IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 29, 2020, accepted April 17, 2020, date of publication April 23, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989749

The Feature Compression Algorithms for Identifying Cytokines Based on CNT Features

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

ABSTRACT As the signaling proteins, cytokines regulate a wide range of biological functions. It is important to distinguish the cytokines from other kinds of proteins. The 188-Dimensional CNT features are presented to identify the cytokines, which contain many redundant features. In this paper, we propose three kinds of feature compression algorithms to exclude the redundant features from the 188D features and keep the accuracy of the algorithm at the same time. The three algorithms are called the genetic based algorithm, the greedy based algorithm and the brute-force based algorithm. Experimental results demonstrate that the brute-force based algorithm gets the highest classification accuracy among the three algorithms. The genetic based algorithm achieves the least number of compressed features among the three algorithms. But they consume much more time than that consumed by the greedy based algorithm. The greedy based algorithm makes a good trade-off among the three factors, which are the classification accuracy, the number of compressed features and the time consumption.

INDEX TERMS Cytokine identification, feature compression, feature selection.

I. INTRODUCTION

Cytokines are a type of proteins, which play an important regulatory role in many cellular activities, such as differentiation, growth and interactions between cells. It has important theoretical and practical significance to study the cytokine identification and classification. The structures and functions of unknown types of cytokines can be understood by accurate recognition of the sequences of cytokines.

Based on the sequence structures and functions of cytokines obtained, authors in paper [1] identify cytokines by manual prediction. Several methods have been proposed over the last decades to identify cytokines, such as the Hidden Markov Model (HMM) based methods [2], [3], the Artificial Neutral Network (ANN) based methods [4]–[7], the Basic Local Alignment Search Tool (BLAST) [8], FASTA [9], [10], CTKPred [11] and CytoPred [12]. In paper [13], Cai *et al.* utilize a set of 188-Dimensional features extracted from the Amino Acids (AAs) composition to identify the cytokines. A common point of all the methods mentioned above is that they all need to extract many features from the cytokines. Are all these features necessary? As we know, the more features the identification algorithms use to identify the cytokines, the more computation resources they consume. Sometimes, irrelevant features can even reduce the accuracy of the identification algorithms. It is necessary to exclude the irrelevant features from the feature set.

Feature selection is an effective method to reduce the number of features in the feature set for classification tasks [14]–[39], which can be very difficult because of a large search space [40]. Given n features, there are 2^n possible feature subsets [41]. As the number of features increases, the feature selection problem becomes even more challenging [42]–[45]. The exhaustive strategy for searching the optimal feature subset is impossible [46], [47]. Various kinds of search strategies have been proposed, such as the complete, the random, the greedy, the heuristic search strategies [48]–[68].

In this paper, we try to compress the 188D CNT feature set [69] by removing the redundant features from it and keep the identification accuracy of the original 188D CNT feature set based method at the same time. We propose three kinds of feature compression algorithms. **The first algorithm is called the genetic based feature compression algorithm.** In the genetic based algorithm, a 188D binary vector (called a solution) is used to represent the 188D feature set. Each bit in the 188D binary vector corresponds to a feature in the 188D CNT feature set. If a feature in the 188D CNT feature

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

set is selected in the compressed feature set, the bit in the vector corresponding to the feature is set to 1, otherwise, the bit is set to 0. A correlation-based algorithm is proposed to produce the initial population with n solutions. Then the population is evolved for several generations by the genetic based algorithm. The classification accuracy is used as the fitness value to evaluate the quality of each solution in the population for each generation. Finally, the genetic based algorithm gets a binary vector that has the largest fitness value among all the other solutions in the population. The final compressed feature set is composed of the features, whose corresponding bits are set to 1 in the vector. **The second algorithm is called the greedy based feature compression algorithm.** All features in 188D feature set can be classified into 9 classes, **called feature classes**, according to the quantities of the AAs (20D), hydrophobicity (21D), polarity (21D), normalized Van der Waals volume (21D), surface tension (21D), charge (21D), polarizability (21D), solvent accessibility (21D) and secondary structure (21D). In the greedy based algorithm, we evaluate the correlation between a feature class and the cytokine. The feature classes are greedily added to the compressed feature set according to their evaluation results from largest to smallest. After adding a class of features to the compressed feature set, the classification accuracy of the new compressed feature set is evaluated by the Support Vector Machine (SVM). **The third algorithm is called the brute-force based feature compression algorithm.** The third algorithm is also based on the feature class. A 9-bit binary vector is used to represent the 9 feature classes of the 188D feature set. Each bit in the vector represents whether a feature class is selected in the compressed feature set. The 9-bit binary vector can be thought of as a decimal number ranging from 1 to 511 (Zero is not included because it means that no feature class is selected). The brute-force based algorithm evaluates all the 511 kinds of conditions and tests the classification accuracy. Finally, the features in the feature classes, corresponding to the decimal number with the largest accuracy, are selected as the compressed feature set by the brute-force based algorithm.

The contributions of the paper are as follows. First, we propose three algorithms to compress the 188D feature set to identify the cytokine proteins, which are the genetic based algorithm, the brute-force based algorithm and the greedy based algorithm. Second, extensive experiments were done to test and compare the performance of the three algorithms. The experimental results show that the genetic based algorithm achieves the best feature compression performance. While it consumes the most time and the classification accuracy of the compressed feature set is not better than that of the 188D feature set. The brute-force based algorithm has the best classification accuracy while it also consumes time. The greedy based algorithm makes a good trade-off among the classification accuracy, the number of compressed features and the time consumption.

The organization of the paper is as follows: in section 2, we introduce the data collection and data preprocessing method. In section 3, we introduce the three kinds of feature compression algorithms in detail. In section 4, we give the experimental results to evaluate the performance of the three algorithms proposed in this paper. Finally, we draw the conclusion.

II. MATERIALS AND METHODS

A. DATA COLLECTION AND PREPROCESSING

Cytokines regulate a wide range of biological functions including hematopoiesis, inflammation and repair by extracellular signaling. It is important to distinguish the cytokines from other kinds of proteins. Cai *et al.* [13] extract 188-Dimensional (188D) features based on the physicochemical properties, distribution and composition of amino acids, which are used to analyze whether a protein is a cytokine. But whether all the 188D features are necessary for the identification is a question. In this paper, we propose three kinds of feature compression algorithms to reduce the number of features contained in the 188D feature set to predict whether a protein is a cytokine.

Figure 1 shows the procedure on how to collect and preprocess the data used in the three kinds of the feature compression algorithm. The whole data set is composed of two parts: the positive instances and the negative instances.

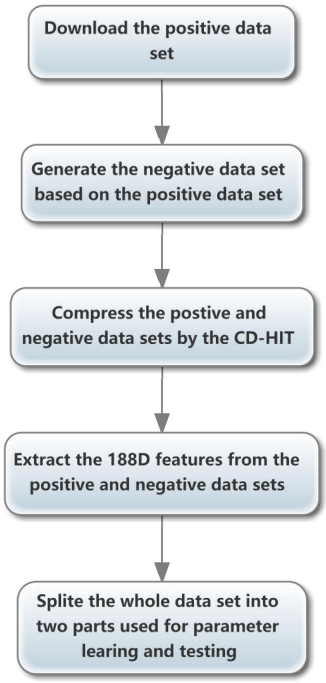

FIGURE 1. Procedure for collecting and preprocessing the cytokine data set.

To get the positive instances, we download the cytokine data set from the Uniprot database [70]–[72]. To get the negative instances, we first list the PFAM families that all positive instances belong to. For each PFAM family, except the PFAM families the positive instances belong to, we extract the longest sequence protein as the negative instance. The CD-HIT program [73] is used to remove the redundant

instances from the positive and negative data sets. Finally, we get a data set with 18944 instances altogether, which contains 9645 positive instances and 9299 negative instances.

B. FEATURE EXTRACTION STRATEGY

In this paper, we want to compress the 188D features proposed in [13]. Now, we briefly introduce how to calculate 188D features [74].

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 [75].

Eight kinds of properties are used to extract the 168 features left from a sequence, including the hydrophobicity, normalized Van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility. 21 features are extracted according to each kind of physicochemical property. Next, we take the hydrophobicity property as an example to show how to calculate the 21 feature values $(FV_{21}, \ldots, FV_{41})$ in the 188D FV.

According to the hydrophobicity property of 20 AAs, they can be classified into three groups, which are the RKEDQN, GASTPHY, and CVLIMFW groups.

The FV_{21}, FV_{22} and FV_{23} 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 the three groups.

The FVs from 24 to 38 are calculated as follows:

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

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

The FV_{39}, FV_{40} and FV_{41} are calculated as follows:

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

where the $FH_i (i = 1, 2, 3)$ represents the respective number of bivalent seeds that contain two amino acids from different groups and $(L^{\cup}1)$ represents the number of bivalent seeds.

A total of 21 features $(FV_{21} - FV_{41})$ are calculated for the hydrophobicity property. After the other seven kinds of physicochemical properties are analyzed in the same way as that of the hydrophobicity, we get a 188D feature vector for a cytokine.

A 188D feature vector is calculated for each cytokine in the positive and negative data set obtained by the CD-HIT program in step 3 of the data preprocessing procedure in Figure 1. And we get a suitable data set to train the machine learning algorithm.

C. SUPPORT VECTOR MACHINE

In this paper, we use the Support Vector Machine (SVM) [76]–[97], as the classification algorithm. Given a set of instance-label pairs $(x_i, y_i), i = 1, \ldots, n$ (called the training set) where x_i is a n dimension vector [4], the SVM calculates the optimal solution of the following problem:

$$\begin{aligned}&\min_{w,b,\xi} \quad \frac{1}{2} w^T w + c \sum_{i=1}^{l} \xi_i \\ &\textit{subject to} \quad y_i \left(w^T \varphi(x_i) + b \right) \geq 1 - \xi_i \qquad \xi_i \geq 0\end{aligned}$$

By mapping the x_i in the training set to a much higher dimensional space, the SVM can find a hyperplane that separates the vectors in the training set with the maximal margin in the new space. Parameter c is the penalty for the classification errors. And the kernel function is defined as $\phi(x_i)^T \phi(x_j)$. Four kinds of kernel functions are often used, which are the radial basis function (RBF) kernel, the sigmoid kernel, the linear kernel and the polynomial kernel.

The RBF kernel [98], [99] is used in this paper, which has two parameters c and γ. For a given problem, the optimal values of the two parameters are not known. We use a grid-based searching strategy to find suitable values for c and γ to make the classifier accurately classify the unknown data. Various pairs of (c, γ) values are sampled from the grid searching space and the one with the highest accuracy is selected.

To learn the optimal values for the parameter (c, γ), the whole data set obtained in section 2.1 is divided into two non-intersecting parts. The first part, called "**Optimal Parameter Searching Data Set (OPSDS)**", is used to search the optimal values for the two parameters (c, γ) of the SVM. A stratified selection method is used to draw 10% of the data from the whole data set. The OPSDS is composed of the selected data. The stratified selection method can ensure the same class distribution in the subset as that of the whole data set. The second part called "**Testing Data Set (TDS)**", which is composed of the 90% data left, is used to test the accuracy of the SVM.

III. THE FEATURE COMPRESSION ALGORITHMS

In this section, three kinds of feature compression algorithms are introduced, which are the genetic based feature compression algorithm, the greedy based feature compression algorithm and the brute-force based feature compression algorithm. As is shown in Figure 2, all three algorithms need to learn the optimal values of (c, γ) by using the OPSDS for any candidate compressed feature set. Then the TDS is used to evaluate the accuracy of the candidate compressed feature set by using the optimal (c, γ) just learned. And the candidate compressed feature set with the highest accuracy will be the final compressed feature set selected by the algorithm.

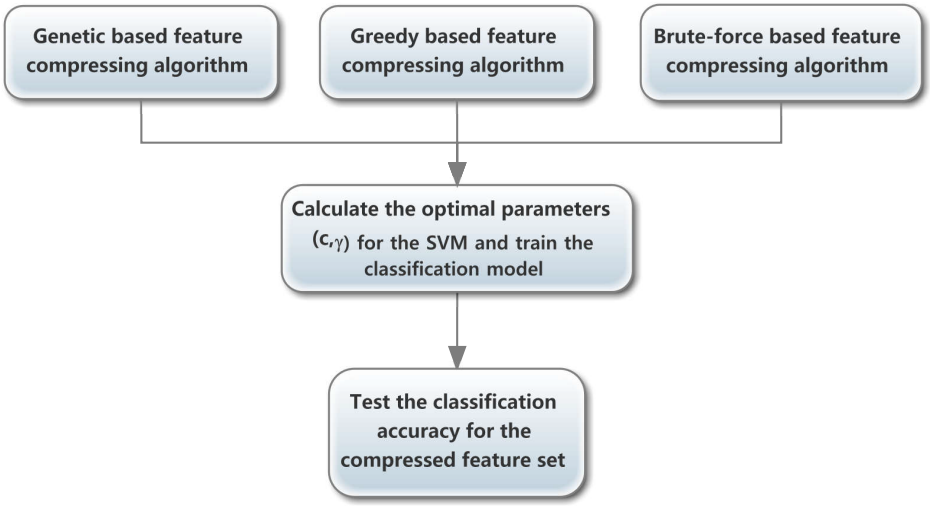

FIGURE 2. Workflow of the feature compression algorithms.

A. THE GENETIC BASED FEATURE COMPRESSION ALGORITHM

In this section, the Genetic based Feature compression Algorithm is introduced. We present each component of the algorithm first, including how to represent the solution of the feature compression problem, how to construct the initial population and how to define the fitness function. Finally, we introduce the whole algorithm.

1) SOLUTION REPRESENTATION

In a genetic algorithm, a set of solutions to the optimization problem is constructed. By evolving the solutions generation by generation, a good solution can be found. The set of solutions is called Population P. We also need a kind of encoding scheme to encode each solution in P. A binary encoding scheme is used in this paper. It means that a solution x in P is a $0-1$ vector with 188 dimensions, which is the number of features to be compressed. If an element x_i in solution x, $(i = 1, 2, \ldots 188)$, is set to 1, then the i^{th} feature is included in the compressed feature set represented by x. Otherwise, the x does not include the i^{th} feature.

2) INITIAL POPULATION CONSTRUCTION ALGORITHM

The quality of the initial population decides the future generations, which is very important, so an initial population construction algorithm is proposed. The algorithm evaluates the value of each feature by calculating the correlation between it and the class according to the Pearson's formula (1). After calculating the worth of each feature, the algorithm constructs an individual in the initial population based on the roulette selection. The probability, calculated by formula (2), decides the chance that a feature is selected or not. It is obvious that the bigger the worth of a feature is, the larger the chance for the feature being selected in an individual of the initial population. The process is repeated M times which is the number of solutions in the initial population. The Initial Population Construction Algorithm is illustrated in Algorithm 1.

Algorithm 1 Initial Population Construction Algorithm

Input: OPSDS+TDS, the population number M
Output: Initial Population Set
1: For each feature $x_i(i = 1, 2, \ldots, 188)$, calculate its correlation $\rho_{X_i,Y}$ with the class according to formula 1 based on the union of OPSDS and TDS
2: Set the initial population set $P = \phi$
3: **while** $|P| \leq M$ **do**
4: Produce a solution x by the roulette selection and add the x into P
5: **end while**
6: **return** P

$$\rho_{X_i,Y} = \frac{\mathrm{cov}(X_i, Y)}{\sigma_{X_i}\sigma_Y} = \frac{E(XiY) - E(Xi)E(Y)}{\sqrt{E\left(Xi^2\right) - E^2(Xi)}\sqrt{E\left(Y^2\right) - E^2(Y)}} \quad (1)$$

$$p_i = \frac{\rho_{XiY}}{\sum_{i=1}^{188} \rho_{XiY}} \quad (2)$$

3) FITNESS FUNCTION

Given a solution $x = (x_1, x_2, \ldots, x_{188})$ in the population P, x is used by the SVM to classify the data in the OPSDS. The classification accuracy, which is defined by the formula (3), is used to be the fitness of the solution x.

$$fit(x) = \frac{TN + TP}{TN + FP + TP + FN} \quad (3)$$

where TP is the true positives, FP is the false positives, TN is the true negatives and FN is the false negatives classified by SVM according to x.

For $\forall d \in PSDS$ or TDS, $d = (d_1, d_2, \ldots, d_{188})$. We must filter d_i according to the selected features in solution $x = (x_1, x_2, \ldots, x_{188})$ according to formula (4).

$$d' = \begin{cases} d_i & if\ x_i = 1 \\ 0 & if\ x_i = 0 \end{cases} \tag{4}$$

The basic idea of the Fitness Calculation Algorithm is to evaluate the value of each solution x in population P according to the fitness function formula (3). To calculate the fitness value of a solution x, we first filter the data in the OPSDS and TDS according to formula (4) and get the filtered data set OPSDS1 and TDS1. Then the OPSDS1 is used to search the optimal value of (c, γ) for the solution x. Finally, we classify the data in TDS1 by using the (c, γ) just learned and the classification accuracy is the fitness value of solution x. The Fitness Calculation Algorithm is shown in Algorithm 2.

Algorithm 2 Fitness Calculation Algorithm

Input: solution x, OPSDS, TDS
Output: fitness of solution x

1: For $\forall d \in$ OPSDS, calculate d' according to formula 4 and get the result data set OPSDS1
2: Search the optimal value for (c, γ) based on the grid search algorithm provided by libSVM by using the OPSDS1
3: Filter the data in TDS in the same way as that of Step 1 and get the result data set TDS1
4: Classify the data in TDS1 based on the SVM by using the (c, γ) in step2 and TDS1
5: Calculate the classification accuracy *acc* of the classification as the fitness of the solution x
6: **return** *acc*

The Genetic based Feature compression Algorithm starts with the initial population of solutions generated by an "initial population construction algorithm" (Algorithm 1). Each solution represents a candidate feature subset to the Feature compression problem, which evolves several generations. During each generation, a fitness function (Algorithm 2) is applied to each solution in the population to determine their qualities. In each generation, the population is updated through crossover and mutation operators. Good solutions are selected according to the Tournament selection method. The Genetic based Feature compression Algorithm invokes the standard one-point crossover and bit mutation to update the current population. The search is terminated when the number of generations exceeds a threshold. The Genetic based Feature compression Algorithm is stated by Algorithm 3.

B. THE GREEDY BASED FEATURE COMPRESSION ALGORITHM

In section II-B, we introduce that the 188D features can be classified into 9 classes according to their physicochemical properties. 20 features belong to the quantities of the AAs.

Algorithm 3 Genetic Based Feature Compression Algorithm

Input: OPSDS, TDS
Output: The compressed feature set

1: **Initialization.** Set the size of Population M, the number of max generations g_{max}. Set the crossover probability $p_c \in (0, 1)$ and the mutation probabilities $p_m \in (0, 1)$. Generate an initial population P_0 by using Algorithm 1.
2: **Parent Selection.** Select a temporary population P_t from the current population by using the Tournament selection method
3: **Crossover.** Make the one-point crossover operation to solutions in P_t, and update P_t.
4: **Mutation.** Make the uniform mutation operation to solutions in P_t, and update P_t.
5: **Survival Selection.** Calculate the fitness value for all solutions generated in the updated P_t by calling the Algorithm 2 and set $P_{t+1} = P_t$.
6: **Stopping Condition.** If $t > g_{max}$, then terminate. Otherwise, set $t = t + 1$, and go to Step 2
7: **return** the solution in the current population who has the maximum fitness value as the best compressed feature set

And 21 features belong to each of the left eight kinds of physicochemical properties.

In the greedy based feature compression algorithm, we consider all the features belonging to the same class as a feature class, so there are altogether 9 feature classes. Once a feature class is selected by the greedy based algorithm, all the features belonging to the class will be included in the final compressed feature set.

The basic idea of the greedy based algorithm is to evaluate the relationship between each feature class with the prediction results of the cytokine data set. The greedy based algorithm greedily adds the features in the feature classes to the compressed feature set one by one according to their influences on the prediction results, until all 188D features are added.

Three different kinds of methods are used to evaluate the relationship between an individual feature and the prediction result, which are the correlation based method, the Info Gain based method and the Gain Ratio based method. By adding all the evaluation results of the features belonging to the same feature class, we can evaluate the relationship between a feature class with the prediction results of the cytokine data set. The correlation based evaluation method is given by formula (1). The Info Gain based evaluation method is given by formula (5). The Gain Ratio based evaluation method is given by formula (6). The greedy based feature compression algorithm is given by Algorithm 4.

The Info Gain is calculated by the following formula:

$$Gain(S, A) = E(S) - E(S|A) \tag{5}$$

where $E(S) = -\sum_{i=1}^{c} p_i log_2(p_i)$ and $E(S|A) = \sum_{i=1}^{c} p_i E(S|A = a_i)$.

Algorithm 4 The Greedy Based Feature Compression Algorithm

Input: OPSDS, TDS
Output: The accuracies calculated for every compressed feature set
1: Evaluate each feature of 188D feature set based on formula (1), (5) or (6) and get the evaluation results set $R = r_1, r_2, \ldots, r_{188}$
2: Calculate the evaluation result R' of the class features by adding the features' evaluation results in set R together that belong to the same class feature
3: Sort the class evaluation results R from largest to smallest
4: **while** $i < 9$ **do**
5: Add the features belonging to the r_i' class feature set to the compressed feature set
6: Filter the data in OPS and TDS and get the data set OPS1 and TDS1
7: Search the optimal value for (c, γ) by using OPS1
8: Classify the data in TDS1 based on the SVM by using the (c, γ)
9: Calculate the classification accuracy *acc* of the classification and store *acc* into an array *ACC*
10: $i = i + 1$
11: **end while**
12: **return** *ACC*

The Gain Ration is calculated by the following formula:

$$GainRation = \frac{Gain(S, A)}{SplitInformation(S, A)} \tag{6}$$

where $SplitInformation(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{S} log_2 \frac{|S_i|}{S}$.

C. THE BRUTE-FORCE BASED FEATURE COMPRESSION ALGORITHM

In this algorithm, one bit is used to represent a feature class. As there are 9 kinds of feature classes, it needs 9 bits altogether. If a kind of feature class is selected, the bit representing the feature class is set to 1, otherwise, the bit is set to 0. There are altogether 511 kinds of feature selection strategy except number zero. The brute-force feature compression algorithm enumerates all kinds of feature selection strategies and selects the one with the highest classification accuracy. According to our experimental results, the strategies with less than 6 kinds of feature classes get poor classification accuracy, so the brute-force based feature compression algorithm only enumerates the feature selection strategies who have more than 6 kinds of feature classes. The brute-force based feature compression algorithm is given by Algorithm 5.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, three experiments are done to test the performance of the three feature compression algorithms proposed in this paper. Finally, we analyze the advantage and disadvantage of different algorithms. Three evaluation standards are used to compare different kinds of algorithms, which are the number of features contained in the final compressed feature set, the classification accuracy and the running time of the algorithms.

Algorithm 5 The Brute-Force Based Feature Compression Algorithm

Input: OPSDS, TDS
Output: The accuracies calculated for every compressed feature set
1: Set the feature strategy number i to 1
2: **while** $i < 512$ **do**
3: **if** the binary value of i has more than six 1-bit value **then**
4: Add the features belonging to the class feature set corresponding to the 1-bit to the compressed feature set
5: **end if**
6: Filter the data in OPS and TDS and get the data set OPS1 and TDS1 according to the compressed feature set
7: classify the data in TDS1 based on the SVM by using the (c, γ)
8: Calculate the classification accuracy *acc* of the classification and store *acc* into an array *ACC*
9: $i = i + 1$
10: **end while**
11: **return** *ACC*

A. PERFORMANCE OF THE GENETIC BASED FEATURE COMPRESSION ALGORITHM

In this experiment, we test the performance of the genetic based feature compression algorithm. Firstly, we produce a population with 25 solutions by Algorithm 1. The crossover probabilities p_c and mutation probabilities p_m are set to (0.5, 0.05), (0.7, 0.05), (0,9, 0.05), (0.5, 0.1), (0.7, 0.1), (0.9, 0.1), (0.5, 0.2), (0.7, 0.2) and (0.9,0.2) respectively. We run the genetic based feature compression algorithm (Algorithm 3) for 20 generations. The maximum fitness value in each generation, calculated by Algorithm 2 for each case of p_c and p_m, is shown in Figure 3, which shows that the

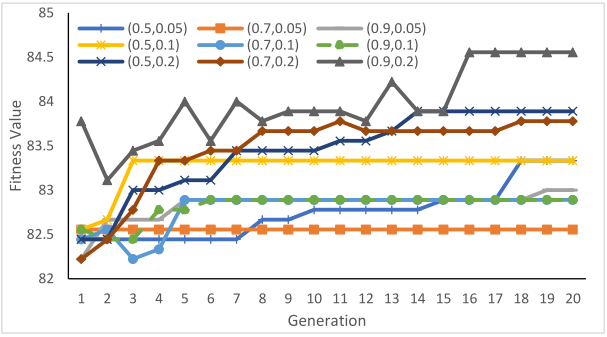

FIGURE 3. The fitness value calculated for n = 25.

fitness values generally become bigger and bigger with the increasing generation. After 20 generations, the fitness value for each pair of (p_c, p_m) is steady and it is the maximum fitness value among all generations. We use the solution with the maximum fitness value as the selected compressed feature set to test the performance of the genetic based feature compression algorithm.

Then we produce a population with 50 solutions by Algorithm 1. The crossover probabilities and the mutation probabilities are set to the values as same as that in Figure 3. We run Algorithm 3 for 10 generations. The fitness values, calculated by Algorithm 2 for each generation, are shown in Figure 4. It also shows that, after 10 generations, we get the solution with the maximum fitness value for each pair of (p_c, p_m) among all generations, which can be used as the selected compressed feature set to test the performance of the genetic based feature compression algorithm.

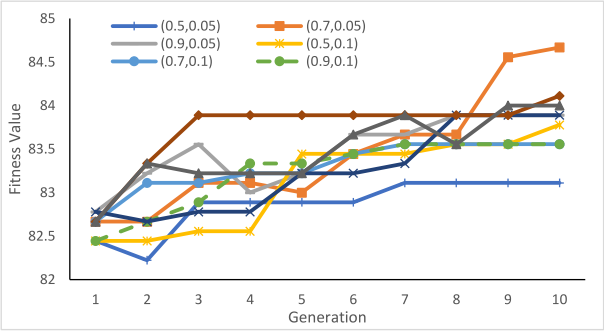

FIGURE 4. The fitness value calculated for n = 50.

After running Algorithm 3 for the case of the initial population with 25 solutions ($n = 25$) for 20 generations, we get 9 solutions, which are 9 sets of compressed features for the 188D features. In the same way, we get another 9 sets of compressed features in the case of 50 solutions in the initial population ($n = 50$). In Figure 5, we compare the number of features contained in the final compressed feature set for $n = 25$ and $n = 50$. On average, the number of compressed features for $n = 25$ is 101, which is less than 104 for $n = 50$.

In Figure 6, we compare the classification accuracy of the 9 groups of compressed feature sets got for $n = 25$ and $n = 50$.

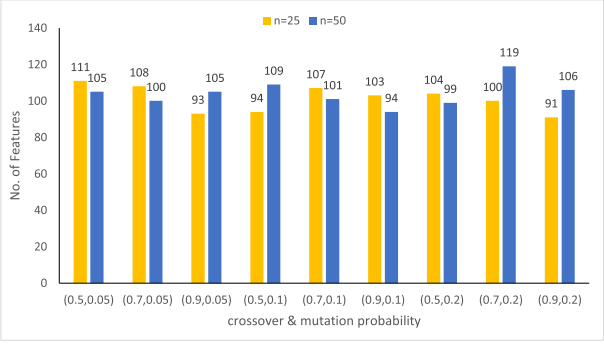

FIGURE 5. Comparison of the number of compressed features between n = 25 and n = 50.

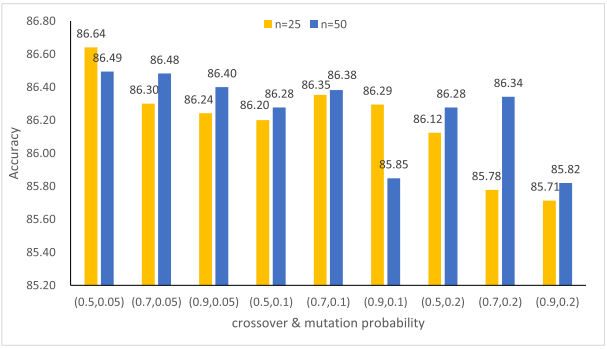

FIGURE 6. Comparison of the classification accuracy between n = 25 and n = 50.

It shows that the maximum accuracy is achieved in case of $n = 25$, $p_c = 0.5$ and $p_m = 0.05$, in which case the number of compressed features is 111.

B. PERFORMANCE OF THE GREEDY BASED FEATURE COMPRESSION ALGORITHM

In this experiment, we test the performance of the greedy based feature compression algorithm. The correlation based, infoGain based and GainRatio based methods are used to evaluate the rank of each feature class. The experimental results are shown in Figure 7. The x axis is the number of feature classes being used to classify the cytokine data. The y axis is the classification accuracy based on the selected features in the feature class. Figure 7 shows that when the number of feature classes selected is few (less than 4), the accuracy of SVM classifier is poor. With the increasing of the number of feature classes selected, the accuracy becomes better and better. Among the three kinds of evaluation methods, the accuracy of the correlation based method is the most steady. The best accuracy is achieved by the InfoGain based method when 8 feature classes are selected, with 167 features. The classification accuracy of the selected features is also better than that of the 188D features.

C. PERFORMANCE OF THE BRUTE-FORCE BASED FEATURE COMPRESSION ALGORITHM

In this experiment, we test the performance of the brute-force based feature compression algorithm. The x axis is the number for a kind of feature selection strategy. The y axis is the

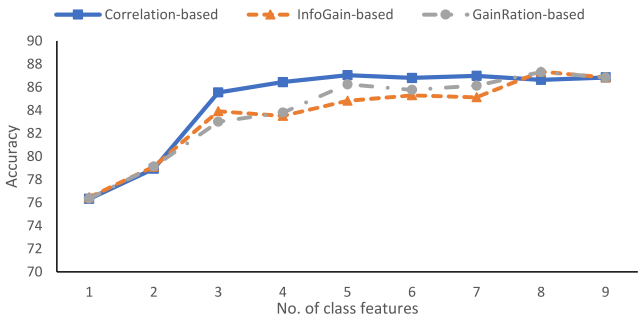

FIGURE 7. Comparison of the classification accuracy among different feature class evaluation methods.

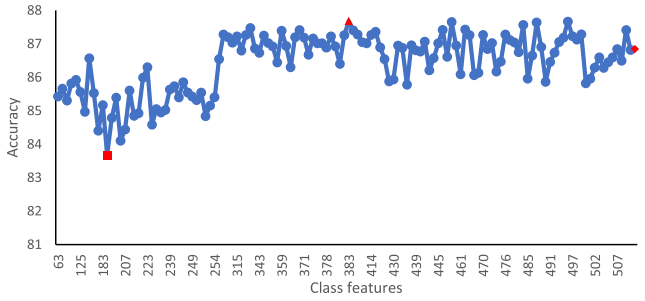

FIGURE 8. Classification accuracy for different compressed feature sets.

classification accuracy corresponding to the feature selection strategy. It shows that when the number is 383, the brute-force based algorithm achieves the maximum accuracy, which is better than the greedy based algorithm and the genetic based algorithm. The decimal number 383 corresponds to the binary number 101111111, which means only the second feature class is not selected as the classification feature by the SVM. The performance of feature compression for the brute-force based algorithm is the same as that of the greedy based algorithm.

D. DISCUSSION

From the three groups of experiments, we can compare the three feature compression algorithms proposed in this paper from three aspects, which are the accuracy, feature compression and runtime.

Among the three algorithms, the genetic based algorithm gets the minimum compressed feature set with the same classification accuracy. But as the search space is very large for the genetic based algorithm, it is hard to find the global optimal accuracy, so in our experiment the best accuracy of genetic based algorithm is not better than the other two kinds of algorithms. As the fitness value is measured by the classification accuracy of the SVM, the genetic based algorithm needs to constantly train the SVM for each solution in the population. It's why the time spent by the genetic algorithm is the longest among the three algorithms.

The brute-force based algorithm achieves the best classification accuracy among the three kinds of algorithms. The number of compressed features is the same as that of the greedy based algorithm. But it consumes much more time than the greedy based algorithm because it needs to train the SVM for hundreds of feature selection strategy.

The greedy based algorithm makes a good trade-off among the accuracy, the number of compressed features and the runtime. It can get better accuracy than the original 188D features with fewer features. The runtime of the greedy based algorithm is much less than that of the other two kinds of algorithms because it only needs to train the SVM for 9 times.

V. CONCLUSION

In this paper, three kinds of feature compression algorithms are proposed to compress a 188D feature set, named the genetic based, the greedy based and the brute-force based feature compression algorithm. The experimental results show that the brute-force based algorithm achieves the highest classification accuracy. The genetic based algorithm selects the least number of features from the 188D features as the compressed feature set. The shortcoming of the two algorithms is that they consume much time because they constantly run the SVM classifier during the procedure of feature selection. The greedy based algorithm makes a good trade-off among the classification accuracy, the number of compressed features and the time consumption.

REFERENCES

[1] Q. Zou, W. Chen, Y. Huang, X. Liu, and Y. Jiang, "Identifying multi-functional enzyme by hierarchical multi-label classifier," *J. Comput. Theor. Nanosci.*, vol. 10, no. 4, pp. 1038–1043, Apr. 2013.
[2] P. K. Papasaikas, P. G. Bagos, Z. I. Litou, and S. J. Hamodrakas, "A novel method for GPCR recognition and family classification from sequence alone using signatures derived from profile hidden Markov models," *SAR QSAR Environ. Res.*, vol. 14, nos. 5–6, pp. 413–420, Oct. 2003.
[3] Y. Yabuki TM, T. Hirokawa, H. Mukai, and M. Suwa, "GRIFFIN: A system for predicting GPCR–G-protein coupling selectivity using a support vector machine and a hidden Markov model," *Nucleic Acids Res.*, vol. 33, no. 2, pp. 148–153, 2005.
[4] C. S. Yu, Y. C. Chen, C. H. Lu, and J.-K. Hwang, "Prediction of protein subcellular localization," *Proteins Struct., Function Genet.*, vol. 64, no. 3, pp. 643–651, 2006.
[5] H. Nielsen, J. Engelbrecht, S. Brunak, and G. V. Heijne, "A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites," *Int. J. Neural Syst.*, vol. 8, nos. 5–6, pp. 581–599, Oct. 1997.
[6] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[7] Q. Zou, P. Xing, L. Wei, and B. Liu, "Gene2vec: Gene subsequence embedding for prediction of mammalian N6-methyladenosine sites from mRNA," *RNA*, vol. 25, no. 2, pp. 205–218, 2019.
[8] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, Oct. 1990.
[9] W. R. Pearson, "Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms," *Genomics*, vol. 11, no. 3, pp. 635–650, 1991.
[10] G. S. Ladics, G. A. Bannon, A. Silvanovich, and R. F. Cressman, "Comparison of conventional FASTA identity searches with the 80 amino acid sliding window FASTA search for the elucidation of potential identities to known allergens," *Mol. Nutrition Food Res.*, vol. 51, no. 8, pp. 985–998, Aug. 2007.
[11] N. Huang, H. Chen, and Z. Sun, "CTKPred: An SVM-based method for the prediction and classification of the cytokine superfamily," *Protein Eng., Des. Selection*, vol. 18, no. 8, pp. 365–368, Aug. 2005.
[12] S. Lata and G. P. S. Raghava, "CytoPred: A server for prediction and classification of cytokines," *Protein Eng. Des. Selection*, vol. 21, no. 4, pp. 279–282, 2008.
[13] C. Z. Cai L. Y. Han, Z. L. Ji, X. Chen, and 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, 2003.
[14] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.
[15] 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.
[16] 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 Bioinf.*, vol. 8, no. 3, pp. 282–293, 2013.
[17] 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, 2018.

[18] 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, 2018.

[19] 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.

[20] 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.

[21] 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. 971–982, 2018.

[22] 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.

[23] 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.

[24] 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, 2017, Art. no. e1005420.

[25] 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.

[26] L. Wei, P. Xing, R. Su, G. Shi, Z. 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, 2017.

[27] 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.

[28] 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, 2018.

[29] 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.

[30] 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.

[31] 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, 2019.

[32] 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.

[33] 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.

[34] 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, 2019, doi, 10.1093/bioinformatics/btz418.

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

[36] 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.

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

[38] 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.

[39] 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.

[40] S. Liang, A. Ma, S. Yang, Y. Wang, and Q. Ma, "A review of matched-pairs feature selection methods for gene expression data analysis," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 88–97, Feb. 2018.

[41] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[42] H. Yang, W. Yang, F.-Y. Dao, H. Lv, H. Ding, W. Chen, and H. Lin, "A comparison and assessment of computational method for identifying recombination hotspots in Saccharomyces cerevisiae," *Briefings Bioinf.*, Oct. 2019, doi: 10.1093/bib/bbz123.

[43] H. Ding and D. Li, "Identification of mitochondrial proteins of malaria parasite using analysis of variance," *Amino Acids*, vol. 47, no. 2, pp. 329–333, 2015.

[44] B. Liu, "BioSeq-analysis: A platform for DNA, RNA and protein sequence analysis based on machine learning approaches," *Briefings Bioinf.*, vol. 20, no. 4, pp. 1280–1294, Jul. 2019.

[45] B. Yu, W. Qiu, C. Chen, A. Ma, J. Jiang, H. Zhou, and Q. Ma, "SubMito-XGBoost: Predicting protein submitochondrial localization by fusing multiple feature information and eXtreme gradient boosting," *Bioinformatics*, vol. 36, no. 4, pp. 1074–1081, 2020.

[46] C. Q. Feng, Z. Y. Zhang, X. J. Zhu, Y. Lin, W. Chen, H. Tang, and H. Lin, "ITerm-PseKNC: A sequence-based tool for predicting bacterial transcriptional terminators," *Bioinformatics*, vol. 35, no. 9, pp. 1469–1477, 2019.

[47] F.-Y. Dao, H. Lv, F. Wang, C.-Q. Feng, H. Ding, W. Chen, and H. Lin, "Identify origin of replication in saccharomyces cerevisiae using two-step feature selection technique," *Bioinformatics*, vol. 35, no. 12, pp. 2075–2083, Jun. 2019.

[48] Y. Liu, F. Tang, and Z. Zeng, "Feature selection based on dependency margin," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1209–1221, Jun. 2015.

[49] H. Liu and Z. Zhao, "Manipulating data and dimension reduction methods: Feature selection," in *Encyclopedia of Complexity and Systems Science*. Berlin, Germany: Springer, 2009, 5348–5359.

[50] H. Liu HM, R. Setiono, and Z. Zhao, "Feature selection, "An ever evolving frontier in data mining," in *Proc. JMLR Feature Sel. Data Mining*, Hyderabad, India, 2010, pp. 4–13.

[51] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[52] A. Unler and A. Murat, "A discrete particle swarm optimization method for feature selection in binary classification problems," *Eur. J. Oper. Res.*, vol. 206, no. 3, pp. 528–539, 2010.

[53] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, 2011.

[54] Y. Shen, J. Tang, and F. Guo, "Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou's general PseAAC," *J. Theor. Biol.*, vol. 462, pp. 230–239, Feb. 2019.

[55] C. Shen, L. Jiang, Y. Ding, J. Tang, and F. Guo, "LPI-KTASLP: Prediction of lncRNA-protein interaction by semi-supervised link learning with multivariate information," *IEEE Access*, vol. 7, pp. 13486–13496, 2019.

[56] Y. Ding, J. Tang, and F. Guo, "Identification of drug-side effect association via multiple information integration with centered kernel alignment," *Neurocomputing*, vol. 325, pp. 211–224, Jan. 2019.

[57] Y. Zhao, F. Wang, S. Chen, J. Wan, and G. Wang, "Methods of MicroRNA promoter prediction and transcription factor mediated regulatory network," *BioMed Res. Int.*, vol. 2017, pp. 1–8, Jun. 2017.

[58] L. Cheng, P. Wang, R. Tian, S. Wang, Q. Guo, M. Luo, W. Zhou, G. Liu, H. Jiang, and Q. Jiang, "LncRNA2Target v2.0: A comprehensive database for target genes of lncRNAs in human and mouse," *Nucleic Acids Res.*, vol. 47, no. 1, pp. D140–D144, Jan. 2019.

[59] X. Zeng, Y. Liao, Y. Liu, and Q. Zou, "Prediction and validation of disease genes using HeteSim scores," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 3, pp. 687–695, May 2017.

[60] L. Yu, J. Zhao, and L. Gao, "Drug repositioning based on triangularly balanced structure for tissue-specific diseases in incomplete interactome," *Artif. Intell. Med.*, vol. 77, pp. 53–63, Mar. 2017.

[61] L. Yu, B. Wang, X. Ma, and L. Gao, "The extraction of drug-disease correlations based on module distance in incomplete human interactome," *BMC Syst. Biol.*, vol. 10, no. 4, p. 111, Dec. 2016.

[62] Y. Qiao, Y. Xiong, H. Gao, X. Zhu, and P. Chen, "Protein-protein interface hot spots prediction based on a hybrid feature selection strategy," *BMC Bioinf.*, vol. 19, no. 1, p. 14, Dec. 2018.

[63] J.-X. Tan, S.-H. Li, Z.-M. Zhang, C.-X. Chen, W. Chen, H. Tang, and H. Lin, "Identification of hormone binding proteins based on machine learning methods," *Math. Biosci. Eng.*, vol. 16, no. 4, pp. 2466–2480, 2019.

[64] X. Zeng, W. Wang, C. Chen, and G. G. Yen, "A consensus community-based particle swarm optimization for dynamic community detection," *IEEE Trans. Cybern.*, early access, Sep. 23, 2019, doi: 10.1109/TCYB.2019.2938895.
[65] H. Xu, W. Zeng, D. Zhang, and X. Zeng, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, Feb. 2019.
[66] H. Xu, W. Zeng, X. Zeng, and G. G. Yen, "An evolutionary algorithm based on Minkowski distance for many-objective optimization," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3968–3979, Nov. 2019.
[67] T. Song, A. Rodriguez-Paton, P. Zheng, and X. Zeng, "Spiking neural P systems with colored spikes," *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 4, pp. 1106–1115, Dec. 2018.
[68] X. Chen, M. J. Pérez-Jiménez, L. Valencia-Cabrera, B. Wang, and X. Zeng, "Computing with viruses," *Theor. Comput. Sci.*, vol. 623, pp. 146–159, Apr. 2016.
[69] 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.
[70] 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, no. 1, pp. 154–159, 2005.
[71] 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, no. 1, pp. 115–119, 2004.
[72] 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, no. 1, pp. 187–191, 2006.
[73] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "CD-HIT suite: A Web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, pp. 680–682, Mar. 2010.
[74] Q. Zou, Z. Wang, X. Guan, B. Liu, Y. Wu, and Z. Lin, "An approach for identifying cytokines based on a novel ensemble classifier," *BioMed Res. Int.*, vol. 2013, pp. 1–11, 2013.
[75] W. Yang, X.-J. Zhu, J. Huang, H. Ding, and H. Lin, "A brief survey of machine learning methods in protein sub-golgi localization," *Current Bioinf.*, vol. 14, no. 3, pp. 234–240, Mar. 2019.
[76] *A Practical Guide to Support Vector Classification*. Accessed: Apr. 15, 2010. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
[77] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.
[78] 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, Mar. 2016.
[79] 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, 1623.
[80] W. Chen, H. Lv, F. Nie, and H. Lin, "I6mA-Pred: Identifying DNA N^6-methyladenine sites in the rice genome," *Bioinformatics*, vol. 35, no. 16, pp. 2796–2800, 2019.
[81] 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.
[82] 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, 2018.
[83] 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.
[84] 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.
[85] 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.
[86] 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.
[87] L. Chao, L. Wei, and Q. Zou, "SecProMTB: A SVM-based classifier for secretory proteins of Mycobacterium tuberculosis with imbalanced data set," *Proteomics*, vol. 19, Aug. 2019, Art. no. e1900007.
[88] 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.
[89] L. Chao, 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.
[90] 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.
[91] 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.
[92] 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.
[93] 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.
[94] 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, Dec. 2017.
[95] 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.
[96] 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, 2016.
[97] 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.
[98] X.-J. Zhu, C.-Q. Feng, H.-Y. Lai, W. Chen, and L. Hao, "Predicting protein structural classes for low-similarity sequences by evaluating different features," *Knowl.-Based Syst.*, vol. 163, pp. 787–793, Jan. 2019.
[99] H. Tang, Y.-W. Zhao, P. Zou, C.-M. Zhang, R. Chen, P. Huang, and H. Lin, "HBPred: A tool to identify growth hormone-binding proteins," *Int. J. Biol. Sci.*, vol. 14, no. 8, pp. 957–964, 2018.

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 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.

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, in 2002, and the M.S. degree and the Ph.D. degree 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 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.