Received June 5, 2020, accepted June 14, 2020, date of publication June 17, 2020, date of current version June 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003086

A KNN Model Based on Manhattan Distance to Identify the SNARE Proteins

XING GAO, (Member, IEEE), AND GUILIN LI
Department of Software Engineering, School of Informatics, Xiamen University, Xiamen 361005, China
Corresponding author: Guilin Li (glli@xmu.edu.cn)

ABSTRACT SNARE proteins, known as membrane fusion proteins, play a primary role to mediate vesicle fusion. Loss of function of the SNARE protein can lead to a variety of diseases. A method to accurately identify the SNARE protein is important and necessary. In this paper, we try different kinds of combinations of sampling methods (the resampling, SMOTE and no sampling), feature extraction approaches (the 188D, K-skip-2-gram and CKSAAP) and distance measurements (Chebyshev distance, Euclidean distance, Manhattan distance and Minkowski distance) to find a suitable model for identifying the SNARE proteins. By doing extensive experiments, we construct a Manhattan distance based KNN model by combining the CKSAAP feature extraction approach with no sampling method, which achieves the best identification performance among all combinations. Finally, we compare our KNN based model with a deep learning based model (called SNARE-CNN) from SN, SP, ACC and MCC four aspects, the experimental results show that the performance of our model is better than that of the SNARE-CNN.

INDEX TERMS Distance measurement, feature representation, SNARE protein identification.

I. INTRODUCTION

SNARE proteins, known as membrane fusion proteins, play a primary role to mediate vesicle fusion [1], [2]. Researchers have found that loss of function of the SNARE protein can lead to a variety of diseases, such as neurodegenerative diseases, mental diseases, cancer, etc. [3], [4]. Therefore, SNARE proteins are very important for human health and it is necessary to construct an accurate model to identify them. Researchers have done a lot of related works to identify the SNARE proteins from different aspects [2], [5]–[12]. But most of the works are from a biological point of view. As we know, machine learning algorithms have been widely used in the classification problem and they have successfully been used to identify different kinds of proteins [13]–[31], [66], [67]. Compared with the biological method, the machine learning method can save time and inexpensive.

In this paper, we propose a machine learning model based on KNN algorithm [32] to accurately identify the SNARE proteins. First, the SNARE dataset constructed by [31] is used to train and test the model. As the dataset is imbalanced, which means the number of negative instances is far more than the number of positive instances. Two kinds of sampling methods are adopted to balance the data set, which are the resampling and SMOTE methods [33]. The resampling method can achieve a balanced data set by enlarging the minority class or condensing the majority class. In this paper, the former is adopted. While the SMOTE method synthesizes some new instances by interpolation based on the original data in the minority class. First, we need to know whether the resampling and SMOTE methods are more effective to identify the SNARE than no sampling method. Second, to train the KNN model, features need to be extracted from the SNARE proteins. There are a lot of feature extraction methods proposed for identifying proteins [34]–[69]. In this paper, three kinds of feature extraction methods are used, which are the 188D [37], K-skip-2-gram [38] and CKSAAP [39], [40]. The 188D method considers both the structure of a SNARE sequence and the physicochemical property of the sequence when extracting the features. The K-skip-2-gram and CKSAAP only consider the structure of a SNARE sequence. We need to determine which extraction approach is the best for the identification. Finally, the KNN algorithm is used to learn the model. As KNN uses the distance between two instances to classify an instance, it needs some distance

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

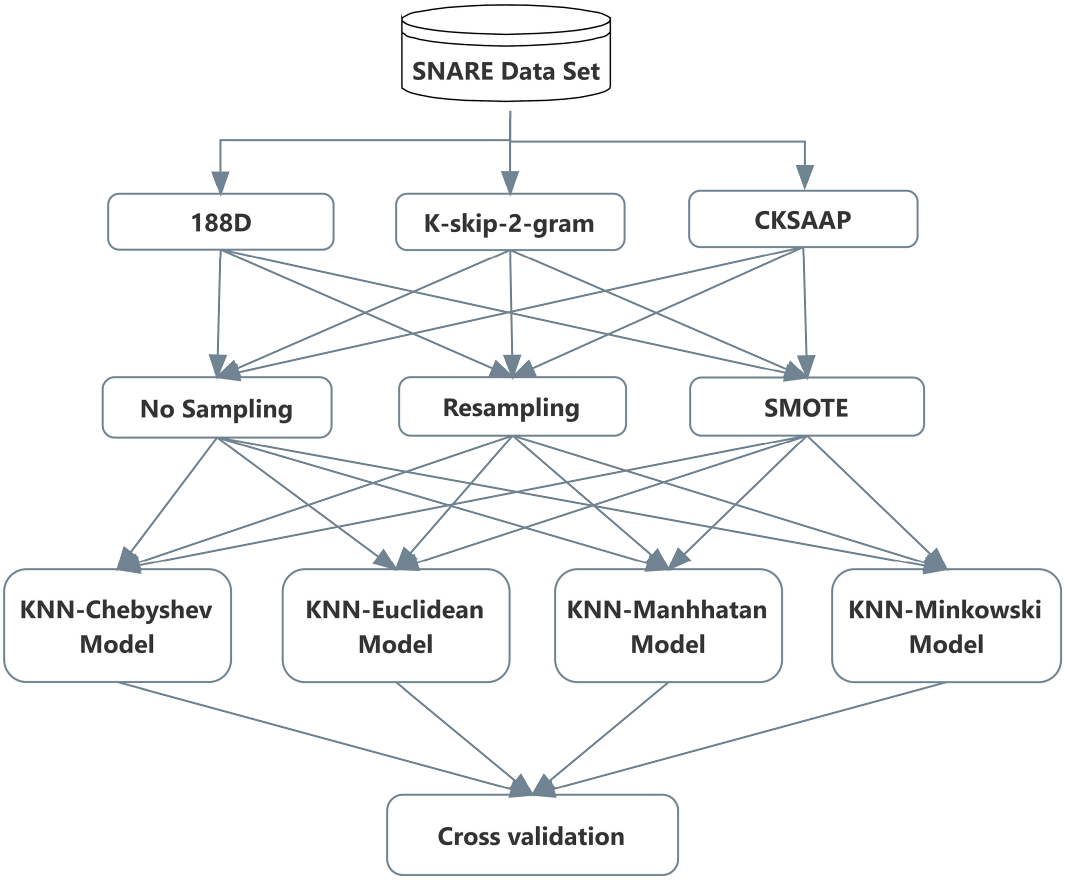


FIGURE 1. Overview of the framework for SNARE protein classification.

measurements. Four kinds of distances are used, which are the Chebyshev distance, Euclidean distance, Manhattan distance and Minkowski distance. We need to determine which one is best to do the SNARE classification.

To solve the three problems, extensive experiments are done. We try different kinds of combinations of the sampling methods, feature extraction approaches and the distance measurements. First, we find that the performance no sampling method is always better than the resampling and SMOTE methods for different feature extraction approaches and distance measurements. The SMOTE method shows better performance when 188D and K-skip-2-gram feature extraction methods are used. The resampling method is more suitable for the CKSAAP method. Second, we find the CKSAAP method is more suitable for the SNARE classification problem. Even though the 188D considers both the structure and physicochemical property of a sequence, the CKSAAP only considers the structure of a sequence. Finally, we find that the Manhattan distance is the best among the four distance measurements for identifying the SNARE proteins. In summary, we constructed a KNN model by combining the Manhattan distance, CKSAAP feature extraction approach with no sampling method, which achieves the best identification performance. Finally, we compare our model with a deep learning based model called SNARE-CNN [31], the experimental results show that the performance of our model is also better than that of the SNARE-CNN.

The contributions of this work include (1) Extensive experiments are done to test the performance of different feature extraction methods, sampling methods and distance measurements of the KNN algorithm to identify the SNARE proteins. (2) Experimental results show that the performance of the KNN model based on Manhattan distance, no sampling method and CKSAAP feature extraction approach is the best one among all models. (3) Compared with a deep learning based model named SNARE-CNN, the performance of our model is better.

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

II. METHODS

Figure 1 shows the framework for the SNARE protein classification. First, three kinds of feature extraction methods, named 188D, k-skip-2-gram and CKSAAP are used to extract the features from the SNARE data set. As the number of positive and negative instances in the dataset is imbalanced, two kinds of sampling methods, which are the SMOTE and resampling methods, are applied to the data set to make the instances in the dataset balance. The original imbalanced data set is also used (no sampling in figure 1). Then, the three data sets are used to train the models by KNN algorithms, respectively. Four kinds of distance measurements are used to compute the distance between two instances in the data set in KNN. Finally, the cross-validation approach is used to evaluate the performance of the models.

A. DATASET

From the UniProt database, a set of SNARE proteins are downloaded, which consists of the positive instances. To build a precise classification model, a set of general proteins are collected as negative instances, which have a similar structure and function with the positive instances. The vesicular transport proteins are chosen, whose number is much larger than the positive instances. Finally, an imbalanced data set is formed and used to construct the KNN models.

B. FEATURE EXTRACTION METHODS

1) 188D

Based on the composition of proteins and physicochemical properties, a 188D Feature Vector FV can be extracted from a SNARE protein. The first 20 features, represented as $FV_1, \ldots, FV_{20}$, are extracted based on the composition of the amino acids and calculated by the following formula:

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

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

The 168 features left are extracted based on eight kinds of physicochemical properties of the protein, such as the hydrophobicity, polarity, polarizability, surface tension, secondary structure etc.. Each property contributes 21 features. For example, Features from FV_{21} to FV_{41} are extracted based on hydrophobicity property and are calculated as follows:

$$(FV_{21}, FV_{22}, FV_{23}) = (\frac{CH_1}{L}, \frac{CH_2}{L}, \frac{CH_3}{L})$$

where CH_1, CH_2, and CH_3 are the size of three groups.

$$\begin{aligned}&(FV_{24}, \ldots, FV_{28}; FV_{29}, \ldots, FV_{33}; FV_{34}, \ldots, FV_{38})\\&\quad = (\frac{DH_{11}}{L}, \ldots, \frac{DH_{15}}{L}; \frac{DH_{21}}{L},\\&\qquad \ldots, \frac{DH_{25}}{L}; \frac{DH_{31}}{L}, \ldots, \frac{DH_{35}}{L})\end{aligned}$$

where the $DH_{ij}(i = 1, 2, 3; j = 1, 2, \ldots, 5)$ represents the sequence length, at which the 1st, 25%, 50%, 75%, and 100% of AAs in three groups are located.

$$(FV_{39}, FV_{40}, FV_{41}) = (\frac{FH_1}{L-1}, \frac{FH_2}{L-1}, \frac{FH_3}{L-1})$$

where $(L-1)$ represents the number of bivalent seeds and the $FH_i(i = 1, 2, 3)$ represents the respective number of bivalent seeds containing two AAs from different groups.

2) K-SKIP-2-GRAM

400 features are extracted based on the k-skip-2-gram model. It considers the composition of any 2 amino acids whose distance is less than k. The distance of two amino acids A_i and A_j in a SNARE sequence is defined as:

$$DT\left(A_i, A_j\right) = j - i - 1$$

$Skip(DT = a)$ of k-skip-2-gram model is defined as:

$$Skip(DT = a) = \{A_i A_{i+a+1} | 1 \le a \le k, 1 \le i \le L - a\}$$

A set $T_{SkipGram}$ is defined as:

$$T_{\mathrm{SkipGram}} = \left\{\bigcup_{a=1}^{k} \mathrm{Skip}(DT = a)\right\}$$

The k-skip-2-gram feature sets can be calculated by the following formula.

$$FV_{SkipGram} = \frac{N\left(A_i A_j\right)}{N\left(T_{SkipGram}\right)}$$

where $N(T_{SkipGram})$ is the total number of all elements in the set $T_{SkipGram}$, and $N\left(A_i A_j\right)$ is the number of amino acid subsequence of 2 length appearing in the set $T_{SkipGram}$.

3) CKSAAP

CKSAAP computes the frequency for a pair of amino acids separated by k other amino acids ($k = 0, 1, \ldots, 5$). For instance, in the case of $k = 0$, two amino acids are successive, which can be denoted as $A_i A_j$. let $f(A_i A_j)$ be the frequency of $A_i A_j$ appearing a protein sequence. As there are 20 kinds of amino acids, there are $20 \times 20 = 400$ possible combinations of each two amino acids including the combination of itself. CKSAAP calculates the frequency of occurrence for each combination of AAs for a 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 $k = 5$, the total number of CKSAAP features will be $400 \times 6 = 2400$.

C. SAMPLING METHODS

As the dataset is imbalanced, which can affect the performance of the classification algorithm, we apply two kinds of methods to balance the dataset named Resample and SMOTE.

1) RESAMPLE

Resample achieves the balance of data set by reducing the number of majority samples which randomly removes some majority samples to reduce the size of the majority class or by increasing the number of samples in the minority class, which oversamples the data in minority class. In this paper, we use the latter one.

2) SMOTE

The Synthetic Minority Oversampling Technique (SMOTE) is a synthetic oversampling technology for minority class, which is an improved solution to the random oversampling algorithm. For each instance x in the minority class, SMOTE calculates the Euclidean distance between x and all the other instances in the minority class to determine its k-nearest neighbors. Based on the imbalance ration, a sampling ration N is set. Several instances are selected for x from its k-nearest neighbors x_n. Finally, a new instance x_{new} is constructed according to x and x_n according to the following formula.

$$x_{new} = x + rand(0, 1) * |x - x_n|$$

D. KNN ALGORITHM

The K Nearest Neighbor (KNN) classification algorithm, is one of the most popular methods in data mining classification technology. The main idea is that if most of the k nearest neighbors of an instance to be classified in the feature space belong to a certain class, then the instance to be classified also belongs to the class. The KNN made the classification decision only based on a small number of adjacent instances. Because the KNN algorithm mainly depends on the limited neighbor instances around, rather than identifying the class domain to determine the classification, it is more suitable for the data set with overlapping among different classes.

The most important thing for KNN algorithm is the method to measure the distance between two instances. Different distance measurements can result in different set of k nearest neighbors. In this paper, we use four kinds of distance measurements, which are the Chebyshev distance, Euclidean distance, Manhattan distance and Minkowski distance.

The Chebyshev distance is given by Formula 1

$$D_{\text{Chebyshev}}(x, y) = \max_i (|x_i - y_i|) \tag{1}$$

The Euclidean distance given by Formula 2 is the most familiar to us.

$$D_{\text{Euclidean}}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

The Manhattan distance is given by Formula 3

$$D_{\text{Manhattan}}(x, y) = \|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

The Minkowski distance is given by Formula 4

$$D_{\text{Minkowski}}(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{4}$$

The x and y from Formula 1 to 4 represent two vectors in the feature space and x_i and y_i are their coordinates respectively.

III. EXPERIMENTS

In this section, we do four groups of experiments to test the classification performance of KNN models combining different kinds of distance measurements, feature extraction methods and sampling methods. Three kinds of feature extraction methods used are the 188D, K-skip-2-gram and CKSAAP. For each feature extraction method, three kinds of sampling methods are used, which are the SMOTE and resampling and no sampling methods. Four kinds of distance measurements are used by the KNN algorithm to measure the distance between two instances in the data set, which are the Chebyshev distance, Euclidean distance, Manhattan distance and Minkowski distance.

The Sensitivity (SN), Specificity (SP), Accuracy (ACC), and Matthew's correlation coefficient (MCC) are used to evaluate the performance of all combinations of KNN models. The SN, defined by Formula (5), calculates the probability of actual positives correctly classified. The specificity, defined by Formula (6), calculates the probability of actual negatives correctly classified. The accuracy, defined by Formula (7), calculates the proportion of correct predictions to the total number of predictions. The Matthews correlation coefficient, defined by Formula (8), is a correlation coefficient between the observed and predicted classifications, whose range is between -1 and $+1$. The larger the value of MCC is, the more match it indicates between the observation and prediction.

$$SN = \frac{TP}{TP + FN} \tag{5}$$

$$SP = \frac{TN}{TN + FP} \tag{6}$$

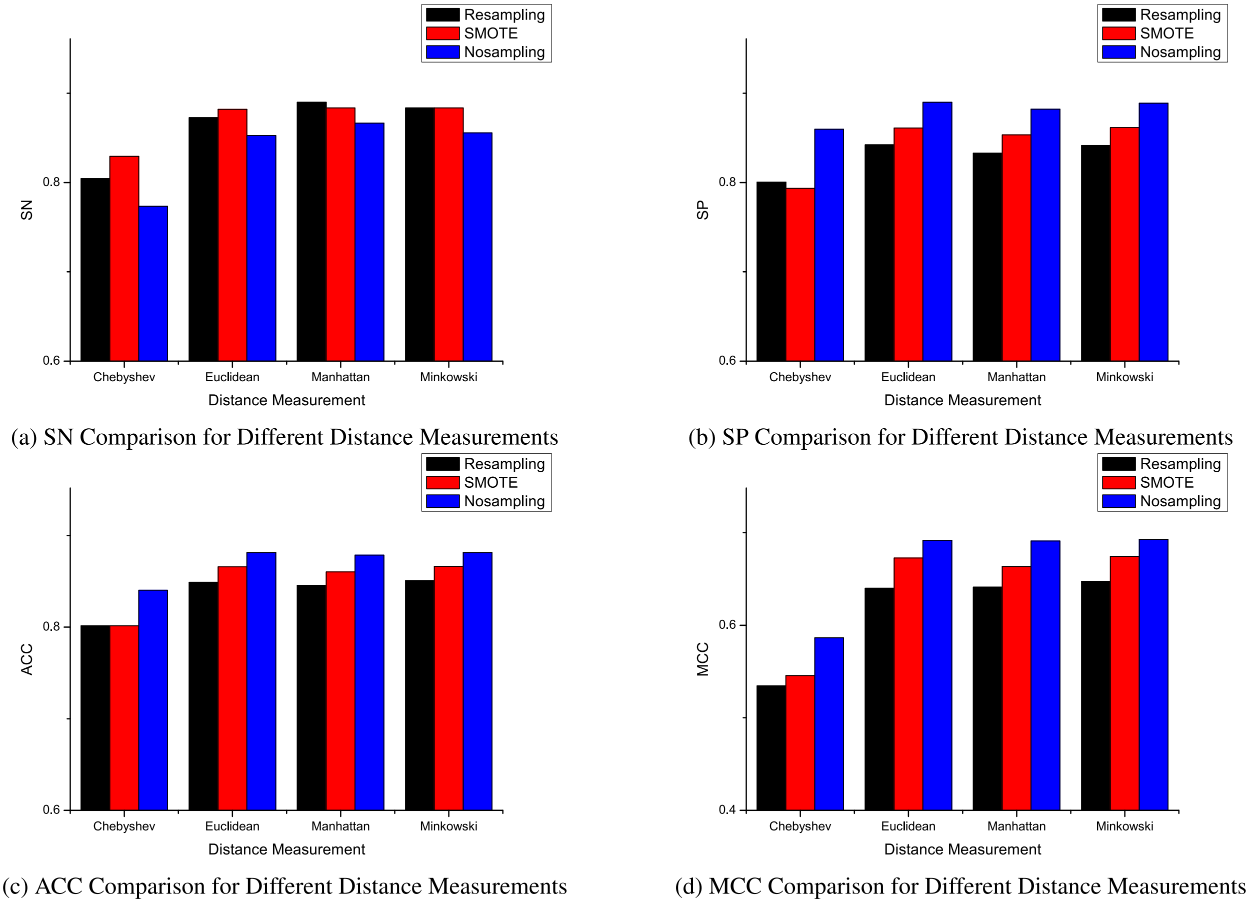
$$ACC = \frac{TN + TP}{TN + FP + TP + FN} \tag{7}$$

$$MCC = \frac{1 - (\frac{FN}{TP+FN} + \frac{FP}{TN+FP})}{\sqrt{(1 + \frac{FP-FN}{TP+FN})(1 + \frac{FN-FP}{TN+FP})}} \tag{8}$$

where TP represents the True Positive, FP represents False Positive, TN represents true negative, and FN represents False Negative.

The 10-fold cross-validation is used to evaluate the performance of the classification results. By dividing 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.

As the SMOTE and resampling methods are imposed on the data set, we need to take some measures to prevent the test data from being contaminated. In our experiments, the SMOTE and resampling methods are only imposed on the training set. After the KNN model is trained by the sampled data set, the test data is used to test the performance of the model just learned.



(a) SN Comparison for Different Distance Measurements

(b) SP Comparison for Different Distance Measurements

(c) ACC Comparison for Different Distance Measurements

(d) MCC Comparison for Different Distance Measurements

FIGURE 2. Performance comparison for 188D of different distance measurements.

TABLE 1. Parameters set for the experiments.

Algorithm	Parameter Name	Value
Resampling	biasToUniformClass	1
	noReplacement	False
	sampleSizePercent	160
SMOTE	nearestNeighbors	1
	percentage	250
KNN	nearestNeighbors	1

Weka, which is a famous machine learning software, is used to do the experiments. Details of the parameters used in the experiments are shown in table 1.

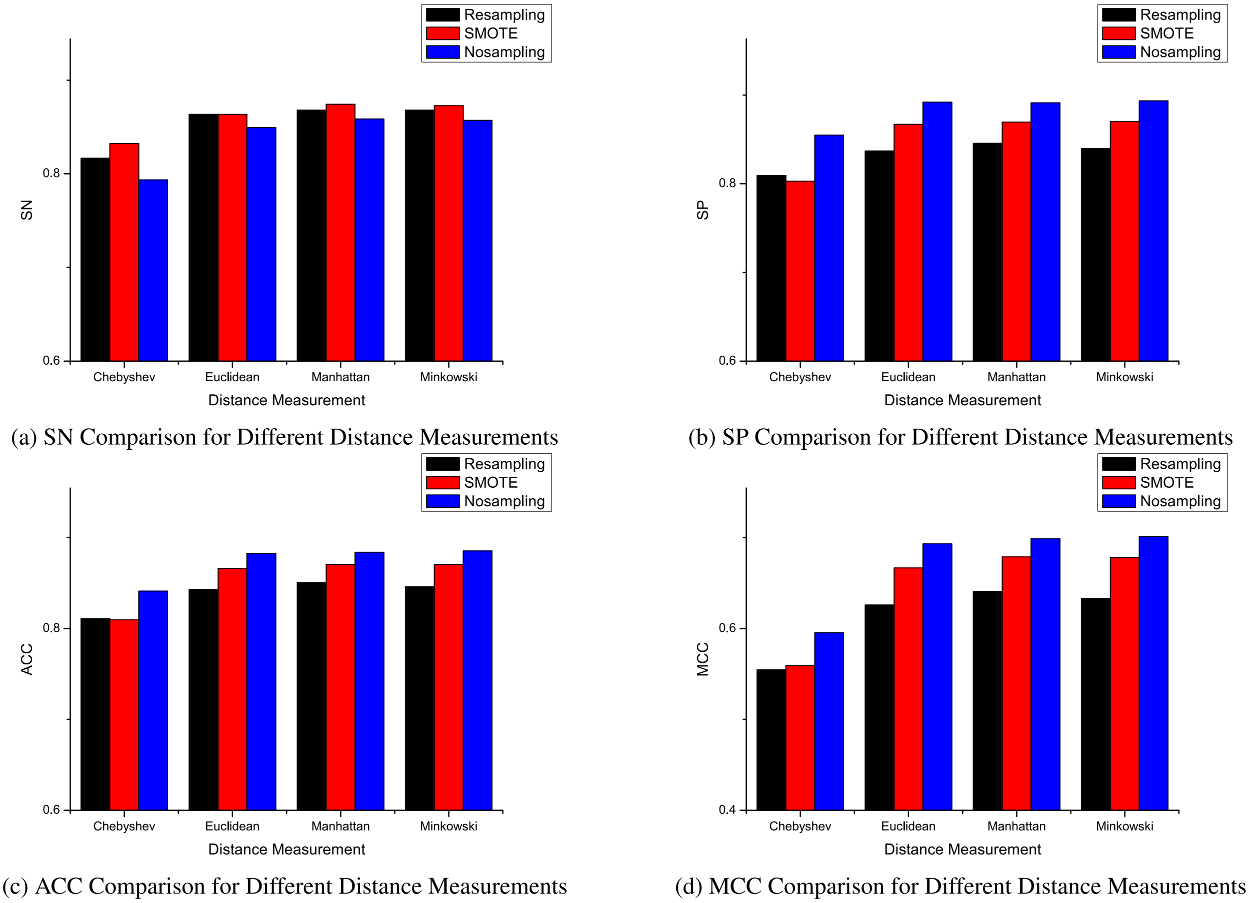
A. PERFORMANCE OF 188D FEATURE SET

In this section, the 188D method is used to extract the features from the SNAREs data set. Then the SMOTE and resampling methods are applied to balance the instances. After that, we get two balanced data sets. Together with the data set with no sampling, we get three data sets. Finally the KNN algorithm is used to classify the three data sets by using different distance measurements, which are the Chebyshev distance, Euclidean distance, Manhattan distance and Minkowski distance. The experimental results are shown in figure 2.

The comparison results for the SN among different combinations of distance measurements and sampling methods based on the 188D feature set are shown in figure 2a. It shows that the performance of Chebyshev is the worst among all distance measurements, while the performance of Manhattan distance is the best. When Manhattan distance is used, the resampled data set achieves the best performance, which is 89%.

The comparison results for the SP among different combinations based on 188D are shown in figure 2b. It shows that the performance of Chebyshev is the worst, while the performance of Minkowski is the best. When Minkowski distance is used, the data set with no sampling achieves the best performance, which is 88.9%. The experimental results show that the SP values computed based on the 188D feature set with no sampling is the best among the three sampling methods.

Figure 2c shows the comparison results of the ACC for different distance measurements and filtering methods based on 188D feature set. It shows that the Minkowski distance achieves the highest accuracy based on the 188D feature set with no sampling, which is 88.1%. The performance of Chebyshev distance is the worst among the four distance measurements. The experimental results also show that the



(a) SN Comparison for Different Distance Measurements

(b) SP Comparison for Different Distance Measurements

(c) ACC Comparison for Different Distance Measurements

(d) MCC Comparison for Different Distance Measurements

FIGURE 3. Performance comparison for K-skip-2-gram of different distance measurements.

ACC values computed based on the 188D feature set with no sampling is best among the three kinds of sampling methods. The SMOTE sampling method is in the second place and the resampling method is the worst.

Figure 2d shows the comparison results of the MCC for different distance measurements and sampling methods based on 188D feature set. It shows that the Minkowski distance achieves the best MCC among the four distance measurements. The highest MCC value is 69.3% in the case of 188D feature set with no sampling. For Manhattan and Euclidean distance, the values of MCC computed is much better than that computed by the Chebyshev distance.

From the experimental results above, we can conclude that for the 188D feature set, the performance of no sampling method is the best among the three kinds of sampling methods. The Minkowski distance achieves the best performance among the four kinds of distances for the KNN algorithm. So the Minkowski distance with no sampling is the best model among all combinations for the 188D feature set.

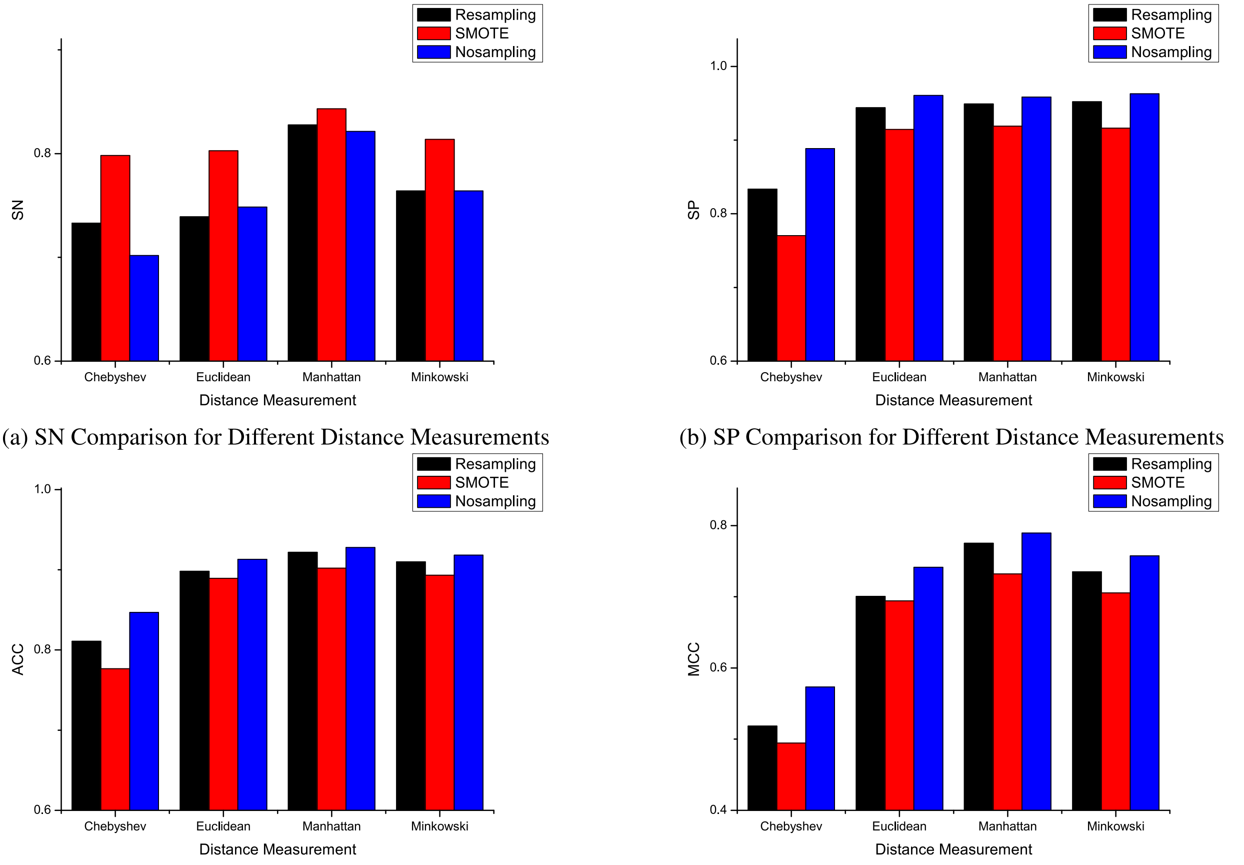
B. PERFORMANCE OF K-SKIP-2-GRAM FEATURE SET

In this section, the K-skip-2-gram method is used to extract the features from the SNAREs data set. After the SMOTE and resampling are applied to balance the instances, the KNN algorithm is used to classify the two kinds of data sets and the data set without sampling by using the four distance measurements. The experimental results are shown in figure 3.

The comparison results for the SN among different combinations of distance measurements and sampling methods based on the K-skip-2-gram feature set are shown in figure 3a. It shows that the performance of Chebyshev is the worst among all distance measurements, while the performance of Manhattan distance is the best. When Manhattan distance is used, the SMOTE data set achieves the best performance, which is 87.4%.

The comparison results for the SP among different combinations for K-skip-2-gram are shown in figure 3b. It shows that the performance of Chebyshev is the worst, while the performance of Minkowski is the best. When Minkowski distance is used, the data set with no sampling achieves the best performance, which is 89.3%. The experimental results also show that the SP values computed based on the K-skip-2-gram feature set with no sampling is the best among the three sampling methods.

Figure 3c shows the comparison results of the ACC for different distance measurements and filtering methods based on K-skip-2-gram feature set. It shows that the Minkowski distance achieves the highest accuracy based on the K-skip-2-gram feature set with no sampling, which is 88.5%. The performance of Chebyshev distance is the



(a) SN Comparison for Different Distance Measurements

(b) SP Comparison for Different Distance Measurements

(c) ACC Comparison for Different Distance Measurements

(d) MCC Comparison for Different Distance Measurements

FIGURE 4. Performance comparison for CKSAAP of different distance measurements.

worst among the four distance measurements. For K-skip-2-gram feature set, the data set with no sampling achieves the best ACC among the three kinds of sampling methods. The SMOTE method is in the second place and the resampling method is the worst.

Figure 3d shows the comparison results of the MCC for different distance measurements and sampling methods based on for K-skip-2-gram feature set. It shows that the Minkowski distance achieves the best MCC among the four distance measurements. The highest MCC value is 70.1% in the case of data set with no sampling. The performance of Chebyshev distance is the worst.

From the experimental results above, we can conclude that for the K-skip-2-gram feature set, the performance of no sampling method is the best among the three kinds of sampling methods. The Minkowski distance achieves the best performance among the four kinds of distances for the KNN algorithm. So the Minkowski distance with no sampling is the best model among all combinations for the K-skip-2-gram feature set.

C. PERFORMANCE FOR CKSAAP FEATURE SET

In this section, the CKSAAP method is used to extract the features from the SNAREs data set. After the SMOTE and resampling methods are applied to balance the instances, the KNN algorithm is used to classify the two kinds of data sets by using different distance measurements. The experimental results are shown in figure 4.

The comparison results for the SN among different combinations of distance measurements and sampling methods based on the CKSAAP feature set are shown in figure 4a. The results show that the SMOTE method achieves the best performance. The Resampling method is in the second place. For the kind of distance measurements, the Manhattan distance achieves the based performance while the Chebyshev is the worst among all distances. The best SN is achieved by the SMOTE method combined with the Manhattan distance, which is 84.3%.

Figure 4b shows the comparison results of the SP for different distance measurements and sampling methods based on CKSAAP feature set. It shows that the performance of no sampling is the best. The resampling method is in the second place and the SMOTE method is the worst. For no sampling method, the SP computed by Minkowski distance is the best, which is 96.3%.

Figure 4c shows the comparison results of the ACC for different distance measurements and sampling methods based on CKSAAP feature set. It shows that the Manhattan distance achieves the highest accuracy based on the CKSAAP feature set with no sampling, which is 92.8%. The performance of

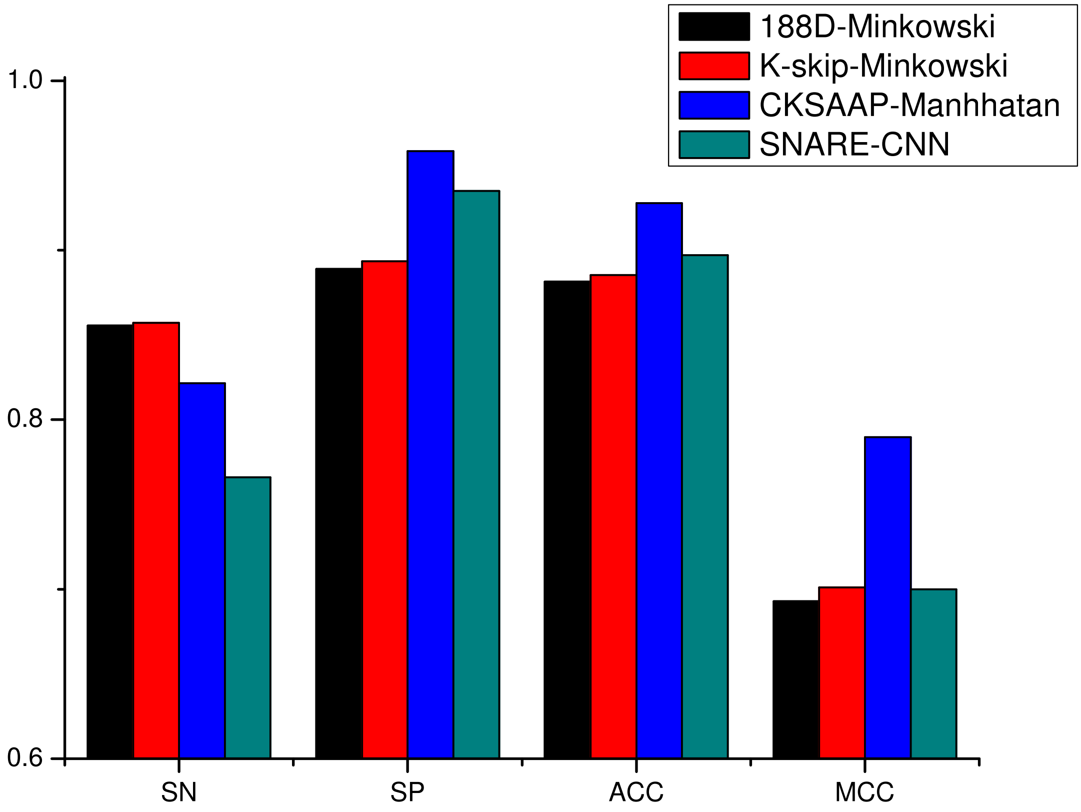


FIGURE 5. Comparison between the KNN models with the SNARE-CNN.

Chebyshev distance is the worst among the four distance measurements. The experimental results also show that the ACC values computed based on the CKSAAP feature set with resampling method is better than that computed based on the SMOTE method.

Figure 4d shows the comparison results of the MCC for different distance measurements and sampling methods based on CKSAAP feature set. It shows that the Manhattan distance achieves the best MCC among the four distance measurements. The highest MCC value is 79% in the case of CKSAAP feature set with no sampling. The results also show that the ACC values computed based on the resample method is better than that computed based on the SMOTE method.

The Manhattan distance achieves the best performance among the four kinds of distances for the KNN algorithm. So the Manhattan distance with no sampling is the best model among all combinations for the CKSAAP feature set.

D. COMPARISON WITH THE OTHER ALGORITHM

Based on the three groups of experiments above, we can conclude that, the data set with no sampling is the best among all sampling method. And we have found three best models for the 188D, K-skip-2-gram and CKSAAP feature sets. The first model is based on the Minkowski distance for 188D feature set. The second model is also based on the Minkowski distance for K-skip-2-gram feature set. The third model is based on the Manhattan distance.

In this section, we compare the performance of the three models with a deep learning method (CNN) proposed by [31], which is based on the PSSA feature extraction method. A CNN network is trained based on the PSSA feature set extracted from the SNARE sequences.

The comparison results for SN, SP, ACC and MCC are show in figure 5. Figure 5 shows that only the model constructed based on the CKSAAP feature and Manhattan distance achieves better performance than the CNN model in all of the four aspects. The performance of data set balancing methods is affected by the feature set. Our experimental results show that the performance of SMOTE filter is better than that of Resample filter in the case of 188D and K-skip-2-gram feature sets, but is worse in the case of CKSAAP feature set. But in this paper, the performance of no sampling method is the best.

IV. CONCLUSION

In this paper, we try different kinds of combinations of distance measurements for KNN algorithm, feature extraction methods and filter methods to classify the SNARE proteins. We find that the model constructed based on the CKSAAP feature with no sampling and Manhattan distance achieves the best performance, which is better than a deep learning based model 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, Sep. 2006.

[2] 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

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[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, p. e1065, Jun. 2015.

[13] W. Chen, H. Ding, P. Feng, H. Lin, and K.-C. Chou, "iACP: A sequence-based tool for identifying anticancer peptides," *Oncotarget*, vol. 7, no. 13, p. 16895, 2016.

[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, no. 1, p. S20, 2012.

[20] 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.

[21] L. Chao, L. Wei, and Q. Zou, "SecProMTB: A SVM-based classifier for secretory proteins of mycobacterium tuberculosis with data set," *Proteomics*, vol. 19, no. 17, 2019, Art. no. e1900007.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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. 5, p. 76, 2017.

[28] 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.

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

[30] 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.

[31] 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.

[32] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976, doi: 10.1109/TSMC.1976.5408784.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[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 S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proc. Nat. Acad. Sci. USA*, vol. 92, no. 19, pp. 8700–8704, 1995.

[36] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, and S.-H. Kim,"Recognition of a protein fold in the context of the SCOP classification," *Proteins*, vol. 35, no. 4, pp. 401–407, 1999.

[37] C. Z. Cai, "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] L. Wei, J. Tang, and Q. Zou, "SkipCPP-pred: An improved and promising sequence-based predictor for predicting cell-penetrating peptides," *BMC Genomics*, vol. 18, no. S7, p. 742, Oct. 2017.

[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] Q. Jiang, G. Wang, S. Jin, Y. Li, and Y. Wang, "Predicting human microRNA-disease associations based on support vector machine," *Int. J. Data Mining Bioinform.*, 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. 19, no. 3, pp. 224–228, 2018.

[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, pp. e127–e127, Nov. 2019.

[59] 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.

[60] 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.

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

[62] 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.

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

[64] 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.

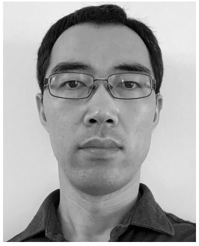
[65] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, ''Subspace clustering guided unsupervised feature selection,'' *Pattern Recognit.*, vol. 66, pp. 364–374, Jun. 2017.

[66] B. Małysiak-Mrozek, T. Baron, and D. Mrozek, ''Spark-IDPP: High-throughput and scalable prediction of intrinsically disordered protein regions with spark clusters on the cloud,'' *Cluster Comput.*, vol. 22, no. 2, pp. 487–508, Jun. 2019, doi: 10.1007/s10586-018-2857-9.

[67] Q. Zou, D. Mrozek, Q. Ma, and Y. Xu, ''Scalable data mining algorithms in computational biology and biomedicine,'' *BioMed Res. Int.*, vol. 2017, Art. no. 5652041, Feb. 2017, doi: 10.1155/2017/5652041.

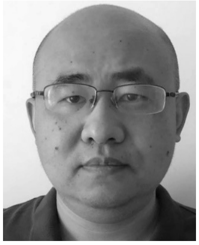
[68] B. Malysiak-Mrozek and D. Mrozek, ''An improved method for protein similarity searching by alignment of fuzzy energy signatures,'' *Int. J. Comput. Intell. Syst.*, vol. 4, no. 1, pp. 75–88, Feb. 2011, doi: 10.1080/18756891.2011.9727765.

[69] D. Mrozek, B. Socha, S. Kozielski, and B. Małysiak-Mrozek, ''An efficient and flexible scanning of databases of protein secondary structures: With the segment index and multithreaded alignment,'' *J. Intell. Inf. Syst.*, vol. 46, no. 1, pp. 213–233, Feb. 2016, doi: 10.1007/s10844-014-0353-0.



XING GAO (Member, IEEE) was born in Yangzhou, Jiangsu, China, in 1980. He received the B.S. degree in computer science and technology from the China University of Mining and Technology, in 2002, and the M.S. and Ph.D. degrees in computer software and theory from the Harbin Institute of Technology, Harbin, Heilongjiang, in 2009.

From 2009 to 2013, he was an Assistant Professor with the Department of Software Engineering, 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.



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 and theory from the Harbin Institute of Technology, Harbin, in 2003 and 2009, respectively.

From 2009 to 2013, he was an Assistant Professor with the Department of Software Engineering, 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.