A Voting-Based Combination System for Protein Cellular Localization Sites Prediction

Hafida Bouziane, Belhadri Messabih, and Abdallah Chouarfa

***Abstract*—During recent years, machine learning techniques have been attracting significant attentions in molecular biology and genomic era. They have become increasingly important to solve real-world problems such as elucidating protein function. An important step in the search for knowledge of protein function is to predict its cellular localization sites. Many computational methods that try to solve this problem have been developed over the years but the imbalanced distribution of proteins in cellular locations enormously influences the behavior of these methods. Hence, the performance and efficiency of the existing prediction methods still need to be improved. A computational method for efficiently predicting protein cellular localization is highly required. In this paper, we explore the use of four supervised machine learning algorithms in predicting the cellular localization sites of proteins from the primary sequence information. Our experiments were performed using Na ïve Bayesian, k-Nearest Neighbor and feed-forward Neural Network classifiers. The experts were evaluated with and without cross-validation on E.coli and Yeast benchmarks and combined using majority voting rule for improving classification accuracy on each dataset. The experimental results show that the proposed combination system significantly outperforms the best individual classifier.**

***Index Terms*—Protein localization, na ïve Bayesian classifier, k-nearest neighbor classifier, neural network classifier, combination of classifiers, E.coli, yeast.**

I. Introduction

Recent advances in large-scale genome sequencing have led to an explosion of newly generated protein sequences. The functional characterization of newly identified proteins remains a challenging problem, especially when these proteins do not have significant homology to proteins of known function. Elucidating the protein function is very relevant for genome annotation and search for novel vaccine or drug discovery. The most reliable way to determine protein structure or function is by direct experimentation.

Unfortunately, it is laborious, expensive and time-consuming to use purely experimental techniques. Hence, in silico methods present alternative approaches to accomplish this task. Machine learning techniques seems to be a more realizable and very promising solution. An important step toward elucidating the protein function is to determine its cellular localization in living cell. Numerous efforts have been made to develop various methods for predicting protein cellular localization. Some attempt to cover a wide variety of localizations, while others focus on a small number of localizations and on specific organisms [1]. Two of the most thoroughly studied single cell organisms are Escherichia coli (E.coli) bacteria and the brewer's Yeast (Saccharomyces cerevisiae). Many studies have detailed both gene and protein expression of these two organisms.

The first approach for predicting the localization sites of proteins from their amino acid sequences was a rule based expert system PSORT developed by Nakai and Kanehisa [2], [3], then the use of a probabilistic model by Horton and Nakai [4] which could learn its parameters from a set of training data, improved significantly the classification accuracy. It achieved an accuracy of 81% on E.coli dataset and 55% on Yeast dataset. Later, the use of standard classification algorithms achieved higher classification accuracy. Among these algorithms, k-Nearest Neighbors (k-NN), binary decision tree and na ïve Bayesian classifier. The best accuracy has been achieved by k-NN classifier, that the classification of the E.coli proteins into 8 classes achieved an accuracy of 86% and Yeast proteins were classified to 10 classes with an average accuracy of 60% by applying cross-validation [5], The accuracies have been improved significantly compared to those obtained before. Since these works, many systems using variety of machine learning techniques have been proposed.

In this study, we focus on the application of four standard supervised machine learning algorithms for predicting protein localization sites from only the amino acid sequence information, namely the Multi-Layer Perceptron (MLP), the Radial Basis Function network (RBF), the k-Nearest Neighbors classifier and the na ïve Bayesian classifier. The four algorithms are evaluated using 5-fold and 10-fold cross-validations on E.coli and Yeast datasets. We additionally try to evaluate whether and how much voting can improve the prediction accuracy using a combination system based on a consensus of predictions. The system proposed combines the four classifiers decisions by applying majority voting rule. The paper is organized as follows. In Section II, we describe the materials and methods used in this work. In Section III, we summarize the experiments and the results obtained by individual classifiers and the combination system proposed. Finally, in Section IV we present discussion and conclusion about our results.

II. Materials and Methods

In this section, we briefly describe the classifiers selected for our study, introduce the datasets used and the evaluation methodologies adopted.

Manuscript received November 15, 2012; revised January 26, 2013.

The authors are with the USTO-MB University, BP 1505 El Mnaouer 3100 Oran ALGERIA (e-mail: h_bouziane@ univ-usto.dz, messabih@ univ-usto.dz, chouarfia@ univ-usto.dz).

A. Classifiers

1) Feed-forward neural networks

The Multi-layer Perceptron (MLP) and the Radial Basis Function (RBF) network are the most commonly used feed-forward Neural Network models in computational biology. The MLP is an improvement of the Perceptron [6] including one or more transition layers known as hidden layers. The units in the input layer are connected to units in the hidden layers, which in turn are connected to units in the output layer. Each connection is associated with a weight. The MLP units take a number of real-valued inputs and generate a single real-valued output, according to an activation function (transfer function) applied to the weighted sum of the outputs of the units in the preceding layer. The most commonly used activation function in this network is a sigmoid function [7]. The learning algorithm can be expressed using generalized Delta rule and back-propagation gradient descent [8]. The MLP in this study consists of an input layer, a hidden layer and an output layer. It is trained using the standard back-propagation algorithm. The hidden and the output layer units have sigmoid activation function.

In RBF network each hidden unit implements a radial activated function, whose outputs are inversely proportional to the distance from the center of the units. The way in which the network is used for data modeling is different when approximating time-series and in pattern recognition. In pattern classification applications the most used radial activated function is the Gaussian [9], [10]. The Gaussian's centers influence the performance of the RBF network. Poggio and Girosi [10] showed that using all the training data as centers may lead to network over-fitting as the number of data becomes too large, and the gradient descent approach used to update the RBF centers moved the centers towards the majority of the data. To avoid these situations, they suggested the use of a clustering algorithm to position the centers. The RBF used in this study is combined with the K-means clustering algorithm [9] for initialization of class centers.

2) K-nearest neighbor classifier

The k-Nearest Neighbor (k-NN) algorithm [11] classifies an example by assigning it the label most frequently represented among the k nearest examples which are closest examples according to a distance-based weighting (Euclidean, Manhattan, etc.). The example is classified by a majority vote of its neighbors k is a user-defined constant (a positive integer value, typically small). The strategy is that classes with the more frequent examples tend to dominate the prediction of the new example. Proper choice of the parameter k dependents on the data, it can be selected by various heuristic techniques e.g. cross-validation. The simplest way of choosing the k value is to run the algorithm many times with different k values and selecting the one with the best performance. In practice, k is usually chosen to be odd. In our study, we use the Euclidean distance to measure the distance between examples.

3) Na ïve Bayesian classifier

The Na ïve Bayesian classifier [12], [13] is a classification method based on Bayes Theorem [14]. It is designed for use when features are independent of one another within each class. To classify an instance x based on a t-uple of attribute values, $x = \{x_1, x_2, \ldots, x_N\}$ into one class $c_j \in C$ with C a set of Q classes, it consists to maximize a posterior hypothesis of assigning the most probable class c_{MAP} to x:

$$c_{MAP} = \arg\max_{c_j \in C} P(c_j \mid x_1, x_2, ..., x_n) \quad (1)$$

$$= \arg\max_{c_j \in C} \frac{P(x_1, x_2, ..., x_N \mid c_j) P(c_j)}{P(x_1, x_2, ..., x_N)} \quad (2)$$

$$= \arg\max_{c_j \in C} P(x_1, x_2, ..., x_N \mid c_j) P(c_j) \quad (3)$$

where $j \in \{1, \ldots, Q\}$. The term $P(x_1, x_2, \ldots, x_N)$ is a constant that can be disregarded. An assumption to simplify the computation of the first term in (3) is that the attributes are conditionally independent, which can yield a result of replacing the term by the product of the conditional probability for the individual attributes. In the Naive Bayes classification, it is possible to use different distributions for different features parameter estimation.

B. Combination of the Selected Classifiers

Combination of predictions from individual classification models has been studied intensively in the last decade. Many methods have been proposed for combining multiple classifiers for a single prediction task and have provided powerful results on diverse applications. In this study, with the performance of each classifier assessed, we set out to explore voting strategy to construct a system combining prediction outputs of the four classifiers attempting majority voting rule to proper exploitation of the combined strengths of the chosen classifiers to produce more accurate predictions than of any individual classifier. We can briefly describe how to obtain the combination of each classifier decisions as follows. Let us consider X a set of N examples and C a set of Q classes. Let us define an algorithm set $S = \{A_1, A_2, ..., A_M\}$ which contains the M classifiers used for the voting. Each example $x \in X$ is assigned to have one of the Q classes. Each classifier will have its prediction for each example. The final class assigned to each example is the class predicted by the majority of classifiers (gaining the majority votes) for this example. This can be formulated as follows. Let $c_l \in C$ denotes the class of an example x predicted by a classifier A_l, and let a counting function F_k defined as:

$$F_k(c_l) = \begin{cases} 1 & c_l = c_k \\ 0 & c_l \neq c_k \end{cases} \quad (4)$$

where c_l and c_k are the classes of C. The count of total votes for class c_k can then be defined as:

$$T_k = \sum_{l=1}^{M} F_k(c_l) \quad (5)$$

The predicted class c for an example x using the algorithm set S is defined to be a class that gains the majority vote as:

$$c = S(x) = \arg\max_{k\in\{1,\ldots,Q\}} T_k \quad (6)$$

If two or more classes gain the same vote, one of them is chosen arbitrarily.

C. Datasets

The datasets used here have been collected from the UCI Machine Learning Data Repository[11] [15]. We describe in what follows their main features, further description can be found in the references [2]-[4].

E.coli dataset:

Escherichia coli is a prokaryotic gram-negative bacterium, it is present in lower gut of humans and animals, some kinds of E.coli have powerful toxic. E.coli dataset has 336 instances divided into 8 classes. Each instance with eight feature values, the first is a string value describing the sequence name and the rest are seven real attributes, that we summarize as follows : mcg (McGeohs method for signal sequence recognition), gvh (Von Heijnes method for signal sequence recognition), lip (Von Heijnes signal peptidase II consensus sequence score), chg (presence of charge on N-terminus of predicted lipoproteins), aac (score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins), alm1 (score of the ALOM membrane spanning region prediction program) and alm2 (score of the ALOM program after excluding putative cleavable signal regions from the sequence). Table I summarizes the distribution of E.coli instances in the eight classes.

TABLE I: DATA DISTRIBUTION OF E.COLI DATASET

Class	abbr	Number
Cytoplasm	cp	143
Inner membrane, no signal sequence	im	77
Periplasm	pp	52
Inner membrane, uncleavable signal sequence	imU	35
Outer membrane non-lipoprotein	om	20
Outer membrane lipoprotein	omL	5
Inner membrane lipoprotein	imL	2
Inner membrane, cleavable signal sequence	imS	2

TABLE II: DATA DISTRIBUTION OF YEAST DATASET

Class	Abbr	Number
Cytoplasm	CYT	463
Nucleus	NUC	429
Mitochondrial	MIT	244
Membrane ptotein, no-N terminal signal	ME3	163
Membrane protein, uncleaved signal	ME2	51
Membrane protein, cleaved signal	ME1	44
Extracellular	EXC	35
Vacuole	Vac	30
Peroxisome	POX	20
Endoplasmic reticulum	ERL	5

Yeast dataset contains 1484 instances divided into 10 classes. Each instance with nine feature values, the first is a string value describing the sequence name and the rest are eight real attributes. The attributes are: mcg, gvh, alm (score of the ALOM membrane spanning region prediction program), mit (score of discriminant analysis of the amino acid content of the N-terminal region of mitochodrial and non-mitochondrial proteins), erl (presence of "HDEL" substring, signal for retention in the endoplasmic reticulum lumen), pox (peroxisomal targeting signal in the C-terminus), vac (score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins) and nuc (indicator of nuclear and non-nuclear proteins). Table II summarizes the distribution of Yeast instances in the ten classes.

D. Performance Evaluation Methods

We have used different methods for performance evaluation of the selected classifiers on E.coli and Yeast datasets. These methods are Cross-Validation, Confusion Matrix and four accuracy measurements. The description of these methods will be given in the following subsections.

Cross Validation

We used cross-validation method to estimate the classification accuracy it consists to split randomly a dataset to several mutually exclusive subsets (partitions or folds). In k-fold cross-validation the dataset is divided into k subsets of approximately equal size. The method holdout proceed in k steps, such as for each step one of the k subsets is used as a test set and all the (k-1) remaining subsets are put to form the training set. Every examples of the dataset appears in test set once and the classification accuracy is estimated by calculating the average accuracy across all the k steps. In our experiments we used 5-fold and 10-fold cross validations.

1) Confusion matrix

In order to compute more easily the statistics for the performance evaluation, a matrix of size Q×Q named confusion matrix (contingency table) has been used, $M = (m_{kl})_{1\le k,l\le Q}$, where m_{kl} denotes the number of examples observed in class k and predicted in class l. The rows indicate different classes observed and the columns show the result of the prediction method for each class. The number of correctly predicted examples is the sum of diagonal elements in the matrix, all others are incorrectly predicted.

2) Prediction accuracy measurements

In this study, we adopted the commonly used measures, namely Recall, Precision and F-measure for evaluating the effectiveness of the prediction for each class and the prediction accuracy for all the classes as performance measures. F-measure is widely used in machine learning algorithms, it has two components which are: the Recall and the Precision. The Recall is the ratio of the number of positive examples correctly predicted of class k and the number of all positive (observed) examples in class k. We can express this ratio using confusion matrix elements as follows:

$$\mathrm{Re}call = 100\times\frac{m_{kk}}{\sum_{l=1}^{Q} m_{kl}}, k\in\{1,\ldots,Q\} \quad (7)$$

The Precision is the ratio of number of correctly predicted examples of class k and the number of examples predicted as belonging to class k, it can formulated as follows:

[1] Web site: http://archive.ics.uci.edu/ml

$$\Pr ecision = 100 \times \frac{m_{kk}}{\sum_{i=1}^{Q} m_{ik}}, k \in \{1,...,Q\} \quad (8)$$

The F-measure is then defined as:

$$F - measure = \frac{2 \times \mathrm{Re}\, call \times \Pr ecision}{\mathrm{Re}\, call + \Pr ecision} \quad (9)$$

The the prediction accuracy is the ratio of number of all correctly predicted examples and the total number of examples (both positive and negative predicted), it is given by:

$$Accuracy = 100 \times \frac{1}{N} \sum_{k=1}^{Q} m_{kk} \quad (10)$$

III. Experiments and Results

We have evaluated the performance of the classifiers individually in predicting the cellular localization sites using E.coli and Yeast datasets. The statistical significance of the results has been analyzed using Precision, Recall and Accuracy measures. In order to estimate the classification error, we have conducted two experiments. In the first experiment the classifiers are tested on the entire datasets. In the second experiment the classifiers were tested using 5fold and 10-fold cross-validations. The MLP is trained using one hidden layer with more than 8 hidden units. We conducted a set of preliminary experiments to find the most suitable architecture in terms of training speed and success rate. We trained the network using various number of hidden units, i.e. 8, 16, 32, 64, 128. The best architectures found were the one with 16 hidden units for the E.coli dataset and the one with 32 hidden units for Yeast dataset. Using k-NN requires to set the *k* value, we have selected this parameter after a preliminary set of experiments. We then used *k*=5 for E.coli and *k*=9 for Yeast.

A. Classification of E.coli Instances Using Full Dataset

It is important to know exactly the performance of each classifier on each individual protein that is included in the dataset. Thus, in this experiment we trained and tested the classifiers on the whole dataset. Table III presents the classification success for each class. The training performance for each classifier is evaluated in terms of Precison, Recall, F-measure and Accuracy. The combination system results are also reported in the same table to make comparison between all the classifiers.

As we can observe, the outer membrane with lipoprotein (omL) proteins were identified perfectly by all the classifiers. The outer membrane (om) proteins were identified only by the RBF with 100% success rate. One observes through the results obtained that both of the classifiers failed to identify the inner membrane with cleavable signal (imS) proteins except the RBF. The RBF succeeded where the others failed. The combination exploits well the successful predictions of all the classifiers. As we can observe in this experiment, the highest classification accuracy was obtained by the combination system. A confusion matrix is given in Table IV for showing the combination system classification success using the whole dataset in learning.

TABLE IV: Confusion Matrix In Training for E.coli Entire Dataset with the Proposed Combination System

Observed	Predicted							
	cp	**im**	**pp**	**imU**	**om**	**omL**	**imL**	**imS**
cp (143)	**141**	0	2	0	0	0	0	0
im (77)	2	**71**	0	4	0	0	0	0
pp (52)	2	0	**50**	0	0	0	0	0
imU (35)	0	3	0	**32**	0	0	0	0
om (20)	0	0	1	0	**19**	0	0	0
omL (5)	0	0	0	0	0	**5**	0	0
imL (2)	0	0	0	0	0	0	**2**	0
imS (2)	0	0	0	0	0	0	0	**2**

B. Classification of E.coli Instances Using 5-Fold and 10-Fold Cross-Validations

In this experiment, we applied 5-fold and 10-fold cross-validations. The E.coli dataset is randomly partitioned to respectively 5 then 10 approximately equally sized subsets. Table V and Table VI summarize the test performance of each classifier for each class.

The results reported for this experiment show that the classification attempts of inner membrane with lipoprotein (imL) and inner membrane with cleavable signal sequence (imS) proteins failed for each classifier and consequently also for the combination system. On the other hand, outer membrane with lipoprotein (omL) proteins were classified 100% success rate only by the k-NN classifier. The cytoplasm (cp) proteins were well classified by almost all classifiers. Here, the best results were also obtained by the combination system it achieved an average classification success of 88.3%. A confusion matrix is given in Table VII for showing the combination system classification success using cross-validation tests. A diagram presenting the comparison between the four classifiers and the combination system on this dataset is given in Fig. 1.

TABLE VII: E.coli Optimal Confusion Matrix In Cross-Validation Tests with the Proposed Combination System

Observed	Predicted							
	cp	**im**	**pp**	**imU**	**om**	**omL**	**imL**	**imS**
cp (143)	**140**	0	3	0	0	0	0	0
im (77)	3	**68**	0	6	0	0	0	0
pp (52)	4	2	**45**	0	1	0	0	0
imU (35)	1	11	0	**23**	0	0	0	0
om (20)	0	1	3	0	**16**	0	0	0
omL (5)	0	0	0	0	0	**5**	0	0
imL (2)	0	1	0	0	0	1	**0**	0
imS (2)	0	1	1	0	0	0	0	**0**

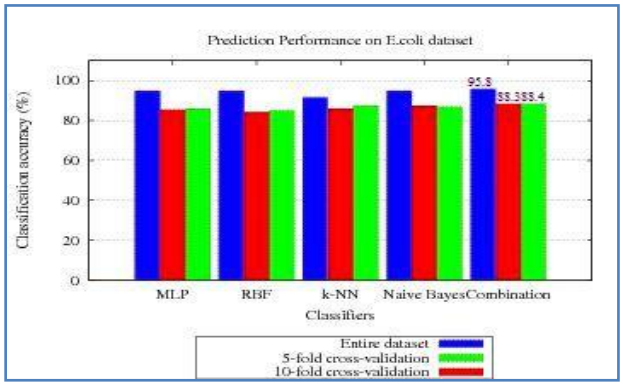

Fig. 1. Accuracy comparison between the four classifiers and the combination system on E.coli dataset.

C. Classification of Yeast Instances Using Full Dataset

In this experiment we trained and tested the classifiers on the whole dataset as previously. Table VIII reports the classification success for each class. The results show that the naïve Bayesian classifier failed to identify the vacuolar proteins (VAC) but classified the endoplasmic reticulum lumen proteins (ERL) with 100% success rate, whereas the MLP failed completely. The proposed system achieved better accuracy than the k-NN, which is the best classifier in this experiment. A confusion matrix is given in Table IX which reports the combination system classification success in learning using entire dataset.

D. Classification of Yeast Instances Using 5-Fold and 10 Fold Cross-Validations

In this experiment, Yeast dataset was divided randomly to five and then to ten different subsets, we proceeded exactly as previously. The prediction results for each classifier are reported in Table X and Table XI. As it can be seen here, all the classifiers failed to recognize the vacuolar proteins this situation is caused by the extremely low number of examples in these classes (one example used for training and one example for testing). The combination system achieved an average classification accuracy of 61.5%. The comparison between the four classifiers and the proposed system on this dataset in term of classification accuracy is given by Fig. 2. The confusion matrix is also given in Table XII for showing the combination system classification success using cross-validation tests.

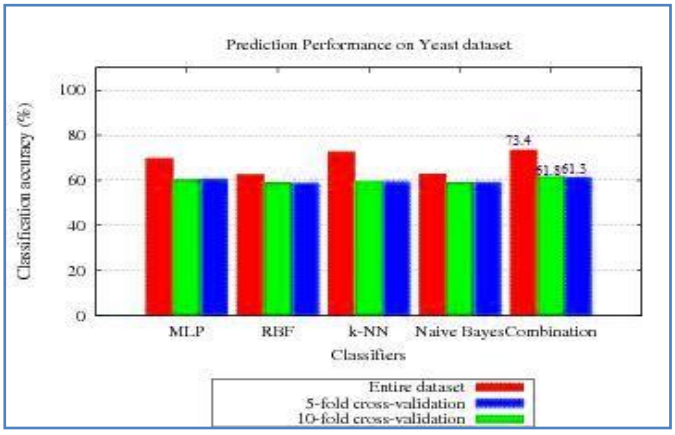

Fig. 2. Accuracy comparison between the four classifiers and the combination system on yeast dataset

TABLE III: LEARNING PERFORMANCE USING E.COLI ENTIRE DATASET

Classifiers	Measures	Classes								Correctly predicted	Accuracy
		cp	*im*	pp	imU	om	omL	imL	imS		
MLP	**Precision**	97.3	94.6	94.3	86.1	100	100	66.7	0	319	**94.9**
	Recall	99.3	90.9	96.2	88.6	95.0	100	100	0		
	F-measure	98.3	92.7	95.2	87.3	97.4	100	80.0	0		
RBF	**Precision**	97.2	91.1	96.1	87.5	100	100	100	100	319	**94.9**
	Recall	98.6	93.5	94.2	80.0	100	100	100	100		
	F-measure	97.9	92.3	95.1	83.6	100	100	100	100		
k-NN	**Precision**	95.3	93.1	90.6	78.9	100	62.5	0	0	308	**91.6**
	Recall	98.6	87.0	92.3	85.7	85.0	100	0	0		
	F-measure	96.9	89.9	91.4	82.2	91.9	76.9	0	0		
Naïve Bayes	**Precision**	97.3	94.6	94.3	86.1	100	100	66.7	0	319	**94.9**
	Recall	99.3	90.9	96.2	88.6	95.0	100	100	0		
	F-measure	98.3	92.7	95.2	87.3	97.4	100	80.0	0		
Combination	**Precision**	97.2	95.9	94.3	88.9	100	100	100	100	322	**95.8**
	Recall	98.6	92.2	96.2	91.4	95.0	100	100	100		
	F-measure	97.9	94.0	95.2	90.1	97.4	100	100	100		

TABLE V: TEST PERFORMANCE USING 5-FOLD CROSS-VALIDATION ON E.COLI DATASET

Classifiers	Measures	Classes								Correctly predicted	Accuracy
		cp	im	pp	imU	om	omL	imL	imS		
MLP	**Precision**	92.7	84.2	84.3	67.6	84.2	66.7	0	0	289	**86.0**
	Recall	97.2	83.1	82.7	71.4	80.0	40.0	0	0		
	F-measure	94.9	83.7	83.5	69.4	82.1	50.0	0	0		
RBF	**Precision**	93.8	79.8	84.3	71.4	84.2	80.0	0	0	286	**85.1**
	Recall	95.1	87.0	82.7	57.1	80.0	80.0	0	0		
	F-measure	94.4	83.2	83.5	63.5	82.1	80.1	0	0		
k-NN	**Precision**	95.3	82.9	85.5	71.9	93.8	55.6	0	0	294	**87.5**
	Recall	98.6	81.8	90.4	65.7	75.0	100	0	0		
	F-measure	96.9	82.4	87.9	68.7	83.3	71.4	0	0		
Naïve Bayes	**Precision**	94.6	80.8	88.5	61.8	100	80.0	0	0	292	**86.9**
	Recall	97.9	81.8	88.5	60.0	90.0	80.0	0	0		
	F-measure	96.2	81.3	88.5	60.9	94.7	80.0	0	0		
Combination	**Precision**	94.6	81.0	86.5	79.3	94.1	83.3	0	0	297	**88.4**
	Recall	97.9	88.3	86.5	65.7	80.0	100	0	0		
	F-measure	96.2	84.5	86.5	71.9	86.5	90.9	0	0		

TABLE VI: TEST PERFORMANCE USING 10-FOLD CROSS-VALIDATION ON E.COLI DATASE

Classifiers	Measures	Classes								Correctly predicted	Accuracy
		cp	im	pp	imU	om	omL	imL	imS		
MLP	**Precision**	94.5	81.0	83.6	62.9	89.5	0	0	0	287	**85.4**
	Recall	96.5	83.1	88.5	62.9	85.0	0	0	0		
	F-measure	95.5	82.1	86.0	62.9	87.2	0	0	0		
RBF	**Precision**	93.7	80.5	84.3	64.9	84.2	80.0	0	0	283	**84.2**
	Recall	93.7	80.5	82.7	68.6	80.0	80.0	0	0		
	F-measure	93.4	80.5	83.5	66.7	82.1	80.0	0	0		
k-NN	**Precision**	95.3	80.8	85.2	65.5	88.2	50.0	0	0	289	**86.0**
	Recall	98.6	81.8	88.5	54.3	75.0	100	0	0		
	F-measure	96.9	81.3	86.8	59.4	81.1	66.7	0	0		
Na ïve Bayes	**Precision**	94.6	81.8	88.7	66.7	100	80.0	0	0	294	**87.5**
	Recall	97.9	81.8	90.4	62.9	90.0	80.0	0	0		
	F-measure	96.2	81.8	89.5	64.7	94.7	80.0	0	0		
Combination	**Precision**	95.2	83.8	84.9	75.8	94.1	80.0	0	0	297	**88.3**
	Recall	97.9	87.0	86.5	71.4	80.0	80.0	0	0		
	F-measure	96.6	85.4	85.7	73.5	86.5	80.0	0	0		

TABLE VIII: LEARNING PERFORMANCE USING YEAST ENTIRE DATASET

Classifiers	Measures	Classes										Correctly predicted	Accuracy
		CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL		
MLP	**Precision**	62.7	70.5	73.2	78.5	80.0	86.4	71.9	50.0	82.4	0	1036	**69.8**
	Recall	75.2	61.3	66.0	93.9	62.7	86.4	65.7	13.3	70.0	0		
	F-measure	68.4	65.6	69.4	85.5	70.3	86.4	68.7	21.1	75.7	0		
RBF	**Precision**	55.8	60.9	68.1	78.6	63.9	79.2	51.2	50.0	92.9	83.3	931	**62.7**
	Recall	65.7	58.7	57.0	81.0	45.1	86.4	62.9	10.0	65.0	100		
	F-measure	60.3	59.8	62.1	79.8	52.9	82.6	56.4	16.7	76.5	90.9		
k-NN	**Precision**	64.8	75.3	76.3	87.2	73.3	87.8	69.7	100	90.9	100	1081	**72.8**
	Recall	83.2	69.5	67.2	79.1	54.9	81.8	65.7	10.0	50.0	100		
	F-measure	72.8	72.2	71.5	83.0	62.9	84.7	67.6	18.2	64.5	100		
Naive Bayes	**Precision**	56.0	67.2	68.8	77.2	52.9	66.7	40.6	0	73.3	100	933	**62.8**
	Recall	71.5	54.1	58.6	85.3	35.3	63.6	74.3	0	55.0	100		
	F-measure	62.8	59.9	63.3	81.0	42.4	65.1	52.5	0	62.9	100		
Combination	**Precision**	63.8	76.9	79.2	83.7	82.1	83.3	74.3	100	93.3	100	1090	**73.4**
	Recall	80.6	65.3	68.9	91.4	62.7	90.9	74.3	10.0	70.0	100		
	F-measure	71.2	70.6	73.7	87.4	71.1	87.0	74.3	18.2	80.0	100		

TABLE IX: CONFUSION MATRIX IN TRAINING FOR YEAST ENTIRE DATASET WITH THE PROPOSED COMBINATION SYSTEM

Observed	Predicted									
	CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL
CYT (463)	**373**	63	22	4	0	0	1	0	0	0
NUC (429)	122	**280**	17	11	9	1	0	0	0	0
MIT (244)	57	9	**168**	5	2	1	1	0	1	0
ME3 (163)	6	5	1	**149**	2	0	0	0	0	0
ME2 (51)	5	2	1	5	**32**	5	1	0	0	0
ME1 (44)	0	0	0	0	0	**40**	4	0	0	0
EXC (35)	4	0	2	0	1	2	**26**	0	0	0
VAC (30)	13	4	1	6	1	0	2	**3**	0	0
POX (20)	5	1	0	0	0	0	0	0	**14**	0
ERL (5)	0	0	0	0	0	0	0	0	0	**5**

TABLE X: TEST PERFORMANCE USING 5-FOLD CROSS-VALIDATION ON YEAST DATASET

Classifiers	Measures	Classes										Correctly predicted	Accuracy
		CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL		
MLP	**Precision**	56.1	59.5	63.1	72.8	43.8	66.0	63.6	0	80.0	0	900	**60.6**
	Recall	59.4	58.5	61.1	87.1	41.2	75.0	60.0	0	40.0	0		
	F-measure	57.7	59.0	62.1	79.3	42.4	70.2	61.8	0	53.3	0		
RBF	**Precision**	53.3	59.2	65.1	74.0	41.2	55.0	48.7	20.0	80.0	100	873	**58.8**
	Recall	66.7	49.0	57.4	82.2	27.5	75.0	54.3	03.3	40.0	100		
	F-measure	59.3	53.6	61.0	77.9	32.9	63.5	51.4	05.7	53.3	100		
k-NN	**Precision**	53.3	59.2	65.1	74.0	41.2	55.0	48.7	20.0	80.0	100	880	**59.3**
	Recall	66.7	49.0	57.4	82.2	27.5	75.0	54.3	03.3	40.0	100		
	F-measure	59.3	53.6	61.0	77.9	32.9	63.5	51.4	05.7	53.3	100		
Naive Bayes	**Precision**	53.5	62.1	65.7	76.0	44.7	58.1	34.3	0	66.7	100	877	**59.1**
	Recall	68.0	50.8	56.6	79.8	33.3	56.8	65.7	0	40.0	60.0		
	F-measure	59.9	55.9	60.8	77.8	38.2	57.5	45.1	0	50.0	75.0		
Combination	**Precision**	55.2	61.6	67.3	75.6	48.8	66.7	48.9	0	71.4	100	910	**61.3**
	Recall	67.4	53.1	59.0	83.4	35.2	72.7	65.7	0	50.0	100		
	F-measure	60.7	57.1	62.9	79.3	43.5	69.6	56.1	0	58.8	100		

TABLE XI: TEST PERFORMANCE USING 10-FOLD CROSS-VALIDATION ON YEAST DATASET

Classifiers	Measures	Classes										Correctly predicted	Accuracy
		CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL		
MLP	**Precision**	56.2	59.6	64.8	73.8	44.4	63.5	52.8	0	70.0	0	897	**60.4**
	Recall	65.2	56.6	56.6	82.8	39.2	75.0	54.3	0	35.0	0		
	F-measure	60.4	58.1	60.4	78.0	41.7	68.8	53.5	0	46.7	0		
RBF	**Precision**	53.6	56.9	67.5	73.0	45.5	62.5	55.0	0	69.2	83.3	877	**59.1**
	Recall	64.1	52.9	56.1	79.8	29.4	79.5	62.9	0	45.0	100		
	F-measure	58.4	54.8	61.3	76.2	35.7	70.0	58.7	0	54.5	90.9		
k-NN	**Precision**	54.1	57.5	64.2	74.4	54.3	71.4	58.3	0	73.3	83.3	886	**59.7**
	Recall	64.8	54.3	57.4	74.8	37.3	79.5	60.0	0	55.0	100		
	F-measure	58.9	55.9	60.6	74.6	44.2	75.3	59.2	0	62.9	90.9		
Naive Bayes	**Precision**	53.5	62.0	65.7	77.4	42.5	60.5	36.4	0	66.7	100	879	**59.2**
	Recall	68.3	51.3	55.7	79.8	33.3	59.1	68.6	0	40.0	40.0		
	F-measure	60.0	56.1	60.3	78.5	37.4	59.8	47.5	0	50.0	57.1		
Combination	**Precision**	55.9	62.7	69.0	76.2	43.9	71.2	48.9	0	69.2	100	918	**61.8**
	Recall	71.3	53.6	57.4	80.4	35.3	72.7	65.7	0	45.0	100		
	F-measure	62.7	57.8	62.6	78.2	39.1	71.9	56.1	0	54.5	100		

TABLE XII: YEAST OPTIMAL CONFUSION MATRIX IN CROSS-VALIDATION TESTS WITH THE PROPOSED COMBINATION SYSTEM

Observed	Predicted									
	CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL
CYT (463)	**330**	96	27	6	1	0	2	0	1	0
NUC (429)	154	**230**	27	14	3	0	1	0	0	0
MIT (244)	63	19	**140**	8	8	1	1	1	3	0
ME3 (163)	10	15	2	**131**	3	2	0	0	0	0
ME2 (51)	8	3	1	7	**18**	7	7	0	0	0
ME1 (44)	0	0	0	0	3	**32**	9	0	0	0
EXC (35)	4	1	2	0	2	3	**23**	0	0	0
VAC (30)	15	2	2	6	2	0	3	**0**	0	0
POX (20)	6	1	2	0	1	0	1	0	**9**	0
ERL (5)	0	0	0	0	0	0	0	0	0	**5**

IV. CONCLUSION AND DISCUSSION

Protein cellular localization sites prediction is one of the most challenging problems in modern computational biology. Various approaches have been proposed and applied to solve this problem but the extremely imbalanced distribution of proteins over the cellular locations make the prediction much more difficult. The objective of this study was to explore and investigate the use of some standard classifiers in predicting the cellular localization sites of proteins using E.coli and Yeast Benchmarks and to try to evaluate whether and how much a combination system based on a consensus of their predictions using majority voting strategy can improve the prediction accuracy. The experiments showed the power of the MLP and the k-NN classifier to identify E.coli and Yeast instances as individual classifiers, and that confirms well the reported results in recent studies dealing with neural networks and k-NN approaches focused to the problem of the protein localization sites prediction [5], [16]-[20].

Finally, the experimental results highlighted the proposed combination system superiority it produced better classification accuracy than the best classifier in the ensemble. An improvement of approximately 2 % was reached. Through this study, we have shown that the classification error can be reduced if we combine several classifiers. The combination of classifiers is an effective tool to improve the classification accuracy and the voting strategy is a robust way to combine predictions. However, further investigations will be carried out using more effective classifiers and different combination strategies to provide a much more improved model performance.

REFERENCES

[1] T. Q. Tungl and D. Lee, "A method to improve protein subcellular localization prediction by integrating various biological data sources," *BMC Bioinformatics*, vol. 10, no. 1, 2009.

[2] K. Nakai and M. Kanehisa, "Expert system for predicting protein localization sites in gram-negative bacteria," *Proteins: Structure, Function, and Genetics*, vol. 11, pp. 95-110, 1991.

[3] K. Nakai and M. Kanehisa, "A knowledge base for predicting protein localisation sites in eukaryotic cells," *Genomics*, vol. 14, pp. 897-911, 1992.

[4] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins," in *Proc. Intelligent Systems in Molecular Biology*, St. Louis, USA, pp. 109-115, 1996.

[5] P. Horton and K. Nakai, *Better Prediction of Protein Cellular Localization Sites with the K Nearest Neighbors Classifier*, AAAI Press. Halkidiki, Greece, pp. 147-152, 1997.

[6] M. Minsky and S. Papert, *Perceptron: An Essay in Computational Geometry*, MIT Press, 1969.

[7] S. Narayan, "The generalized sigmoid activation function: competitive supervised learning," *Information Sciences*, vol. 99, no. 1-2, pp. 69-82, 1997.

[8] D. E. Rumellart, G. E. Hinton, and R. J. Williams, *Learning Internal Representation by Errors Propagation*, MIT Press, Cambridge, vol. 1, pp. 318-362, 1986.

[9] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281-294, 1989.

[10] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proc. IEEE*, vol. 78, no. 9, pp. 1481-1497, 1990.

[11] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[12] I. J. Good, *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, MIT Press, 1965.

[13] P. Langley, W. Iba, and K. Thompson, "An analysis of bayesian classifiers," in *Proc. Tenth National Conference on Artificial Intelligence*, Menlo park, pp. 223-228, 1992.
[14] T. Bayes, "An essay toward solving a problem in the doctrine of chances," *Philosophical Transactions of the Royal Society of London*, vol. 53, pp. 370-418, 1763.
[15] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, 1998.
[16] A. D. Anastasiadis and G. D. Magoulas, "Analysing the localisation sites of proteins through neural networks ensembles," *Neural Computing and Applications*, vol. 15, no. 3-4, pp. 277-288, 2006.
[17] A. D. Anastasiadis, G. D. Magoulas, and X. Liu, "Classification of protein localisation patterns via supervised neural network learning," in *Proc. Fifth Symposium on Intelligent Data Analysis (IDA-03), LNCS 2810*, pp. 430-439, 2003.
[18] A. Reinhardt and T. Hubbard, *Using Neural Networks for Prediction of the Subcellular Location of Proteins*, Nucleic Acids Research, vol. 26, no. 9, pp. 2230-2236, 1998.
[19] M. Avci and T. Yildirim, "Classification of Escherichia Coli bacteria by artificial neural networks," in *Proc. First international IEEE Symposium in Intelligent Systems, IEEE Xplore*, vol. 3 pp. 13-16, 2002.
[20] I. Turkoglu and E. D. Kaymaz, "A hybrid method based on artificial immune system and k-NN algorithm for better prediction of protein cellular localization sites," *Applied Soft Computing*, vol. 9, pp. 497-502, 2009.