Hierarchical Learning Architecture With Automatic Feature Selection for Multiclass Protein Fold Classification

Chuen-Der Huang*, Chin-Teng Lin, *Senior Member, IEEE*, and Nikhil Ranjan Pal, *Senior Member, IEEE*

***Abstract*—The structure classification of proteins plays a very important role in bioinformatics, since the relationships and characteristics among those known proteins can be exploited to predict the structure of new proteins. The success of a classification system depends heavily on two things: the tools being used and the features considered. For the bioinformatics applications, the role of appropriate features has not been paid adequate importance. In this investigation we use three novel ideas for multiclass protein fold classification. First, we use the *gating neural network*, where each input node is associated with a gate. This network can select important features in an online manner when the learning goes on. At the beginning of the training, all gates are almost closed, i.e., no feature is allowed to enter the network. Through the training, gates corresponding to good features are completely opened while gates corresponding to bad features are closed more tightly, and some gates may be partially open. The second novel idea is to use a *hierarchical learning architecture* (HLA). The classifier in the first level of HLA classifies the protein features into four major classes: all alpha, all beta, alpha + beta, and alpha/beta. And in the next level we have another set of classifiers, which further classifies the protein features into 27 folds. The third novel idea is to induce the *indirect coding features* from the amino-acid composition sequence of proteins based on the N-gram concept. This provides us with more representative and discriminative new local features of protein sequences for multiclass protein fold classification. The proposed HLA with new indirect coding features increases the protein fold classification accuracy by about 12%. Moreover, the gating neural network is found to reduce the number of features drastically. Using only half of the original features selected by the gating neural network can reach comparable test accuracy as that using all the original features. The gating mechanism also helps us to get a better insight into the folding process of proteins. For example, tracking the evolution of different gates we can find which characteristics (features) of the data are more important for the folding process. And, of course, it also reduces the computation time.**

***Index Terms*—Feature extraction, gating network, N-gram coding, protein sequence, radial basis function network (RBFN), Structure Classification of Protein (SCOP), support vector machine (SVM).**

Manuscript received June 3, 2003; revised August 19, 2003. This work was supported by the Brain Research Center, University System of Taiwan, under Grant 91B-711. *Asterisk indicates corresponding author.*

*C.-D. Huang is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C., and also with the Department of Electrical Engineering, Hsiuping Institute of Technology, Taichung 412, Taiwan R.O.C. (e-mail: cdhuang@mail.hit.edu.tw).

C.-T. Lin is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: ctlin@mail.nctu.edu.tw).

N. R. Pal is with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, WB 700108, India (e-mail: nikhil@isical.ac.in).

Digital Object Identifier 10.1109/TNB.2003.820284

I. Introduction

LARGE-SCALE sequencing projects produce a massive number of putative protein sequences. However, the growing of the number of known three-dimensional (3-D) protein structures is much slower than the sequence determined. This situation makes the need to extract structural information from the sequence database more imperative. Since the 3-D coordinate structures provide insight into the function, mechanism and evolution of protein, there are several famous classification databases such as Structure Classification of Protein (SCOP), Class, Architecture, Topology, and Homologous superfamily (CATH), DIAL-derived domain database (DDBASE), Entrez, and 3Dee, which imbue the structures with context and analysis. These different classification databases of proteins focus on their own characteristics. For example, comprehensive protein classification, such as SCOP, provides a detailed description of the structural and evolutionary relationships of the proteins of known structure. A more recent scheme, CATH is also a hierarchical classification of protein domain structure, which reveals the prominent features of protein structure space [1]–[5].

To classify databases of proteins which imbue the structures with context and analysis is very important for understanding the functions of proteins, and also essential for the discovery of new medication and therapies. In early days, such databases were made by factitious or semiautomatic procedure, such as SCOP or CATH. But recently, protein classification and protein fold prediction have been solved by the aid of computer with the strong ability of computation [6]–[8]. Computational methods have been developed for the assignment of a protein sequence to a folding class in the SCOP, where 83 folds are distinguished in 3D_ALI database and 128 folds are distinguished in the SCOP database [9]–[12]. In [9] and [11], the researchers have used primary global protein sequence in terms of three descriptors as physical, chemical, and structural properties of the constituent amino acids to code the sequences. Machine learning methods have been further induced into this complex classification problem.

Neural networks (NNs) and support vector machines (SVMs) are very widely used tools in machine learning strategy; these two algorithms should be very useful for such the complex problems of bioinformatics [6], [7]. The NN method, which has been widely used for decade, was a powerful tool for nonlinear and chaotic data. The SVM method, which has the advantage of fast convergence, was combined with the decision tree algorithm for

multiclass protein folds recognition in order to get higher classification accuracy [9], [11]. In particular, there have been several attempts to use NNs for prediction of protein folds. Dubchak *et al.* [9] point out that when we want a broad structural classification of protein—say, into four classes, all alpha (α), all beta (β), alpha + beta ($\alpha + \beta$), and alpha/beta (α/β)—it is easy to get more than 70% prediction accuracy using simpler feature vector for representing a protein sequence [7], [8], [10]. However, the problem becomes more and more difficult as we demand more refined classification into more classes. Dubchak *et al.* [9] used a multilayer perceptron network for predicting protein folds using global description of the chain of amino acids representing proteins. They used various combinations of the global features describing the physical, chemical, and structural properties of the constituent amino acids, and trained networks to find a good set of features. In [9], Dubchak *et al.* proposed an NN-based scheme for protein fold classification into 27 classes. This method like the one in [9], [11] also uses global descriptors of the primary sequence. They used proteins from the Protein Data Bank (PDB), where two proteins have no more than 35% sequence identity. For each fold an NN is trained. This procedure was repeated seven times for each fold, and each time only one set of features computed from a particular attribute was used. Then a voting mechanism was used to decide on the fold of a given protein. All these investigations clearly suggest that the choice of the right features is very important for a better classification of protein folds.

Although the bioinformatics researchers acknowledged the importance of feature analysis, no systematic efforts to find the best set of features have been done—mostly authors have used enumeration techniques. Feature analysis is more important for bioinformatics applications for two reasons: the class structure is highly complex and the data are usually in very large dimension. Most of the feature analysis techniques available in the pattern recognition literature are offline in nature. It is known that all features that characterize a data point may not have the same impact with regard to its classification, i.e., some features may be redundant and also some may have derogatory influence on the classification task. Thus, selection of a proper subset of features from the available set of features is important for design of efficient classifiers. There are methods for selecting good features on the basis of feature ranking, etc. [13]–[17].

In this investigation we use three novel ideas. First, we use NNs where each input node is associated with a gate. At the beginning of the training all gates are almost closed, i.e., no feature is allowed to enter the network. During the training, depending on the requirements, gates are either opened or closed. At the end of the training, gates corresponding to good features are completely opened while gates corresponding to bad features are closed more tightly. And of course, some gates may be partially open. Hence, the network can not only select features in an online manner when the learning goes on, but it also does some feature extraction. The second novel idea is to propose a new hierarchical learning architecture (HLA) to cope with the multiclass protein fold classification problem. At the first level of HLA, the network classifies the data into four major classes: α, β, $\alpha + \beta$, and α/β. And in the second level we have another set of networks, which further classifies the data into 27 folds. The proposed architecture can house a set of either NNs or SVMs as basic building blocks, with each being a multiclass classifier inherently. This is in contrast to the original approaches in [9] and [11], where a series of two-class classifiers and a voting scheme must be used to solve the same problem and avoided the derivative problem, i.e., the "false positive" problem. The third novel idea is to induce the indirect coding features from the amino-acid composition sequence of proteins based on the N-gram concept. In addition to the aforementioned traditional global features, we derive new local features describing the chain of amino acids representing proteins using the bigram and new spaced-bigram coding methods. These kinds of features can well describe the interactions among neighboring amino acids locally in a 3-D structure of the amino-acid composition sequence of proteins. This provides us with more representative and discriminative new features of protein sequences for the problems of multiclass protein fold classification.

The proposed HLA with new N-gram coded features increases the protein fold classification accuracy by about 12% than the conventional methods. Moreover, the gating network is found to reduce the number of features drastically. Using only half of the original features selected by the gating network can reach comparable test accuracy as that using all the original features. The process also helps us to get a better insight into the folding process. For example, tracking the evolution of different gates we can find which characteristics (features) of the data are more important for the folding process. And, of course, it reduces the computation time. The experiments on the same datasets and protein characteristics also show that the proposed HLA can achieve higher classification accuracy with smaller number of classifiers and lower computation overhead. Furthermore, due to the removal of the voting mechanism, the numerical output value of the classifiers in the proposed HLA can indicate the reliability and confidence of the prediction. Since each protein is classified with different reliability, such a reliability score is necessary for practical prediction systems.

The rest of this paper is organized as follows. Section II introduces the protein datasets used in the target problem of this research. Section III introduces the conventional global features as well as the proposed local features describing the chain of amino acids representing proteins. The proposed HLA housing NNs or SVMs is described in Section IV. The online feature selection scheme through gating NNs is proposed in Section V. The accuracy measurement indices of protein fold classification are discussed in Section VI. The experimental results and discussions are given in Section VII, and conclusions are made in Section VIII.

II. Protein Datasets

The SCOP is a famous protein databank, which uses the evolution and similarity of proteins to classify the structure of proteins. The data structure of SCOP is found according to the hierarchical structure of proteins, where the hierarchical classification scheme is widely used in bioinformatics such as SCOP and CATH. In the SCOP database, the main classes are divided into several classes. The main classes, with most numbers of

TABLE I
PATTERN NUMBERS OF EACH CLASSES IN SCOP WHICH WAS PICKED UP TO BE TRAINING AND TESTING PATTERNS IN THIS STUDY

Classes	Pattern Number (Training Data)	Pattern Number (Testing Data)
All Alpha	55	61
All Beta	109	117
Alpha/Beta	115	145
Alpha+Beta	34	62
Total Number	313	385

TABLE II
FOLD NUMBERS OF EACH CLASS AND PATTERN NUMBERS OF EACH FOLD IN SCOP WHICH WAS PICKED UP TO BE TRAINING AND TESTING PATTERNS IN THIS STUDY

Classes	Fold number per class (Training pattern per fold)		Fold number per class (Testing pattern per fold)	
All Alpha	6	13,7,12,7,9,7	6	6,9,20,8,9,9
All Beta	9	30,9,16,7,8,13,8,9,9	9	44,12,13,6,8,19,4,4,7
Alpha/Beta	9	29,11,11,13,10,9,10,11,11	9	48,12,13,27,12,8,14,7,4
Alpha+Beta	3	7,13,14	3	8,27,27
Total Number	27		27	

protein, are α, β, α/β, and $\alpha+\beta$. The other classes in SCOP such as multidomain proteins, membrane and cell surface proteins, and small proteins are less than the four main classes in amount. These four classes are named by the structure of proteins [2]–[5]. The protein classification in SCOP was performed manually or semiautomatically, which takes a great amount of time for such a complex task. It has been a pushing research topic to classify and predict the multiclasses of proteins by machine learning methods [2], [3], [9], [11].

A. Training Dataset

This training dataset was built for the prediction of 128 protein folds based on the PDB selected sets. The data set was selected by their characteristics so that all proteins in the data set have less than 35% of the sequence identity for the aligned subsequences longer than 80 residues. Following the prior published papers [3], [9], [11], the training data number is 313 and they should be divided into four classes with 27 folds according to their structures representing all major structural classes.

B. Testing Dataset

An independent dataset was also taken for testing the effect of prediction. The testing dataset was based on PDB-40D set developed by the authors of the SCOP database [2]–[5]. A total number of 385 proteins with identity less 40%, same as those used by Dubchak and Ding, were selected for testing in our study. Table I shows the numbers of proteins in the training and testing datasets for different protein classes used in our experiments. Table II shows the numbers of proteins in the training and testing datasets for different folds of each protein class used in our experiments, where there are 27 folds for the four classes in total.

III. GLOBAL AND LOCAL FEATURES DESCRIBING THE AMINO-ACID SEQUENCES

Before applying the machine learning methods to handle the bioinformatics problems, the features extraction of the analyzed data is a very important task, since different extracted features may cause different classification results, better or worse. Two major approaches, the direct coding method and the indirect coding method, are used in bioinformatics to extract features from experimental data. The direct coding method contains position-depend, sequence-length-depend, and a vector per residue. On the other hand, the indirect coding method is position independent, length invariant, and a vector per sequence.

A. Global Features—Physical/Chemical Characteristics

In the previous studies [9] and [11], several features have been considered for predicting protein folds using global description of the chain of amino acids representing proteins. These descriptors were computed from the physical, chemical, and structural properties of the constituent amino acids. Different properties of the amino acids were used as features such as the relative hydrophobicity of amino acids. The information about the predicted secondary structure and predicted solvent accessibility was also used. They divided the amino acids into three groups based on hydrophobicity, three groups based on secondary structure, and four groups based on solvent accessibility. A protein sequence was then described based on three global descriptors: composition (C), transition (T), and distribution (D) [9], [11]. These descriptors essentially describe the frequencies with which the properties change along the sequence and their distribution on the chain. In addition to the three amino-acid attributes described above, three more attributes were usually used: normalized Van Der Walls volume, polarity, and polarizability. They also used the percent composition of amino acids as feature vectors. Let there be M folds in the data set. For each fold, the data set were divided into two groups, one containing points from the fold and the other containing the rest. So there are M such partitions.

In this study, we also adopt the aforementioned six kinds of physical or chemical characteristics (attributes) of proteins for fold classification. There are composition (C), predicted secondary structure (S), hydrophobicity (H), normalized Van Der Waals volume (V), polarity (P), and polarizability (Z). The six kinds of protein sequence information (PSI) are extracted from the provided open protein database. Except for the first PSI, C, the same set of descriptors is used for all the other PSIs resulting in a feature parameter vector in 21 dimensions for each of S, H, V, P, and Z. The first kind of PSI, C, is the sequence composition of amino acids. It is known that there are totally 20 types of amino acids; therefore, these 20 kinds of amino acids are corresponding to a 20-dimensional feature vector. Table III shows the symbols, descriptors, and dimensions of these six PSIs used in our experiments.

TABLE III
THE DESCRIPTORS AND FEATURE DIMENSION SIZES OF EACH OF THE SIX PROTEIN ATTRIBUTES

Characteristics	Descriptors			Feature Size
Composition (C)	20 kinds of amino acids			20
Predicted Secondary Structure (S)	Alpha	Beta	Loop	21
Hydrophobicity (H)	Positive	Neural	Negative	21
Volume (V)	Large	Middle	Small	21
Polarity (P)	Positive	Neural	Negative	21
Polarizability (Z)	Strong	Middle	Weak	21
Total Number				125

We feed these features to our classifiers from single PSI to multiple PSIs progressively. In the experimental reports of this study, the symbol "+" denotes the combination of feature information. It means that we feed more than one PSI into the classifiers once. The summed dimensions of the PSIs are corresponding to the input nodes of the NN classifiers or the input variables of the SVM. In our experiments, we used different combinations of PSIs as input features to each classifier. Hence, while we used the physical or chemical characteristics, the two extreme cases are: 1) the use of the composition of amino acids only and 2) the use of all six PSIs. In the first case, the feature dimension is 20, and in the second case, the feature dimension is up to 125 (20+21+21+21+21+21).

B. Local Features—N-Gram Coding

The six types of PSI introduced above are kinds of global features extracted by the direct encoding method. They emphasize more on the global properties and structures of the amino-acid sequences, and less on the local interactions among neighboring amino acids. In this section, we shall induce the indirect coding features from the amino-acid composition sequence of proteins based on the N-gram concept. We shall develop new local features describing the chain of amino acids representing proteins using the bigram and new spaced-bigram coding methods. These kinds of features can well describe the interactions among neighboring amino acids locally in a 3-D structure of the amino-acid composition sequence of proteins. In extracting such local features, in addition to the traditional bigram coding scheme, we also propose the new spaced bigram coding scheme, which can better describe the 3-D protein structure caused by the mutual interactions among interleaving (every other) neighboring amino acids in a protein sequence. This provides us with more representative and discriminative new features of protein sequences for the problems of multiclass protein fold classification.

For a sequence composed of M alphabets, a bigram coding scheme applied on it will produce a new sequence (i.e., feature vector) with M^2 dimensions. Each element in the feature vector represents the number of appearance of a specific pairwise combination of the M alphabets in the neighboring two amino acids of the sequence. Since a protein sequence is composed of 20 kinds of general amino acids represented by 20 alphabets, respectively, and other types of amino acids represented by a common alphabet B or Z, it is a sequence composed of 21 alphabets. Hence, after the bigram coding, we obtain a feature vector with 411 dimensions for a protein sequence. Similar to the bigram coding, the newly proposed spaced bigram coding is to detect the appearance frequency of any two-alphabet pair in every other (interleaving) neighboring amino acids of a protein sequence. Hence, the spaced bigram coding on a protein sequence also produce a feature vector with 441 dimensions, each representing the number of appearance of a specific pairwise combination of the 21 alphabets in the every other neighboring two amino acids of the sequence.

Consider a segment of the amino-acid sequence of the protein with ID number 1pga: MTYKLILNG as an example. In the bigram coding, we count the numbers of the pairs (MT), (TY), (YK), (KL), etc., respectively. In the spaced bigram coding, we count the numbers of the pairs (MY), (TK), (YL), (KI), etc., respectively. It is believed that the mutual interactions between every two neighboring amino acids, and also the mutual interactions between every other two neighboring amino acids play the key roles in the 3-D structure of a protein sequence. It was even claimed that the effect of the latter type of interactions is stronger than the former type of interaction.

From both the direct and indirect coding schemes in the above two subsections, we have now obtained eight types of PSI. The first six types belonging to global features represent the physical–chemical characteristics of a protein sequence, and the other two types belonging to local features represent the mutual interactions between neighboring amino acids. If we use all of these features at once, the feature space will be of 1007 (20+21+21+21+21+21+441+441) dimensions, which is a large number. This motivates the study of automatic feature selection for protein fold classification in this research.

IV. HIERARCHICAL LEARNING ARCHITECTURE

In Section II and from Tables I and II, we find that the fold characteristics of the proteins are separated into four mainly typical classes named as all α, all β, α and β (α/β), and α plus β ($\alpha+\beta$), respectively. Within each class, it contains several different numbers of folds in it, with a total number of 27 folds. The purpose of this work, multiclass protein fold classification, is to classify each of the proteins into one of the 27 folds. According to the classification characteristics of the protein data, a novel HLA including two-level of classifiers is proposed, as shown in Fig. 1. In the first level, a multiclass classifier for recognizing the four protein classes is used. In the second level, we perform detailed classification on each class. There are four independent multiclass classifiers used in the second level for finer protein fold recognition, from four classes to 27 folds (see Table II). The proposed HLA is an effective learning structure, in the sense of reducing the numbers of classifiers, avoiding the voting scheme, and increasing the accuracy of protein fold recognition.

In Fig. 1, we illustrate how the proposed HLA is used in actual experimental data to handle input features. In the first level,

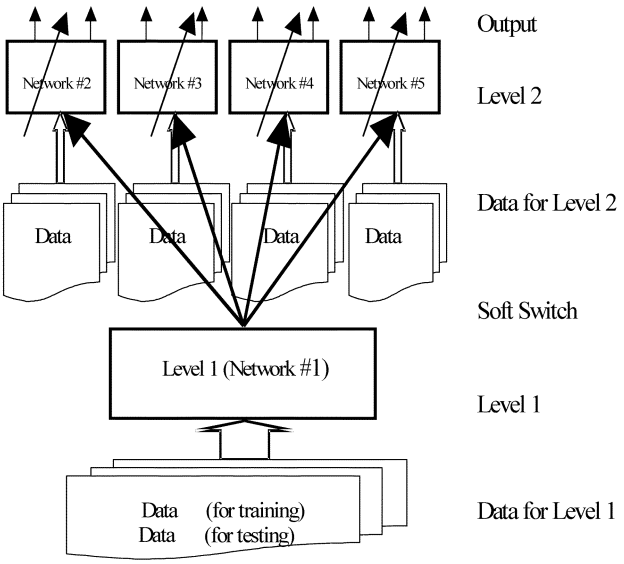

Fig. 1. Proposed HLA for protein folds classification.

a multiclass classifier (labeled as Classifier #1) is used to distinguish input proteins data into four classes, denoted as I, II, III, and IV (α, β, α/β, and $\alpha+\beta$, correspondingly). Here we shall adopt proper PSI introduced in Section III as the inputs of this classifier in our experiments. The second level in the HLA consists of four smaller independent multiclass classifiers (labeled as Classifiers #2 to #5), each for the fold recognition of different class of protein data classified by the Level 1 classifier. In other words, Classifier #2 is to classify the protein data, which are classified as Class I by Classifier #1, into one of six fold types. Similarly, Classifier #3 is to classify the Class II protein data into one of nine fold types, Classifier #4 is to classify the Class III protein data into one of nine fold types, and Classifier #5 is to classify the Class IV protein data into one of three fold types. So, totally 27 (6+9+9+3) folds are recognized by the Level 2 classifiers.

In general, a q-dimensional data set is used to train the HLA represented by Level 1 and Level 2 in Fig. 1. Let the training data be $X_{Tr} = X_1 U X_2 U X_3 U X_4$, where X_i is the training data corresponding to class i. First we train the Level 1 classifiers using X. The Level 1 classifier divides the data into four classes. Note that the division of X made by the Level 1 classifier may not exactly correspond to $X_{Tr} = X_1 U X_2 U X_3 U X_4$. The Level 2 classifiers are independently trained; the ith Level 2 classifier is trained with X_i. Once the training of the second level classifiers is over, the system is ready to be tested. A q-dimensional data point is now fed into the Level 1 classifier which will classify the point to one of the four classes; say, it is classified to Class 3. Then the training data point is fed to the third classifier (Classifier #4) in the second level. It should be noted here that, for such architecture, if the Level 1 classifier makes any mistake, then Level 2 classifiers cannot recover the same. The proposed HLA is quite general in nature and, hence, for both Level 1 and Level 2, we can use any classification network; in fact, we can use any nonneural classifier, too.

The concept of the proposed HLA is neither the same as the cascade network nor as the divide-and-conquer network. The constituents of the HLAs are all independent networks. It likes a sieve to sieve the data out of the input training data to several different groups. In fact, this HLA is suitable for data sets that can be grouped into a smaller number of classes, where each class can further be divided into a set of clusters. The problem we handle, multiclass and multifold classification of protein structures, has this kind of characteristic. Also, since the proposed HLA houses a set of multiclass classifiers as the basic building blocks, it dose not need a stochastic voting mechanism after a long series of two-class classifications normally used in bioinformatics.

In our experiments, we shall use NN and SVM, all belonging to the machine learning family, as the basic building blocks of our HLA. We shall introduce these classifiers and the experimental results in Sections IV-A and IV-B. No matter what kind of classifiers we choose, the overall classification results are better than those of the one-versus-others method (OvO) method with NN, and are even better than those of the existing modified OvO method [9]. Such higher classification accuracy is obtained by using fewer classifiers with smaller network size. The extra decision mechanism such as voting scheme is also avoided.

A. Neural Networks

NNs have been developed for many years and been used well in various applications. Many researchers continue to apply different algorithms and develop different structures to enhance the ability of NNs. Here we use NN models as the multiclass classifiers in the HLA. Some brief introductions about two popular NN models are given below.

1) Multilayer perceptron (MLP) is a classic and widely used NN model. Such a network can solve nonlinear regression, and construct global approximation to the nonlinear input–output mapping [18], [19].
2) The radial basis function network (RBFN) is a three-layer network. The hidden layer nodes use a basis function, the Gaussian function, as the activation function. Unlike the MLP network, the output nodes are linear. The RBFN, suggested by Moody [20], is very suitable to be used as classifier. The RBFN used here can grow its hidden nodes automatically. When data are fed into the network, the sum square error (SSE) will be calculated with the cost function, and the backpropagation (BP) learning rule is used to minimize the SSE until the restrict number of nodes or the preset value of SSE arrived [21], [22].

B. Support Vector Machines

An SVM is a new-generation learning algorithm based on recent advances in statistical learning theory. In the early 1990s, their introduction leads to a recent explosion of applications and deepening theoretical analysis. Basically, the SVM is a typical two-class classifier and a kind of universal feedforward network which was developed by Vapnik and his colleagues at Bell Laboratories, and has been improved by other researchers. It is a kind of machine learning algorithm, while in operation, the SVM will construct a hyperplane in a high-dimensional features space as the decision surface between positive and negative patterns. The

structural risk minimization ability makes the SVM a very efficient classifier in various applications including biosequences analysis, etc. [23]–[26].

With the further improvements by other researchers recently, the SVM has the ability to do multiclass classification directly [27], which is the model adopted here in our HLA as the constituent multiclass classifiers. In practice, the SVM algorithm is used in three types of learning machine: 1) polynomial learning machines; 2) RBFNs; and 3) two-layer perceptrons (MLPs). In this study, we choose the RBFNs for the SVM algorithm and act as the kernels (building blocks) of the proposed HLA.

V. Online Feature Selection Through Gating

Due to the large number of input dimensions in the multifold classification of protein structures, especially for the combined global and local features, it is essential to perform important feature selection automatically. In general, feature selection methods could be classified into two major categories. One is based on statistical information of features; the other is based on classifiers. These two major methods have their differences in concepts. The former is based on statistics criteria to find out the optimal subset, and the latter is based on the learned weights to find out the useless features or point out the most importance features by the preset criteria. The latter methods commonly use NNs to complete the feature selection work [14]–[17].

In a standard MLP network, the effect of some features (inputs) can be eliminated by not allowing them into the network, i.e., by equipping each input node (hence, each feature) with a gate and closing the gate. For good features the associated gates can be completely opened. On the other hand, if a feature is partially important, then the corresponding gate should be partially opened. Pal and Chintalapudi suggested a mechanism for realizing such a gate so that "partially useful" features be identified and attenuated according to their relative usefulness [13], [16], [17]. In order to model the gates, we consider an attenuation function for each feature such that for a good feature the function produces a value of one or nearly one; while for a bad feature, it should be nearly zero. For a partially effective feature, it should have a value that is intermediate to these extremes. To model the gate, we multiply the input feature value by its gate function value and the modulated feature value is passed into the network. The gate functions attenuate the features before they propagate through the network, so we may call these gate functions attenuation functions. A simple way of identifying useful gate functions is to use sigmoidal functions with a tunable parameter, which can be learned using training data. To complete the description of the method, we define the followings in connection with a MLP network.

Let $F_i : R \rightarrow [0, 1]$ be the gate or attenuation function associated with the ith input feature, F_i have an argument w_i, $F_i'(w_i)$ be the value of derivative of the attenuation function at w_i; μ be the learning rate of the attenuation parameter; v be the learning rate of the connection weights, x_i be the ith input of an input vector, x' be the attenuated value of x, i.e., $x' = xF(w)$, w_{ij}^0 be the weight connecting the jth node of the first hidden layer to the ith node of the input layer, and δ_j^i be the error term for the jth node of the first hidden layer. It can be easily shown that except for w_{ij}^0, the update rules for all weights remain the same as that for an ordinary MLP. Assuming that the first hidden layer has q nodes, the update rules for w_{ij}^0 and w_i are

$$w_{ji,\text{new}}^0 = w_{ji,\text{old}}^0 - vx_i\delta_j^1 F(w_i) \tag{1}$$

$$w_{i,\text{new}} = w_{i,\text{old}} - \mu x_i \left(\sum_{j=1}^{q} w_{ji}^0 \delta_j^1 \right) F'(w_i). \tag{2}$$

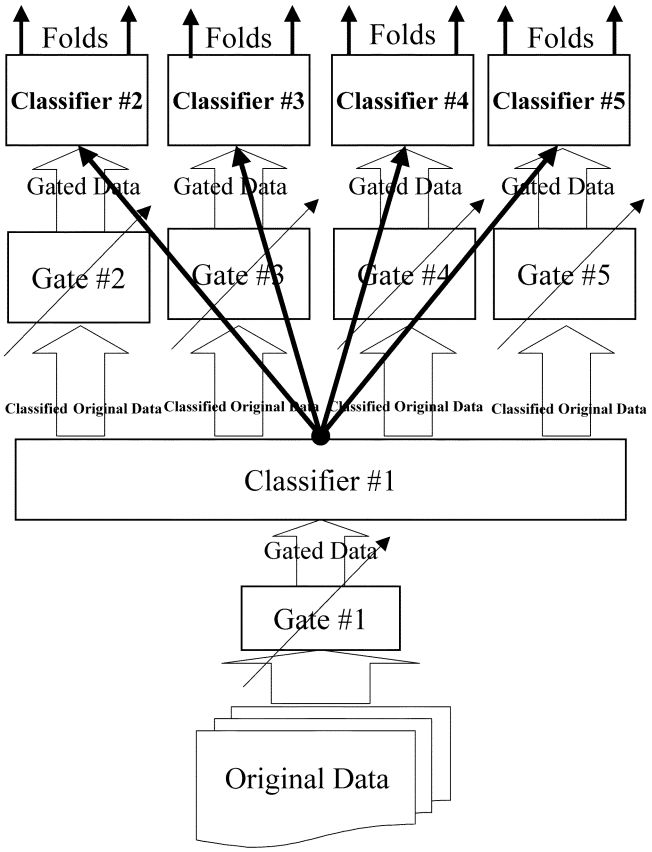

Fig. 2. Proposed HLA with gating network for online feature selection. The arrows on gates represent the variable online feature selection function.

Although for the gate function, several choices are possible, we use here the sigmoidal function $F(w) = 1.0/(1 + e^{-w})$. The q gate parameters are so initialized that when the training starts, $F(w)$ is practically zero for all gates; i.e., no feature is allowed to enter the network. As the backpropagation learning proceeds, gates for the features that can reduce the error faster are opened. Note that the learning of the gate function continues along with other weights of the network. At the end of the training, important features can be picked up based on the values of the attenuation function [13], [16], [17].

This feature selection mechanism is put in front of every multiclass classifier in the HLA such that each classifier can select the most important features for its respective classification problem as shown in Fig. 2. In other words, before a set of training data are sent into a classifier in the HLA for training, they are passed into the feature selection mechanism (i.e., the gating network) first. According to the results of feature selection, only the training data corresponding to the selected important features are used for the training of the classifier, which is

RBFN or SVM in our HLA. Also, in the testing phase, only the same selected important features are fed into the classifiers. It is noted that since every classifier in the HLA aims at different classification job, the important features selected for each classifier might not be the same, although every classifier faces the same original input training data before feature selection.

VI. Measurement Indices of Classification Accuracy

In bioinformatics, because the two-way classifiers are usually used, several different accuracy measurement methods were proposed to account for the confusing situations of "true positive" or "false positive" [7]. In our HLA classification approach, such confusing conditions will not happen. Therefore, the accuracy measurement in our experiments is quite clear and simple. Let us use a function A (accuracy) to indicate the classification correctness of a protein pattern fed into the HLA. Then the total number of correctly classified proteins can be expressed as

$$\begin{aligned} C &= A(\text{level 2}|\text{level 1}) \\ &= A(\text{level 2})A(\text{level 1}I\ \text{level 2}) \end{aligned} \tag{3}$$

where A is a conditional function whose value is one only when a protein pattern is correctly classified by the classifiers in both Level 1 and Level 2 of the HLA, and is zero otherwise.

Based on the above concepts, the accuracy measurement of the proposed approach is defined as follows. If the number of testing proteins belonging to the F_ith fold is n_i, but the tested classifier only recognizes c_i proteins as the F_ith fold, then the accuracy rate of this tested classifier is set as c_i/N_i for the F_ith fold. In addition to the calculation of individual accuracy, the total classification accuracy can be briefly calculated as follows:

$$\begin{aligned} N &= n_1 + n_2 + n_3 + \cdots + n_i \\ &= \sum_{i=1} n_i \ (\text{in this case, } i = 27, N = 385) \end{aligned} \tag{4}$$

$$\begin{aligned} C &= c_1 + c_2 + c_3 + \cdots + c_i \\ &= \sum_{i=1} c_i \ (\text{in this case, } i = 27) \end{aligned} \tag{5}$$

$$Q = \frac{C}{N} \tag{6}$$

where N is the total number of testing proteins data, C is the total number of correctly classified proteins in (3), and Q is the classification (prediction) accuracy.

VII. Experimental Results

To test and demonstrate the proposed techniques for multiclass protein fold classification, several experiments are designed and performed and the results are reported and discussed in this section. These experiments are based on the protein database, SCOP, introduced in Section II. To demonstrate the three novelties of the proposed techniques, our experiments are divided into three parts focusing on the proposed HLA, new local features of protein sequences, and automatic feature selection mechanism, respectively, in Sections VII-A–C.

TABLE IV
Protein Fold Classification Accuracy of Various Single-Level Classification Approaches, Where the Input PSIs Fed Into the Classifier Are C + S + H + P + V + Z

Accuracy \ Classifier	MLP	GRNN	RBFN	SVM
Q(C+S+H+P+V+Z) (%)	48.8	44.2	49.4	51.4

A. Experiments on HLA

In the experiments of this subsection, we shall perform extensive tests on the effectiveness of the proposed HLA with different constituent classifiers fed with different combinations of conventional global PSI features. In the experiments, we use four different multiclass classifiers as the basic building blocks in the proposed HLA, respectively. They are MLP, RBFN, General Regression Neural Network (GRNN) [18]–[22], and SVM, introduced in Section IV. The used MLP has three hidden layers with 40, 80, and 40 sigmoid nodes, respectively. The used RBFN has only one hidden layer, where various numbers of hidden nodes are tested as stated below. The used GRNN also has only one hidden layer. The used SVM is the multiclass SVM proposed in [27]. In each case, the whole HLA is trained completely. Especially, the nodes and training epochs of NNs are chosen carefully during the experiments to avoid the overfitting problem.

In our experiments, the proposed HLA with different basic building blocks (classifiers) is used to recognize the protein folds given in the SCOP database. Six different combinations of features were used as the input vectors to the classifiers, respectively. They are C, C + S, C + S+ H, C + S + H + P, C + S + H + P + V, and C + S + H + P + V + Z, where each character represents a kind of PSI defined in Table III, and "+" means combination. For performance comparisons, we also use each of MLP, RBFN, GRNN, and SVM to classify the proteins into 27 folds directly without using the proposed HLA, where 27 output nodes are used in each NN model. We call this the single-level approach. Table IV lists the classification rates of various single-level approaches, where the full set of PSIs are used as input features. It is observed that the average classification accuracy Q is only about 50%. The classification accuracies of the proposed HLA with various NN or SVM classifiers with respect to different combinations of PSIs are listed in Table V. It is observed that the HLA can increase the classification accuracy Q by about 7%. Also, more PSIs result in higher Q values.

The results obtained by the proposed HLA are also better than those by the OvO method, unique OvO method (uOvO), and all-versus-all method (AvA) methods proposed in [9]. These methods require a series of SVMs or NNs and a voting mechanism. The comparison results are given in Table VI. Table VI shows that the overall classification results of the proposed approach are normally better than those of the compared counterparts. Especially, the proposed HLA with the RBFN classifiers achieves the best classification accuracy, 56.4%, which is higher than the best result (53.9%) achieved by the AvA method with the two-class SVM classifiers [AvA(SVM)] proposed in [9]. The higher classification accuracy of the proposed approach

TABLE V
PROTEIN FOLD CLASSIFICATION ACCURACY OF THE PROPOSED HLA WITH VARIOUS NN OR SVM SUBCLASSIFIERS, WHERE DIFFERENT COMBINATIONS OF PSIS ARE TESTED. THE CORRESPONDING CLASSIFICATION ACCURACIES OF VARIOUS SINGLE-LEVEL CLASSIFICATION APPROACHES ARE ALSO SHOWN FOR COMPARISONS

Classifiers & PSIs / Accuracy(%)		Single-level Learning Architecture	Hierarchical Learning Architecture
RBFN	C	48.6	44.9
	C+S	50.7	53.8
	C+S+H	52.0	53.3
	C+S+H+P	50.7	54.3
	C+S+H+P+V	49.1	55.3
	C+S+H+P+V+Z	48.4	56.4
GRNN(C+S+H+P+V+Z)		44.2	45.2
SVM (C+S+H+P+V+Z)		51.4	53.8

TABLE VI
PROTEIN FOLD CLASSIFICATION ACCURACY COMPARISONS OF THE PROPOSED HLA AND THE EXISTING APPROACHES, WHERE "OVO" STANDS FOR THE ONE-VERSUS-OTHERS METHOD, "UOVO" FOR THE UNIQUE ONE-VERSUS-OTHERS METHOD, AND "AVA" FOR THE ALL-VERSUS-ALL METHOD

Features(Accuracy) / Classifiers	C (%)	C+S (%)	C+S+H (%)	C+S+H+P (%)	C+S+H+P+V (%)	C+S+H+P+V+Z (%)
OvO (NN)*	20.5	36.8	40.6	41.1	41.2	41.8
OvO (SVM)*	43.5	43.2	45.2	43.2	44.8	44.9
uOvO(SVM)*	49.4	48.6	51.1	49.4	50.9	49.6
AvA (SVM)*	44.9	52.1	56.0	56.5	55.5	53.9
RBFN (Single-level)**	40.3	48.6	50.1	52.0	49.1	49.4
HLA (MLP)	32.7	48.6	47.5	43.2	43.6	44.7
HLA (RBFN)	44.9	53.8	53.3	54.3	55.3	56.4
HLA (GRNN)	-----	----	----	----	----	45.2
HLA (SVM)	-----	----	----	----	----	53.2

Note: * Data from the paper (Dubchak *et al.*, 2001 [9]).
** Using RBFN directly to classify the proteins into 27 folds (i.e., single-level approach).

is obtained by using fewer classifiers with smaller size. Also, the extra decision mechanism such as the voting scheme is also avoided.

In Table VI, the architecture of the used RBFN-based HLA consists of five RBFN classifiers as shown in Fig. 1, with a total of 366 hidden nodes. The RBFN can find the proper number of hidden nodes by itself during the training process. In our HLA, the largest RBFN is Classifier #1 in Level 1, which contains 145 hidden nodes. The smallest RBFN is Classifier # 5 in Level 2, which contains only 13 hidden nodes. More detailed information about the node numbers are given in Table VII. For comparisons, the total number of hidden nodes used in the single-level RBFN is 125, which achieves 49.4% classification accuracy, and cannot be better even with more hidden nodes. Also, in the AvA(SVM) method proposed in [9], which achieved 53.9% classification accuracy, a total of 351 two-way SVM classifiers were used. In another experiment, we further compare the classification accuracy of the RBFN-based HLA with those of the single-level RBFN. In this experiment, we switched the roles of training data and testing data used in the previous experiments. The results are listed in Table VIII indicating the superiority of the proposed approach again.

TABLE VII
NUMBER OF NODES USED IN THE RBFNS OF THE PROPOSED RBFN-BASED HLA

Level in HLA	Level 1 RBFN #1	Level 2				Total
		RBFN #2	RBFN #3	RBFN #4	RBFN #5	
Number of Hidden Nodes	145	38	101	69	13	366
Number of Output Nodes	4	6	9	9	3	27

TABLE VIII
COMPARISONS OF THE PROTEIN FOLD CLASSIFICATION ACCURACY OF THE PROPOSED RBFN-BASED HLA AND THE SINGLE-LEVEL RBFN WITH TRAINING DATA AND TESTING DATA EXCHANGED, WHERE THE FULL SET OF PSIS ARE USED

Method / Classification	Single-level RBFN	RBFN-Based HLA
Total Number	174/313	183/313
Accuracy(%)	55.6	58.5

TABLE IX
REQUIRED TRAINING TIME OF THE SINGLE-LEVEL RBFN AND EACH RBFN IN THE PROPOSED HLA, WHERE THE TOTAL SIX PSIS ARE USED AS NETWORK INPUTS, AND THE TRAINING IS PERFORMED IN A PERSONAL COMPUTER WITH INTEL PENTIUM IV CPU UNDER 1-GHZ CLOCKS

Classifier	CPU Time (sec.)		
Single-level	95.6		
RBFN-Based HLA	Level 1 RBFN # 1		126.9
	Level 2	RBFN #2	1.1
		RBFN #3	15.3
		RBFN #4	8.4
		RBFN #5	0.3
	Total CPU Time		152.0

In Table IX, we also show the training time required by the single-level RBFN, and the training time required by each RBFN in the HLA, where the training was performed in a personal computer with Intel Pentium IV CPU under 1-GHz clocks. The results indicate that although Level 1 RBFN in the HLA consumed longer training time, the training of each Level 2 RBFN converged very quickly. This reflects the underlined "divide-and-conquer" philosophy of the proposed HLA. Although the total training time of the RBFN-based HLA is longer than that of the single-level one, this is the expense paid for higher classification accuracy. It is worthy to mention that the single-level RBFN could not perform better even more training time were taken in our experiments.

TABLE X
CLASSIFICATION ACCURACIES OF THE RBFN-BASED HLA WITH VARIOUS COMBINATIONS OF GLOBAL FEATURES (C + H + S + P + V + Z) AND LOCAL FEATURES [BIGRAM-CODED FEATURE (B) AND SPACED BIGRAM-CODED FEATURE (SB)]

RBFN-Based HLA					
Features		Global features (6 PSIs)	Local feature B	PSIs+B	PSIs+B+ SB
No. of Features		125	441	125+441	125+441 +441
Accuracy of Level 1		81.6	79.2	83.1	83.6
Accuracy of Level 2 (%)	Group 1	67.2	59.0	77.0	73.8
	Group 2	52.1	56.4	62.4	63.2
	Group 3	58.6	60.0	62.8	69.0
	Group 4	48.4	56.5	54.8	53.2
Overall Accuracy (%)		56.4	58.2	63.7	65.5

TABLE XI
CLASSIFICATION ACCURACIES OF THE SVM-BASED HLA WITH VARIOUS COMBINATIONS OF GLOBAL FEATURES (C + H + S + P + V + Z) AND LOCAL FEATURES [BIGRAM-CODED FEATURE (B) AND SPACED BIGRAM-CODED FEATURE (SB)]

SVM-Based HLA					
Features		Global features (6 PSIs)	Local feature B	PSIs+B	PSIs+B+ SB
No. of Features		125	441	125+441	125+441 +441
Accuracy of Level 1		81.3	77.9	83.4	84.4
Accuracy of Level 2 (%)	Group 1	60.7	57.4	73.8	73.8
	Group 2	49.6	53.8	59.0	60.7
	Group 3	56.6	60.0	64.8	65.5
	Group 4	45.2	59.7	52.6	58.1
Overall Accuracy(%)		53.2	57.7	62.3	64.2

As compared to the popular OvO method, and the modified uOvO method and AvA methods proposed in [9], the proposed HLA with embedded multiclass classifiers has another important advantage. Due to the removal of the voting mechanism required by the OvO, uOvO, and AvA methods, the numerical output value of the classifiers in the proposed HLA can indicate the reliability or confidence of the prediction. Since each protein is predicted with different reliability, such a reliability score is necessary for practical classification/prediction systems. For example, a low reliability score for a new protein may indicate that it does not belong to any fold in the system.

B. Experiments on New Protein Features

In Section VII-A, we find that the HLA housing RBFNs or SVMs achieved the best results among the compared counterparts. In this subsection, we shall focus on the HLA with these two building blocks fed with the combination of the conventional global features and the new local features of the amino-acid sequences of proteins proposed in Section III. In additional to the six types of PSI describing the physical/chemical characteristics of proteins used in Section VII-A, two new sets of local features obtained by the bigram coding and spaced bigram coding schemes are considered to add to the input vectors of the HLA. Table X shows the classification accuracies of the RBFN-based HLA with these combined features. Four different combinations of input features are tested: 1) the conventional six types of PSI (i.e., C + S + H + P + V + Z) (125 dimensions); 2) the bigram-coded feature vector (441 dimensions); 3) the combination of 1) and 2) (125 + 441 dimensions); and 4) the combination of 3) and the spaced bigram-coded feature vector (125 + 441 + 441 dimensions). Table X shows that the new local features did improve the accuracies of protein fold classification, obviously. It is observed that the addition of the bigram-coded feature to the original six PSI features increase the accuracy by 7.3%, which is even 9.8% higher than the result reported in [9]. Especially the full set of features including the global and local features improves the accuracy by 11.7% and achieves 65.5% classification rate in total. Table XI shows the experimental results of SVM-based HLA corresponding to those on Table X. This table also shows the advantages of the newly proposed local protein features in Section III.

TABLE XII
CLASSIFICATION ACCURACIES OF THE RBFN-BASED HLA WITH DIFFERENT GLOBAL FEATURES SETS (C + H + S + P + V + Z) SELECTED BY THE GATING NETWORK

RBFN-Based HLA		Number of Feature Selected			
		50	67	80	125
Accuracy of Level 1 (%)		79.2	80.3	80.8	81.6
Accuracy of Level 2 (%)	Class 1	47.5	50.8	73.8	67.2
	Class 2	47.9	51.3	56.4	52.1
	Class 3	51.0	53.1	54.5	58.6
	Class 4	48.4	54.8	87.1	48.4
Overall Accuracy (%)		49.1	52.5	53.0	56.4

C. Experiments on Automatic Feature Selection Scheme (Gating Network)

Sections VII-A and VII-B clearly indicate that the protein fold recognition problems always contain large feature dimensions, from 125 to 125 + 441 + 441. In this subsection, we shall test the automatic feature selection scheme proposed in Section V to reduce the feature dimensions for HLA. We shall first consider the HLA fed with the conventional six types of PSI (i.e., C + S + H + P + V + Z) (125 dimensions). Table XII presents classification performance of the RBFN-based HLA with different feature sets. The performance of Level 1 HLA (see Fig. 1) shows that with 67 features (50% reduction), the decrease in performance is only 1.26% while with 65% features the test accuracy is reduced by only 0.76%. This clearly suggests that the gating network can do an excellent job of selecting important features. Let us now consider the overall classification

TABLE XIII
VALUES OF THE GATING FUNCTIONS FOR THE MOST IMPORTANT 15 FEATURES AFTER DIFFERENT ITERATIONS

Feature Number	Gating function values after 1000 iterations	Feature Number	Gating function values after 1000 iterations	Feature Number	Gating function values after 1000 iterations
30	0.002657	82	0.002903	103	1.0
81	0.002677	98	0.002995	22	1.0
41	0.002774	79	0.003050	26	1.0
40	0.002952	83	0.003197	28	1.0
77	0.002964	92	0.003634	29	1.0
103	0.002970	40	0.003697	30	1.0
82	0.003042	81	0.004338	31	1.0
92	0.003211	41	0.004585	33	1.0
98	0.003256	103	0.007582	35	1.0
27	0.003500	22	1.0	38	1.0
31	0.0041.6	26	1.0	41	1.0
22	0.008275	29	1.0	59	1.0
26	1.0	30	1.0	75	1.0
29	1.0	31	1.0	81	1.0
35	1.0	35	1.0	83	1.0

TABLE XIV
PERFORMANCE OF ORDINARY MLP ON DIFFERENT SUBSETS OF FEATURES AT LEVEL 1 OF HLA TO SHOW THE IMPORTANCE OF EACH PSI

MLP	C	C+S	C+S+H	C+S+H+P	C+S+H+P+V	C+S+H+P+V+Z
Correct Classified Number	243	308	305	301	302	309
Accuracy (%)	63.1	80.0	79.2	78.2	78.4	80.3

TABLE XV
CLASSIFICATION ACCURACIES OF THE RBFN-BASED HLA WITH VARIOUS COMBINATIONS OF GLOBAL FEATURES (C + H + S + P + V + Z) AND LOCAL FEATURES [BIGRAM-CODED FEATURE (B) AND SPACED BIGRAM-CODED FEATURE (SB)] SELECTED BY THE GATING NETWORK

RBFN-Based HLA (Gated)						
Features		Global features (6 PSIs)			PSIs + B	PSIs +B + SB
No. of Features		50	67	80	67+242	67+242+205
Accuracy of Level 1 (%)		79.2	80.3	80.8	80.3	82.1
Accuracy of Level 2 (%)	Class 1	47.5	50.8	60.7	73.8	82.1
	Class 2	47.9	51.3	46.2	56.4	58.1
	Class 3	51.0	53.1	53.1	66.2	67.6
	Class 4	48.8	54.8	58.4	41.9	48.4
Overall Accuracy (%)		49.1	52.5	53.0	60.5	62.6

performance (with 27 folds). For this case we get 53% test accuracy with 67% features, which is just 3% less than what we can achieve taking into account all 125 features.

We have made several runs of the gating networks and results reported corresponding to some typical output. We emphasize the fact that depending on the initialization, two different sets of features may be picked up by the gating network in two different runs. This is absolutely fine, since if there are two correlated features, a and b, the net may pick up feature a in run 1 and feature b in run 2. Moreover, depending on the choice of the threshold, the number of selected features may be different. Table XIII shows 15 of the most important features of a typical run of the gating network after 1000, 1500, and 4000 iterations. It is interesting to note that after 1000 iterations, eight of the top most 15 important features come from the predicted secondary structure. Of these eight, one of the features, number 27, disappears from the list of important features with iterations. Probably the gate corresponding to some other correlated feature opened faster. After 4000 iterations, of the important 15 features, nine come from the predicted secondary structure. This clearly tells that the local secondary structure, as expected, has a strong impact on the final folds. In this list of 15 important features, we have representation from polarity, polarizability, volume, and hydrophobicity. In this investigation, we initialized the gating function with a value of 0.000 124.

Table XIV depicts the classification performance at level 1 (into four classes) by the MLP network with different sets of features. Table XIII reveals the fact that use of more features is not necessarily good. It also says that the distribution of predicted secondary structure and composition constitutes a good set of features. This is also consistent with the results obtained from the gating network.

We shall now apply the gating network scheme to the enlarged input vectors combining the global and local features. The resulting classification accuracies of the RBFN-based HLA are presented in Table XV. With the same preset threshold value, the gating network reduces the dimension of the six types of PSI to 67 from 125, the dimension of the bigram-coded feature to 242 from 441, and the dimension of the spaced bigram-coded feature to 205 from 441. The dimension reduction reduces the classification accuracy of the RBFN-based HLA by 2.9% using only about half of the original features.

In the above experiments, we used the same threshold value in the gating networks for all the basic classifier units of HLA, which will produce different input vector sizes for different classifiers at different levels of HLA. In another experiment, we try to use the same size of feature vector for each classifier of the HLA though the gating networks. The results and comparisons with different combinations of protein features are given in Table XVI. It is observed that the classification accuracy is further improved. Table XVII shows the required number of nodes in each RBFN classifier of the RBFN-based HLA with respect to different gated features. The total number of the required nodes is found to be quite small. This demonstrates the efficiency the proposed HLA with automatic feature selection mechanism (gating network).

TABLE XVI
PERFORMANCE COMPARISON OF THE RBFN-BASED AND SVM-BASED HLA WITH GATING NETWORK USING FIXED THRESHOLD OR FIXED SIZE OF FEATURE DIMENSION

Learning Architecture / Features	RBFN-Based HLA Overall Accuracy (%)		SVM-Based HLA Overall Accuracy (%)	
	Fixed Threshold	Fixed Dimension	Fixed Threshold	Fixed Dimension
Global features (PSIs)	56.4	55.6	47.3	51.7
PSIs + Bi-gram coded features	56.9	60.5	58.2	58.4
PSIs + Bi-gram coded features + Spaced Bi-gram coded features	61.3	62.6	61.0	62.6

TABLE XVII
REQUIRED NODE NUMBERS IN EACH RBFN OF THE RBFN-BASED HLA FOR DIFFERENT GATED GLOBAL FEATURES

HLA / Feature Size (PSIs)		50	67	80	125
Level 1	Classifier #1	136	124	112	145
Level 2	Classifier #2	22	49	42	38
	Classifier #3	68	56	71	101
	Classifier #4	77	92	88	69
	Classifier #5	12	6	2	13
Total Number		315	327	315	366

VIII. CONCLUSION

In this paper, we proposed a new HLA with online feature selection mechanism to solve the multiclass protein fold classification problem. We also derived new local features from the protein sequences to enhance the classification rate. The proposed HLA is a general learning concept, which can integrate a set of baseline classifiers (such as NN or SVM) in an efficient way to attack highly complex classification problems. Furthermore, the proposed modified bigram coding scheme for protein sequences are based on a concept of entropy, which can well describe the cubic structures of proteins in space. Such kinds of information were usually missing in the conventional global features of protein sequences.

The extensive experimental results based on the SCOP database demonstrated the superiority of the proposed protein fold classification scheme, in both learning mechanism and new protein features. The classification accuracy of the novel scheme is also higher than that of the popular OvO method, the modified uOvO method and AvA method. In addition, due to the use of the multiclass classifiers as the basic building blocks, the proposed HLA does not need a large number of two-class classifiers and a voting scheme. As a result, the computation time for a prediction can be reduced and each prediction can be associated with a numerical value to assess the reliability or confidence of the prediction.

The experimental results also showed that the online feature selection mechanism in HLA was quite effective in reducing the dimensionality of the input data features. Such online feature selection capability can give a better insight into the folding process. So far the bioinformatics researchers did not have any tools for such online feature selection and, consequently, they are used to consider different intuitive combination of features. Since consideration of all possible subset is computationally not feasible, it is often impossible to find the best set of features. The proposed system opens up the possibility of computing many more features from the amino-acid sequence and then allowing the system to pickup the desirable ones. Its application domain is extended to all other areas of bioinformatics also.

REFERENCES

[1] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, "CATH—A hierarchic classification of protein domain structure," *Structure*, vol. 5, no. 8, pp. 1093–1108, 1997.

[2] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequence and structures," *J. Mol. Biol.*, vol. 247, pp. 536–540, 1995.

[3] 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, pp. 401–407, 1999.

[4] L. L. Conte, B. Ailey, T. J. Hubbard, S. E. Brenner, A. G. Murzin, and C. Chothia, "SCOP: A structural classification of proteins database," *Nucleic Acid Res.*, vol. 28, no. 1, pp. 257–259, 2000.

[5] L. L. Conte, S. E. Brenner, T. J. Hubbard, C. Chothia, and A. G. Murzin, "SCOP database in 2002: Refinements accommodate structural genomics," *Nucleic Acids Res.*, vol. 30, no. 1, pp. 264–267, 2002.

[6] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*. Cambridge, MA: MIT Press, 1998.

[7] C. H. Wu, *Neural Networks and Genome Informatics*. Amsterdam, The Netherlands: Elsevier, 2000.

[8] J. Yang, R. Parehk, V. Honavar, and D. Dobbs, "Data driven theory refinement algorithms for bioinformatics," in *IJCNN '99 Int. Joint Conf.*, vol. 6, 1999, pp. 4064–4068.

[9] I. Dubchak and C. H. Q. Ding, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, no. 4, pp. 349–358, 2001.

[10] K. C. Chou and C. T. Zhang, "Prediction of protein structural classes," *Critical Rev. Biochem. Mol. Biol.*, vol. 30, no. 4, pp. 275–349, 1995.

[11] 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.*, vol. 92, pp. 8700–8704, 1995.

[12] U. Hobohm and C. Sander, "Enlarged representative set of protein structures," *Protein Sci.*, vol. 3, no. 3, pp. 522–524, 1994.

[13] N. R. Pal and K. Chintalapudi, "Connectionist system for feature selection," *Neural, Parallel Sci. Comput.*, vol. 5, no. 3, pp. 359–381, 1997.

[14] K. L. Priddy, S. K. Rogers, D. W. Ruck, G. L. Tarr, and M. Kabrisby, "Bayesian selection of important features for feed-forward neural network," *NeuroComputing*, vol. 5, pp. 91–103, 1993.

[15] A. Verikas and M. Bacauskiene, "Feature selection with neural networks," *Pattern Recogn. Lett.*, vol. 23, pp. 1323–1335, 2002.

[16] N. R. Pal, "Soft computing for feature analysis," *Fuzzy Sets Syst.*, vol. 103, pp. 210–221, 1999.

[17] N. R. Pal and V. K. Eluri, "Two efficient connectionist schemes for structure preserving dimensionality reduction," *IEEE Trans. Neural Networks*, vol. 9, pp. 1142–1154, Nov. 1998.

[18] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[19] S. Lee and R. M. Kil, "Multilayer feedforward potential function network," in *Proc. Int. Joint Conf. Neural Networks*, vol. 1, 1988, pp. 161–171.

[20] J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Comput.*, vol. 1, no. 2, pp. 281–294, 1989.

[21] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function network," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.

[22] J. A. Leonard, M. A. Kramer, and L. H. Ungar, "Using radial basis functions to approximate a function and its error bounds," *IEEE Trans. Neural Networks*, vol. 3, pp. 624–627, July 1992.
[23] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[24] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst.*, vol. 13, pp. 18–28, Jul./Aug. 1998.
[25] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Proc. 1997 IEEE Workshop Neural Networks for Signal Processing*, pp. 276–285.
[26] M. Niranjan, "Support vector machines: A tutorial overview and critical appraisal," in *IEE Colloq. Applied Statistical Pattern Recognition*, 1999, p. 2/1.
[27] C. J. Lin and C. W. Hsu, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, pp. 415–425, Mar. 2002.

Chuen-Der Huang received the B.S. degree in electrical engineering and the M.S. degree in automatic control engineering from Feng Chia University, Taichung, Taiwan, R.O.C., in 1980 and 1983, respectively.

He is currently with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., and the Department of Electrical Engineering, Hsiuping Institute of Technology, Taichung, Taiwan, R.O.C. His research interests include bioinformatics, machine learning, fuzzy control, and data mining.

Chin-Teng Lin (S'88–M'91–SM'99) received the B.S. degree in control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 1986 and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1989 and 1992, respectively.

Since 1992, he has been with the College of Electrical Engineering and Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., where he is currently the Associate Dean of the college and a professor of Electrical and Control Engineering Department. From 1998 to 2000, he served as the Director of the Research and Development Office of the National Chiao-Tung University and the Chairman of Electrical and Control Engineering Department from 2000 to 2003. His current research interests are neural networks, fuzzy systems, cellular neural networks (CNNs), fuzzy neural networks (FNNs), VLSI design for pattern recognition, intelligent control, and multimedia (including image/video and speech/audio) signal processing, and intelligent transportation system (ITS). He is the Coauthor of *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1996) and the Author of *Neural Fuzzy Control Systems with Structure and Parameter Learning* (Singapore: World Scientific, 1994). He is currently an Associate Editor of *International Journal of Speech Technology* and the *Journal of Automatica*. He has published over 70 journal papers in the areas of neural networks, fuzzy systems, multimedia hardware/software, and soft computing, including 52 IEEE journal papers.

Dr. Lin is a Member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi. Since 1998, he has been the Executive Council Member (Supervisor) of Chinese Automation Association. From 1994 to 2001, he was the Executive Council Member of the Chinese Fuzzy System Association of Taiwan (CFSAT). Since 2002, he has been the Society President of CFSAT. He has won the Outstanding Research Award granted by the National Science Council (NSC), Taiwan, from 1997 to present, the Outstanding Electrical Engineering Professor Award granted by the Chinese Institute of Electrical Engineering (CIEE) in 1997, the Outstanding Engineering Professor Award granted by the Chinese Institute of Engineering (CIE) in 2000, and the 2002 Taiwan Outstanding Information-Technology Expert Award. He was also elected to be one of the 38th Ten Outstanding Rising Stars in Taiwan, R.O.C., in 2000. He is also a Member of the IEEE Circuit and Systems Society (CASS), the IEEE Neural Network Society, the IEEE Computer Society, the IEEE Robotics and Automation Society, and the IEEE Systems, Man, and Cybernetics Society. He is a Member and the Secretary of the Neural Systems and Applications Technical Committee (NSATC) of IEEE CASS and will join the Cellular Neural Networks and Array Computing (CNNAC) Technical Committee soon. From 2000 to 2001, he was the Chairman of the IEEE Robotics and Automation Society, Taipei chapter. He is the Distinguished Lecturer representing the NSATC of IEEE CASS from 2003 to 2004. He is an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS and IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Nikhil Ranjan Pal (M'91–SM'00) received the Ph.D. degree in computer science from the Indian Statistical Institute, Calcutta, India, in 1991.

He is currently with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta. He is an Associate Editor of the *International Journal of Fuzzy Systems* and the *International Journal of Approximate Reasoning*. He coauthored the book *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing* (Norwell, MA: Kluwer, 1999). His research interests include image processing, fuzzy theory, neural networks, and genetic algorithms.