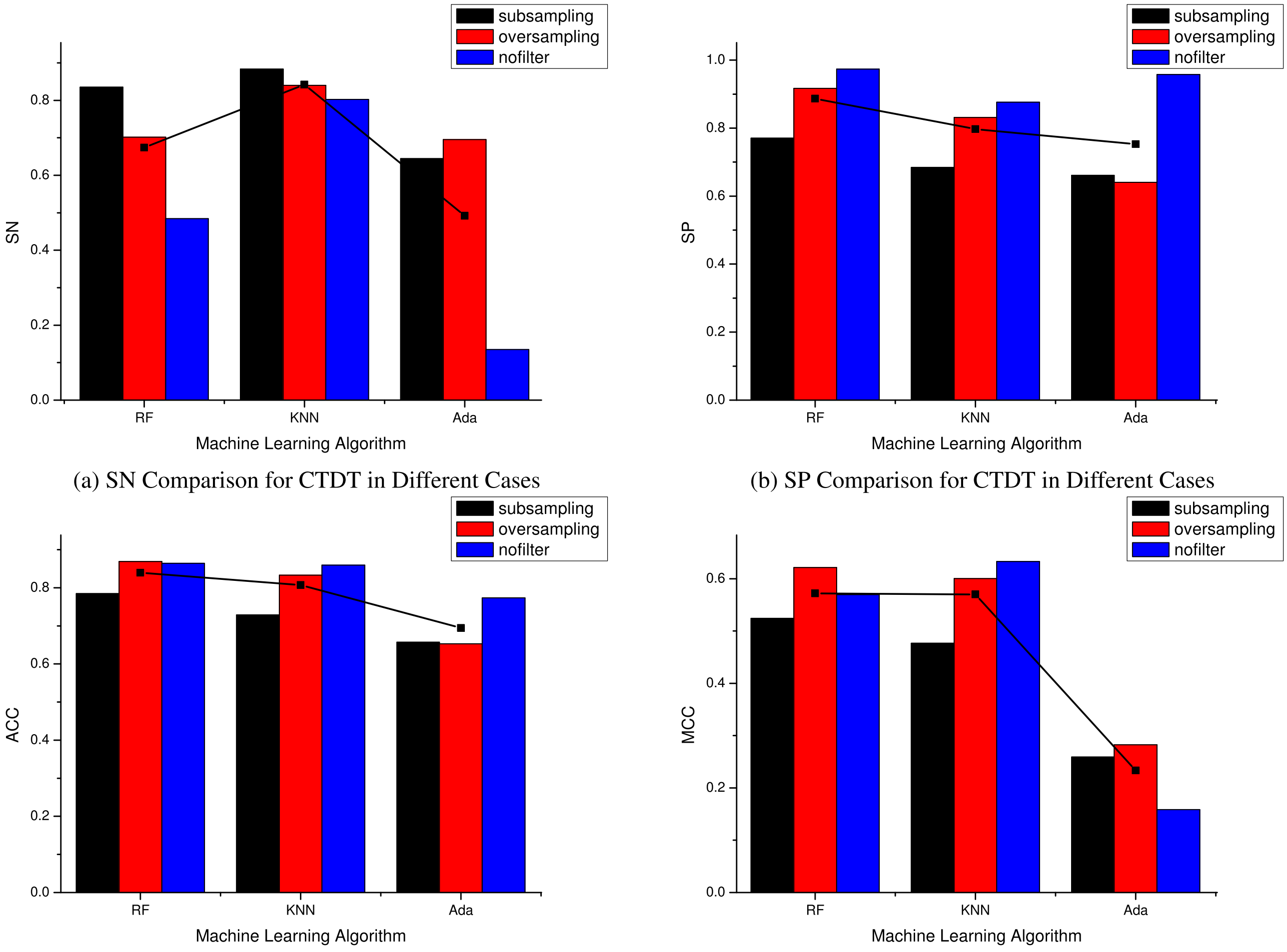
Algorithm	Parameter Name	Paramete Value
Oversampling	biasToUniformClass	1
	noReplacement	False
	sampleSizePercent	160
Subsampling	distributeSpread	1
Adaboost	classifier	DecisionStump
	numDecimalPlaces	2
	numIterations	10
	weightThreshold	100
KNN	nearestNeighbors	1
	distanceFunction	EuclideanDistance
Random Forest	bagSizePercent	100
	numDecimalPlaces	2
	numIterations	100
	numExecutionSlots	1

feature sets, which are the adaboost, KNN and random forest algorithms. The experimental results are shown in figure 2.

The comparison results of the SN for different combinations of the filtering methods and classification algorithms based on GAAC are shown in figure 2. It shows that subsampling method achieves the best performance in all the machine learning algorithm. The oversampling method is in the second place. The no filter method is the worst. To evaluate the performance of a particular machine learning algoirhtm, we calculate the average SN of all three filtering methods for each machine learning algorithm. The average SN values calcuated for each machine learning algorithm are ploted by the line chart in figure 2. The line chart shows that the KNN algorithm is the best among all three machine learning algorithms. The Random Forest algorithm is in the second place.

Figure 2b shows the comparison results of the SP for different filtering methods and classification algorithms based on GAAC. It shows that the performance of the data set with no filter is the best among the three filtering methods for all machine learning algorithms. The oversampling method is in the second place. The line chart in figure 2b shows the random forest achieves the best average SP among the three machine learning algorithms. The KNN algorithm is in the second place.

Figure 2c shows the comparison results of the ACC for different filtering methods and classification algorithms based



(a) SN Comparison for CTDT in Different Cases

(b) SP Comparison for CTDT in Different Cases

(c) ACC Comparison for CTDT in Different Cases

(d) MCC Comparison for CTDT in Different Cases

FIGURE 3. **Performance comparison for CTDT in different cases.**

on GAAC. It shows that the performance of the data set with no filter is the best among the three filtering methods for all machine learning algorithms. The random forest achieves the best average ACC among the three machine learning algorithms.

Figure 2d shows the comparison results of the MCC for different filtering methods and classification algorithms based on GAAC. It shows that the performance of the data set with no filter is the best among the three filtering methods for the random forest and KNN algorithms. The random forest achieves the best average MCC among the three machine learning algorithms.

From the experimental results above, we can find that the random forest and no filtering method is the most suitable model (represented by GAAC-RF-nofilter) for the GAAC to identify the SNARE proteins.

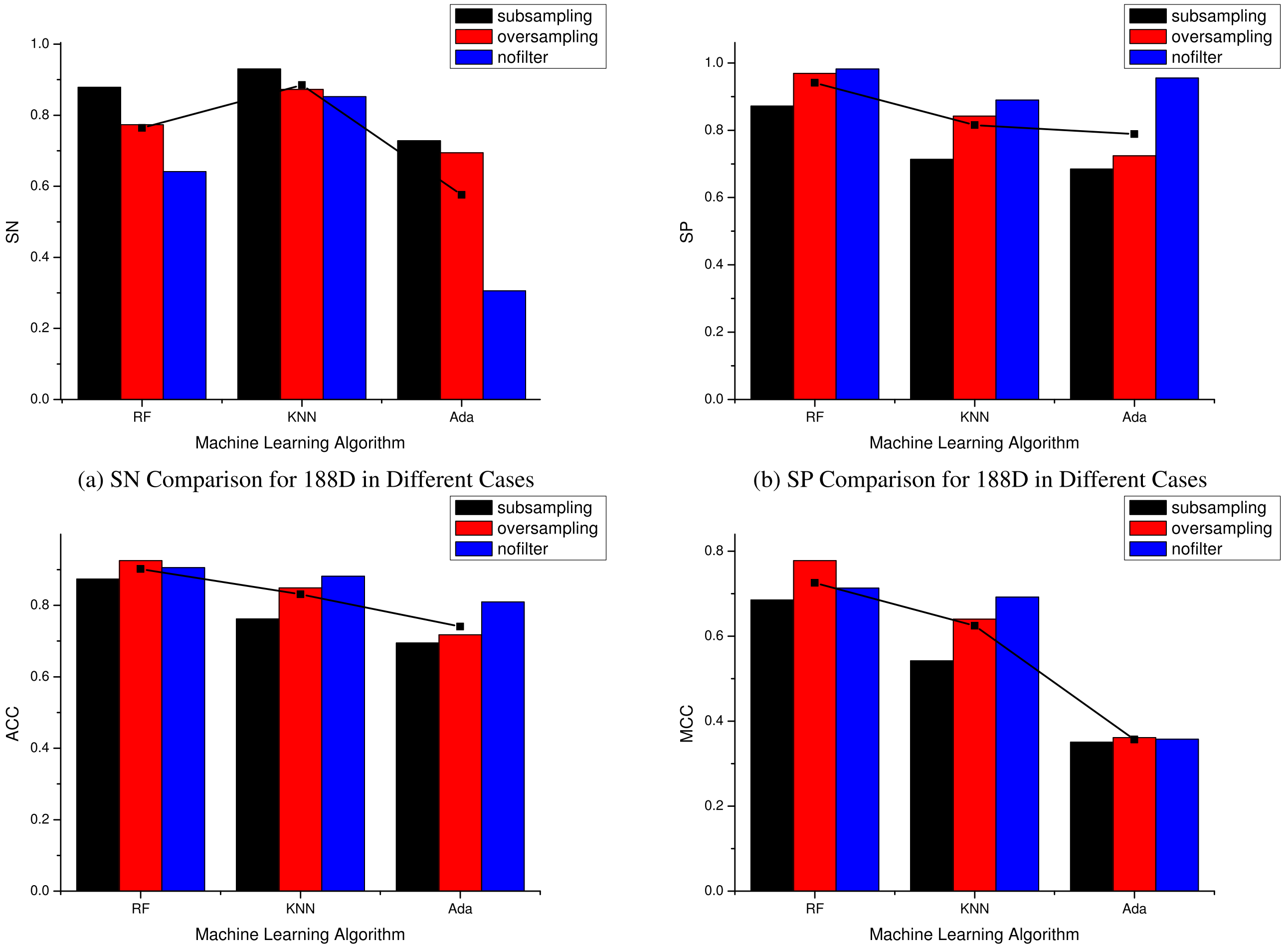
B. PERFORMANCE OF THE CTDT

In this section, the CTDT method is used to extract the features from the SNAREs data set. The subsampling, oversampling filters and no filter are applied to the extracted CTDT features. The adaboost, KNN and random forest classification algorithms are used to classify the filtered feature sets.

The experimental results are shown in figure 3. The comparison results of the SN for different combinations of the filtering methods and classification algorithms based on CTDT are shown in figure 3a. The line chart shows that the KNN algorithm can achieve the best SN when the subsampling filter is applied to the feature set. The performance of subsampling filter is the best among the three filtering methods for random forest and KNN algorithms.

Figure 3b shows the comparison results of the SP for different filtering methods and classification algorithms based on CTDT. It shows that the performance of the data set with no filter is the best among the three filtering methods for all machine learning algorithms. The line chart in figure 3b shows the random forest achieves the best average SP among the three machine learning algorithms.

Figure 3c shows the comparison results of the ACC for different filtering methods and classification algorithms based on CTDT. The line chart shows that the random forest



(a) SN Comparison for 188D in Different Cases

(b) SP Comparison for 188D in Different Cases

(c) ACC Comparison for 188D in Different Cases

(d) MCC Comparison for 188D in Different Cases

FIGURE 4. Performance comparison for 188D in different cases.

algorithm can achieve the best ACC when the oversampling filter is applied to the feature set.

Figure 3d shows the comparison results of the MCC for different filtering methods and classification algorithms based on CTDT. The line chart shows that the random forest algorithm can achieve the best MCC when the oversampling filter is applied to the feature set.

From the experimental results above, we can find that the random forest and oversampling filter method is the most suitable model for the CTDT to identify the SNARE proteins, represented by CTDT-RF-oversample.

C. PERFORMANCE FOR THE 188D

In this section, the 188D method is used to extract the features from the SNAREs data set. The subsampling, oversampling filters and no filter are applied to the extracted CTDT features. The adaboost, KNN and random forest classification algorithms are used to classify the filtered feature sets. The experimental results are shown in figure 4.

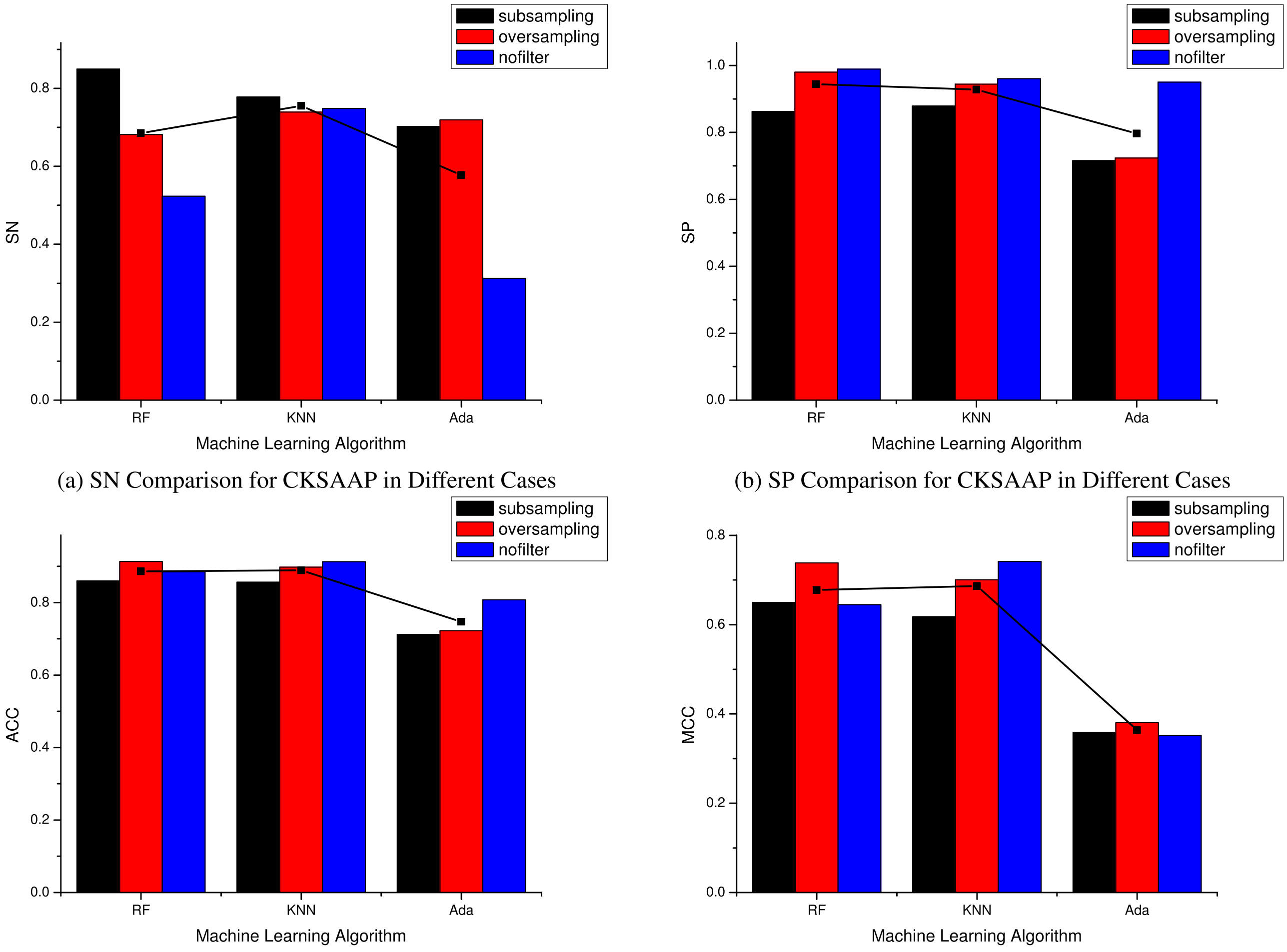
The comparison results of the SN for different combinations of the filtering methods and classification algorithms based on 188D are shown in figure 4a. The line chart shows that the KNN algorithm is the best among all three machine learning algorithms. The performance of subsampling filter is the best among the three filtering methods

Figure 4b shows the comparison results of the SP for different filtering methods and classification algorithms based on 188D. The line chart shows that the random forest algorithm is the best among all three machine learning algorithms. The performance of no filter is the best among the three filtering methods

Figure 4c shows the comparison results of the ACC for different filtering methods and classification algorithms based on 188D. The line chart shows that the random forest algorithm can achieve the best ACC when the oversampling filter is applied to the feature set.

Figure 4d shows the comparison results of the MCC for different filtering methods and classification algorithms based on 188D. The line chart shows that the random forest algorithm can achieve the best MCC when the oversampling filter is applied to the feature set.

From the experimental results above, we can find that the random forest and oversampling filter method is the most



(a) SN Comparison for CKSAAP in Different Cases

(b) SP Comparison for CKSAAP in Different Cases

(c) ACC Comparison for CKSAAP in Different Cases

(d) MCC Comparison for CKSAAP in Different Cases

FIGURE 5. **Performance comparison for CKSAAP in different cases.**

suitable model for the 188D to identify the SNARE proteins, represented by 188D-RF-oversample.

D. PERFORMANCE FOR THE CKSAAP

In this section, the CKSAAP method is used to extract the features from the SNAREs data set. The subsampling, oversampling filters and no filter are applied to the extracted CTDT features. The adaboost, KNN and random forest classification algorithms are used to classify the filtered feature sets.

The experimental results are shown in figure 5. The comparison results of the SN for different combinations of the filtering methods and classification algorithms based on CKSAAP are shown in figure 5a. The line chart shows that the KNN algorithm can achieve the best SN when the subsampling filter is applied to the feature set.

Figure 5b shows the comparison results of the SP for different filtering methods and classification algorithms based on CKSAAP. The line chart shows that the random forest algorithm is the best among all three machine learning algorithms. The performance of no filter is the best among the three filtering methods

Figure 5c shows the comparison results of the ACC for different filtering methods and classification algorithms based on CKSAAP. The line chart shows that the KNN algorithm can achieve the best ACC when the no filter is applied to the feature set.

Figure 5d shows the comparison results of the MCC for different filtering methods and classification algorithms based on CKSAAP. The line chart shows that the KNN algorithm can achieve the best MCC when the no filter is applied to the feature set.

From the experimental results above, we can find that the KNN and no filter method is the most suitable model for the CKSAAP to identify the SNARE proteins, represented by CKSAAP-KNN-nofilter.

E. COMPARISON WITH THE OTHER ALGORITHM

In this section, we compare the performance of the 4 models found by the above experiments with the deep learning model SNARE-CNN. The CNN based method is based on the PSSA feature extraction method. After extracting the PSSA features from the SNARE sequences, a CNN network is trained based on the feature set.

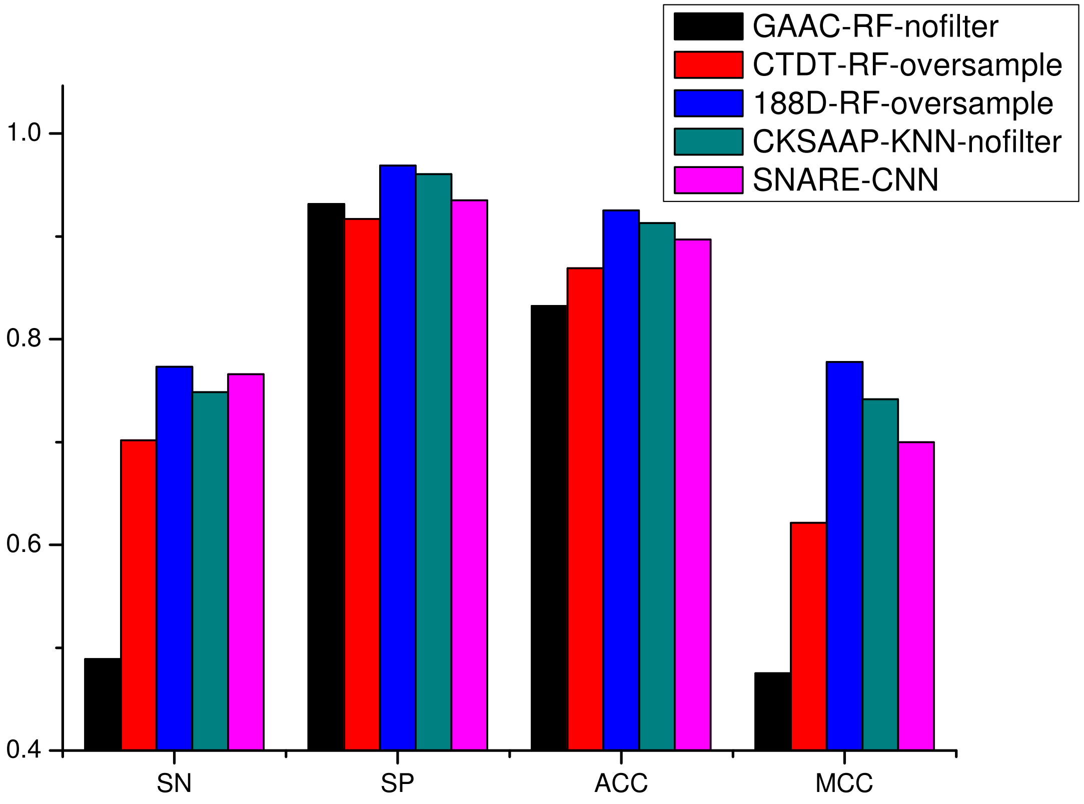


FIGURE 6. Performance comparison among the hybrid model with the SNARE-CNN.

The 10-fold cross validation approach is used to compared the performance of our hybrid model with SNARE-CNN. The comparison reults are shown in figure 6. It shows that, for SN, only the performance of 188D-RF-oversample model is better than the CNN model. For SP, ACC and MCC, the performance of 188D-RF-oversample and CKSAAP-KNN-nofilter is better than the CNN model.

From the results we can see that the performance of the random forest algorithm with oversampling filter based on 188D feature set is better than that of the SNARE-CNN. Our final hybrid model is the random forest algorithm with oversampling filter based on 188D.

IV. CONCLUSION

In this paper, we propose a novel hybrid model, which tries different kinds of combinations of feature extraction methods, filter methods and classification algorithms to identify the SNARE proteins. We find that the performance of the random forest algorithm with oversampling filter and 188D feature extraction method is the best among all combinations, whose performance is better than that of the SNARE-CNN.

REFERENCES

[1] R. Jahn and R. H. Scheller, "SNAREs engines for membrane fusion," *Nature Rev. Mol. Cell Biol.*, vol. 7, pp. 631–643, Aug. 2006.

[2] C. Hou, Y. Wang, J. Liu, C. Wang, and J. Long, "Neurodegenerative disease related proteins have negative effects on SNARE-mediated membrane fusion in pathological confirmation," *Frontiers Mol. Neurosci.*, vol. 10, p. 66, Mar. 2017.

[3] W. G. Honer, P. Falkai, T. A. Bayer, J. Xie, L. Hu, H.-Y. Li, V. Arango, J. J. Mann, A. J. Dwork, and W. S. Trimble, "Abnormalities of SNARE mechanism proteins in anterior frontal cortex in severe mental illness," *Cerebral Cortex*, vol. 12, no. 4, pp. 349–356, Apr. 2002.

[4] J. Meng and J. Wang, "Role of SNARE proteins in tumourigenesis and their potential as targets for novel anti-cancer therapeutics," *Biochim. et Biophys. Acta (BBA) Rev. Cancer*, vol. 1856, no. 1, pp. 1–12, Aug. 2015.

[5] Q. Sun, X. Huang, Q. Zhang, J. Qu, Y. Shen, X. Wang, H. Sun, J. Wang, L. Xu, X. Chen, and B. Ren, "SNAP23 promotes the malignant process of ovarian cancer," *J. Ovarian Res.*, vol. 9, no. 1, p. 80, Dec. 2016.

[6] T. Weimbs, S. H. Low, S. J. Chapin, K. E. Mostov, P. Bucher, and K. Hofmann, "A conserved domain is present in different families of vesicular fusion proteins: A new superfamily," *Proc. Nat. Acad. Sci. USA*, vol. 94, no. 7, pp. 3046–3051, Apr. 1997.

[7] A. C. Yoshizawa, S. Kawashima, S. Okuda, M. Fujita, M. Itoh, Y. Moriya, M. Hattori, and M. Kanehisa, "Extracting sequence motifs and the phylogenetic features of SNARE-dependent membrane traffic," *Traffic*, vol. 7, no. 8, pp. 1104–1118, Aug. 2006.

[8] A. D. J. van Dijk, D. Bosch, C. J. F. ter Braak, A. R. van der Krol, and R. C. H. J. van Ham, "Predicting sub-golgi localization of type II membrane proteins," *Bioinformatics*, vol. 24, no. 16, pp. 1779–1786, Aug. 2008.

[9] T. H. Kloepper, C. N. Kienle, and D. Fasshauer, "An elaborate classification of SNARE proteins sheds light on the conservation of the eukaryotic endomembrane system," *Mol. Biol. Cell*, vol. 18, no. 9, pp. 3463–3471, Sep. 2007.

[10] T. H. Kloepper, C. N. Kienle, and D. Fasshauer, "SNAREing the basis of multicellularity: Consequences of protein family expansion during evolution," *Mol. Biol. Evol.*, vol. 25, no. 9, pp. 2055–2068, Jun. 2008.

[11] X. Shi, P. Halder, H. Yavuz, R. Jahn, and H. A. Shuman, "Direct targeting of membrane fusion by SNARE mimicry: Convergent evolution of legionella effectors," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 31, pp. 8807–8812, Aug. 2016.

[12] B. Lu, "The destructive effect of botulinum neurotoxins on the SNARE protein: SNAP-25 and synaptic membrane fusion," *PeerJ*, vol. 3, Jun. 2015, Art. no. e1065.

[13] W. Chen, H. Ding, P. Feng, H. Lin, and C. Kuo-Chen, "iACP: A sequence-based tool for identifying anticancer peptides," *Oncotarget*, vol. 7, no. 13, 2016, Art. no. 016895.

[14] Y. Ding, J. Tang, and F. Guo, "Identification of protein–protein interactions via a novel matrix-based sequence representation model with amino acid contact information," *Int. J. Mol. Sci.*, vol. 17, no. 10, p. 1623, Sep. 2016.

[15] W. Chen, H. Lv, F. Nie, and H. Lin, "I6 mA-pred: Identifying DNA N6-methyladenine sites in the rice genome," *Bioinformatics*, vol. 35, no. 16, pp. 2796–2800, Aug. 2019.

[16] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identification of human MicroRNAs by incorporating a high-quality negative set," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 1, pp. 192–201, Jan. 2014.

[17] L. Wei, C. Zhou, H. Chen, J. Song, and R. Su, "ACPred-FL: A sequence-based predictor using effective feature representation to improve the prediction of anti-cancer peptides," *Bioinformatics*, vol. 34, no. 23, pp. 4007–4016, Jun. 2018.

[18] Y. Xiong, Q. Wang, J. Yang, X. Zhu, and D.-Q. Wei, "PredT4SE-stack: Prediction of bacterial type IV secreted effectors from protein sequences using a stacked ensemble method," *Frontiers Microbiol.*, vol. 9, p. 2571, Oct. 2018.

[19] Y. Xiong, J. Liu, W. Zhang, and T. Zeng, "Prediction of heme binding residues from protein sequences with integrative sequence profiles," *Proteome Sci*, vol. 10, p. S20, Jun. 2012.

[20] X. Zhu, J. He, S. Zhao, W. Tao, Y. Xiong, and S. Bi, "A comprehensive comparison and analysis of computational predictors for RNA N6-methyladenosine sites of saccharomyces cerevisiae," *Briefings Funct. Genomics*, vol. 18, no. 6, pp. 367–376, Oct. 2019.

[21] Z. Liao, D. Li, X. Wang, L. Li, and Q. Zou, "Cancer diagnosis through IsomiR expression with machine learning method," *Current Bioinf.*, vol. 13, no. 1, pp. 57–63, Feb. 2018.

[22] L. Chao, L. Wei, and Q. Zou, "SecProMTB: A SVM-based Classifier for Secretory Proteins of Mycobacterium tuberculosis with Data Set," *Proteomics*, vol. 19, Aug. 2019, Art. no. e1900007.

[23] H. Bu, J. Hao, J. Guan, and S. Zhou, "Predicting enhancers from multiple cell lines and tissues across different developmental stages based on SVM method," *Current Bioinf.*, vol. 13, no. 6, pp. 655–660, Nov. 2018.

[24] C. Meng, S. Jin, L. Wang, F. Guo, and Q. Zou, "AOPs-SVM: A sequence-based classifier of antioxidant proteins using a support vector machine," *Frontiers Bioeng. Biotechnol.*, vol. 7, p. 224, Sep. 2019.

[25] L. Wei, Q. Zou, M. Liao, H. Lu, and Y. Zhao, "A novel machine learning method for cytokine-receptor interaction prediction," *Combinat. Chem. High Throughput Screening*, vol. 19, no. 2, pp. 144–152, Jan. 2016.

[26] B. Liu, C.-C. Li, and K. Yan, "DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks," *Briefings Bioinf.*, Oct. 2019, doi: 10.1093/bib/bbz098.

[27] B. Liu, S. Chen, K. Yan, and F. Weng, "IRO-PsekGCC: Identify DNA replication origins based on pseudo k-tuple GC composition," *Frontiers Genet.*, vol. 10, p. 842, Sep. 2019.

[28] Y. Cao, S. Wang, Z. Guo, T. Huang, and S. Wen, "Synchronization of memristive neural networks with leakage delay and parameters mismatch via event-triggered control," *Neural Netw.*, vol. 119, pp. 178–189, Nov. 2019.

[29] X. Zeng, N. Ding, A. Rodríguez-Patón, and Q. Zou, "Probability-based collaborative filtering model for predicting gene-disease associations," *BMC Med. Genomics*, vol. 10, no. S5, p. 76, Dec. 2017.

[30] X. Zhang, Q. Zou, A. Rodriguez-Paton, and X. Zeng, "Meta-path methods for prioritizing candidate disease miRNAs," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 1, pp. 283–291, Jan. 2019.

[31] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: A survey," *Briefings Funct. Genomics*, vol. 15, no. 1, pp. 55–64, 2015.

[32] R. Cao, Z. Wang, Y. Wang, and J. Cheng, "SMOQ: A tool for predicting the absolute residue-specific quality of a single protein model with support vector machines," *BMC Bioinf.*, vol. 15, no. 1, p. 120, 2014.

[33] N. Q. K. Le and V.-N. Nguyen, "SNARE-CNN: A 2D convolutional neural network architecture to identify SNARE proteins from high-throughput sequencing data," *PeerJ Comput. Sci.*, vol. 5, p. e177, Feb. 2019.

[34] T.-Y. Lee, S.-A. Chen, H.-Y. Hung, and Y.-Y. Ou, "Incorporating distant sequence features and radial basis function networks to identify ubiquitin conjugation sites," *PLoS ONE*, vol. 6, no. 3, Mar. 2011, Art. no. e17331.

[35] I. Dubchak, I. Muchnik, S. R. Holbrook, and and S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proc. Nat. Acad. Sci. USA*, vol. 92, pp. 8700–8704, Sep. 1995.

[36] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S.-H. Kim, "Recognition of a protein fold in the context of the Structural Classification of Proteins (SCOP) classification," *Proteins*, vol. 35, pp. 401–407, Jun. 1999.

[37] C. Z. Cai, L. Y. Han, Z. L. Ji, X. Chen, Y. Z. Chen, "SVM-prot: Web-based support vector machine software for functional classification of a protein from its primary sequence," *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3692–3697, Jul. 2003.

[38] K. Chen, Y. Jiang, L. Du, and L. Kurgan, "Prediction of integral membrane protein type by collocated hydrophobic amino acid pairs," *J. Comput. Chem.*, vol. 30, no. 1, pp. 163–172, Jan. 2009.

[39] K. Chen, L. Kurgan, and M. Rahbari, "Prediction of protein crystallization using collocation of amino acid pairs," *Biochem. Biophys. Res. Commun.*, vol. 355, no. 3, pp. 764–769, Apr. 2007.

[40] K. Chen, L. A. Kurgan, and J. Ruan, "Prediction of flexible/rigid regions from protein sequences using k-spaced amino acid pairs," *BMC Struct. Biol.*, vol. 7, no. 1, p. 25, 2007.

[41] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.

[42] G. Wang, Y. Wang, W. Feng, X. Wang, J. Y. Yang, Y. Zhao, Y. Wang, and Y. Liu, "Transcription factor and microRNA regulation in androgen-dependent and -independent prostate cancer cells," *BMC Genomics*, vol. 9, no. 2, p. S22, 2008.

[43] J. Qh, W. Gh, J. Sl, Y. Li, and Y. Wang, "Predicting human microRNA-disease associations based on support vector machine," *Int. J. Data Mining Bioinf.*, vol. 8, no. 3, pp. 282–293, 2013.

[44] L. Xu, G. Liang, S. Shi, and C. Liao, "SeqSVM: A sequence-based support vector machine method for identifying antioxidant proteins," *Int. J. Mol. Sci.*, vol. 19, no. 6, p. 1773, Jun. 2018.

[45] L. Xu, G. Liang, L. Wang, and C. Liao, "A novel hybrid sequence-based model for identifying anticancer peptides," *Genes*, vol. 9, no. 3, p. 158, Mar. 2018.

[46] L. Dou, X. Li, H. Ding, L. Xu, and H. Xiang, "Is there any sequence feature in the RNA pseudouridine modification prediction problem?" *Mol. Therapy Nucleic Acids*, vol. 19, pp. 293–303, Mar. 2020.

[47] L. Yu, S. Yao, L. Gao, and Y. Zha, "Conserved disease modules extracted from multilayer heterogeneous disease and gene networks for understanding disease mechanisms and predicting disease treatments," *Frontiers Genet.*, vol. 9, p. 745, Jan. 2019.

[48] L. Yu, J. Zhao, and L. Gao, "Predicting potential drugs for breast cancer based on miRNA and tissue specificity," *Int. J. Biol. Sci.*, vol. 14, no. 8, pp. 971–980, 2018.

[49] L. Yu and L. Gao, "Human pathway-based disease network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 4, pp. 1240–1249, Jul. 2019.

[50] W. Chen, P. Feng, T. Liu, and D. Jin, "Recent advances in machine learning methods for predicting heat shock proteins," *Current Drug Metabolism*, vol. 20, no. 3, pp. 224–228, May 2019.

[51] X. Zeng, W. Lin, M. Guo, and Q. Zou, "A comprehensive overview and evaluation of circular RNA detection tools," *PLOS Comput. Biol.*, vol. 13, no. 6, Jun. 2017, Art. no. e1005420.

[52] L. Wei, P. Xing, G. Shi, Z. Ji, and Q. Zou, "Fast prediction of protein methylation sites using a sequence-based feature selection technique," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 4, pp. 1264–1273, Jul. 2019.

[53] L. Wei, P. Xing, R. Su, G. Shi, Z. S. Ma, and Q. Zou, "CPPred-RF: A sequence-based predictor for identifying cell-penetrating peptides and their uptake efficiency," *J. Proteome Res.*, vol. 16, no. 5, pp. 2044–2053, May 2017.

[54] 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," *Artif. Intell. Med.*, vol. 83, pp. 67–74, Nov. 2017.

[55] J. He, T. Fang, Z. Zhang, B. Huang, X. Zhu, and Y. Xiong, "PseUI: Pseudouridine sites identification based on RNA sequence information," *BMC Bioinf.*, vol. 19, no. 1, p. 306, Dec. 2018.

[56] Q. Xu, Y. Xiong, H. Dai, K. M. Kumari, Q. Xu, H.-Y. Ou, and D.-Q. Wei, "PDC-SGB: Prediction of effective drug combinations using a stochastic gradient boosting algorithm," *J. Theor. Biol.*, vol. 417, pp. 1–7, Mar. 2017.

[57] J. Zhang and B. Liu, "A review on the recent developments of sequence-based protein feature extraction methods," *Current Bioinf.*, vol. 14, no. 3, pp. 190–199, Mar. 2019.

[58] B. Liu, X. Gao, and H. Zhang, "BioSeq-Analysis2.0: An updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches," *Nucleic Acids Res.*, vol. 47, no. 20, p. e127, Nov. 2019.

[59] B. Liu, Y. Zhu, and K. Yan, ''Fold-LTR-TCP: Protein fold recognition based on triadic closure principle,'' *Briefings Bioinf.*, Dec. 2019, doi: 10.1093/bib/bbz139.

[60] Y. Wang, S. Yang, J. Zhao, W. Du, Y. Liang, C. Wang, F. Zhou, Y. Tian, and Q. Ma, ''Using machine learning to measure relatedness between genes: A multi-features model,'' *Sci. Rep.*, vol. 9, no. 1, p. 4192, Dec. 2019.

[61] X. Zeng, S. Zhu, X. Liu, Y. Zhou, R. Nussinov, and F. Cheng, ''DeepDR: A network-based deep learning approach to in silico drug repositioning,'' *Bioinformatics*, vol. 35, no. 24, pp. 5191–5198, Dec. 2019, doi: 10.1093/bioinformatics/btz418.

[62] P. Zhu, Q. Hu, Q. Hu, C. Zhang, and Z. Feng, ''Multi-view label embedding,'' *Pattern Recognit.*, vol. 84, pp. 126–135, Dec. 2018.

[63] P. Zhu, Q. Hu, Y. Han, C. Zhang, and Y. Du, ''Combining neighborhood separable subspaces for classification via sparsity regularized optimization,'' *Inf. Sci.*, vols. 370–371, pp. 270–287, Nov. 2016.

[64] P. Zhu, Q. Xu, Q. Hu, and C. Zhang, ''Co-regularized unsupervised feature selection,'' *Neurocomputing*, vol. 275, pp. 2855–2863, Jan. 2018.

[65] P. Zhu, Q. Xu, Q. Hu, C. Zhang, and H. Zhao, ''Multi-label feature selection with missing labels,'' *Pattern Recognit.*, vol. 74, pp. 488–502, Feb. 2018.

[66] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, ''Subspace clustering guided unsupervised feature selection,'' *Pattern Recognit.*, vol. 66, pp. 364–374, Jun. 2017.

[67] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, and M. J. Martin, ''The universal protein resource (UniProt),'' *Nucleic Acids Res.*, vol. 33, pp. 154–159, Jan. 2005.

[68] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, and M. J. Martin, ''UniProt: The universal protein knowledgebase,'' *Nucleic Acids Res.*, vol. 32, pp. 115–119, Jan. 2004.

[69] C. H. Wu, R. Apweiler, A. Bairoch, D. A. Natale, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, and M. Magrane, ''The universal protein resource (UniProt): An expanding universe of protein information,'' *Nucleic Acids Res.*, vol. 34, pp. 187–191, Jan. 2006.

GUILIN LI was born in Harbin, Heilongjiang, China, in 1979. He received the B.S. and M.S. degrees in computer science and technology and the Ph.D. degree in computer software from the Harbin Institute of Technology, Harbin, in 2003 and 2009, respectively.

From 2009 to 2013, he was an Assistant Professor with the Software Department, Xiamen University, Fujian, China. Since 2013, he has been an Associate Professor with the School of Informatics, Xiamen University. He is the author of more than 30 articles. His research interests include bioinformatics, feature engineering, machine learning, and deep learning.