Semi-Supervised Boosting Fold Recognition Algorithm (SB-FR) for Protein Fold Recognition

Wessam H. El-Behaidy, Aliaa A. A. Youssif, and Atef Z. Ghalwash
Helwan University/Faculty of Computers and Information, Cairo, Egypt
Email: w_behaidy@yahoo.com,{aliaay, atef.ghalwash}@hq.helwan.edu.eg

***Abstract*—Protein structure prediction is a very challenging problem in drug discovery and computational biology. To achieve better multi-class classification model for fold recognition problem, a combination between semi-supervised and boosting techniques is proposed into Semi-supervised Boosting Fold Recognition (SB-FR) algorithm. In addition, a testing method "TreeTest" is introduced for improving the overall accuracy of SB-FR algorithm. To benchmark the performance of the proposed SB-FR algorithm, a famous challengeable "Ding and Dubchak" dataset is used for training and testing. In addition, different parameters are applied to the same random sets of labeled and unlabeled sequences. To benchmark the "TreeTest" testing method, All-versus-All (AvA) testing method is used for comparison. Finally, using the proposed SB-FR algorithm along with the proposed "TreeTest" method for fold recognition multi-class classification, a 5.6% improvement is recorded in the overall accuracy for three-class and 8.2% for five-class classifications when compared to the base classifier.**

***Index Terms*—protein structure prediction, multi-class classification, fold recognition, semi-supervised, boosting, Ding and Dubchak dataset, All-vs-All**

I. Introduction

Protein structure methods have appeared to reduce the gap between gene sequence generation and protein structure determination, i.e. sequence-structure gap. These methods can predict protein structure in a relatively fast and cheap way. Fold recognition (FR) approach is one of the most promising methods for protein structure prediction [1]. FR is a template- based method, which finds a suitable template for model building in the absence of a significantly similar sequence with known structure.

The machine learning (ML) techniques are the most used in fold recognition problem; as the Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and the Neural Networks (NN). However, boosting algorithms have the potential to build efficient classification models in a very fast manner [2].

Fold recognition problem has been treated from different perspectives. Classification-based methods recognize those templates by selecting a similar fold as the target sequence [3], while others rank the template by the better alignment the template has to the target sequence [4]. An estimation function has also been used to estimate the alignment accuracy for each sequence-template alignment, and the one that has the best alignment is chosen [5].

In this paper, the fold recognition problem is treated as a semi-supervised learning (SSL) problem. If the already determined proteins' structure in PDB is treated as known data and the huge availability of proteins' sequence with unknown structure as unknown data, then we are facing a SSL problem.

SSL paradigm is based on using large amount of unlabeled data (unknown class) with a small amount of labeled data (known class). Since the unlabeled data contains relevant information, SSL gives higher accuracy in classification [6].

Here, a boosting framework in addition to a number of unlabeled examples is applied on fold recognition problem to improve the classification accuracy.

II. The Proposed Algorithm

The proposed algorithm Semi-supervised Boosting fold recognition (SB-FR) is described into the pseudo code shown in Fig. 1. Each function mentioned in the pseudo-code is explained later on.

TrSeq ←***DataSet_Extension*** *(**D-B, Ext-DB**)*
(TrData, TstData) ←***Feature_extraction*** *(TrSeq, TstSeq)*
SmMtrx ← ***Calculate_Similarity*** *(TrSeq)*
For Loop=1 to 20 // For generalization & stabilization
(L-Set, U-set) ← ***Separate_Sets*** *(TrData)*
NoClassifiers ← ***Calculate_NoClassifiers***
(NoClasses)
For i=1 to NoClassifiers-1
For j=i+1 to NoClassifiers
NwLSet ← ***Cut2Classes*** *(L-Set)*
(Ensbl_classifiers, weights) ←***BinarySB***
(NwLSet, U-Set, SmMtrx, NoSampled)
End
End
Overall_Percentage ←***Testing*** *(TstData,*
Ensbl_classifiers, weights)
End

Figure 1. Pseudo-code of SB-FR proposed algorithm.

During training phase, several steps are applied. In DataSet_Extension function, each class in Ding and

Manuscript received July 31, 2013; revised September 10, 2013.

doi: 10.12720/ijoee.2.1.1-6

Dubchak (D-B) dataset is appended by extra sequences from extended data set (Ext-DB) for both training sequences (TrSeq) and testing sequences (TstSeq) (see Section III.A). The amino acid composition feature vectors are extracted from training and testing sequences as in [7] using Feature_extraction function. The resulted features are referred as training data (TrData) resulted features are referred as training data (TrData) and testing data ***(TstData).*** After that, the ***TrData*** is passed to ***Calculate_Similarity*** function to calculate the similarity matrix ***(SmMtrx)*** between any two sequences (either labeled or unlabeled) using one minus P-distance (1-P). *P-distance* is used to calculate the pairwise distance between sequences using (1). In (1), ***seq1≠seq2*** means that the two sequences are different at this site and ***h*** contains the locations that need to be scored in the pairwise alignment, by excluding sites with double gaps. *P* is close to 1 for poorly related sequences, and *P* is close to 0 for similar sequences.

$$P = \frac{sum\left(\left(seq1 \neq seq2\right)\&h\right)}{sum\left(h\right)} \quad (1)$$

The ***SeparateSets*** function is the start point from where a repetition for 20 times is performed to check the generalization and stability of SB-FR algorithm (see, Section III.C). This function separates the training data ***(TrData)*** into Labeled set ***(L-Set)*** and Unlabeled set ***(U-Set)*** randomly.

Since protein fold recognition is typically a multi-class problem, the SB-FR uses the All-versus-All (AvA) method for training [3]. So, the number of classifiers will equal to K(K-1)/2 for K classes ***(NoClasses)*** calculated by ***Calculate_NoClassifiers*** function. Also, the ***Cut2Classes*** function cuts the required two specified classes from labeled set ***(L-Set)*** and changes their labels into 1 and -1.

The ***BinarySB*** function is based on SB algorithm [8]. During this function, the labels of unlabeled set (U-Set) are predicted as pseudo-labels. Each pseudo-label (and its confidence) for each unlabeled example is calculated by using existing ensemble, and the calculated pairwise similarity matrix ***(SmMtrx)***. The confidence of pseudo-labels is sorted in descending order and the highest confident pseudo-labeled examples are sampled and combined with the labeled samples. The number of sampled examples is determined by a predefined value, i.e. ***NoSampled***. Therefore, a component classifier $h_t(x)$ is trained using the base classifier with this new training set. Given the component classifier $h_t(x)$ with an appropriate weight α_t [8], the ensemble classifier (***Ensbl_classifiers***) H(x) is updated by combining it linearly with the previous ones, as in (2), to make improved predictions. This whole process from predicting pseudo-labels and train a component classifier is repeated for a predefined number of iterations (here, it is 20), or when the weight classifier becomes less than zero.

$$H\left(x\right) = \sum_{t-1}^{T} a_t h_t\left(x\right) \quad (2)$$

In testing phase, the ensemble classifiers ***(Ensbl_classifiers)*** H(x) with their corresponding ***weights*** are applied to test data ***(TstData)*** to get the overall accuracy in percentage ***(Overall_percentage)*** as in [3]. During testing phase, the All-versus-All (AvA) method [3] is used, where the class of query sequence is determined by getting the highest vote for this class. A proposed "Tree-Test" method (Section IV), is also used in determining the class for query sequence.

III. Methods

A. Datasets

Ding and Dubchak (D –B) Dataset [3] is a challenging one. The *training set* consists from 311 proteins sequences that have less than 35% identity between each pair of two proteins in any aligned subsequence longer than 80 residues. The independent *testing set* consists from 385 proteins sequences that have less than 40% identity between each other and not more than 35% identity to any protein in the training set. The proteins used for training and testing belong to 27 different folds representing all major structural classes; all α, all β, α/β, α + β and small proteins.

Extended SCOP dataset [7], will be referred as ***(Ext-DB)***, was formed by populating each fold in D-B dataset with additional protein examples chosen from ASTRAL SCOP 1.71, where sequences have less than 40% identity to each other.

B. Fold Classifier

In this work, the Support Vector Machine (SVM) was used as the base classifier, which is extensively used for classification and regression problems. WEKA software [9] was used to implement the SVM with the sequential minimal optimization (SMO) algorithm using its default parameters.

C. Performance Measures

The performance of proposed algorithm was evaluated by computing overall accuracy (Q) [3] which is the most commonly used in multi-way classification methods. To check the generalization and stability of the method, randomly chosen sets for labeled and unlabeled are selected for twenty times and the average of their overall accuracy was used to construct the final curve.

IV. Proposed "Tree-Test" Method

In this proposed method, the number of trained two-way classifiers is K(K-1)/2 for K classes as in [3]. This method has the shape of a binary tree. At each node, a smaller part of the test set is tested against a new classifier which is different from AvA method [3]. At each level in this binary tree, each classifier divides the coming part of test dataset into smaller sets. This idea will be more clarified in Fig. 2. Assuming that there are 3 classes, so there are 3 classifiers. The classifier 1-2 is applied to the whole testing dataset. This classifier divides this dataset into two smaller sets; set 1 at level 1 (S1-L1), and set 2 at level 1(S2-L1). By assuming that

classifier 1-2 has the capability to differentiate between class 1 and 2, so S1-L1 will not include sequences that belong to class 2 and in turn, S2-L1 will not include sequences that belong to class1. But each set can includes other sequences belong to other classes (class3), and that what will be determined by the next levels. By applying classifier 1-3 to S1-L1, it excludes the sequences of class 3 that are included into S1-L1and divides the sequences into another 2 sets; S1-L2 and S3-L2. For the other branch, classifier 2-3 is applied to S2-L1 and divides it into S2-L2 and S3-L2. Notice that, there are two sets that have the same name S3-L2; these two sets are merged to collect all the sequences belong to class 3.If any processing will be done on these merged sets, the merging must be done at first. The final results will be the sets; Set1, Set2 and the combined Set3.

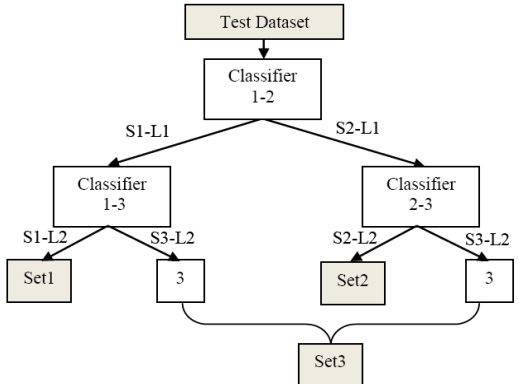

Figure 2. How to specify the class for the test dataset using proposed "Tree-test" method.

The proposed "TreeTest" testing method is faster than AvA method. Suppose that, for ***k*** classes there will be number of classifiers as in (3). In addition, each classifier will needs ***t*** units to test ***n*** sequences. So, AvA will takes ***t*** units equals to the number of classifiers, while **"**TreeTest**"** will takes ***t*** units less than the number of classes by one**.** Thus, the "TreeTest" is faster than ***AvA*** by a factor of ***k***/2.

$$No.\ of\ classifiers\ of\ k\ classes = \frac{k(k-1)}{2} \qquad (3)$$

V. Results and Discussion

This section answers the questions: Does SB-FR algorithm improve the performance of base classifier? Has SB-FR a stability performance related to the number of unlabeled sequences? This section started by the differentiating parameters applied to SB-FR algorithm, then the results are mentioned and an analysis is finally performed.

A. Differentiate Parameters

To benchmark SB-FR algorithm, different parameters are applied to the same random sets from labeled and unlabeled resulted from ***SeparateSets*** function in Section II.

In training phase, the parameters are varied in:

- **Number of classes:** SB-FR algorithm is tested for multi-class classification using three-class and five-class.
- **Number of labeled sequences for each class:** Different group numbers from Labeled set (***L-Set***) are used to specify how much labeled sequences will be dedicated for each class. These group numbers are 5, 10, 20, 50, and 100, which will be referred respectively as B5, B10, B20, B50 and B100.
- **Number of unlabeled sequences for each class:** Different group numbers from Unlabeled set (***U-Set***) are used **to** specify how much unlabeled sequences will be sampled from the U-Set during training. These groups range from 10 to 200 with increased step of 10.

In training phase, the substitution matrix ***BLOSUM62*** is used during the aligning of pairwise sequences, which will be helpful in similarity matrix calculation. Here, the gap opening and gap extension are 10 and 1 respectively.

In testing phase, both ***All-Versus-All (AVA)*** and ***"TreeTest"*** testing methods are applied on trained classifiers with their weights to determine the class of query sequence.

B. Curves Fitting

For three-class and five-class, the results have exponential shape for that "One-Phase association" curve fitting is calculated using GraphPad Prism 5.04 software. This curve starts at Y0 then goes up to Plateau with one phase. This plateau value demonstrates the expected highest overall accuracy. Also, the rate constant is calculated which expressed in reciprocal of the X axis units.

C. Three-Class Results

The three classes are chosen upon the highest number of sequences available in the Ext-DB dataset. These chosen classes are class 3, 7 and 16 with respectively 328, 415 and 351 sequences. The total number of sequences is 1094 for training and 112 for testing.

Table I shows the three-class classification results. It shows the expected highest overall accuracy and rate constant for the calculated fitting curves and their corresponding overall accuracy of the base classifier referred as "reference line values" for different groups from Labeled set (***L-Set***).

By using AvA method in testing phase, the expected highest overall accuracy column in Table I demonstrates that as the number of labeled sequences increases, the overall accuracy increases. Furthermore, the overall accuracy almost stabilizes after using 10 labeled/class (B10); as the improvement in overall accuracy from B10 to B100 is only 0.31%. By comparing the expected highest overall accuracy with their corresponding reference values, it is shown that the reference line starts to reach overall accuracy over 87% after being trained by 50 labeled/ class (B50), while this overall accuracy is reached by only using 5 labeled/class (B5) when using AvA testing method. From rate constant column, it is shown that its range varies between 0.0296 and 0.0577.

By using TreeTest method in testing phase, the expected highest overall accuracy column in Table I demonstrates that as the number of labeled sequences increases, the overall accuracy increases. Additionally, the overall accuracy almost stabilizes after using 10 labeled/class (B10); as the improvement in overall accuracy from B10 to B100 is only 0.39%. By comparing the expected highest overall accuracy with their corresponding reference values, it is shown that the reference line starts to reach overall accuracy over 89% after being trained by 100 labeled/ class (B100), while this overall accuracy is reached by only using 10 labeled/class (B10) when using TreeTest testing method. From rate constant column, it is shown that its range varies between 0.0531 and 0.0788.

Fig. 3 shows a comparison between AvA and TreeTest methods. It demonstrates that the TreeTest overall accuracy is always higher than AvA method; as the improvement in TreeTest overall accuracy ranges from 1.58% to 1.85% with an average of 1.78%. Also, the expected highest overall accuracy for both testing methods are usually higher than the reference line, but when a much higher number of labeled sequences are being used in training, the overall accuracy of AvA method can't reach the reference line, while on the other hand the TreeTest method even crosses it, as in Fig. 3(e).

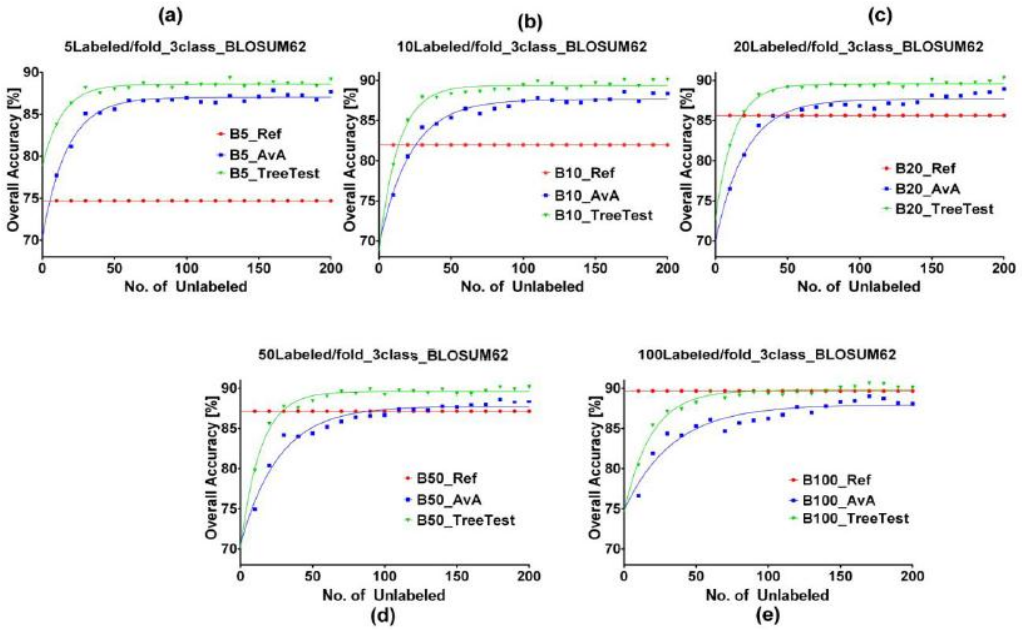

Figure 3. Three-class results comparison between the 5 groups of labeled set (B5—> B100) using both AvA and "TreeTest" methods during testing phase.

Also, Table I shows an improvement in rate constants of TreeTest method over AvA method. This improvement ranges from 25.6% to 89.74%, with an average of 64.35%. As a consequence of this improvement, TreeTest's overall accuracy converges to the expected highest overall accuracy with less added unlabeled sequences than AvA method. This rapid converge can also be concluded from Fig. 3.

TABLE I. THREE-CLASS CLASSIFICATION RESULTS OF EXPECTED HIGHEST OVERALL ACCURACY AND RATE CONSTANTS FOR THE FITTING CURVES USING ALL-VERSUS-ALL (AVA) AND "TREETEST" IN TESTING PHASE.

Labeled-Set Groups	Reference line values	Three-class classification			
		Expected highest overall accuracy		Rate constant	
		AvA	TreeTest	AvA	TreeTest
B5	74.69	87.02	88.6	0.058	0.073
B10	81.96	87.63	89.38	0.046	0.075
B20	85.63	87.65	89.55	0.048	0.079
B50	87.13	87.75	89.6	0.038	0.072
B100	89.66	87.94	89.77	0.03	0.053

D. Five-Class Results

The five classes are the three previously chosen classes in addition to two others; class 20 and 26 with 237 and 331sequences available in them respectively. The total number of sequences is 1737 for training and 151 for testing.

Table II shows the five-class classification results. It shows the expected highest overall accuracy and rate constant for the calculated fitting curves and their corresponding accuracy of the base classifier referred as "reference line values" for different groups from Labeled set (***L-Set***).

By using AvA method in testing phase, the expected highest overall accuracy column in Table II demonstrates that as the number of labeled sequences increases, the overall accuracy usually increases, but when a much higher number of labeled sequences are being used in training, the overall accuracy starts to stabilize. The overall accuracy almost stabilizes after using 10 labeled/class (B10). By comparing the expected highest overall accuracy with their corresponding reference values, it is shown that the overall accuracy is always below the reference line. From the rate constant column, it is shown that it varies from 0.1076 to 0.2599.

By using TreeTest method in testing phase, the expected highest overall accuracy column in Table II demonstrates that as the number of labeled sequences increases, the overall accuracy increases. As well, the overall accuracy almost stabilizes after using 10 labeled/class (B10); as the improvement in overall accuracy from B10 to B50 is only 0.96%. By comparing the expected highest overall accuracy with their corresponding reference values, it is shown that the reference line starts to reach overall accuracy over 65% after being trained by 50 labeled/ class (B50), while this overall accuracy is reached by only using 10 labeled/class (B10) when using TreeTest testing method. From the rate constant column, it is shown that its range varies from 0.1010 to 0.1606.

Fig. 4 shows a comparison between AvA and TreeTest methods. This comparison demonstrates that the TreeTest overall accuracy is always higher than AvA method; as the improvement in TreeTest overall accuracy ranges from 29.88% to 32.42% with an average of 31.29%.

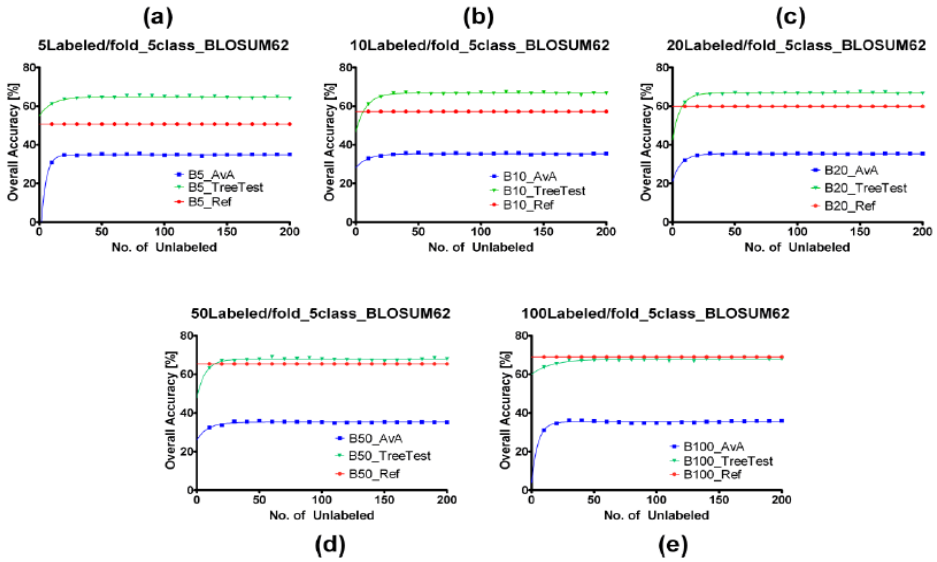

Figure 4. Five-class results comparison between the 5 groups of labeled set (B5—> B100) using both AvA and "TreeTest" methods during testing phase.

TABLE II. FIVE-CLASS CLASSIFICATION RESULTS OF EXPECTED HIGHEST OVERALL ACCURACY AND RATE CONSTANTS FOR THE FITTING CURVES USING ALL-VERSUS-ALL (AVA) AND "TREETEST" IN TESTING PHASE.

Labeled-Set Groups	Reference line values	Five-class classification			
		Expected highest overall accuracy		Rate constant	
		AvA	TreeTest	AvA	TreeTest
B5	50.73	34.89	64.77	0.26	0.101
B10	57.24	35.35	66.77	0.108	0.126
B20	59.91	35.38	66.80	0.144	0.161
B50	65.34	35.31	67.73	0.114	0.15
B100	68.99	35.49	67.77	0.198	0.064

Also, Table II shows an improvement in rate constants of TreeTest method over AvA method. This improvement ranges from 11.68% to 32.04%, with an average of 20.24%.

VI. Conclusion

During the SB-FR algorithm training and testing, different parameters were applied to the same random sets of labeled and unlabeled sequences to benchmark the performance of the proposed algorithm.

It was found that adding more unlabeled sequences to the training set results in a higher overall accuracy until it reaches a stabilized accuracy. While adding more labeled sequences per class, the overall accuracy would stabilize faster (less number of unlabeled sequences).

Also, the "TreeTest" results always have a higher overall accuracy and it stabilizes faster than AvA. In addition, the "TreeTest" method always improve the base classifier (reference line), on the other hand AvA method sometimes fails to reach the base classifier accuracy.

Finally, using SB-FR algorithm for multi-class classification with the proposed the "TreeTest" method for testing has achieved an improvement in overall accuracy with 5.6% for three-class and with 8.2% for five-class classification compared to the base classifier.

References

[1] P. R. Daga, R. Y. Patel, and R. J. Doerksen, "Template-based protein modeling: Recent methodological advances," *Curr Top Med Chem*, vol. 10, pp. 84-94, 2010.

[2] Y. Krishnaraj and R. C. K., "Boosting methods for protein fold recognition: An empirical comparison," in *Proc. 2008 IEEE Inter.l Conf. on Bioinform. and Biomed*., 2008, pp. 393-396.

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

[4] J. Xu, "Fold recognition by predicted alignment accuracy," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 2, no. 2, pp. 157-165, April 2005.

[5] F. Jiao, J. Xu, L. Yu, and D. Schuurmans, "Protein fold recognition using the gradient boost algorithm," in *Proc. Comput. Syst. Bioinformatics Conf.*, 2006, pp. 43-53.

[6] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, 2006.

[7] M. T. A. Shamim, M. Anwaruddin, and H. A. Nagarajaram, "Support vector machine-based classification of protein folds using the structural properties of amino acid residues and amino acid residue pairs," *Bioinformatics*, vol. 23, no. 24, pp. 3320-3327, 2007.

[8] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "SemiBoost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Analysis and Mach. Intell.*, vol. 31, no. 11, pp. 2000-2014, Nov 2009.

[9] L. J. McGuffin, "The ModFOLD server for the quality assessment of protein structural models," *Bioinformatics*, vol. 24, no. 4, pp. 586-587, Feb. 2008.

Wessam H. El-Behaidy. She received her B.Sc. is in Computer Science, from faculty of Computers and Information, Helwan University, Cairo, Egypt in 2000. Also, she earned her M.Sc. and Ph.D. in Computer Science, Helwan University, Cairo, Egypt in 2004 and 2012 respectively. Her major field is pattern recognition. She is currently a teacher assistant in faculty of Computers and Information, Helwan University, Cairo, Egypt. Her research interests also include structural bioinformatics, protein structure prediction, machine learning, and image processing. Dr. Wessam just published 5 papers in international conferences and Springer.

Atef Z. Ghalwash, He received his MSc from Faculty of Engineering, Cairo University, 1980. He was awarded his PhD from Faculty of Engineering, Maryland University, USA, 1988. He is currently a PROFESSOR in the Faculty of Computers and Information, Helwan University, Cairo, Egypt. Fields of research interest: Artificial Intelligence, Expert Systems, Computer Security, DSS and Systems Developments. Prof. Atef published more than 150 papers in the related fields.

