Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Protein Fold Classification Using Sequence Features

T.Divya Sravani*, K.Suvarna Vani
Department of Computer Science and Engineering
V.R.Siddhartha Engineering College
Vijayawada, Andhra Pradesh, India

Abstract—Protein fold classification is one of the challenging problems in bioinformatics. It classifies the protein fold in the given protein sequences without depending on sequence similarity. Many traditional machine learning techniques like Support Vector Machines (SVM), K-Nearest Neighbor (KNN) and Neural Networks (NN) are used for this problem. In this paper, we use different meta classifiers for the fold classification problem. The main objective of this paper is to analyze the different meta classifiers on this problem here we found that in-adequate classes are present in the dataset. Support Vector Machine (SVM) which can be effectively extended from a binary to a multi-class classifier does not perform well on this problem. In order to tackle this problem we use different boosting algorithms, rotation forest and random forest in which it has better accuracy than the other classifiers in the existing system.

Keywords— Protein fold classification, Support Vector Machine, MultiboostAB, AdaBoost.M1, LogitBoost, Rotation forest, Random forest.

I.Introduction

Proteins are one of the most important biological macromolecules and have an important role in life sustaining processes. Proteins are polypeptide chains consisting of a large number of amino acid residues which are covalently linked together via amide bonds or peptide bonds. These amino acids are made of organic compounds which have two functional groups they are amino group and carboxyl group. These proteins are classified into different structures. They are primary, secondary, tertiary, quaternary structures. Primary structure or primary sequence is the linear arrangement of naturally occurring 20 amino acids. Those 20 naturally occurring amino acids are Isoleucine, Valine, Leucine, Phenylalanine, Cysteine, Methionine, Alanine, Glycine, Threonine, Serine, Tryptophan, Tyrosine, Proline, Histidine, Glutamine, Asparagine, Glutamic, Aspartic, Lysine, Arginine. The secondary structure of a protein is formed by folding of amino acids chain (or residues) of a protein into local secondary structures like alpha helices, beta strands, and non-regular coils. The tertiary structure is the local conformation of three-dimensional arrangement of the amino acids and they represent by the x, y and z coordinates of all the atoms of a protein or by the coordinates of the backbone atoms. The secondary structures structures are folded into tertiary structure depending on hydrophobic forces and side chain interactions, such as hydrogen bonding between amino acids. The quaternary structure is described by the coordinates of all the atoms, or all the backbone atoms which are associated with all the chains participating in the quaternary organization. This protein quaternary structure represents the protein complexes. Functioning of a protein in biological reactions not only depends on its amino acid sequence (primary structure) but also crucially relies on its three-dimensional configuration (tertiary structure). The classification of the tertiary structure of a protein from its amino acid sequence still remains as an unsolved issue in bioinformatics and molecular biology. Protein fold recognition and classification are one of the major research problems in the computational biology. This protein fold classification does not depend on the sequence similarity. Proteins are said to have a common fold if they have the same major secondary structure in the same arrangement with the same topology, whether or not they have a common evolutionary origin. The proteins are said to be structurally similar if they have relatively same physical and chemical properties resulting certain arrangements and topologies.

The goal of this paper is to analyze the different meta classifiers on the protein fold data in order to know which classification algorithms perform better. We conduct different experiments on the data using different classification algorithms for different parameters and analyzed that these algorithms work better than the others. For each individual classifier we analyzed different parameters and analyzed that some parameters work better than the others based on the time complexity and accuracy wise.

II.Related Literature

Ding et. al [1] developed six feature sets which are extracted independently from protein sequences and these feature sets are classified using SVMs and neural networks based on three multi-classification methods (OvO, uOvO, AvA). They concluded that SVMs performance is better than neural networks. Okun [2] uses K-Local Hyperplane Distance Nearest

Neighbor Algorithm (HKNN) for the protein fold problem he uses the concept of nearest neigbours in this method. This algorithm forms a linear local hyperplane for each class in the data set. Here the distance between the proteins and their local hyperplanes are calculated to decide the protein class. Zhao et. al [3] deal with the protein sequence classification. They consider the pair wise similarity between the sequences in which they apply the ensemble classifiers for classifying the data. Chinnasamy et. al [4] developed BAYESPROT which is a tree augmented navies bayesian classifier based on the theory of learning Bayesian networks. In which it consists of a class node connecting to all child nodes each representing a feature and for all features we have class nodes and combine them using the voting scheme to decide the proteins by class and fold. Marsolo et. al [5] has done classification on the protein by considering the fold, class, and multi level strategy in which they used two classification algorithms, Naives bayes classifier and Boosted C4.5 algorithm and found that classification by class and multi level strategy has more accuracy than the fold. Chung et. al [6] proposed a hierarchical learning architecture (HLA) with automatic feature selection. They considered two levels in the first level they classify the data into the four major structures of protein. In the second level, they use a group of networks which again classify the structures into the 27 folds. Bologna et.al [7] proposed first ensemble method for the protein fold prediction problem. They used a 131-dimensional feature vector and an ensemble of four-layer discretized interpretable multi-layer perceptrons (DIMLP), where each network learns all protein folds simultaneously. Shamim et. al [8] developed a new method for protein fold recognition using structural information of amino acid residues and amino acid residue pairs. They developed a SVM based classifiers that combines secondary structural state and solvent accessibility state frequencies of amino acids and amino acid pairs as feature vectors. Wieslaw et. al [9] applies the multi-class support vector machine classifier for protein fold recognition problem. Zimek et. al [10] used Ensembles of Nested Dichotomies (ENDs) method to handle the protein dataset in which it takes a random sample from all different trees for a given n-class problem and forms class probability estimates for a given instance x by averaging the estimates obtained from the individual ensemble members to handle the multiclass in which it turns them to the binary classes.

III.Feature Extraction

Feature extraction is an important step in the classification of protein sequences. Here in order to apply the machine learning algorithms, we have to convert these amino acid sequences of different length to the numerical feature vectors with equal length. These feature vectors are being generated based on physical and chemical properties of amino acids. The properties are amino acids composition (C), predicted secondary structure based on Normalized Frequency of alpha-helix (S), Hydrophobicity (H), normalized VanDer Waals volume (V), Polarity (P) and Polarizability (Z).

For these properties except for the amino acid composition (C) feature vectors have three descriptors they are composition, transition and distribution. These descriptors are known as the global sequence descriptors. Composition is used to describe the global composition of a given amino acid property in a protein. Transition is used to describe the frequencies with which the property changes along the entire length of the protein. Distribution is used to describe the distribution pattern of the property along the sequence. Except for the Amino acid composition, feature vectors for the above five properties are constructed in two steps.

A. Step1

For each protein sequence, every amino acid was replaced by the index 1, 2 and 3 depending on its grouping based on the Table 1. These twenty amino acids are divided into three groups i.e. group1, group2 and group3 according to their properties.

For example consider the protein 1PPBL protein sequence:

TFGSGEADCGLRPLFEKKSLEDKTERELLESYIDGR.

Based on the hydrophobicity division of amino acids, 232221213231233111231112111331223121 is the encoded sequence for the above protein.

TABLE I
DIVISION OF AMINO ACIDS INTO DIFFERENT GROUPS

Attribute	Group1	Group2	Group3
Secondary structure	Helix E,A,L,M,Q,R,K,H	Strand V,I,Y,C,W,F,T	Coil G,N,S,P,D
Hydrophobicity	Polar R,K,E,D,Q,N	Neutral G,A,S,T,P,H,Y	Hydrophobic C,V,L,I,M,F,W
Polarity	(4.9-6.2) L,I,F,W,C,M,V,Y	(8.0-9.2) P,A,T,G,S	(10.4-13.0) H,Q,R,K,N,E,D
Van derWaals volume	(0-0.108) G,A,S,D,T	(0.128-0.186) C,P,N,V,E,Q,I,L	(0.219-0.409) K,M,H,F,R,Y,W
Polarizability	(0-2.78) G,A,S,C,T,P,D	(2.95-4.0) N,V,E,Q,I,L	(4.43-8.08) M,H,K,F,R,Y,W

B. Step 2

For each converted sequences calculated in step1 three descriptors Composition(c), Transition (t) and Distribution (d) are calculated.

1) *Composition:* It is the percent composition of each group present in the given protein sequence. The formula is:

$c_i = n_i/N$

where c_i represents the percent composition of each groupi,

n_i represents total number of groupi residues in the sequences,

N represents the length of the sequence

Eg: For the above encoded sequence we calculate the composition descriptor for group1.

n_1=Total number of group1 residues = 15

N=Total length of sequence = 36

c_1=15/36=41.66%

Similarly we calculate composition for the other groups of the sequence for different attributes.

2) *Transition:* It is represented by the percent frequency with which groupi is followed by groupj or groupj followed by groupi where i, j takes the values 1, 2 and 3.The formula is:

$t_{ij}= (n_{ij}+n_{ji})/ (N-1)$

where n_{ij}, n_{ji} is the number of dipeptide encoded as ij, ji.

N represents the length of the sequence.

Eg: For the above encoded sequence we calculate the Transition for the group1

n_{12}=Total number of dipeptides encoded as '12' = 6

n_{21}=Total number of dipeptides encoded as '21' = 4

t_{12}= (6+4)/ (36-1) =28.57%

Similarly we calculate transition for the other groups of the sequence for different attributes

3) *Distribution:* It consists of the five values for each of the three groups: the fractions of the entire sequence, where the first residue of a given group is located, and where 25%, 50%, 75%, and 100% of each group contained in the sequence.

Eg: For the above encoded sequence we calculate the Distribution for the group1.

Here we require the positions of the group1 in order to calculate the distributions.

The total number of group1 residues is: 15

25% of the group1 residues=3

50% of the group1 residues=11

75% of the group1 residues=7

Position of first residues of group1 located in the above encoded sequence is 6

Position of 25% residues of group1 contained in the above encoded sequence is 12

Distribution of the first residues of group1:

(Position of first residues of group1 located in the above encoded sequence)/

(Length of the sequence) =6/36=16.6%

Distribution of the 25% residues of group1 contained:

(Position of 25% residues of group1 located in the above encoded sequence)/

(Length of the sequence) =12/36=33.3%

Similarly we calculate for 50%, 75%, 100% for the group1 and the other groups of the sequence.

Each feature vector excluding amino acid composition contains 21 features. They are 3 composition features, 3 transition features and 5* 3 distribution features. Feature vector is of length 126 which is constructed by concatenating 21 all 5 attribute vectors of length 105 (5*21=105), amino acid composition vector of length 20 and the sequence length of length 1.

IV.Machine Learning Algorithms

In this paper we use different meta classifiers and tree classifiers for the classification of the protein data. The different classifiers are Adaboost, Logitboost, Multiboost, Rotation forest and Random forest and we compared with the base learner J48 algorithm. Adaboost, Logitboost, Multiboost and Random forest are ensemble methods. These ensemble methods are effective because it combines many models to improve the overall prediction accuracy. Boosting is an ensemble method which is simple and effective method to improve the performance of any learning algorithm. The main idea is to concentrate on the misclassified instances rather than the correctly classified this is done by assigning more weights to the misclassification instances. AdaBoost, Adaptive Boosting [11] is a popular boosting algorithm. In this it assigns weights to each instance. Initially, for all instances we assign equal weight and then we apply the base learner for that it classifies the instances into correctly and misclassified instances. The correctly classified weights are reduced and the misclassified instances weights are increased. Then again we apply the base learner in which it focuses more on the misclassified. This is done until we reduce the number of misclassified instances.AdaBoost.M1 is the most straightforward generalization of boosting algorithm. It is adequate when the weak learner is strong enough to achieve high accuracy. It doesn't require prior knowledge about the weak learner. It minimizes the exponential loss function.

LogitBoost [12] is a boosting algorithm which directly implements logistic regression method on the forward stage wise additive modeling. Logistic regression or logit regression is used for predicting the outcome of the response variable based on the other predictor variables. Here the forward stage wise additive modeling is building a model on the logistic regression for the training data and the errors which are produced by the first model is being reduced on the second model this continues until we get the minimum number of errors for the ensemble model. It generally performs better than the Adaboost algorithm in multi class case.AdaBoost.M1 optimizes the exponential loss where as the LogitBoost minimizes the logistic loss. MultiBoostingAB [13] is another technique of the boosting methods. Here it can be considered as wagging committees which are formed by AdaBoost. Wagging is a variant of bagging in which each classifier is trained on the entire training set, but each instance is assigned a weight. Bagging uses re-sampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses re-weighting for each training instance. It reduces high bias and variance. It produces decision committees with lower error than AdaBoost and wagging. Random Forest [14] is an ensemble method in which it uses a set of decision trees. In this it builds a dataset using the sampling with replacement method for training a group of base classifiers. Then it randomly selects a subset of features instead whole set of features at each step of construction to train each classifier

Rotation Forest [15] is an ensemble method in which it builds a set of decision trees. For each tree, the bootstrap samples extracted from the original training set are adopted to construct a new training set. Then the feature set of the new training set is randomly split into some subsets, which are transformed with a linear transformation method individually. Consequently, a full feature set is reconstructed with all the transformed features for each tree in the ensemble. Since a small rotation of axes builds a complete different tree, the diversity of the ensemble system can be guaranteed by the transformation. Finally, the outputs of all trees are fused by the average rule. Here the diversity is promoted by using Principal Component Analysis on feature extraction for each base classifier.

V.Dataset

The Protein Data Bank [16] (PDB) is the primary repository for experimentally determined 3D protein structures. These structures were created using crystallography methods. PDB entries contain additional information such as references, structure details and other features. SCOP (Structural Classification of Proteins) protein database [17], in which proteins are classified in a hierarchical manner that reflects their structural and evolutionary relationship. The main levels of the hierarchy are Family (based on the proteins evolutionary relationships), Superfamily (based on some common structural characteristics) and Fold (based on secondary structure elements).

The training and testing datasets were taken from ding and dubchak dataset [1]. The original training dataset is based on the PDB set. This dataset contains 313 proteins with less than 30% sequential similarity of 27 most populated folds Table 2. The original test dataset is based on the SCOP database. This dataset contain 385 proteins that have less than 35% sequence similarity. Recently, two proteins (i.e. 2SCMC and 2GPS) in the training dataset and two proteins (2YHX1 and 2YHX2) in the testing dataset were removed from this dataset due to the lack of sequence information. By this we have 311 proteins for training and 383 proteins for testing dataset.

TABLE II
27 different folds present in the data set

Fold	Index	Training instances	Test instances
Globin-like	1	13	6
Cytochrome c	3	7	9
DNA binding 3helical	4	12	20
4helical up and down	7	7	8
4 helical cytokines	9	9	9
Alpha EF hand	11	7	9
Immunogliobin	20	30	44
Cupredoxins	23	9	12
Viralcoat	26	16	13
ConA	30	7	6
SH3-like barrel	31	8	8
OB-fold	32	13	19
Trefoil	33	8	4
Trypsin	35	9	4
Lipocalin	39	9	7
(TIM)-barrel	46	29	48

Fad	47	11	12
Flavodoxin	48	11	13
NADP-binding	51	13	27
P-loop	54	10	12
Thiroredoxin	57	9	8
Ribonuclease	59	10	14
Hydrolase	62	11	7
Periplasmic	69	11	4
Beta grasp	72	7	8
Ferredoxin	87	13	27
Smallinhibitors	110	12	27

According to this dataset proteins are classified into one of the following four structural classes: All Alpha, All Beta, alpha/beta, and alpha+beta. Structural class categorizes various proteins into groups which share similarities in the local folding patterns. The all-alpha and all-beta classes represent structures that consist of mainly alpha-helices and beta-strands, respectively. The alpha/beta classes contain both alpha-helices and beta-sheets alternating in protein structure and alpha+beta classes contain both alpha-helices and beta-sheets located in specific parts of the structure. Of these 27 most populated fold types, types 1-6 belong to the Alpha structural class, types 7-15 to the Beta class, types 16-24 to the Alpha/Beta class, and types 25-27 to the Alpha+Beta class.

VI. Tools And Parameter Tuning

Here we test the different feature datasets using the data mining toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.6.6. WEKA is an open source toolkit and it consists of collection of machine learning algorithms for solving data mining problems [18]. Here for each parameter set we perform 10 fold cross validation on the training set to build a model and for that model we evaluate the independent test set. In 10 fold cross validation 9 sets of data were trained and the remaining set is being compared with the last set. In our experiment we first train the data and then we perform crossvalidation on the independent test data. Next, we tune the parameters for different classifiers.

For AdaboostM1, default parameters of WEKA were changed to perform 100 iterations with re-sampling using J48 base classifier. For LogitBoost, 100 iterations were done with re-sampling and decision stump was used as the weak learner. For MultiboostAB, 100 iterations were done with re-sampling and J48 was used as the weak learner. For Rotation forest and Random forest we considered 80 base classifiers and for J48 default parameters were used. Here we consider the standard cross validation on independent test data.

VII. Evaluation Measures

The standard Q percentage accuracy [19] is used to measure the prediction accuracy of the algorithms. Suppose we have X number of classes with N number of proteins such that n_1 is number of proteins observed in class X_1, n_2 is number of proteins in class X_2 and so on. Let a_1 be the number of proteins that are correctly classified in class X_1, a_2 be the number of proteins that are correctly classified in class X_2 and so on.

The total number of proteins N can be expressed as $N = [n_1 + n_2, .., n_x]$. The total number of proteins that are correctly classified can be expressed as $A = [a_1 + a_2, .., a_x]$.

The classification accuracy of the ith fold is calculated by:

$Q_i = a_i/n_i$

where n_i denotes the number of proteins in the ith fold

a_i denotes the number of proteins which are correctly classified.

The overall accuracy Q is calculated by:

$Q=A/N$

where N is the total number of proteins,

A is the total number of correctly classified proteins,

Q is the overall classification accuracy.

Here we use tabulated the accuracy values only. The other evaluation measures are based on the confusion matrix. A confusion matrix contains information about the actual and predicted classifications done by a classification machine learning algorithm. Performance of algorithm is evaluated using the data available in the confusion matrix.

VIII. CLASSIFICATION RESULTS

We perform different experiments on classification by fold and classification by class by applying different meta and tree classifiers on the data set for different combination of features. By results shown in the Fig. 1 and Fig. 2, we can analyze that the classification by class has higher accuracy than the classification by fold.

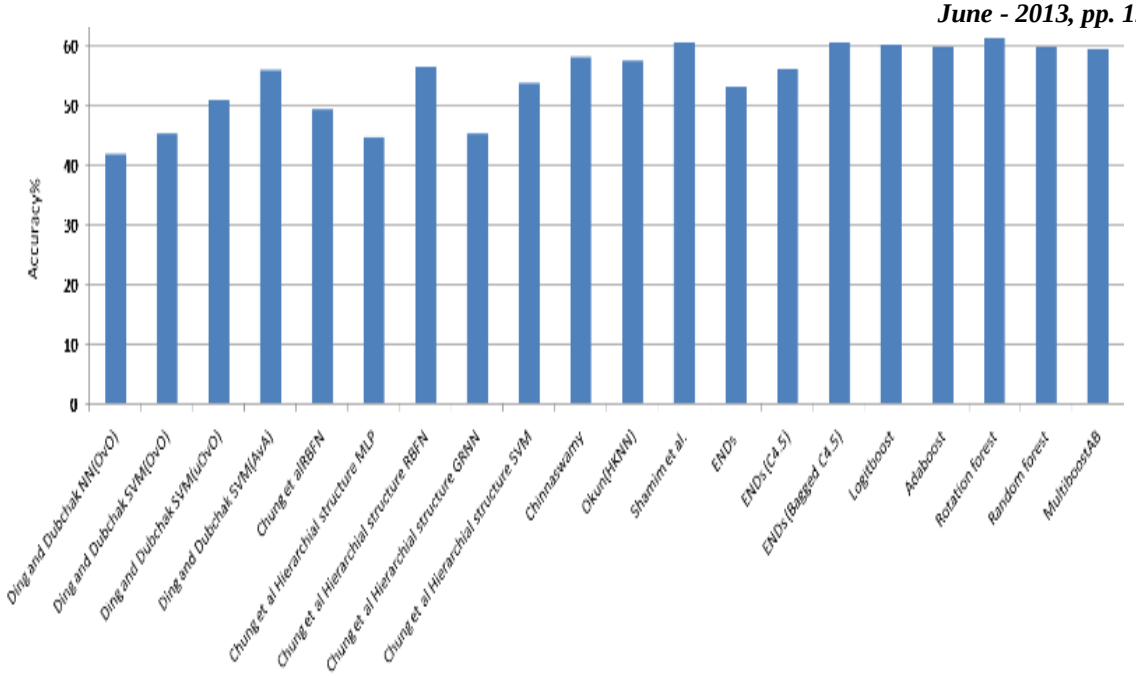

Fig. 1 Accuracy comparison with existing classification methods for 27 folds

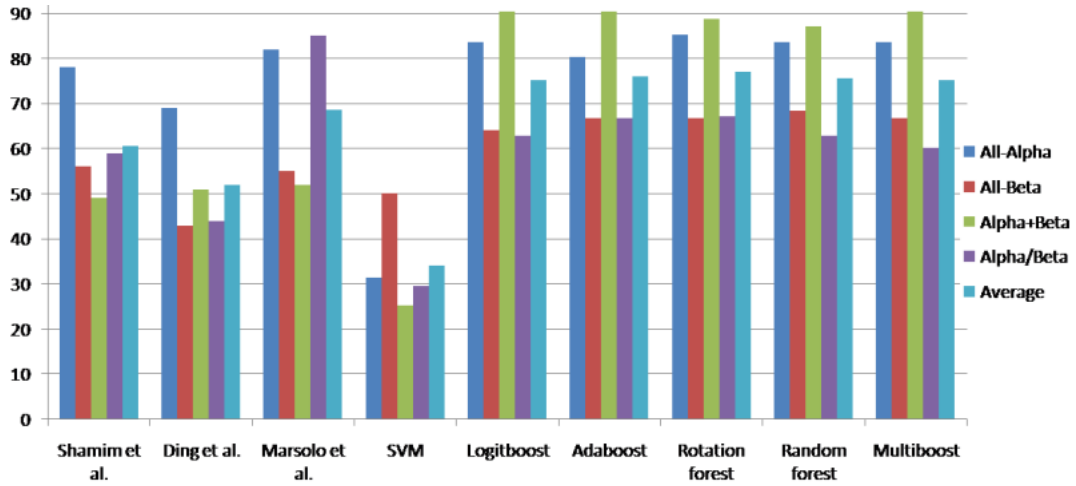

Fig. 2 Accuracy comparison with existing classification methods for structural classes

In classification by class the alpha+beta in Table 6 class has more accuracy than the other classes and the least class accuracy is for the alpha/beta class in Table 5. We achieved the highest accuracy 90.3% for different boosting algorithms in different combination of parameters for alpha+beta class. Here we provide the results for classification by class for different combination of features with considering length as another parameter. Here by adding the length as other parameter the classification accuracy is being increased. Here the results we obtained for the classification by class is more than the existing methods in Table 12.

TABLE III
CLASSIFICATION OF ALL ALPHA CLASS FOR DIFFERENT COMBINATIONS OF FEATURES USING DIFFERENT CLASSIFIERS

Classifier	C	CS	CSH	CSHP	CSHPV	CSHPVZ
Logitboost	73.8	73.8	83.6	78.7	78.7	80.3
AdaboostM1	80.3	78.7	75.4	80.3	78.7	75.4

Rotationforest	85.2	80.3	85.2	85.2	80.3	82
Randomforest	82	78.7	83.6	82	78.7	80.3
J48	73.8	65.6	63.9	59	59	59
MultiboostAB	80.3	82	80.3	83.6	82	82

TABLE IV
CLASSIFICATION OF ALL BETA CLASS FOR DIFFERENT COMBINATIONS OF FEATURES USING DIFFERENT CLASSIFIERS

Classifier	**C**	**CS**	**CSH**	**CSHP**	**CSHPV**	**CSHPVZ**
Logitboost	62.4	58.1	58.1	56.4	59	64.1
AdaboostM1	65.8	65.8	65	66.7	65	64.1
Rotationforest	66.7	59	64.1	62.4	65.3	64.1
Randomforest	68.4	63.2	65	63.2	67.5	64.1
J48	56.4	48.7	47.9	47.9	47	47.9
MultiboostAB	64.1	65.8	65	65.8	66.7	64.1

TABLE V
CLASSIFICATION OF ALL ALPHA/BETA CLASS FOR DIFFERENT COMBINATIONS OF FEATURES USING DIFFERENT CLASSIFIERS

Classifier	**C**	**CS**	**CSH**	**CSHP**	**CSHPV**	**CSHPVZ**
Logitboost	54.3	62.2	60.8	62.9	57.3	55.9
AdaboostM1	65.8	65.8	65	66.7	65	64.1
Rotationforest	62.2	64.3	67.1	64.3	62.9	60.8
Randomforest	59.4	61.5	62.9	62.2	62.9	59.4
J48	50.3	53.8	49.7	49.7	49.7	44.1
MultiboostAB	60.1	58	57.4	55.2	58.7	56.6

TABLE VI
CLASSIFICATION OF ALL ALPHA+BETA CLASS FOR DIFFERENT COMBINATIONS OF FEATURES USING DIFFERENT CLASSIFIERS

Classifier	**C**	**CS**	**CSH**	**CSHP**	**CSHPV**	**CSHPVZ**
Logitboost	82.3	90.3	88.7	87.1	87.1	85.5
AdaboostM1	87.1	87.1	87.1	90.3	82.3	88.7
Rotationforest	83.9	85.5	85.5	87.1	83.9	88.7

Randomforest	83.9	87.1	87.1	82.3	85.5	87.1
J48	85.5	83.9	83.9	83.9	83.9	83.9
MultiboostAB	83.9	85.5	90.3	88.7	85.5	85.5

In classification by fold the accuracies are low but the fold results are comparatively more than the SVM, neural networks and the nearest neighbours Table 11. Here we perform experiments considering the individual features Table 7 and Table 8 and combination of features Table 9 and Table 10 and we found that combination of features shows more accuracy than the previous one. For the classification by fold for different combination of features we achieve the accuracy for Logitboost is 60.1%, AdaboostM1 is 59.8%, Rotationforest is 61.4%, Randomforest is 59.8%, J48 is 42.6%, MultiboostAB is 59.5% in Table 10 and these results are more than without feature length Table 9. In our analysis we found that amino acid composition, secondary structure and hydrophobicity and length are effective features for the classification.

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT PROPERTIES OF 27 FOLDS USING DIFFERENT CLASSIFIERS WITHOUT LENGTH AS FEATURE

Property	**J48**	**Randomforest**	**Logitboost**	**MultiboostAB**	**Rotationforest**	**Adaboost**
C	33.7	50.4	46.2	50.7	57.4	51.9
S	36.3	48.6	45.4	47	48.6	47.5
H	26.6	41.3	36.3	41.5	46	43.9
P	26.9	41	36.6	40.7	45.4	40.7
V	28.2	40.5	34.5	41.8	43.9	43.3
Z	21.1	40.5	35	39.4	40.7	39.9

TABLE VIII
CLASSIFICATION RESULTS OF DIFFERENT PROPERTIES OF 27 FOLDS USING DIFFERENT CLASSIFIERS WITH LENGTH AS FEATURE

Property	**J48**	**Randomforest**	**Logitboost**	**MultiboostAB**	**Rotationforest**	**Adaboost**
C	39.7	57.2	53.5	54.3	58.2	54.8
S	39.2	50.9	47.8	48	53	49.1
H	31.6	49.9	43.9	46.5	45.7	47
P	32.6	47	40.5	45.4	46	45.4
V	31.1	48	40.2	45.4	46.2	44.9
Z	31.9	44.4	39.7	41.3	43.6	43.1

TABLE IX
CLASSIFICATION OF 27 FOLDS FOR DIFFERENT COMBINATIONS OF FEATURES USING DIFFERENT CLASSIFIERS WITHOUT CONSIDERING LENGTH AS FEATURE

Classifier	**C**	**CS**	**CSH**	**CSHP**	**CSHPV**	**CSHPVZ**
Logitboost	46.2	57.7	58.2	57.1	57.1	57.2
AdaboostM1	46.2	56.4	58.4	58.7	60.3	56.1
Rotationforest	50.7	60.1	55.9	56.4	55.9	56.9
Randomforest	50.4	57.4	56.9	56.9	57.7	57.2
J48	33.7	41	35	35.2	36.8	37.1
MultiboostAB	50.7	60.1	55.9	56.4	55.9	56.9

TABLE X
CLASSIFICATION OF 27 FOLDS FOR DIFFERENT COMBINATIONS OF FEATURES USING DIFFERENT CLASSIFIERS WITH CONSIDERING LENGTH AS FEATURE

Classifier	**C**	**CS**	**CSH**	**CSHP**	**CSHPV**	**CSHPVZ**
Logitboost	53.5	60.1	59.0	58.5	57.2	58.7
AdaboostM1	54.8	57.4	57.2	59.8	57.4	57.7

Rotationforest	58.2	59.8	60.1	59.3	61.4	60.6
Randomforest	57.2	59.8	59.8	56.4	57.2	58.7
MultiboostAB	54.3	57.7	59.5	59.3	58.2	59.0
J48	39.7	42.6	40.7	40.5	40.5	41.0

TABLE XI
COMPARISON TABLE OF DIFFERENT METHODS FOR 27 FOLDS

Approach	**Accuracy**
Ding and Dubchak NN (OvO)	41.8
Ding and Dubchak SVM (OvO)	45.2
Ding and Dubchak SVM (uOvO)	51.1
Ding and Dubchak SVM (AvA)	56.0
Chung RBFN	49.4
Chung Hierarchical structure MLP	44.7
Chung Hierarchical structure RBFN	56.4
Chung Hierarchical structure GRNN	45.2
Chung Hierarchical structure SVM	53.8
Chinnaswamy	58.2
Okun(HKNN)57.4	
Bologona	59.1
Shamim	60.5
ENDs	53.0
ENDs(C4.5)	56.1
ENDs(Bagged C4.5)	60.5
Logitboost	60.1
Adaboost	59.8
Rotationforest	61.4
Randomforest	59.8
MultiboostAB	59.5

TABLE XII
COMPARISON TABLE OF DIFFERENT METHODS BASED ON THE ACCURACY FOR THE SCOP STRUCTURAL CLASSES

Class	**Shamim**	**Ding**	**Marosolo**	**SVM**	**Logitboost**	**Adaboost**	**Rotation forest**	**Random forest**	**Multiboost**
All-Alpha	78	69	82	31.5	83.6	80.3	85.2	83.6	83.6
All-Beta	56	43	55	50	64.1	66.7	66.7	68.4	66.7
Alpha+Beta	49	52	52	25.3	90.3	90.3	88.7	87.1	90.3
Alpha/Beta	59	44	85	29.6	62.9	66.7	67.1	62.9	60.1
Average	60.5	52	68	34.1	75.2	76	76.9	75.5	75.2

IX. Conclusion

In this work, we studied several important issues for protein fold classification from protein sequence features. In this context large number of protein folds using different machine learning algorithms are studied and analyzed. Here the classification by class shows much accuracy than the classification by fold and in that All alpha and Alpha+Beta shows more results than Beta. These methods are not enough to increase the classification accuracy because there is an imbalance classes among the dataset. It can be seen that some of the structural classes like Alpha/Beta have very low performance accuracy inspite of applying the boosting algorithms. It is still desirable to extract relevant features which are part of the ongoing study. Our future work is to handle the unbalanced among the data set and to perform classification on the balanced data set.

REFERENCES

[1] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," Bioinformatics, vol. 17, pp. 349-358, 2001.

[2] O. Okun, "Protein fold recognition with k-local hyperplane distance nearest neighbor algorithm", Proceedings of the 2nd European Workshop on Data Mining and Text Mining for Bioinformatics, pp. 47–53, 2004.

[3] Xing Zhao, Xin Li Luonan Chen, and Kazuyuki Aihara, "Protein classification with imbalanced data," *Proteins* , vol. 70, pp. 1125–1132, 2008.

[4] A. Chinnasamy, W. K. Sung, and A. Mittal, " Protein structure and prediction using tree augumented navies Bayesian classifier," Pacific *Symposium on Biocomputing*, pp. 387–398.

[5] Marsolo Keith, Srinivasan Parthasarathy , and H.Q. Ding "A multi-level approach to SCOP fold recognition," in *Proc.Fifth IEEE Symposium on Bioinformatics and Bioengineering*, 2005.

[6] Chung, C. T. Lin, and N. R. Pal, "Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification," *IEEE transactions on NanoBioscience*, vol. 2(4), pp. 221–232, 2003.

[7] G. Bologna, and R. D. Appel, "A comparison study on protein fold recognition," in *Proc. Ninth International Conference on Neural Information Processing*, 2002, vol. 5, pp. 2492–2496.

[8] 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(24), pp. 3320-3327, 2007.

[9] W. Chmeilnicki, and K. Stapor "An efficient multi-class support vector machine classifier for protein fold recognition," *IWPACBB*, 2010, pp. 77–84.

[10] Zimek, B. Fabian, E. Frank, K. Stefan, "A study of hierarchical and flat classification of proteins," *IEEE transactions on computational biology and bioinformatics* ,vol. 7(3), pp.563–571, 2010.

[11] R. E. Schapire, "The strength of weak learnability," Machine Learning 5, 197–227, 1990.

[12] J. Friedman, T. Hastie, and R. Tibshirani "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28(2), pp. 337–374, 2000.

[13] I. Geoffrey Webb, "MultiBoosting: A Technique for Combining Boosting and Wagging," *Machine Learning*, vol. 40(2).

[14] L. Breiman, "Random forests," *Machine Learning*, pp. 5–32 ,2001.

[15] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso,"Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(10), pp. 1619–1630, 2006.

[16] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P.E. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, pp. 235-242, 2000.

[17] L. Lo Conte, B. Ailey, T. J. P. Hubbard, S. E. Braner, A. G. Murzin, and C. Chothia, "SCOP a structural classification of proteins database," vol. 28(1), pp. 257–259, 2000.

[18] I. H. Witten, and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco, 2005.

[19] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, and H. Nielsen,"Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16(5), pp. 412–424, 2000.

