A two Stage Approach towards Protein secondary structure classification

Authors: Kushal Kanti Ghosh, Soulib Ghosh, Sagnik Sen, Ram Sarkar, Ujjwal Maulik

Abstract:

Protein secondary structure (PSS) describes the local folded structures which form inside a polypeptide due to interactions among atoms of the backbone. Generally, globular proteins are divided into four classes namely all-α, all-β, $\alpha + \beta$ and α/β. Classification of PSS is important in terms of different biological functions that include protein fold recognition, tertiary structure prediction, prediction of DNA-binding sites, and reduction of the conformation search space among others. In this paper, we have proposed a machine learning based model for secondary structure classification of proteins into said four classes. In doing so, we have considered both sequence based and structure based features. At first, mutual information, a filter based feature selection method, is used to remove the redundant features, and then these features are used to train three different classifiers – Random Forest, K-Nearest Neighbour (KNN), and Multi-layer Perceptron (MLP). After that some standard classifier combination approaches applied to integrate the decision made by the said classifiers and it has been found that weighted product rule performs the best among all. The proposed model outnumbers some state-of-the-art methods used for classifications of PSS.

Keywords:

Protein, Secondary structure, Protein sequence, Feature selection, Classifier combination

1. Introduction:

Protein plays a vital role in various physio-chemical process in human being. Functioning of immunity system, Na-K ion channels, energy regulation, drug delivery are some tasks controlled by proteins [1]. Protein Secondary Structure Classification (PSSC) was first proposed by Levitt and Chothia [2]. They divided 31 globular proteins in four classes all-α, all-β, $\alpha + \beta$ and α/β. PSSC is useful in many aspects like protein fold recognition, tertiary structure prediction, analysis of protein function for drug discovery, prediction of protein folding rates, prediction of DNA-binding sites, reduction of the conformation search space [3] etc. In structural classification of proteins (SCOP) [4], the classification is done manually [5] based on proteins with known tertiary structure. In the latest version of the SCOP, the number of classes has been increased to 11. Since, almost 90% of proteins fall into the mentioned four classes, these are mainly considered for computational classification purposes [6] . However, rapid growth of genomics and proteomics has resulted in huge data on amino acid sequences. So, manual classification method is not a viable option. Herein lies the importance of computational prediction of structural classes of proteins. PSSC system broadly has two stages: Feature modelling and Classification. Feature modelling is the task of identifying the relevant features which contribute to predict PSS accurately. The features used for PSSC are broadly classified into two groups [7] : sequence based and structure based. Sequence based features are extracted following the alphabetic sequence of the amino acids. These types of features yield high accuracy when tested on high sequence similarity datasets. But they fail in case of low similarity datasets. To remove this deficiency, structural features are introduced in the literature. Structural features are extracted based on predicted secondary structure of proteins.

Till date, researcher have applied several classifiers for PSSC. However, the accuracies achieved by these are highly affected by the sequence similarity within the train and test datasets [8]. Due to the presence of homologous proteins, overestimation of prediction accuracy happens in case of high similarity datasets (this means we get high accuracy for high similarity datasets). The existing sequence representation methods and classification algorithms have been extensively reviewed in [9][5]. During the last few decades, a lot of computational methods have been applied in PSSC. The main focus of these methods is on the sequence based features. Several features based on amino acid composition (AAC) [10], pseudo amino acid composition (pseAAC) [10] have been applied for PSSC. But they do not perform well in case of low sequence similarity datasets [10]. So, many researchers have used structure based features. But prediction accuracies for $\alpha + \beta$ and α/β are still unsatisfactory. In the paper [11], Liu et al. have used AAC and dipeptide composition from the PSI-BLAST profile. Costantini et al. in their work [8] have used polypeptide composition, whereas in [12] Chou et al. have adopted functional domain composition and Yang et al. in [13] have availed

amino acid sequence reverse encoding. In [14], Kurgan et al. have introduced features based on the predicted secondary structure and PSI-BLAST profile. Some other works based on structural features can be found in [15] [16]. In [17], Dai et al. have come up with sequential and structural features and compared each other using Support Vector Machine (SVM) classifier and 4 datasets. Zhang et al. in [10] have introduced 14 structural features and tested on two datasets with SVM classifier. Bao et al. have put forward a new method based on structural features for PSSC in [18] and applied on three benchmark datasets.

In the present work, we have followed a two-stage approach with the goal of improving prediction accuracy for protein secondary structural classes. In the first stage, we have extracted both sequence based and structure based features and applied feature selection method named Mutual Information (MI) on the extracted features. The selected features are then passed through the second stage. In this stage, we have combined the outcomes of three classifiers, namely, Random Forest (RF) [22], K-Nearest Neighbour (KNN) [23] and Multi-Layer Perceptron (MLP) [24], using the rule of weighted product. From them results described in Section 4, we can clearly see that the overall prediction accuracy has increased. The entire paper is divided into five main sections. The first section deals with introduction and related work. The second section covers the detailed explanation of methods and methodologies, required for the present work. This section contains the description of feature extraction, feature selection and classifier combination. Third section describes the proposed method that contains the justifications of the proposed approach and the model design. Next section illustrates the results obtained and the comparisons with the state-of-the art methods. The final section demonstrate the conclusion and future scope.

2. Methods and Methodologies

2.1) Feature Extraction:

A protein sequence is made from various combinations of 20 different amino acids [36] which are denoted as $\Sigma = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. The $n-gram$ features are defined as a pair of values (v_i, c_i) where, v_i is the feature i and c_i is the count of the feature i in the amino acid sequence, for $i = 1,2,\ldots, 20^n$. These features are the all possible combinations of the characters from the set Σ mentioned earlier in this section. Another commonly used information for protein classification is 6-letter exchange group [37]. The six letter group actually contains 6 combinations of the letters from the set Σ. These combinations are: $A = \{H, R, K\}$, $B = \{D, E, N, Q\}$, $C = \{C\}$, $D = \{S, T, P, A, G\}$, $E = \{M, I, L, V\}$ and $F = \{F, Y, W\}$. We denote $n-gram$ features of Σ as a_i and that of 6 letter exchange groups as e_i. We have used a_1, e_1 and e_2 in

our work. The features have been scaled to avoid skewness in the count for different protein chains by using the formula 1:

$$\bar{x} = \frac{x}{L-n+1} \text{ …….. (1)}$$

Where x = count of the gram features, L = length of protein sequence, n = size of the gram feature and $\bar{x}$ is the normalized feature value.

Number of a_1 features: 20, e_1 features: 6 and $e_2$features: 36. So we have 62 features from here. a_1 features: $[A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y]$. e_1 features are $[A, B, C, D, E, F]$.

To differentiate a_1 and e_1 features, we have denoted e_1 features as $[xA, xB, xC, xD, xE, xF]$, whereas e_2 features are as follows:

$[AA, AB, AC, AD, AE, AF, BA, BB, BC, BD, BE, BF, CA, CB, CC, CD, CE, CF, DA, DB, DC, DD, DE, DF, EA, EB, EC, ED, EE, EF, FA, FB, FC, FD, FE, FF]$

Every amino acid in a protein sequence is predicted as one of the three secondary structures: E (strand), H (helix) , C (coil) using STRIDE [10]. In ß and α proteins, the effect of α-helix and ß-strand is very minimal respectively. On the other hand, the effect of both α-helix and ß-strand on α+ß and α/ß proteins are quite prominent. α-helix and ß-strand are usually interspersed in α/ß, however, they are generally separated in α+ß. The prediction is well explained in the paper [38] using a method named PSIPRED. Hence, every amino acid sequence in a protein can be represented using H, E and C, which is known as SSS (Secondary Structure Sequence). Particularly to deal with the α+ß and α/ß proteins, another modified sequence is calculated from SSS known as SS (Simplified Sequence). To calculate SS from SSS, firstly all the consecutive sequences of H, E and C are replaced by a single α, ß and C respectively. After that, all the C is removed from the so obtained intermediate sequence to get hold of the SS. As an example, if the SSS is**:** **EECEEECEECCHHCCHHCCEECCHHCEE** then the SS becomes**:** **ßßßßßßααααßßααßß.** In our proposed method, 25 features are obtained among them 20 are from the SSS and 5 from SS.

Two basic nomenclatures are introduced which will be used throughout the article: - N – Length of SSS, N' – Length of SS.

1-gram features from these sequences have been considered. Number of features in this case is three corresponding to: $[E, H, C]$.

It has been already mentioned that the effect of H and E is very predominant in α and ß proteins respectively. In any SSS, if the presence of H is substantially higher than E, it can be concluded that there is a high possibility that the protein is α-protein. Similarly in any ß-protein, presence of E is considerably higher than H [14]. In our proposed, $P(H)$, $P(E)$ and $Diff$ are used

as features. $P(H)\ and\ P(E)$ are probability of occurrence of H and E in SSS respectively. $Diff$ is the absolute value of the difference between $P(H)\ and\ P(E)$. Difference is additionally is used to strengthen the variation between the presence of H and E.

$$Diff = abs\big(P(E) - P(H)\big) \dots\dots\dots\dots\dots\dots\dots (2)$$

In the previous case, for feature extraction, only the presence of H, E and C is considered. Now, we consider the spatial arrangements of those H, E and C [14]. Three features namely $CV_H, CV_E\ and\ CV_C$ are calculated using equations 3, 4 and 5 respectively.

$$CV_H = \frac{\sum_{j=1}^{N_H} P_{H_j}}{N(N-1)} \dots\dots\dots\dots\dots\dots\dots (3)$$

$$CV_E = \frac{\sum_{j=1}^{N_E} P_{E_j}}{N(N-1)} \dots\dots\dots\dots\dots\dots\dots (4)$$

$$CV_C = \frac{\sum_{j=1}^{N_C} P_{C_j}}{N(N-1)} \dots\dots\dots\dots\dots\dots\dots (5)$$

$N_H, N_E\ and\ N_C$ are the number of H, E and C in SSS. $P_{H_j}, P_{E_j} and\ P_{C_j}$ are the j^{th} position of H, E and C.

In real scenario, H, E and C have some three dimensional shape which gives rise to the protein structure. As α-protein contains mainly H, so we can find so many long helix structures which can be useful in differentiating it from other proteins. Similarly, ß-protein contains so many long E. In case of α+ß and α/ß proteins, the length of H and E is not so large. To include three dimensional spatial information, four features are taken from the SSS based on H and E sequences [39]. First of all, the lengths of the consecutive H and E are stored in two separate arrays. The normalized mean and maximum of the two arrays are considered as features which are formulated as shown in equations 6, 7, 8 and 9.

$$ME = Max(Seg_E\ array)/N \dots\dots\dots\dots\dots\dots\dots (6)$$

$$MH = Max(Seg_H\ array)/N \dots\dots\dots\dots\dots\dots\dots (7)$$

$$MeE = Mean(Seg_E\ array)/N \dots\dots\dots\dots\dots\dots (8)$$

$$MeH = Mean(Seg_H\ array)/N \dots\dots\dots\dots\dots\dots (9)$$

Till now the features are designed to distinguish between α and ß proteins. The main difference between α+ß and α/ß proteins is that the E are parallel in α/ß protein, whereas, they are anti-parallel in α+ß protein. The next features are made to distinguish between α+ß and α/ß proteins.

α+ß and α/ß proteins contain both H and E in significant amount which is a notable distinguishing property from α and ß proteins. Due to the parallel and anti-parallel nature of E-strands, the distance between H and E segments of SSS is different in α+ß and α/ß proteins [10]. The distances between H segment and E are stored in an array. The normalized mean and maximum

of that array are taken as features [10]. Similarly, the distances between E and H segments are kept in an array. The features taken are normalized mean and maximum of that array. The features are formulated in the equations 10, 11, 12 and 13.

$$MEH = Max(Dist_{EH}\ array)/N \ldots\ldots\ldots\ldots\ldots (10)$$
$$MHE = Max(Dist_{HE}\ array)/N \ldots\ldots\ldots\ldots\ldots (11)$$
$$MeEH = Mean(Dist_{EH}\ array)/N \ldots\ldots\ldots\ldots\ldots (12)$$
$$MeHE = Mean(Dist_{HE}\ array)/N \ldots\ldots\ldots\ldots\ldots\ldots (13)$$

Specifically to highlight the parallel and anti-parallel issues of E in α+ß and α/ß proteins, we concentrate on the E-strand segments of the SSS. Two distance probabilities from SSS based on the E-strand segments are considered as features which are formulated in equations 14 and 15.

$$P_{D_E} = \frac{N_1}{N_1 + N_2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$
$$P'_{D_E} = \frac{N_2}{N_1 + N_2} \ldots\ldots\ldots\ldots\ldots\ldots (15)$$

First, we construct a set D_E that contains the distances between two E-strand segments. Here, N_1 is the number of elements in D_E whose value is greater than or equal to 5 and N_2 is the number of elements in D_E having value less than 5.

The length of the SSS also varies in different protein structures. So, length of SSS is also considered as a feature in our proposed method.

Finally, five probabilistic features are extracted from the SS. The normalized occurrence of ‘αα’, ‘ßß’, ‘ßαß’, ‘ααα’ and ‘ßßß’ are used as features to distinguish between α+ß and α/ß proteins. The formulae for the said features are given in equations 16, 17, 18, 19 and 20 respectively.

$$Prob_{\alpha\alpha} = \frac{N_{\alpha\alpha}}{N'} \ldots\ldots\ldots\ldots (16)$$
$$Prob_{ßß} = \frac{N_{ßß}}{N'} \ldots\ldots\ldots\ldots (17)$$
$$Prob_{ß\alpha ß} = \frac{N_{ß\alpha ß}}{N'} \ldots\ldots\ldots (18)$$
$$Prob_{\alpha\alpha\alpha} = \frac{N_{\alpha\alpha\alpha}}{N'} \ldots\ldots\ldots (19)$$
$$Prob_{ßßß} = \frac{N_{ßßß}}{N'} \ldots\ldots\ldots\ldots (20)$$

In this work, total 87 features have been extracted from various sequence representation of a protein. The number of features obtained from each sequence is represented in a tabular format in the following Table 1.

Table 1: Description of the number of features taken from individual protein sequence.

Feature Descriptor Sequence	Number of Features
Amino Acid Sequence	20
Exchange Group Sequence	42
Secondary Structure Sequence	20
Simplified Sequence	5
Total Features	**87**

2.2) Feature Selection:

The performance of a classifier is closely dependent on intrinsic structure of the training set. There may be many redundant and irrelevant features which may lead to low classification accuracy. Redundancy in dataset also results in space wastage and slows down the classification process. Feature selection is a pre-processing step which is used to identify relevant features and remove redundant and irrelevant features from the dataset. Brute force feature selection requires to select all possible combinations of the features and evaluating them. Obviously, the cost is too high, making it an unfeasible option. Forward [25] and backward [25] feature selection methods, though computationally less expensive than the brute force approach, result in low classification accuracy, because they fail to consider all possible feature combinations. In contemporary research [25], feature selection is done in three different ways: filter methods, wrapper methods and hybrid methods. Filter methods select features using the intrinsic properties of the data. These are fast, scalable, light weight and do not use any classifier. On the other hand, wrapper methods use classifiers to evaluate the feature subsets. Wrappers beat filters in terms of accuracy but at the cost of high computational complexity due to the usage of classifiers. Hybrid methods try to combine the pros of the above two methods. But the proper choice of filter-wrapper combination requires exhaustive experimentation. Since filter methods are fast and unbiased towards the selection of classifier, they are used widely for feature dimensionality reduction, especially for large datasets [25]. Taking into consideration all the above mentioned facts, we have applied filter methods for feature selection.

1. Entropy based: These methods use Shannon's information theory [26], and compute information gain by calculating the entropy of a feature w.r.t. the label. Symmetric Uncertainty [26], MI [27], Minimum Redundancy Maximum Relevance [28], gain ratio [29] etc. fall into this category.

2. Distance based: Normally the methods of this type are used for two-class problems, although they can be extended for multiclass problems. If a feature X causes more difference in the conditional probability of the class label than feature Y, then X is more preferable. One well known technique from this category is ReliefF [30].

3. Dependence based: These methods use the ability to predict one variable from another variable. We can measure the correlation between a feature and class label using various dependence measures. Chi-square [31], Fisher score [32] etc. are some well-known methods of this category.

2.3) Classifier combination:

The main concept of classifier combination evolved from the idea that each classifier operates in different way to obtain the final classification result. In case of using only one classifier, the output of the selected classifier is computed, and only this will affect the resulting decision. So, selecting only one classifier may not be the ideal since potentially valuable data can be lost by discarding the results of the other classifiers. In order to avoid this kind of loss of information, outcome of the various classifiers are considered for making the final decision. In [19], Nagi et al. have used classifier combination technique in microarray dataset. They have analysed the result of individual classifier with the combination technique like bagging and boosting. Doss et al. in [20] have come up with an idea of classifier combination technique using dynamic entropy based solution for sentence segmentation. In [21], Rohlfing et al. have proposed two approaches that assess the performance of the individual classifiers and combine the result by weighting them according to their estimated performance.

Kittler et al. in [33] have proposed various techniques to combine the prediction capabilities of different classifiers. On the basis of the outputs of different classifiers used in the combining process, we can divide the classifier combination techniques broadly in three types [34]:

1. Abstract level: Where individual classifier only outputs a label. Examples of combiners which only require this information are, for instance, voting combiners like Boosting and Bagging.

2. Rank level: Each classifier not only provides the best predicted level but also the ranks list of the labels based on the probability. Combiners like Borda count rule uses this type of information.

3. Measurement level: In this case, every classifier outputs the measurement (or probability) for each label. The probabilities obtained from different classifiers are aggregated to calculate the final result. Combiners like – sum, product etc. adopt this type of information.

Depending on the architecture of the combination techniques, the classifier combination methods can be of three types [34] :

1) Serial: Classifiers are cascaded in a linear sequence.

2) Parallel: In parallel model, classifiers are invoked independently and the results are combined. Most methods in the literature are parallel model [34].

3) Hierarchical: In hierarchical model, classifiers are combined following a tree like structure similar to decision tree. Each tree node can be associated with complex classifiers needful of large number of features. High flexibility and efficiency in exploiting the discriminant capabilities of various types of features are the prime advantages of this method.

Let us consider M classifiers, C classes ($w_1, w_2, w_3, \dots, w_c$) and a pattern (feature) Z that generates the feature vector x_i for classifier i. Posteriori probability for a class j given by i^{th} classifier is denoted as $p(w_j/x_i)$. Different techniques for combining classifiers are obtained by Bayes theorem and certain hypothesis [33]:

Some of the well-known combination technique found in the literature are product rule, sum rule, majority voting, weighted sum and weighted product. A brief description, mainly the formulation of the above mentioned methods are given below.

Product Rule: It assigns $Z \rightarrow w_j$ [33] if the equation 21 holds. Product rule is very sensitive toward low probability value.

$$p^{1-M}(w_j)\prod_{i=1}^{M} p(w_j|x_i) = \max_k p^{1-M}(w_k)\prod_{i=1}^{M} p(w_k|x_i) \dots\dots\dots\dots (21)$$

Sum Rule: In case of sum rule, we assign $Z \rightarrow w_j$ [33] when the given relation in equation 22 becomes true. Sum rule provides average performance in the cases where majority does not provide accurate result.

$$(1-M)p(w_j) + \sum_{i=1}^{M} p(w_j|x_i) = \max_k[(1-M)p(w_k) + \sum_{i=1}^{M} p(w_k|x_i)] \dots\dots\dots\dots\dots (22)$$

Majority Voting: This is used when classifiers are combined on decision level, not on the score level. We assign $Z \rightarrow w_j$ [33] in those cases where equation 23 holds. It provides average performance when majority provide wrong estimation. Unlike other combine techniques, majority voting only takes care of decision outcomes rather than a combination of a posterori probabilities [19].

$$\sum_{i=1}^{M} \Delta_{ji} = \max_k \sum_{i=1}^{M} \Delta_{ki} \dots\dots\dots\dots\dots (23)$$

Where,

$$\Delta_{ki}=\begin{cases}1\ if\ p(w_k|x_i)=\max_j p(w_j|x_i)\\ 0\ otherwise\end{cases}$$

Weighted Sum: The formulation of weighted sum is given in the equation 24. We assign $Z \rightarrow w_j$ [35] if equation 23 turns out to be true. Weighted sum rule provides promising result only if the weights are assigned precisely, however, calculating appropriate weights is not an easy task [19].

$$(1-M)p(w_j)+\sum_{i=1}^{M}\alpha_i p(w_j|x_i)=\max_k\left[(1-M)p(w_k)+\sum_{i=1}^{M}\alpha_i p(w_k|x_i)\right] \ldots\ldots\ldots\ldots (24)$$

We have to find $\alpha_i, i = 1\ to\ M$, in such a way that $e=\frac{1}{N}\sum_{k=1}^{N}\eta(\theta_k)$, where

$\eta(\theta_k)=\begin{cases}0\ if\ \beta_k=Z_k\\ 1\ otherwise\end{cases}$ is minimized. β_k is the true class pattern and $Z_k\ and\ \theta_k$ are assigned by our model.

Weighted Product: Weighted product is formulated as equation 8. Assign $Z \rightarrow w_j$ if the given equation holds.

$$p^{1-M}(w_j)\prod_{i=1}^{M}p(w_j|x_i)^{\alpha_i}=\max_k p^{1-M}(w_k)\prod_{i=1}^{M}p(w_k|x_i)^{\alpha_i} \ldots\ldots\ldots\ldots\ldots (25)$$

α_i is same as above mentioned in equation 24.

3. Proposed Method:

The proposed method is divided into two sections. The first section describes the justification of the proposed approach. The subsequent section explains the model design of our proposed approach.

3.1) Justification of the proposed method:

We have used both sequence and structure based features. From the literature, we have taken 62 sequence based features and 17 structure based features. Furthermore, we have introduced 8 structure based features. The prediction accuracy of protein classes highly depends on the selection of appropriate features. There may be many not-so-informative and/or redundant features leading to over fitting of the classifier model which in turn cause many misclassification. Moreover, feature selection gives us a scope to assess the relevance of the features to the dataset. This we consider in the first stage of PSSC, i.e., feature modelling. We have shown how we achieve high accuracy with proper selection of features in section 4, and on the other hand, accuracy decreases when misleading/redundant features are considered.

Now comes the second stage i.e., classification. To classify, or to properly identify a pattern, we need to maximize certain criteria, normally related to the pattern recognition problem. This is mostly done by comparing the present pattern recognition methods and selecting the best among them. Different classifiers perform the task of classification differently. But some recognition errors made by a method, can be resolved by another method. If we combine their abilities by some means, obviously the prediction accuracy can be increased. If the chosen classifiers produce complementary results (i.e. if a classification model wrongly classifies a pattern but another one properly classifies the same pattern), and the chosen classifiers are properly combined, then most of the patterns can be classified properly.

3.2) Model Design:

The proposed model is divided into three main parts – feature extraction, feature selection and classifier combination. The first section deals with the feature extraction methods which are used in the present work. The second section describes the feature selection methods. The final section describes the used classifiers and the classifier combination techniques. The pictorial representation of the proposed model is shown in figure 1.

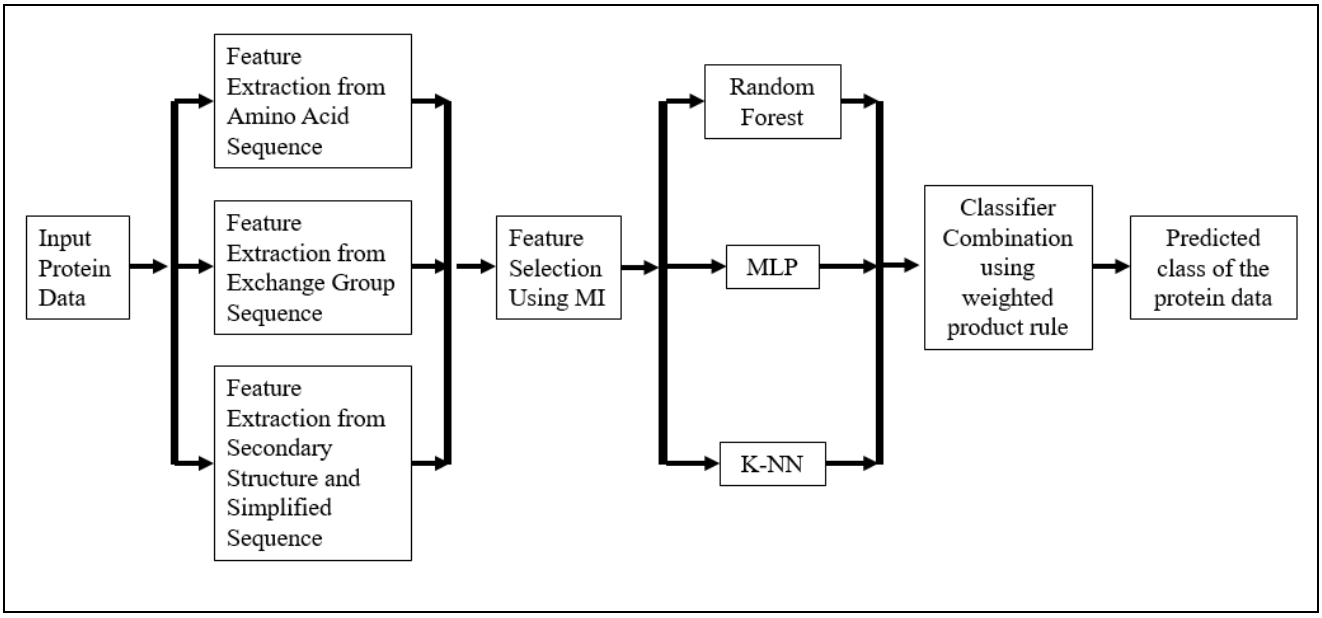


Figure 1: Illustration of the proposed PSSC model.

3.2.1) Feature Extraction

The feature extraction methodologies are performed on the four sequences of a protein data – amino acid sequence, exchange group sequence, secondary structure sequence and simplified sequence. The detailed description of the feature extraction techniques are discussed in section 2.1.

3.2.2) Feature Selection:

We have used Mutual Information in the next stage for selecting features. MI is an entropy based filter method which can detect non-linear relationship between feature and class label [26]. This gives an edge to MI compared to other traditional filter methods. According to information theory, entropy measures the uncertainty of a variable. Entropy [26] of a variable X is obtained using the equation 26.

$$H(X) = -\sum_{x \in X} p_x \log p_x \ldots\ldots\ldots\ldots (26)$$

Where, p_x is considered as the probability of x.

Now, conditional entropy of $X\ given\ Y$, is estimated by the following equation 27 [25]:

$$H(X|Y) = \sum_{x \in X, y \in Y} p_x \log\left(\frac{p(y)}{p(x,y)}\right) \ldots\ldots\ldots\ldots (27)$$

Conditional entropy implies the information needed to describe the outcome of the variable X given another variable Y. MI between two variables X and Y, is the measure of mutual dependence between the two variables which is formulated in equation 28.

$$MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \ldots\ldots\ldots\ldots\ldots\ldots (28)$$

Where $p(x,y)$ is the joint probability of $X\ and\ Y$.

3.2.3) Classifier Combination:

The various combination techniques are explained, however, in the present work, weighted product is selected as the final classification technique. Some of the reasons behind choosing this are the following. It is beneficial to weight each of the classifiers so that the final ensemble reflects the knowledge of the reliability of each of the classifiers. In other words, one can assign different weights to different classifiers in order to achieve a more satisfactory ensemble of classifiers. In case of majority voting, the classifiers are combined on decision level which ignores the confidence score for each classifier. Unlike that, weighted product works on posterori probabilities. For those reasons, weighted product is selected as the final combination technique.

To obtain the benefit of classifier combination, we have selected three classifiers wisely which function in different ways. Three selected classifiers are RF, KNN and MLP - RF is tree based, KNN is feature similarity based and MLP is neural networks based classifiers. Since they have completely different working principles, so it can be concluded that they provide complementary information about the patterns they classify, and can be applied in our classifier combination technique. So, we have combined the outcomes of RF, KNN and MLP using weighted product.

4. Results and Discussion:

The first section describes the dataset used in this work to validate our method. The subsequent sections contain the detailed study of the accuracy obtained in various datasets using our method and an elaborative comparative study with state-of-the-art methods.

4.1) Dataset description:

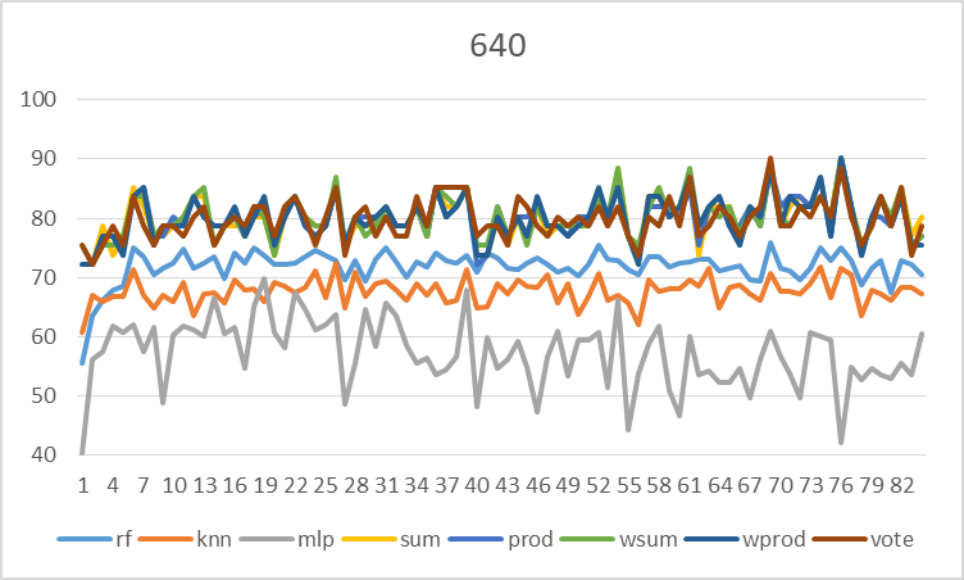
To evaluate the proposed model, we have used four benchmark datasets with low sequence similarity (< 40%), namely 640, 1189, 25pdb and fc699. The number of proteins present in each dataset is shown in Table 2. 1189 dataset has originally 1189 protein structure sequences. But after removing redundant sequences, it has 1092 sequences. Total number of samples in each class for individual dataset is shown in Table 2.

Table 2: Detailed description of the dataset used in our work.

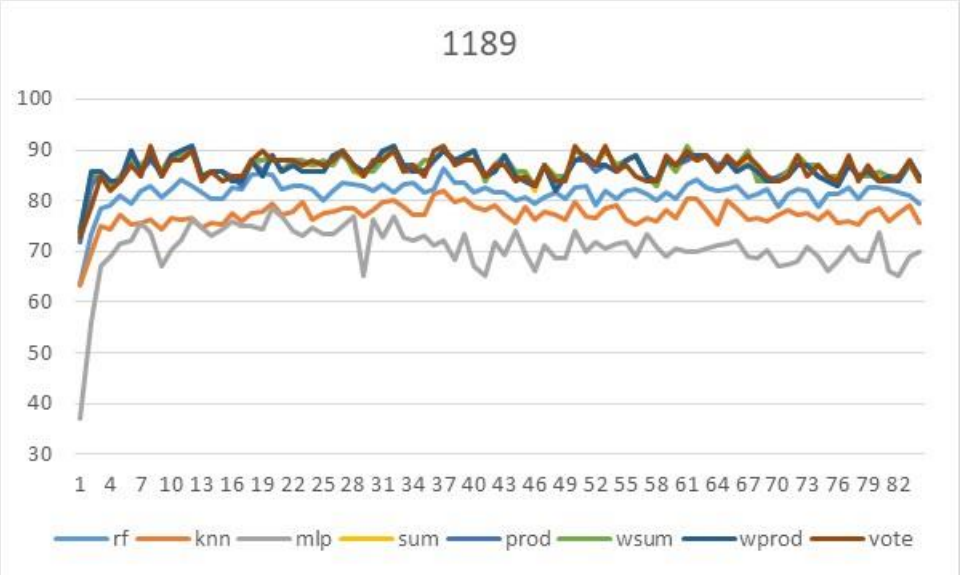
Dataset	**All-α**	**All-β**	**α+β**	**α/β**	**Total**
640 [18]	138	154	171	177	640
1189 [18]	223	294	334	241	1092
25pdb[7]	443	443	346	441	1673
fc699[7]	130	269	377	82	858

` 4.2) Results:

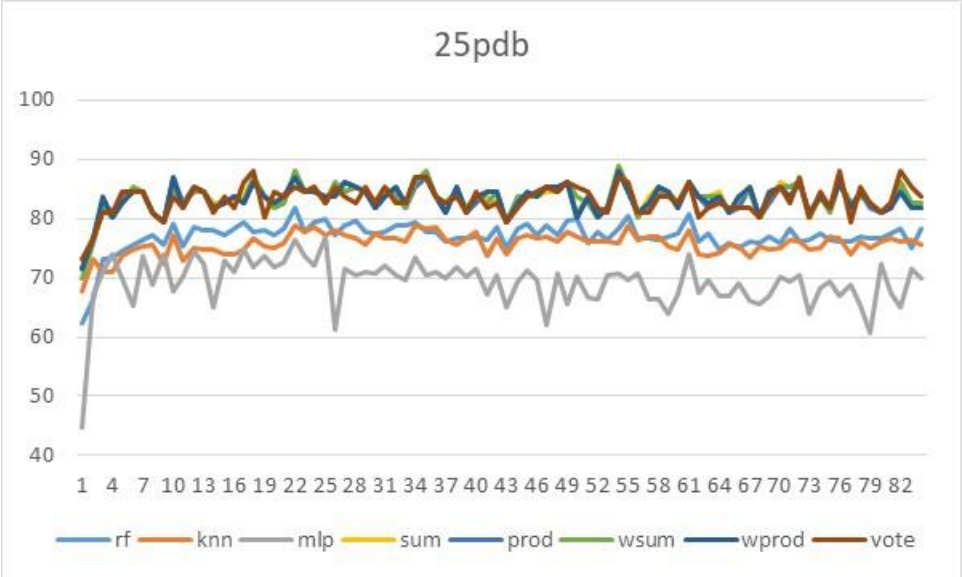
The following graphs (Graph 1, 2, 3, 4) show accuracy vs. number of features for different classifier combination techniques for each dataset. We have shown the graphs for the chosen classifiers as well as all the combination techniques like sum, product, majority voting, weighted sum and weighted product. It can be observed that the accuracies obtained in classifier combination techniques outperform the performance of individual classifier. Rather, it is achieved by less number of features, which indicates the rest features do not contribute much to classification or they may even be misleading indicated by decreasing accuracy.



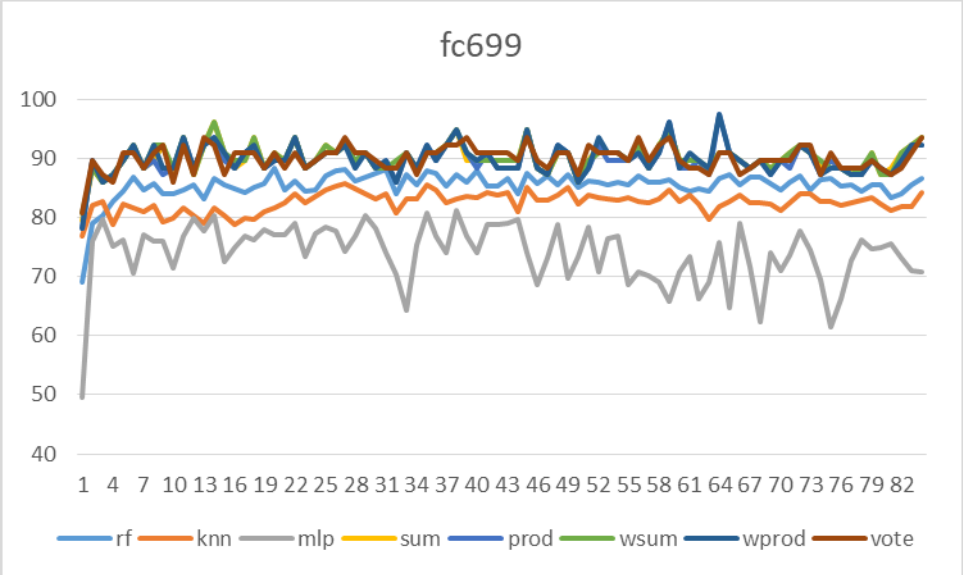
Graph 1: Accuracy vs. Number of features for 640 dataset using various methods.



Graph 2: Accuracy vs. Number of features graph for 1189 dataset using various methods.



Graph 3: Accuracy vs. Number of features for 25pdb dataset using various methods.



Graph 4: Accuracy vs. Number of features for fc99 dataset using various methods.

4.3) Comparison:

In the result section, the proper use of weighted product rule for classifier combination is justified theoretically. In this section the same is compared with weighted sum, sum, product and majority voting to uphold the use of it experimentally for each of the four datasets. Besides, we have compared the outcomes of our model with different protein structural class prediction models available in the literature. The detailed comparative study for 640, 1189, 25pdb and fc699 have appeared on Table 3, Table 4, Table 5 and Table 6 respectively. For the mentioned tables OA signifies *Overall Accuracy.*

Table 3: Comparison of accuracy (in %) of our proposed method with various state-of-the-art methods and mentioned classifier combination techniques on the 640 dataset:

Model	**All-α**	**All-β**	**α+β**	**α/β**	**OA**
RKS-PPSC [15]	89.10	85.10	71.40	88.10	83.10
PSIPRED [18]	93.72	84.01	66.39	83.53	83.44
cc-2&3 [40]	76.92	81.25	83.87	94.73	84.51
[18]	77.52	81.57	**93.27**	85.57	86.20
[41]	92.0	86.4	88.1	74.9	85.0
Sum	**93.75**	**91.67**	86.67	77.78	86.89
Prod	**93.75**	**91.67**	81.25	82.35	86.89
Voting	86.67	87.5	82.35	**100**	**88.52**
Wsum	**93.75**	**91.67**	86.67	77.78	86.89
Wprod	**93.75**	**91.67**	81.25	82.35	86.89

Table 4: Comparison of accuracy (in %) of our proposed method with various state-of-the-art methods and some classifier combination techniques which follow the same processing technique till feature selection in the 1189 dataset:

Model	**All-α**	**All-β**	**α+β**	**α/β**	**OA**
RKS-PPSC [18]	89.10	85.10	71.40	88.10	83.10
PSIPRED [18]	93.72	84.01	66.39	83.53	83.44
cc-2&3[40]	72.49	82.65	77.24	**93.04**	82.56
[18]	74.35	84.15	85.47	92.61	84.42
[10]	92.4	84.4	84.4	73.4	83.6
[37]	92.3	87.1	87.9	65.4	83.5
[39]	92.4	87.4	82.0	71.0	83.2
[3]	93.7	84.0	83.5	66.4	82.0
Sum	96.15	**96.55**	92	89.47	**93.94**
Prod	96.15	93.33	**95.83**	84.21	92.93
Voting	**96.3**	96.43	92	84.21	92.93
Wsum	**96.3**	93.33	**95.83**	88.89	**93.94**
Wprod	96.15	93.33	**95.83**	84.21	92.93

Table 5: Comparison of accuracy (in %) of our proposed method with various state-of-the-art methods and some classifier combination techniques which follow the same processing technique till feature selection in the 25pdb dataset**:**

Model	**All-α**	**All-β**	**α+β**	**α/β**	**OA**
LLSC-PRED[14]	75.20	67.50	62.10	44.00	62.20
AAD-CGR [13]	64.30	65.00	65.00	61.70	64.00
AADP-PSSM [15]	83.30	78.10	76.30	54.40	72.90
AAC-PSSM-AC [42]	85.20	81.30	73.70	55.20	73.90
[6]	86.10	80.80	80.60	60.10	76.70
[7]	92.70	78.90	71.90	74.50	79.79
[10]	95.7	80.8	82.4	75.5	83.7
[37]	92.3	83.7	81.2	68.3	81.4
[39]	95.0	85.6	81.5	73.2	83.9
[3]	95.0	81.3	83.2	77.6	84.3

[41]	91.9	87.8	80.2	79.3	84.4
Sum	93.33	**93.33**	90.91	85.29	90.52
Product	**96.55**	**93.33**	**91.3**	85.29	**91.38**
Voting	96.43	90	82.14	**86.67**	88.79
Wsum	93.33	**93.33**	90.91	85.29	90.52
Wprod	**96.55**	**93.33**	**91.3**	85.29	**91.38**

Table 6: Comparison of accuracy (in %) of our proposed method with various state-of-the-art methods and some classifier combination techniques which follow the same processing technique till feature selection in the fc699 dataset:

Model	**All-α**	**All-β**	**α+β**	**α/β**	**OA**
CBF-PSSE [17]	84.62	91.45	93.90	34.50	86.01
PBF-PSSE [17]	88.46	81.41	88.86	80.49	85.66
[7]	96.40	92.50	**95.10**	65.10	91.61
[41]	96.9	89.2	89.4	89.0	90.4
Sum	**100**	**100**	87.8	**100**	93.59
Prod	**100**	**100**	87.8	**100**	93.59
Voting	**100**	88.0	94.59	**100**	93.59
Wsum	**100**	**100**	87.8	**100**	93.59
Wprod	**100**	**100**	92.31	83.33	**94.87**

5. Conclusion and Future Scope:

In this paper, a machine learning based model has been proposed for secondary structure classification of proteins into four classes - all-α, all-β, $\alpha+\beta$ and α/β. We have used both sequence based and structure based features for the classification. At first, a filter based feature selection method called MI is applied to eliminate all redundant features, and then the selected features are fed to three different classifiers – RF, KNN, and MLP. In the final stage, some popularly used classifier combination approaches found in the literature are used to unite the decision made by the said classifiers, among them weighted product is found the best one. The proposed model has been compared with some state-of-the-art methods and it has been observed that our model outperforms the others.

From the graphs present in result section (graph 1, graph 2, graph 3, graph 4), we can clearly see the fluctuating accuracy. A decrease in accuracy at a point and increase in the next point indicate that though a feature can be misleading, it can result in good accuracy when present with some other features. That is where feature subset selection beats feature ranking. In our study, we have used feature ranking. In future we aim for doing the same work with feature subset selection. We can also work with features of higher order i.e., $a_j, j \geq 2$ and $e_i, i \geq 3$. In the next stage, the optimum feature

subset will be fed to the classifier combination techniques. About the second stage, we can use other classifiers like SVM [43], Naïve Bayes [44] etc.

Reference:

[1] B. Panda and B. Majhi, "A novel improved prediction of protein structural class using deep recurrent neural network," *Evol. Intell.*, vol. 0, no. 0, p. 0, 2018.

[2] M. Levitt and C. Chothia, "Structural patterns in globular proteins," *Nature*, vol. 261, no. 5561, p. 552, 1976.

[3] S. Ding, S. Zhang, Y. Li, and T. Wang, "A novel protein structural classes prediction method based on predicted secondary structure," *Biochimie*, vol. 94, no. 5, pp. 1166–1171, 2012.

[4] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *J. Mol. Biol.*, vol. 247, no. 4, pp. 536–540, 1995.

[5] L. A. Kurgan and L. Homaeian, "Prediction of structural classes for protein sequences and domains — Impact of prediction algorithms , sequence representation and homology , and test procedures on accuracy," vol. 39, pp. 2323–2343, 2006.

[6] A. Dehzangi, K. Paliwal, A. Sharma, O. Dehzangi, and A. Sattar, "A Combination of Feature Extraction Methods with an Ensemble of Different Classifiers for Protein Structural Class Prediction Problem," pp. 1–14, 2013.

[7] S. Bankapur, "Protein Secondary Structural Class Prediction using Effective Feature Modeling and Machine Learning Techniques," *2018 IEEE 18th Int. Conf. Bioinforma. Bioeng.*, pp. 18–21, 2018.

[8] S. Costantini and A. M. Facchiano, "Prediction of the protein structural class by specific peptide frequencies," *Biochimie*, vol. 91, no. 2, pp. 226–229, 2009.

[9] K. Chou, "Progress in Protein Structural Class Prediction and its Impact to Bioinformatics and Proteomics," no. 1, pp. 423–436, 2005.

[10] L. Zhang, X. Zhao, and L. Kong, "A protein structural class prediction method based on novel features," *Biochimie*, vol. 95, no. 9, pp. 1741–1744, 2013.

[11] T. Liu, X. Zheng, and J. Wang, "Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile," *Biochimie*, vol. 92, no. 10,

pp. 1330–1334, 2010.

[12] K. Chou and Y. Cai, “Predicting protein structural class by functional domain composition,” vol. 321, pp. 1007–1009, 2004.

[13] J. Yang, Z. Peng, Z. Yu, R. Zhang, V. Anh, and D. Wang, “Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation,” vol. 257, pp. 618–626, 2009.

[14] L. Kurgan, K. Cios, and K. Chen, “SCPRED : Accurate prediction of protein structural class for sequences of twilight-zone similarity with predicting sequences,” vol. 15, pp. 1–15, 2008.

[15] T. Liu and C. Jia, “A high-accuracy protein structural class prediction algorithm using predicted secondary structural information,” *J. Theor. Biol.*, vol. 267, no. 3, pp. 272–275, 2010.

[16] J. Yang, Z. Peng, and X. Chen, “Prediction of protein structural classes for low-homology sequences based on predicted secondary structure,” no. January, 2010.

[17] Q. Dai, Y. Li, X. Liu, Y. Yao, Y. Cao, and P. He, “Comparison study on statistical features of predicted secondary structures for protein structural class prediction : From content to position,” 2013.

[18] W. Bao, D. Wang, and Y. Chen, “Classification of Protein Structure Classes on Flexible Neutral Tree,” vol. 5963, no. c, pp. 1–12, 2016.

[19] S. Nagi and D. K. Bhattacharyya, “Classification of microarray cancer data using ensemble approach,” *Netw. Model. Anal. Heal. Informatics Bioinforma.*, vol. 2, no. 3, pp. 159–173, 2013.

[20] M. Magimai-Doss, D. Hakkani-Tur, O. Cetin, E. Shriberg, J. Fung, and N. Mirghafori, “Entropy based classifier combination for sentence segmentation,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*, 2007, vol. 4, p. IV-189.

[21] T. Rohlfing, D. B. Russakoff, and C. R. Maurer, “Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation,” *IEEE Trans. Med. Imaging*, vol. 23, no. 8, pp. 983–994, 2004.

[22] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[23] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[24] F. Rosenblatt, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms,” CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

[25] M. Ghosh, S. Adhikary, K. K. Ghosh, A. Sardar, S. Begum, and R. Sarkar, “Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods,” *Med. Biol. Eng. Comput.*, Aug. 2018.

[26] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, no. July 1928, pp. 379–423, 1948.

[27] H. Uğuz, “A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm,” *Knowledge-Based Syst.*, vol. 24, no. 7, pp. 1024–1032, 2011.

[28] N. S. Mohamed, S. Zainudin, and Z. A. Othman, “Metaheuristic approach for an enhanced mRMR filter method for classification using drug response microarray data,” *Expert Syst. Appl.*, vol. 90, pp. 224–231, 2017.

[29] A. G. Karegowda, A. S. Manjunath, and M. A. Jayaram, “Comparative study of attribute selection using gain ratio and correlation based feature selection,” *Int. J. Inf. Technol. Knowl. Manag.*, vol. 2, no. 2, pp. 271–277, 2010.

[30] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Proceedings of the ninth international workshop on Machine learning*, 1992, pp. 249–256.

[31] X. Jin, A. Xu, R. Bie, and P. Guo, “Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles,” in *International Workshop on Data Mining for Biomedical Applications*, 2006, pp. 106–115.

[32] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in *Advances in neural information processing systems*, 2006, pp. 507–514.

[33] J. Kittler, I. C. Society, M. Hatef, R. P. W. Duin, and J. Matas, “On Combining Classifiers,” vol. 20, no. 3, pp. 226–239, 1998.

[34] J. Fierrez, A. Morales, R. Vera-rodriguez, and D. Camacho, “Multiple classifiers in biometrics . Part 1 : Fundamentals and review,” vol. 44, pp. 57–64, 2018.

[35] J. Kittler, “Combining Classifiers : A Theoretical Framework,” pp. 18–27, 1998.

[36] J. Cao and L. Xiong, “Protein Sequence Classification with Improved Extreme Learning Machine Algorithms,” vol. 2014, 2014.

[37] M. J. Mizianty and L. Kurgan, “Modular prediction of protein structural classes from

sequences of twilight-zone identity with predicting sequences," vol. 24, pp. 1–24, 2009.

[38] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol.*, vol. 292, no. 2, pp. 195–202, 1999.

[39] S. Zhang, S. Ding, and T. Wang, "High-accuracy prediction of protein structural class for low-similarity sequences based on predicted secondary structure," *Biochimie*, vol. 93, no. 4, pp. 710–714, 2011.

[40] W. Bao, Y. Chen, and D. Wang, "Prediction of protein structure classes with flexible neural tree," no. May, 2017.

[41] L. Liu, J. Cui, and J. Zhou, "A Novel Prediction Method of Protein Structural Classes Based on Protein Super-Secondary Structure," pp. 54–62, 2016.

[42] T. Liu, X. Geng, and X. Zheng, "Accurate prediction of protein structural class using auto covariance transformation of PSI-BLAST profiles," pp. 2243–2249, 2012.

[43] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[44] M. E. Maron, "Automatic indexing: an experimental inquiry," *J. ACM*, vol. 8, no. 3, pp. 404–417, 1961.