JJBS

Volume 13, Number 4, December 2020
ISSN 1995-6673
Pages 483 - 492

Jordan Journal of Biological Sciences

Prediction of Protein Secondary Structure from Amino Acid Sequences by Integrating Fuzzy, Random Forest and Feature Vector Methodologies

Sivagnanam R. Mani Sekhar[1, 2,*] , Siddesh G. Matt [2] and Sunilkumar S Manvi[1]

[1]*School of Computing & Information Technology, Reva University, Bengaluru, Karnataka,* [2]*Department of Information Science & Engineering, Ramaiah Institute of Technology, Bengaluru, India.*

Received: October 17, 2019; Revised: November 27, 2019; Accepted: December 28, 2019

Abstract

The study of protein structure is an important research area in computational biology. Several algorithms have been used to predict the structure of the protein, but still it is a time consuming and challenging task as the dataset is increased day by day. The Proposed Work Enhanced Fuzzy Random Forest (EFRF) scrapes information from various websites allowing us to get class labels for our unsupervised data set. Afterward, Feature Vectors have been used to generate a transformed view of the protein sequences, which are then used as input to the proposed EFRF classifiers for prediction of secondary structure like alpha, Beta sheet, and Coil. Subsequently, Nave Bayers (NB), Support Vector Machine (SVM) classifiers have been used to compare and contrast precision and accuracy. The experiment shows that the proposed solution EFRF achieves an accuracy of 96% compared to the SVM 75 % and NB 41%.

Keywords: Protein, Machine learning, Protein secondary structure, Random forest, Fuzzy, Bioinformatics and Feature Vector.

1. Introduction

Proteins are the multifaceted and essential building blocks for living organisms. They show a vital role in the development of cell body, structure, and functions. Different Proteins are made up of different structures, resulting in unique functionality. They are made from peptide bonds and amino acids sequence. Amino acid is made up of the carboxylic and amino groups. They help in the formation of a peptide bond by releasing H2O. Mainly, proteins are formed from twenty different amino acids, and each twenty amino acids have been represented by a Universal coding scheme. When each of these amino acids joins together, they form a chain called polypeptide chain. Out of twenty different amino acids, nine are marked as vital for the human body, as nine of these amino acids need to be taken as supplements. As drug development requires a particular knowledge of the binding sites of candidate compounds, a well-predicted structure helps in the computational screening and optimizing candidate compounds. Identifying the mechanism by which a protein functions and how it folds is of great curiosity for researchers and developers.

Prediction of protein function and structure is a challenging task in bioinformatics (Kumar, 2015). A suitable structure prediction mechanism helps the researcher in finding the essential functions. In the past, different computation techniques have been developed for prediction of primary, secondary, tertiary, and quaternary structure (Quan et al., 2016; Brender and Zhang, 2015; Carnevali et al., 2003; Lee et al., 1996; Mandal and Jana, 2012; Benítez and Lopes, 2010). While extracting protein secondary structure, selection of the right algorithm and feature extraction techniques are very important. Many statistical methods have been proposed, but their computation performance is not sufficient for huge and multifaceted biological datasets. Meanwhile, as the data size is increasing by date, still it is a challenging task for prediction of protein secondary structure, resulting in incessant growth of high throughput analytical model. However, identification of protein secondary structure helps in the understanding of protein tertiary structure and also offers perception of protein function.

Currently, researchers are working on Machine learning and template-based learning methodologies for structure prediction. A multi-classifier can perform quite better than the single classifier and allows it to handle a complex and huge dataset. Subsequently, extend its support in handling missing data and reducing noise level Bonissone et al.(2008a, 2008b).

Random Forest works on the principle of the decision tree. It is one of the most popular algorithms in bioinformatics, as it is comparatively easy to use and robust against imperfect records for experimental biological problems (Yang, 2010; Smolarcyzk and Stapor, 2018; Cao et al., 2016; Jo and Cheng, 2014). In (Bankapur and patil, 2018), authors have used SXGbg and CE approaches for feature extraction. After successful retrieval of the essential features, the author has used different machine learning for classification like KNN, SVM, and RF for structure prediction. As sequence finding and resolution play an important role in protein structure, prediction author (Hu, et al., 2018) used loop with 2-15

* Corresponding author e-mail: manisekharsr@gmail.com.

amino acids and matrix score to cover more area for protein structure computation. (Li et al., 2011) has used RF algorithm for secondary structure prediction. They have developed a method called ProC_S3, working on an RR contact map, and with top 600 features. (Jai and Hu, 2011) Proposed a method for predicting β -hairpin motifs using the RF algorithm by incorporating several properties. There result shows that RF performs better compared to other algorithms.

In proposed work author has presented a method called EFRF for the identification of protein secondary structure. The work focuses on the creation of multi-classifier by incorporating random forest (RF) technique (Breiman, 2001), subsequently feature vector has been generating. For inadequate values, random forest is constructed using fuzzy techniques. The integration of Random forest with fuzzy logic makes system more dynamic and helps in overcoming ambiguous data (Bonissone et al., 2010), but the accuracy of system also depends upon the selected features; if the selected features are not effective even the good algorithm can result in poor accuracy. In this work we have selected features which have direct correlation with 3D structure of the protein; later, these features are combined in a matrix to improve the efficiency of the vector.

This paper is organized as follows: section II provides a brief discuss about the different approaches used in protein structure prediction. In section III, a proposed EFRF methodology and its architecture for protein secondary structure are proposed and discussed. Section IV illustrates the implementation of the proposed work. Later, Dataset and achieved result are discussed in section V. Finally, section VI discusses the conclusion part.

2. Related Work

As evolutionary and syntactical based evidence is not adequate for extraction of valuable feature from the protein sequences, (Sudha et al., 2018) proposed an Enhanced Artificial Neural Network (ANN) for prediction of protein Structural Class and Fold Recognition. They have used physic chemical information and FCS technique for feature extraction by integrating FCS methodology. The result shows that Enhanced ANN performs well in RDD, EDD, TG, DD datasets. Here computation is based on the limited features. Performance can be further increased by introducing evolutionary and syntactical feature and also by latest feature extraction techniques.

A bio-inspired computing approach for prediction of protein secondary structure is presented in (Yavuz et al., 2018). Here the computation is performing in two different stages; in first stage, they used clonal selection algorithm (CSA) for data training. Later in the second stage, they used a deep learning technique called multilayer perceptron for classification. The result shows that dataset trained from CSA performs well. The proposed solution aims to improve by introducing fuzzy logic in classification.

(Hasic et al., 2017) uses a multi neural network method and consensus function for prediction of proteins secondary structure. These methods result in lowering the hypothesis space, which in turn helps in finding the best result. They have focused more on identifying and prediction of alpha helices and beta sheets from the CB513 and 25PDB datasets. (Kathuria et al., 2018) uses a machine learning techniques for identification of unknown protein structures. They used Amide frequencies and RF classifier for prediction of protein secondary structure. The result shows that the model performs better in amides dataset. ROC curve and area have been used for validation of model. Multi classification techniques can be involved during secondary protein structure prediction to achieve more accuracy.

The work of (Zhang et al., 2016) used chaos game concept for prediction of protein secondary structure from the given sequence of proteins. The accuracy of structure is depending upon the likeness of protein data. This issue can lead to the unwanted structure prediction. They used a time series technique, feature vector of 36 dimension and CGR to overcome this issue. The prediction accuracy can be further increased by incorporating Random tree learning techniques.

3. Proposed Solution

Machine learning techniques have been universally used in Bioinformatics domain and other related areas. They provide a platform for developers in creation of automatically learning system with the capability of improvement from experience. Decision Tree is one of the most widely used analytic methods in Machine learning. Collection of Decision trees is known as a Random Forest. However, RF can work effectively when applied to large dataset, whereas they can be unstable when training value deals with small distribution. To overcome this issue, fuzzy logic has been incorporated in tree construction (Bonissone et al., 2010).

Steps followed in the proposed approach to predict secondary protein structure are as follows:

Step 1: *Parallelized collection and analysis of data:* Protein Data is collected parallel and stored in local driver for structure prediction. Later cleaning and optimization procedure is applied by removing DNA and RNA from the stored dataset.

Step 2: *Design and generation of feature vectors:* This step illustrates the process of identification & selection of necessary features; subsequently it combines in a matrix to improve the efficiency of vector.

Step 3: *Prediction of protein secondary structure using EFRF techniques:* The classifier takes a vector as an input, subsequently integrating fuzzy concept and RF for protein structure prediction.

The steps given above are showed in Figure 1. Here data is extracted from protein data bank parallel. Later, the protein dataset is cleaned and analysis for efficient computation of the model. Subsequently, feature vector is computed from the given sequence data by incorporating 3 compositions, 3 to 15 transition values with the given frequency and protein length. Finally, the protein structure is predicted using proposed EFRF, SVM & NB.

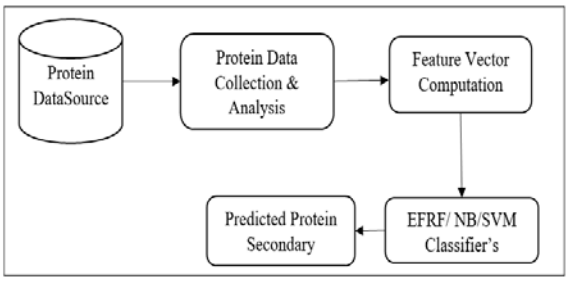

Figure 1. Proposed EFRF Architecture for protein structure prediction

3.1. Parallelized collection and analysis of data

Selenium is used for extracting protein data from websites by processing the HTML Web Page and extracting data for manipulation to a local storage. Once the protein sequences are stored locally, the application can run without an internet connection. Protein Data Banks contain millions of sequences and the whole process was parallelized using Java and multithreading to increase computational performance. The sequence data that is collected from the internet had a mix of protein and non-protein data (such as DNA and RNA) which was filtered and cleansed as per requirement. Figure 2, describes the Data preprocessing stages. Here the unstructured data is extracted from protein data bank, subsequently cleaned with required attributes using parallel computation and drives. Later Post processed data is stored in a local machine with features and labels.

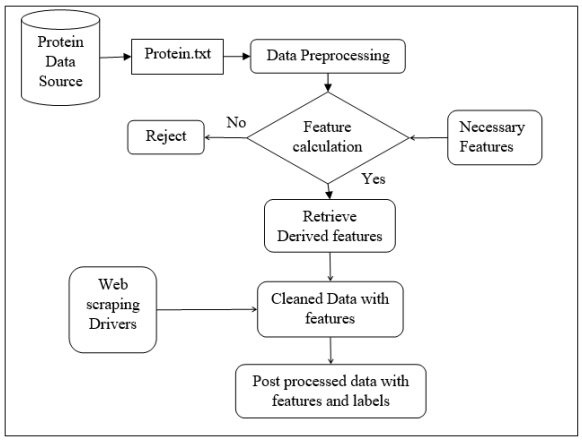

Figure 2. Proposed parallelized data proposing system design in EFRF

3.2. Design and Generation of feature vectors

Design and Development of Feature Vector is a measurable entity that is used to describe a feature of the objects. Selection of feature vectors is key factor in improving the performance of the predictor algorithm. For structure prediction, the features selected need to have direct correlation to the 3D structure of the protein. Here, four different features are selected for structure prediction: these are Hydrophobicity, Polarizability, Polarity and Van der Waals Volume. These features are combined in a matrix to improve the efficiency of the vector.

3.2.1. Generating Feature Vector:

The proposed work involves a single dimensional feature vector that is unique to each protein sequence. A vector of length 105 is generated for each protein sequence based on values of certain physical and chemical properties. Here, sequence of protein is converted from a heterogeneous size to a feature vector of homogeneous size.

The twenty amino acids are segregated into three different categories based on their values corresponding to the properties. The physical and chemical properties taken into account are Hydrophobicity, Polarizability, Polarity, and Van Der Waals Volume. These 20 amino acids are categories into three clusters (Chinnasamy et al., 2005) corresponding to their values and properties, as shown in Table 1.

Table 1. Secondary structure classes attribute with corresponding classes

Attribute	Cluster 1	Cluster 2	Cluster 3
Class	Coil	Helix	Strand
Hydrophobicity	C, F, I, L , M V, W	D, E, K, N, Q, R	A, G, H, P, S, T, Y
Polarizability	F, H, K, M, R, W, Y	A, C, D, G, P, S, T	E, I, L, N, V, Q
Polarity	D, E, H, K, N, Q, R	C, F, I, L, M, V, W, Y	A, G, S, T, P
Van der Waals Volume	F, H, K, M, R, W, Y	A, D, G, S, T	C, E, I, L, N, P, Q, V

The feature vectors 23 made up of separate individual feature vectors that are as follows:

Composition Feature Vector (Comp$_i$): The composition feature vector is computed as follows

$$Comp_i = ((Tg_i)/SeqLen)100; \quad (1)$$

In equation 1, "Comp$_i$" represents the percent composition of each group, "Tg" tells group total and "SeqLen" denotes the sequence length

Transition Feature Vector (Trs$_{ij}$): *Trs$_{ij}$* shows the group occurrence percentage foe group i to j for the value of one, two, and three.

Transition Feature Vector (Tij): Tij is characterized by the t frequency percen with which group i is followed by group j or vice versa where i, j takes the values 1, 2 or 3 respectively.

Distribution Feature Vector (DFV): The DFV comprises five values from three groups that represent the sections of the given sequence value, also specify the first residue of a given group is located, and where other are located.

Percentage Frequency Feature Vector (PFFV): The PFFV defines the length as 20 and lists out the different percentage of these 20 amino acids in the given protein sequence data.

3.2.2 Calculations for Feature Vectors:

This section shows the computation of feature vector values based on their properties. A property can have 3 compositions, 3 to 15 transition values with the given frequency and protein length. The section below elucidates the calculation of the feature vector.

- 3 composition values with 4 properties: 3 x 4 = 12
- 3 transition values with 4 properties: 3 x 4 = 12
- 15 transition values with 4 properties: 15 x 4 = 60
- Percentage frequency of each amino acid: 20 x 1 = 20
- Length of protein: 1
- Feature Vector: 12 + 12 + 60 + 20 + 1 = 105

3.3. Prediction of protein secondary structure using Enhanced Fuzzy Random Forest methodology

The proposed EFRF system architecture is showed in Figure 3; it illustrates the process of protein secondary structure prediction. The stored dataset is converted from amino acid sequence to feature vector by using selected features and feature vector algorithm. Subsequently in parallel, protein data is extracted in FASTA format with amino acid sequence & feature vector. Then, these optimized values are given as an input with required features. Afterwards, proposed EFRF, SVM, & NB classifiers are applied on it for protein secondary class prediction. The workflow of the system is a monolithic architecture. The data set comprises eight thousand cleansed sequences stored locally. The user interface takes a protein sequence as input and based on the feature vectors a scoring matrix is generated. Subsequently, the Feature Vector is used as input for the classifiers.

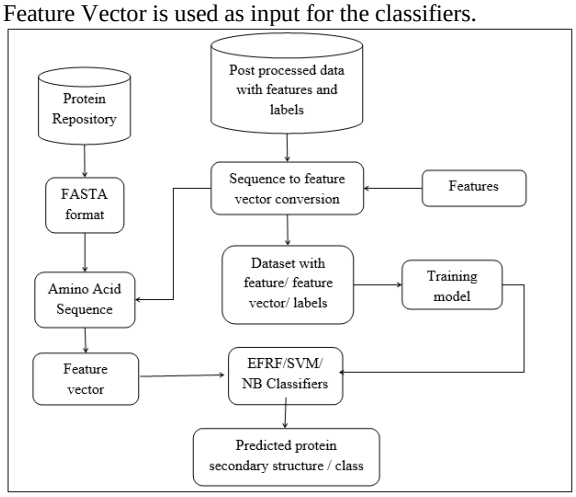

Figure 3. Proposed EFRF Data Post-Processing System Design for secondary structure prediction

(Jang, 1994; Janikow, 1998; K M Lee et al., 1999; Pulkkinen and Koivisto, 2008) illustrate the different methodologies in which fuzzy logic is combined positively with decision tree. According to RF (Breiman, 2001) progression of each node step by step and subsequently during the tree construction process each node will get split randomly with the available attributes. Finally, new process of split will perform based on random selection.

Fuzzy Random forest (FRF) (Bonissone et al., 2010) can be used for protein structure prediction; the proposed work uses fuzzy partition method for each inner node (INod) of the trees (Cadenas, Garrido and Martınez, 2009). $INod_1$, $INod_2$, $INod_3$,…. $INod_f$, are the state values generated from membership method (μ_{INodi}) as shown in equation 2. Here the construction of the tree size is a key point.

$$\forall x \in E \sum_{i=1}^{INodf} \mu_{INod\,i} \quad (x) = 1 \qquad (2)$$

According to (Bonissone, 2010; Chinnasamy, 2005) split of tree depends on numerical values and membership values ($\chi_{Tr,\ Nd}$); these values help in finding the tree (Tr) with node (Nd). The membership value helps in the splitting process of the tree (Tr). If the membership degree value is more than zero, the split will happen according to equation 3. Here 'CN' is represented as child node, 'Tr' is represented as tree

$$\chi_{(Tr,\ CN)}\ (\text{sample}) * \chi_{(Tr,\ node)}\ (\text{sample}) * \mu_{\ fuzz\_set\_prt}\ (\text{sample}) \qquad (3)$$

Fuzzy random forest matrix (MFRF) 11 is used in classification problems; it classifies data to a given class and also generates the state of the node leaf (Le) and tree (Tr), subsequently also supporting in decision making 11 as shown in equation 4

$$\text{FuzzyAggre}(\text{class}_i,\text{MFRF}) = \sum_{tree=1}^{\text{Total tree}} \sum_{leaf_{index}=1}^{\text{no.\_of leafnode}} (MFRF)_{\text{tree,leafIndex,clas}} \qquad (4)$$

4. Implementation

In this work, two different classified NB (Robles et al., 2004) and SVM (Cortes and Vapnik, 1995; Cai et al. 2002) are compared with the proposed EFRF algorithm. The proposed model accepts primary protein sequences from the user; subsequently, Feature Vectors are generated from the entered sequence and then analyzed to predict class labels for each entered sequence. The protein sequences vary in length and can be categorized based on various physical and stereo chemical properties; these properties determine the feature vector generation. The attributes used to describe the sequence in the article include Polarity, Hydrophobicity, Polarizability and Van der Waals Volume.

The proposed EFRF model is categorized into three sub modules: *initial, intermediate and final modules*

Initial sub-Module: Initially used an unsupervised data set from the Protein Data bank which needed a lot of data preprocessing before actually using the data present in the data set. The data set contained about four hundred thousand sequences which were a mixture of DNA, RNA and Protein sequences. Firstly, extract the protein sequences from the data set since those records were the only meaningful records for this project. The other parameters were Sequence ID, Sequence Type, Sequence Length, Sequence Name and the Primary Sequence itself. This model requires the Class Label, the Secondary Structure of Protein. This unsupervised dataset had to be converted into a supervised dataset.

Intermediate Sub-Module: It displayed the results of the sequence with their frequency percentages. The percentage in the output is responsible for categorizing them into the final secondary structure.

The dataset (PDB) used had about eight thousand protein sequences to handle. So, this manual process took around two hours for two hundred sequences. Further, Selenium driver is used for web scraping. This allowed us to automate the retrieval process and increase efficiency of the conversion. The total time it took to retrieve and analyze all the sequence and to create a class label for each and every sequence out of the eight thousand protein sequences was about seven hours. Since this was not optimal performance, we converted the script so as to run on six different threads on six different Google Chrome tabs so as to achieve parallelization. The whole retrieval and data aggregation process took about two hours.

Final Module: After the conversion of the unsupervised dataset into a supervised data set, three classifiers namely NB, SVM and EFRF were used for predicting the secondary structure of protein. These classifiers are operating extremely differently from each other.

The attributes sent to these classifiers were the feature vectors and the class label with the classifiers for prediction. These classifiers take up to 75% of the dataset for training, and the rest for testing to provide an insight of how accurate the predictions turn out to be. Here, Feature Vector is used to create a Vector having consistent values to the chemical properties of the protein sequence. Algorithm 1 focuses on overall protein secondary structure prediction; the algorithm takes protein primary sequence as an input and generates corresponding protein secondary structure. Lines 2 to 4 compute the necessary feature vector score by incorporating secondary structure attributes as shown in algorithm 2. Finally, feature vector score is input to the classifier for protein secondary prediction as shown in line 5 to 7.

Algorithm 1: Proposed EFRF with Feature Vector Scoring

Input: Primary Protein Sequence
Output: Protein secondary structure
1. **begin**
2. **Compute** feature vector score
3. Apply Algorithm 2
4. **return** vector score
5. **Generate** protein secondary structure
6. Apply Algorithm 3
7. **return** protein class
8. **end**

Algorithm 2 shows the proposed Feature Vector Scoring process. The algorithm takes Protein sequence as an input and subsequently computes Composition, Transition and distribution values are as shown in line number 2 to 7. Line 8 computes the frequency of the given sequence of each array. Finally, line 12 to 14 computes the vector score for the given sequence of array. Similarly, the process is applied to the different amino acid sequences.

Algorithm 2: Proposed Feature Vector Scoring process

Input: Primary Protein Sequence
Output: A vector score for the Sequence
1. Procedure find Feature Vector (sequenceSQ)
2. **begin**
3. **for** each property **do**
4. divideAA --> 3Groups
5. calcComposition(SQ) --> return compArr
6. calcTransition(SQ) --> return transArr
7. calcDistribution(SQ) --> return distArr
8. **for** each AA **do**
10. calcf requencyof AAinSQ
11. **return** PFArr
12. **for** each in array **do**
13. fv[]+ = array
14. **return** fv
15. **end**

Algorithm 3 discusses the procedure for protein structure prediction such as Helix, Coil and strand class. Here features vector is given as an input; subsequently for the corresponding feature vector and protein sequence a necessary structure will be predicted. Line 1 focuses on the calling part of the training model for protein class prediction; subsequently random features is selected in line 3. Next, line 4 stores the node split values to a variable called "d". Later, line 6 performs the node split using fuzzy; afterwards, line 8 builds a RF using fuzzy concept, consequently predicting the class using training model. The predicted Helix, Coil, & Strand is stored in a target as shown in line 9. Further, for each outcome of the random tree, vote is calculated as show in line 11. Finally lines 12 to 17 compute the votes for each attributes resulting in protein structure prediction.

Algorithm 3: Enhanced Fuzzy Random Forest

Input: Feature Vector of size 1 x 105
Output: Predicted Protein Structure
1. TrainingModel ()
2. **begin**
3. k =RandomSelectFeatures() (From m features, k << m)
4. d = FuzzyBestSplit(k)
5. i = 0
6. **while** ((i < n) && (n! =1)) **do**
7. d = FuzzyBestSplit(d)
8. BuildFuzzyForest(d, i)
9. PredictionFromTrainedModel()
10. targetOutcome[n] = RandomDecisionTree(k) {AlphaHelix, RandomCoil, EtendedStrand}
11. votes[m] =⇒ CalculateVotes(targetOutcome[i]){i = 0, 1, n =size}{m << n}
12. max = 0
13. i = 0
14. **while** (i < n) **do**
15. if (max < votes[j] {j = 0, 1, ..m})
16. max = votes[j]
17. **return** max
18. **end**

5. Results and Discussions

5.1. Dataset

A sample dataset of protein sequence extracted from the protein data bank (PDB) is shown in Figure 4. The dataset is split into three parts: Initial Dataset, Intermediate Supervised Dataset, and Supervised Dataset. The section below illustrates the step by step process of data conversion.

5.1.1. Initial Dataset

This data is directly obtained from the data bank. It requires preprocessing in order to be useful in sequence prediction. The files "pdbseq.res.txt" and "pdbNS.txt" focus on the necessary sequences after removing RNA and DNA sequences.

```
>101m_A mol:protein length:154  MYOGLOBIN
MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRVKHLKTEAEMKASEDLKKHGVT
>102l_A mol:protein length:165  T4 LYSOZYME
MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTKSPSLNAAAKSELDKAIGRNTNGVITKDEAEKLF
>102m_A mol:protein length:154  MYOGLOBIN
MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKTEAEMKASEDLKKAGVT
>103l_A mol:protein length:167  T4 LYSOZYME
MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTKSPSLNSLDAAKSELDKAIGRNTNGVITKDEAEK
>103m_A mol:protein length:154  MYOGLOBIN
MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKTEAEMKASEDLKKAGVT
>1041_A mol:protein length:166  T4 LYSOZYME
MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTKSPSLNAAKSAAELDKAIGRNTNGVITKDEAEKL
>1041_B mol:protein length:166  T4 LYSOZYME
```

Figure 4. Sample Protein dataset from PDB (Protein Data Bank)

pdb_seqres.txt: It contains the following information: Protein Sequence ID, Class, Molecule Type, Protein, Length of the sequence, Name of the sequence, Primary Protein Sequence as displayed in figure 5.

101m_A mol: protein length: 154 MYOGLOBIN
MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRVKHLK
TEAEMKASEDLKKHGVTVLTALGAILKKKGHHEAELKPLAQSHATKHKIPIK
YLEFISEAIIHVLHSRHPGNFGADAQGAMNKALELFRKDIAAKYKELGYQG

Figure 5. pdb sequence.txt file

Useful attributes: Protein Sequence ID, Type, Primary Protein Sequence

Cleansed Dataset : Considers only Protein Sequence after discarding RNA and DNA sequences from the data bank.

pdb_NS.txt: It contains the following information: Name of the sequence, Primary Protein Sequence as showed in figure 6.

MYOGLOBIN

MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRVKHLKTE
AEMKASEDLKKHGVTVLTALGAILKKKGHHEAELKPLAQSHATKHKIPIKYLEFI
SEAIIHVLHSRHPGNFGADAQGAMNKALELFRKDIAAKYKELGYQG

Figure 6. Sample pdb NS.txt file

5.1.2. Intermediate Supervised Dataset:

Intermediate dataset is an unsupervised dataset obtained directly from the Protein Data Bank. It does not have a class label for the sequence of amino acids and various other parameters as shown in "contenttest.txt" file.

ContentTest.txt : It contains the following information: Name of the sequence, Primary Protein Sequence, Class Label (Secondary Structure Prediction), Percentage of the Secondary Structure Percentage in the Sequence as displayed in figure 7.

MYOGLOBIN
MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRVKHLKTE
AEMKASEDLKKHGVTVLTALGAILKKKGHHEAELKPLAQSHATKHKIPIKYLEFI
SEAIIHVLHSRHPGNFGADAQGAM NKALELFRKDIAAKYKELGYQG

Alpha helix: 75.32

Figure 7. Sample ContentTest.txt file

5.1.3. Supervised Dataset:

The supervised dataset has been created by web scraping from enter here the website from where we scraped and the class label obtained is used as the class label for the amino acid sequence which along with the sequence length forms our supervised dataset. As shown in "contentTest.csv" file.

ContentTest.csv : It contains the following information: Name of the sequence, Primary Structure Sequence, Class Label (Secondary Structure Prediction) as presented in figure 8.

MYOGLOBIN,MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDR
VKHLKTEAE
MKASEDLKKHGVTVLTALGAILKKKGHHEAELKPLAQSHATKHKIPIKYLEFISEAII
HVLHSRHPG NFGADAQGAMNKALELFRKDIAAKYKELGYQG, Alpha helix

Figure 8. Sample ContentTest.csv file

5.2. Feature Vector Input

The feature vector of length 105 is constructed by taking into the properties of the amino acid sequence which includes Hydrophobicity, Polarity, Polarizability and Van Der Waals Volume. Individual Composition, Transition and Distribution values for each of the properties are generated and together with the sequence length and percentage frequency of each amino acid are combined to constitute the feature vector. An array of floating point values that are fed as attributes to the classifiers are displayed in Figure 9.

[0.25609756, 0.42682928, 0.31707317, 0.40243903, 0.34146342, 0.25609756, 0.37804878, 0.3292683, 0.29268292,.., 0.085365854, 0.085365854, 0.0121951215, 0.0121951215, 0.0121951215, 0.024390243, 82.0]

Figure 9. An array of floating point values

Now the sequence, after validation, is converted into a vector which consists of floating point values. This vector is called a feature vector. The feature vector is then provided as input to the classifiers for prediction purposes. The use of Feature Vector is to generate a Vector containing corresponding values to the chemical properties of the protein sequence.

5.3. Classifier's for prediction of protein secondary structure

In this work, authors use three different classifiers for prediction of protein secondary structure. Initially, NB (Chinnasamy et al., 2005) classifier is used for structures prediction, as it performs well in multi class environment. Subsequently, as the dataset contains large dimension feature vector; authors have also used a SVM (Cortes and Vapnik, 1995; Cai, 2002) for prediction of structures. Finally, the proposed model EFRF is integrated using Fuzzy concept, RF and Feature vector concepts for prediction of protein structures. This section focuses on the accuracy measures, precision measures and Normalized Confusion Matrix for Proposed EFRF, SVM & NB.

5.3.1. Accuracy-Measures for Proposed EFRF, SVM & NB

Figure 10 & Figure 11 shows the two graphs generated from the dataset with splits of 60% of the dataset used for training in the first case and 75% of the dataset used for training in the other case. The Accuracy of NB classifier varies within the range size of approximate 0.27 with the variation in the size of datasets from 1000 to 8000. The observed variations are against the expected behavior that the increase in size of training dataset should lead to increase in accuracy. This concludes that our Feature Vector does not work well for the NB Classifier.

The Accuracy of SVM varies within range size of 0.02 with the variation in the size of dataset. This behavior is

parallel to the expected behavior as accuracy increases with the increase in size of dataset. Although there may be a slight decrease in the values of accuracy with the increase in the dataset size, this may be neglected due to the reason that this small variation can sometimes occur as a result of overfitting. Overall, the classifier has a good accuracy of 76%. Thus, SVM is a suitable classifier for the feature vector.

The Accuracy of proposed EFRF varies within range size of 0.03 with the variation in the size of dataset. The graphs show almost a linear relationship. This behavior sharply coincides with our expected behavior as accuracy increases with the increase in size of dataset. As a result, the classifier does an excellent work in predicting all the three structures. Overall, the classifier has a good accuracy of 96%. Thus, EFRF is the most suitable classifier for the feature vector.

Table 2 and figure 10 show the dataset of size 1000, 2000, 3000,4000, 5000, 6000, 7000, 8000 with 60:40 data split proposed EFRF give an accuracy of 92.9%, 95.2%, 94.4%,95.0%,95.0%, 96.5%, 95.7%, 95.5% whereas SVM give an accuracy of 77.3%, 76.9%, 75.8%, 75.5%, 78.0%, 77.6%, 76.0%, 75.8% and NB gives an accuracy of 77.5%, 49.8%, 48.5%, 45.1%, 45.3%, 46.5%, 46.5%, 41.8%. The result show that proposed EFRF perform better compare to other algorithms.

Table 2. Accuracy-Measure: Comparative Results among SVM, NB & the proposed methods EFRF with 60:40 Data split.

Dataset size	Enhanced Fuzzy Random Forest (EFRF)	Support Vector machine (SVM)	Naïve Baye's (NB)
1000	93	77.3	77.5
2000	95	76.9	49.8
3000	94.4	75.8	48.5
4000	95.1	75.5	45.1
5000	95	78	45.3
6000	96.5	77.6	46.5
7000	95.7	76	46.5
8000	95.5	75.8	41.8

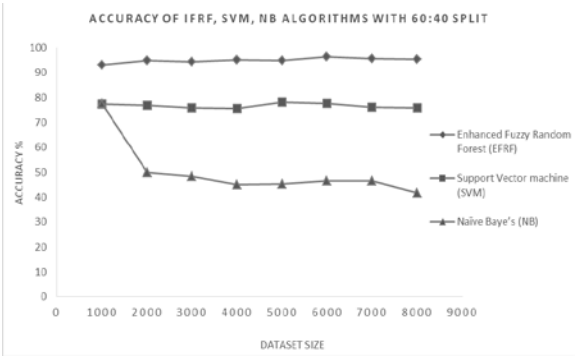

Figure 10. Graph with Accuracy-Measure: Outcomes Among SVM, NB & proposed methods EFRF with 60:40 Data split.

Similarly, table 3 and figure 11 illustrate the dataset of size 1000, 2000, 3000,4000, 5000, 6000, 7000, 8000 with 75:25 data split the proposed EFRF give an accuracy of 92.9%, 96.0%, 97.0%, 96.1%, 96.8%, 96.2%, 96.1%, 96.2% whereas SVM give an accuracy of 76.0%, 78.8%, 75.3%, 76.7%, 77.5%, 78.0%, 75.5%, 75.3% and finally NB gives an accuracy of 74.8%, 49.0%, 44.1%, 42.6%, 45.4%, 49.9%, 49.0%, 41.5%. The result show that proposed EFRF performs better compare to other algorithms.

Table 3. Accuracy-Measure: Comparative Results among SVM, NB & proposed methods EFRF with 75:25 Data split.

Dataset size	Enhanced Fuzzy Random Forest (EFRF)	Support Vector machine (SVM)	Naïve Baye's (NB)
1000	92.8	76	74.8
2000	96	78.8	49
3000	96.9	75.3	44.1
4000	96.1	76.7	42.6
5000	96.8	77.5	45.4
6000	96.2	78	49.9
7000	96.2	75.5	49
8000	96.3	75.3	41.5

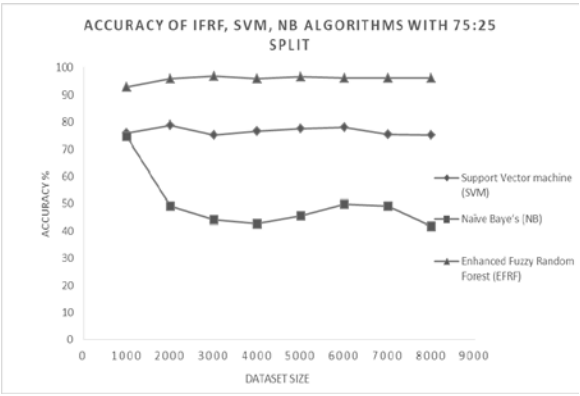

Figure 11. Graph with Accuracy-Measure: Outcomes Among SVM, NB & proposed methods EFRF with 75:25 Data split.

The section 5.3.1 provides a brief discussion on how the accuracy and results varied across the classifiers. In this NB, classifier yielded an accuracy of approximately 47%. This low accuracy is the consequence of poor prediction of sequences which are Random Coil. In order to find a better classifier which suits the given feature vector, SVM is incorporated. SVM produced an overall accuracy of 78%, hence showing competence with our developed feature vector. But still, the classifier fails when predicting the structure of most of the Alpha Helix sequences. Finally, the EFRF classifier gives an excellent accuracy of 96% showing a strong capability of success with the Feature Vector.

5.3.2. Precision-Measures for Proposed EFRF, SVM & NB

The precision measures for the Proposed EFRF, SVM & NB are shown in table 4. The Precision value of the proposed EFRF system is 96.2% whereas for the SVM and NB is 72.9% & 41.5% for the dataset of size 8000. This shows that proposed EFRF performs well when compared with SVM & NB. Similarly, Table 4 and figure 12 illustrate the dataset of size 1000, 2000, 3000,4000, 5000, 6000, 7000, 8000 with 60:40 data split the proposed EFRF give a precision of 92.9%, 95.2%, 94.4%, 95%, 95%, 96.5%, 95.7%, 95.5% whereas SVM give a precision of 74.5%, 75.7%, 73%, 72.6%, 75.7%, 74.7%, 72.7%, 73.7% and later NB gives a precision of 77.5%, 49.8%, 48.5%, 45.1%, 45.3%, 46.5%, 46.5%, 41.8%. The result tells that proposed EFRF performs well compared to other algorithms.

Table 4. Precision -Measure: Comparative Results among SVM, NB & proposed methods EFRF with 60:40 Data split.

Dataset size	Enhanced Fuzzy Random Forest (EFRF)	Support Vector machine (SVM)	Naïve Baye's (NB)
1000	92.9	74.5	77.5
2000	95.2	75.7	49.8
3000	94.4	73	48.5
4000	95	72.6	45.1
5000	95	75.7	45.3
6000	96.5	74.7	46.5
7000	95.7	72.7	46.5
8000	95.5	73.7	41.8

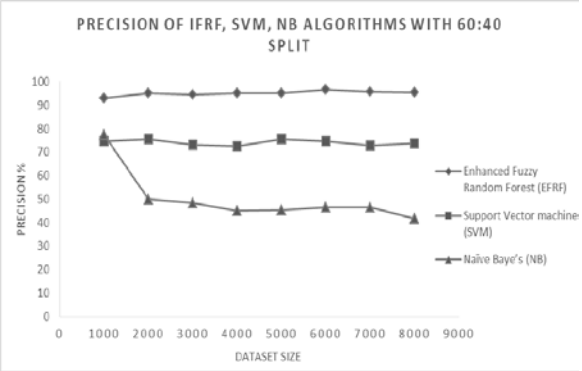

Figure 12. Graph with Precision -Measure: Outcomes Among SVM, NB & proposed methods EFRF with 60:40 Data split.

Table 5 and figure 13 discuss the dataset of size 1000, 2000, 3000,4000, 5000, 6000, 7000, 8000 with 75:25 data split the proposed EFRF give a precision of 92.9%,96%, 97%, 96.1%, 96.8%, 96.2%, 96.1%, 96.2, % whereas SVM give a precision of 73.8%, 78.3%, 72.9%, 74.2%, 75.1%, 75.8%, 71.9%, 72.9% and later NB gives a precision of 74.8%, 49%, 44.1%, 42.6%, 45.4%, 49.9%, 49%, 41.5%. The result tells that proposed EFRF performs well compared to other algorithms.

Table 5. Precision -Measure: Comparative Results Among SVM, NB & proposed methods EFRF with 75:25 Data split. Best Results Are Bolded Per Row

Dataset size	Enhanced Fuzzy Random Forest (EFRF)	Support Vector machine (SVM)	Naïve Baye's (NB)
1000	92.9	73.8	74.8
2000	96	78.3	49
3000	97	72.9	44.1
4000	96.1	74.2	42.6
5000	96.8	75.1	45.4
6000	96.2	75.8	49.9
7000	96.1	71.9	49
8000	96.2	72.9	41.5

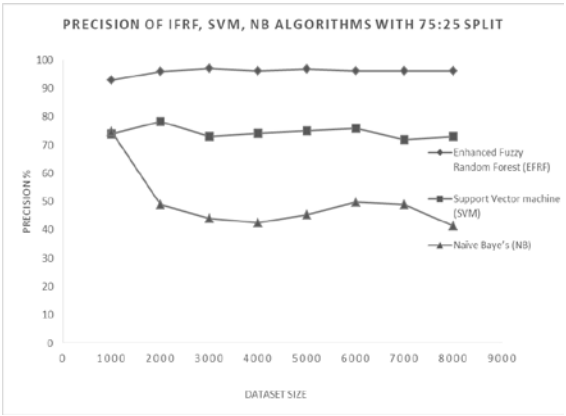

Figure 13. Graph with Precision -Measure: Outcomes among SVM, NB & the proposed methods EFRF with 75:25 Data split.

5.3.3. Normalized Confusion Matrix for Proposed EFRF, SVM & NB

The Confusion Matrix for the NB Classifier 25 with the predicted values normalized between 0 and 1 is shown in Figure 14. From the matrix, it can be concluded that the classifier does a good work in predicting 87% of Alpha helix structure correctly. Also, the classifiers performance in predicting the Extended Strand sequences correctly is above average at 69%. However, when it comes to predicting the Random Coil sequences, the classifier Performance is very bad with only 3 percent of the testing sequences predicted correctly.

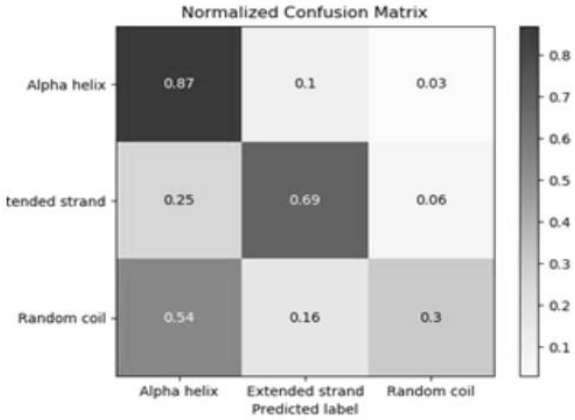

Figure 14. Normalized Confusion Matrix for Naive Bayes

SVM is very effective in cases when the number of dimensions is very large. The Confusion Matrix for the SVM Classifier with the predicted values normalized between 0 and 1 is shown in Figure 15. From the matrix, it can be analyzed that the classifier does an excellent work in predicting the structure of Extended Strand and Random Coil correctly for almost all the testing sequences. However, the performance of the classifier fails when it comes in predicting the structure of Alpha helix sequences. Consequently, only 28% of the testing sequences are predicted correctly as Alpha helix. With an overall accuracy of 76%, SVM was a good classifier for the feature vector.

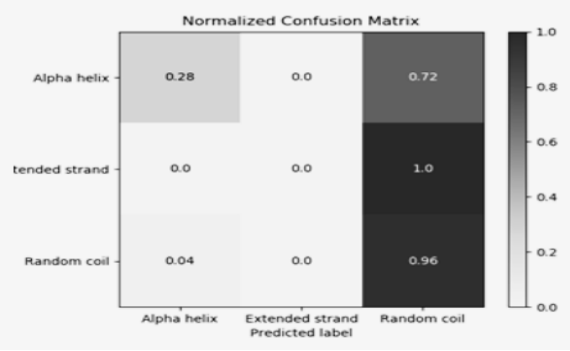

Figure 15. Normalized Confusion Matrix for Support Vector Machines

In the prospect for a better classifier to improve the accuracy, the proposed work EFRF integrates Fuzzy concept with RF Classifier. The Confusion Matrix for the EFRF Classifier with the predicted values normalized between 0 and 1 is shown in Figure 16. From the confusion matrix, it can be inferred that the classifier does an excellent work in correctly predicting all the three structures for almost all the testing sequences.

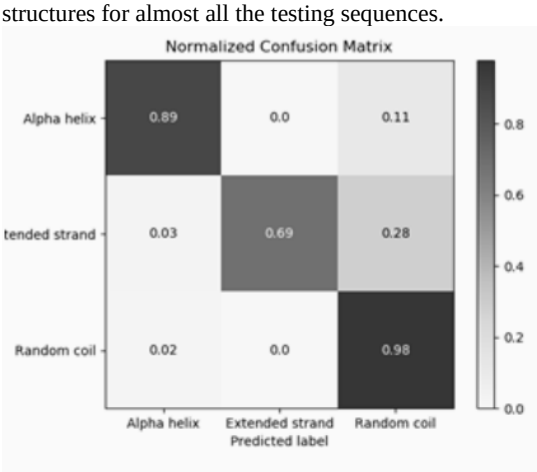

Figure 16. Normalized Confusion Matrix for Enhanced Fuzzy Random Forest

5.4. Evaluation of models

There are different measures used in the validation of predicted values. The section below discusses the performance measure using Root Mean squared error, and Correlation.

Root Mean squared error (RMSE): RMSE is used to compute the error rate of a model. It calculates the differenence between expected result (ER) and observed result (OR) with its square root of the same. RMSE computation process is showed in equation 5, here 'ER' expected result, 'OR' observed result and 'k' is the number of instance.

$$\text{RMSE}=\sqrt{\frac{\sum_{i=1}^{k}(OR\ i-ER\ i)\text{^}2}{k}} \quad (5)$$

Correlation: Correlation helps in computation of statistical relation between expected result (ER) and observed result (OR). Correlation computation process is shown in equation 6, here 'ER' expected result, 'OR' observed result and 'k' is the number of instance.

$$\text{Correlation} = \frac{\sum_{i=1}^{k}(ERi-Mean\ ERi)(ORi-Mean\ ORi)}{\sqrt{\sum_{i=1}^{k}(ERi-Mean\ ERi)^2\ \sum_{i=1}^{k}(ORi-Mean\ ORi)^2}} \quad (6)$$

Table 6 shows the computed RMSE, correlation and predicted accuracy for the proposed model EFRF, NB and SVM. The result shows that EFRF outperform when compared with the NB and SVM for the value of RMSE and Correlation. Here the proposed EFRF model has the lowest RMSE value of 0.33 whereas for SVM is 0.99 and NB is 1.5. Similarly, EFRF has the highest correlation value of 0.95 whereas for SVM is 0.55 and NB is 0.38.

Table 6. Performance Measure among NB, SVM, & proposed method EFRF

Model	RMSE	Correlation	Accuracy %
Naïve Baye's (NB)	1.5	0.38	41.5
Support Vector Machine (SVM)	0.99	0.55	75.3
Enhanced Fuzzy Random Forest (EFRF)	0.33	0.95	96.3

6. Conclusion

The proposed work EFRF uses a Machine Learning model in prediction of the two-dimensional structures of the protein from their amino acid sequences. The model takes the primary protein sequence as input and outputs the structural class through which the protein folds. These can be used in various Drug developments, which in particular requires knowledge of the binding sites of the candidate compounds, a well-predicted structure helps in the computational screening and optimizing candidate compound. The unsupervised dataset to train our Machine Learning model is a linear chain of amino acids that forms the primary structure of the protein. The extracted information from various websites allows us to get class labels for our unsupervised data set. Subsequently generates a unique Feature Vectors for each protein sequence based on Polarity, Hydrophobicity, Polarizability, and Van der Waals Volume. Later, various classification models are used to analyze and predict class labels for each sequence. For the given dataset, NB classifier yielded an accuracy of approximately 47%. This low accuracy is the consequence of poor prediction of sequences, which are Random Coil. Whereas SVM produced an overall accuracy of 78%, hence shows competence with our developed feature vector. But still, the classifier fails when predicting the structure of most of the Alpha Helix sequences. Later, the proposed EFRF classifier gives an accuracy of 96%, showing a strong capability of success with the developed Feature Vector. Finally, the model is further validated using RMSE and correlation. The computed result shows that the EFRF model has the lowest RMSE value of 0.33, whereas for SVM is 0.99, and NB is 1.5. Similarly, EFRF has the highest correlation value of 0.95, whereas for SVM is 0.55, and NB is 0.38.

References

Bankapur, S. and Patil, N., 2018, October. Protein Secondary Structural Class Prediction Using Effective Feature Modeling and Machine Learning Techniques. In 2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE) (pp. 18-21). IEEE.

Benítez C M V and Lopes H S. 2010. Protein structure prediction with the 3D-HP side-chain model using a master–slave parallel genetic algorithm. *Journal of the Brazilian Computer Society*, *16*(1), 69-78.

Bonissone P P, Cadenas J M, Garrido M C, and Diaz-Valladares R A. 2008b. Combination methods in a fuzzy random forest. In *2008 IEEE International Conference on Systems, Man and Cybernetics* (pp. 1794-1799). IEEE.

Bonissone P P, Cadenas J M, Garrido M C, and Dıaz-Valladares R A. 2008a. A fuzzy random forest: Fundamental for design and construction. In *Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'08)* (pp. 1231-1238).

Bonissone P, Cadenas J M, Garrido M C, and Díaz-Valladares R A. 2010. A fuzzy random forest. *International Journal of Approximate Reasoning*, *51*(7), 729-747.

Breiman L. 2001. Random forests. *Machine learning*, *45*(1), 5-32.

Brender J R, and Zhang Y. 2015. Predicting the effect of mutations on protein-protein binding interactions through structure-based interface profiles. *PLoS computational biology*, *11*(10), e1004494.

Cadenas, J. M., Garrido, M. C., & Martınez, R. (2009). Una estrategia de particionamiento fuzzy basada en combinación de algoritmos. In *Proceedings in XIII Conferencia de la Asociación Espaòola para la Inteligencia Artificial, Sevilla, Spain* (pp. 379-388).

Cai Y D, Liu X J, Xu X B and Chou K C. 2002. Prediction of protein structural classes by support vector machines. *Computers & chemistry*, *26*(3), 293-296.

Cao, R., Jo, T. and Cheng, J., 2016. Evaluation of protein structural models using random forests. arXiv preprint arXiv:1602.04277.

Carnevali P, Tóth G, Toubassi G, and Meshkat, S N. 2003. Fast protein structure prediction using Monte Carlo simulations with modal moves. *Journal of the American Chemical Society*, *125*(47), 14244-14245.

Chinnasamy A, Sung W K, and Mittal A. 2005. Protein structure and fold prediction using tree-augmented naive Bayesian classifier. *Journal of Bioinformatics and computational Biology*, *3*(04), 803-819.

Cortes C and Vapnik V. 1995. Support-vector networks. *Machine learning*, *20*(3), 273-297.

Hasic H, Buza E, and Akagic A. 2017. A hybrid method for prediction of protein secondary structure based on multiple artificial neural networks. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1195-1200). IEEE.

Hu, X.Z., Long, H.X., Ding, C.J., Gao, S.J. and Hou, R., 2018. Using random forest algorithm to predict super-secondary structure in proteins. The Journal of Supercomputing, pp.1-12.

Jang J S. 1994. Structure determination in fuzzy modeling: a fuzzy CART approach. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference* (pp. 480-485). IEEE.

Janikow C Z. 1998. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *28*(1), 1-14.

Jia, S.C. and Hu, X.Z., 2011. Using random forest algorithm to predict β-hairpin motifs. Protein and peptide letters, 18(6), pp.609-617.

Jo, T. and Cheng, J., 2014, December. Improving protein fold recognition by random forest. In BMC bioinformatics (Vol. 15, No. 11, p. S14). BioMed Central.

Kathuria C, Mehrotra D, and Misra N K. 2018. Predicting the protein structure using random forest approach. *Procedia computer science*, *132*, 1654-1662.

Kumar M. 2015. An enhanced algorithm for multiple sequence alignment of protein sequences using genetic algorithm. *EXCLI journal*, *14*, 1232.

Lee B, Kurochkina N and Kang H S. 1996. Protein folding by a biased Monte Carlo procedure in the dihedral angle space. *The FASEB journal*, *10*(1), 119-125.

Lee K M, Lee K M, Lee J H and Lee-Kwang H. 1999. A fuzzy decision tree induction method for fuzzy data. In *FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No. 99CH36315)* (Vol. 1, pp. 16-21). IEEE.

Li, Y., Fang, Y. and Fang, J., 2011. Predicting residue–residue contacts using random forest models. Bioinformatics, 27(24), pp.3379-3384.

Mandal S, and Jana N D. 2012. Protein structure prediction using 2D HP lattice model based on integer programming approach. In *Proceedings of 2012 International Congress on Informatics, Environment, Energy and Applications* (pp. 17-18).

Protein Data bank (2018). Retrieved from: https://pdb101.rcsb.org/learn/guide-to-understanding-pdb-data/primary-sequences-and-the-pdb-format.

Pulkkinen P, and Koivisto H. 2008. Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms. *International Journal of Approximate Reasoning*, *48*(2), 526-543.

Quan L, Lv Q, and Zhang Y. 2016. STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*, *32*(19), 2936-2946.

Robles V, Larrañaga P, Peña J M, Menasalvas E, Pérez M S, Herves V and Wasilewska A. 2004. Bayesian network multi-classifiers for protein secondary structure prediction. *Artificial Intelligence in Medicine*, *31*(2), 117-136.

Smolarczyk, T. and Stąpor, K., 2018. Random Forest Classifier for Early-Stage Protein Structure Prediction. Studia Informatica, 39.

Sudha P, Ramyachitra D, and Manikandan P. 2018. Enhanced artificial neural network for protein fold recognition and structural class prediction. *Gene Reports*, *12*, 261-275.

Yang, P., Hwa Yang, Y., B Zhou, B. and Y Zomaya, A., 2010. A review of ensemble methods in bioinformatics. Current Bioinformatics, 5(4), pp.296-308

Yavuz B Ç, Yurtay N, and Ozkan O. 2018. Prediction of protein secondary structure with clonal selection algorithm and multilayer perceptron. *IEEE Access*, *6*, 45256-45261.

Zhang L, Kong L, Han X, and Lv J. 2016. Structural class prediction of protein using novel feature extraction method from chaos game representation of predicted secondary structure. *Journal of theoretical biology*, *400*, 1-10.