Received: March 17, 2017

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Distributed Tree-based Ensemble Learning Approach for Efficient Structure Prediction of Protein

Leo Dencelin Xavier[1*] **Ramkumar Thirunavukarasu[2]**

[1]*Department of Computer Science & Engineering, Sathayabama University, India*
[2]*School of Information Technology & Engineering, VIT University, India*
* Corresponding author's Email: ldency6@gmail.com

Abstract: Knowledge of a protein's secondary structure, in turn, contributes to our understanding of the functions of the protein is vital to many aspects of living organisms such as those of enzymes, hormones, and structural material, etc. It also helps in designing new drugs for critical disease. In this paper, we have advocated a distributed approach to identify the Protein Secondary Structures using an ensemble method on protein primary sequences. The Ensemble based Random Forest algorithm has been adopted to build the three-way predictive model. Based on the amino acid features of each protein and decision tree parameters, the classification model allows us to assign protein structures as 'α helix', 'β sheet', or a coil. Also the proposed model is implemented in a distributed computing environment, SPARK. Experiments have been carried out using cross-validation tests on RS126 and CB513 benchmark datasets. Our results clearly confirm that ensemble approach in classifying protein secondary structures scores better accuracy with improved performance when it will be implemented in the distributed environment.

Keywords: Amino acids, Protein structure prediction, Ensemble techniques, Distributed framework, Apache SPARK.

1. Introduction

Proteins are macromolecules, and consist of combinations of amino acids in peptide linkages, that contain carbon, hydrogen, oxygen, nitrogen, and sulphur atoms. There are only 20 different types of amino acids, and they can be combined to generate an infinite number of sequences. In reality, only a small subset of all possible sequences appears in nature. In the study of proteins, there are three important attributes of proteins: sequence, structure, and function. The sequence is essentially the string of amino acids which comprises the protein. The structure of the protein is the way the protein is spread in the three dimensional space. The most important thing is the protein function. The protein function is its actual role in the specific organism in which it exists. Understanding the protein function is critical for most applications, such as drug design, genetic engineering, or pure biological research.

Machine learning is the field of making computers smarter and able to learn from the data rather than static instructions [1]. This paper specifically discusses the application of machine learning to a common problem in biology field [2]. The problem is predicting the secondary structure of proteins from its primary sequence using an efficient ensemble based technique. In machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance that could be obtained from any of the constituent learning algorithms alone. Ensemble methods are techniques that create multiple models and then combine them to produce improved results. These methods usually produce more accurate solutions than a single model would. This has been the case in a number of machine learning competitions like Kaggle, where the winning solutions used ensemble methods. Various common ensemble learning techniques are (i) Bagging: Building multiple models (typically of the same type) from different subsamples of the training dataset. In generalized bagging, we can use different learners on different population and helps us to reduce the variance error. (ii) Boosting: Building multiple models (typically of the same type) each of which learns to fix the prediction errors of a prior model in the chain. If an observation was classified incorrectly, it tries to increase the weight of this observation and vice versa. Boosting in general decreases the bias error and builds strong predictive models. (iii) Stacking: This is a very interesting way of combining models. Building multiple models (typically of differing types) and supervisor model that learns how to best combine the predictions of the primary models. This can lead to decrease in either bias or variance error depending on the combining learner we use.

Random Forests are an ensemble learning method which is a combination of tree predictors where each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest [3]. The basic principle is that a group of “weak learners” can come together to form a “strong learner”. Introducing the right kind of randomness makes them accurate classifiers and repressors [4]. Decision nodes are added to each of the trees without pruning, until classification for every instance is obtained. This divide and conquer strategy combined with the randomized attribute selection strategy automatically weights the domain pairs based on the probability that they share a particular structural level. For every domain pair, a majority voting algorithm collects the classification decision from each of the trees to make the final prediction [5].

It is appropriate to utilize the big data computing framework in the knowledge curation of protein structure prediction problem because of the following reasons: (i) Volume - The high volume nature of protein data bank (Greater than Terra Byte), since efficient structure prediction algorithms require huge volume of protein data for the improved accuracy. (ii) Velocity - The dynamic nature of amino acid sequence requires a real-time processing framework and it is being achieved through big data computing framework such as Spark. (iii) Variety - The amino acid sequence data are highly un-structured in nature. For handling such unstructured data, programming models based on big data framework are the apt choice [6]. In recent years, the open source Apache Spark project, which adopts Resilient Distributed Datasets (RDD) framework and distributed file system [7], brings bioinformatics researchers a good platform and opportunities to obtain a scalable, fault tolerance, efficient and reliable computing performance on Linux clusters. In this paper, we have proposed a distributed approach based on Apache Spark, which will work for huge amount of protein data and for predicting the secondary structures in a large scale with increased accuracy using ensemble based classifier.

The rest of the paper has been organized as follows: In section 2, protein structure prediction concepts and techniques have been discussed. The distributed ensemble classifier approach using spark has been introduced in section 3. The performance and accuracy evaluation of our proposed approach with respect to various measures are present in Section 4, where conclusion and scope for future work are accounted in Section 5.

2. Structure prediction: Concepts and techniques

A protein is made up of building blocks called amino acids. A protein is a large molecule that plays critical roles in the body, and is required for structure, function, and regulation of bodily tissues. There are hundreds of different kinds of proteins, such as enzymes, antibodies, and structural. These proteins take many different shapes and forms as well. When discussing proteins, it is important to remember that structure follows function, meaning however the molecule is shaped gives a hint about what it does in the cell. This just goes to show that knowing a structure of a protein can give important information.

These amino acids are bonded together through physio-chemical reactions to create the protein’s backbone. Once the backbone is completed, the atoms in the amino acids begin to interact with one another, and the backbone begins to fold on itself. Protein folding works in a manner similar to origami, but there are many other factors that play a role in protein folding which are outside the realm of this paper. Even though every protein is different, one similarity between all proteins is that the sequences fold up into 1 of 3 different structures, known as α helix, β sheet, and a coil. These are all of varying sizes, but the same structure. The one problem that scientists face is, determining this secondary structure of the protein sequence. It is much easier to obtain a protein’s amino acid sequence, and then use computational methods to determine the structure. However, even determining the structure accurately using computational methods is hard to do [8].

Computational methods for PSSP, mostly based on machine learning methods, can be schematically grouped in the following three families: sequence-based methods; network-based methods [9]; hierarchical ensemble methods [10]. Some methods provided predictions of a relatively small set of functional classes [11], while others considered predictions extended to larger sets, using Support Vector machines [12] , HMM Algorithm, artificial neural networks, Bayesian networks, Decision Trees or methods that combine functional linkage networks with learning machines using a logistic regression model or simple algebraic operators. Also perceptron based techniques are giving reasonable results, by automatically using the input vectors thereby reducing the feature extraction part [13].

The purpose of this paper is to present a plausible method to determine protein secondary structure by classifying amino acids to one of the three secondary structures.

3. Proposed work – Distributed ensemble classifier in Spark environment

Many attempts on predicting secondary structure on the basis of machine learning techniques use a fixed window, or some assumption that there is no randomness in the data. In order to do this, the following method on constructing the decision trees using random subset of data is proposed. K-Fold cross-validation is used in this experiment where the dataset is partitioned into K sets of equal size. We have used K=10 for our proposed work. Nine of these sets are combined and used for training, while the remaining one is used for testing. Then the process is repeated with nine different sets combined for training and so on until all the ten individual partitions have been used for testing. The final accuracy of an algorithm will be the average of the ten trials. The datasets were obtained from Protein Data Bank (PDB) Repository. The first step is to create features and select the important features from the protein sequence. The features are simply the amino acids in the protein sequence, since this is ultimately the determining factor when discussing the secondary structure of a protein. In the second step the calculation of effective parameters for the Random Forest Classifier is performed. As the final step we implement this entire approach (Step 1 and Step 2) in a distributed environment SPARK for better performance.

3.1 Feature engineering

The original training and test dataset were taken from Protein Data Bank (PDB) of Barton. It consists of 502 non-homologous protein chains with more than 83000 residues and less than 25% homology. It is generated by Cuff and Barton [14]. The proteins (2JZ6, 2RPJ and 2K2D) were removed from the dataset due to lack of sequence Information. Amino Acid sequences are commonly transformed to numerical feature vectors of an equal length and applied to machine learning algorithms. Variety of features have been extracted from the sequence and given as input. Features such as PSSM profiles, Amino Acid Factors, Physio-Chemical properties and Solvent accessibility have been used from the Input Protein Primary sequence [15]. (i) **PSSM profiles**: A 20-dimensional vectors for all residues composed a matrix called the position specific scoring matrix (PSSM) [16]. (ii) **Amino Acid Factors**: The 20 amino acids have different and specific properties, the composition of these properties of different residues within a protein can influence the specificity and diversity of the protein structure and function. (iii) **Physicochemical and biochemical properties**: These properties include polarity, hydrophobicity, secondary structure, molecular volume, and electrostatic charge. It has been obtained from AAIndex DB [17]. (iv) **Solvent accessibility**: The solvent accessibility of each amino acid 'buried' or 'exposed' are encoded as 10 and 01 respectively [18]. The features used in this work along with their dimension size are shown in Table 1.

One of the advantages of Random Forest method over other machine learning technique is that the importance of input features can be readily obtained during the training. Random forests provide two straight forward methods for feature selection. (i) ***Gini Importance or Mean decrease in Impurity (MDI)***: Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure, based on which the (locally) optimal condition chosen is called impurity.

Table 1. The Features used in this work and their dimension size

Features Used	**Dimension**
PSSM profile	20
Secondary Structure	21
Hydrophobicity	21
Polarity	21
Molecular Volume	21
Electrostatic Charge	21
Solvent Accessibility	2

Table 2. Impurity measures of classification problem

Impurity	**Task**	**Formula**	**Description**
Gini impurity	Classification	$\sum_{i=1}^{C} fi(1 - fi)$	f_i is the frequency of label *i* at a node and *C* is the number of unique labels.
Entropy	Classification	$\sum_{i=1}^{C} - fi \log(fi)$	f_i is the frequency of label *i* at a node and *C* is the number of unique labels.

Table 3. Pseudo code for random forest algorithm

```
Stacking Algorithm

Learn a function that combines the predictions of the individual classifiers

Train n different classifiers C1...Cn (the base classifiers)
❖ obtain predictions of the classifiers for the training examples
    ▪ Use cross-validation for efficient combination of data!
❖ form a new data set (the meta data)
    ▪ classes
        ♦ which should be the same as the original dataset
    ▪ attributes
        ♦ one attribute for each base classifier
        ♦ value is the prediction of this classifier on the example
❖ train a separate classifier M (the meta classifier)
```

For classification related problem, it is typically either Gini impurity or information gain/entropy and for regression trees it is variance. Thus when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure. (ii) ***Permutation Importance or Mean Decrease in Accuracy (MDA)***: This method directly measures the impact of each feature on accuracy of the model. This is assessed for each feature by removing the association between that feature and the target. It has been achieved by randomly permuting the values of the feature and measuring the resulting increase in error. The influence of the correlated features is also removed.

3.2 Model creation – Random forecast

In our study, each decision tree in the random forest has been trained to predict if the class labels contain any of the three secondary structures, α helix, β sheet, and a coil. A number of features used to construct each decision tree were randomly selected from the total input features. Random forest grows a large number of decision trees based on a subset of randomly selected features and a fraction of randomly selected training data points. All the trained trees are applied to a new data point to make prediction. The majority vote of the ensemble of trained decision trees is used as the final prediction for the data point. The decision trees were trained by the standard decision tree training algorithm that maximized the information gain in selecting a feature to partition the training data. After training, each tree (T) was able to predict the probability of each class. The average probability predicted by these trees was calculated and the class with higher predicted probability was the prediction. The average decision based on a large number of decision trees makes random forest robust against noisy data, irrelevant features, and unbalanced class distribution. The random forest method was implemented by using the ***randomForest, H2O*** packages using R programming.

The node impurity is a measure of the homogeneity of the labels at the node. The two impurity measures for classification are Gini impurity and entropy are shown in Table 2.

We have considered the below aspects during tree construction, for Performance and **Accuracy Improvement**:

- Similar classifier usually makes similar errors, so forming an ensemble with similar classifiers would not improve the classification rate.

- Poorly performing classifier may cause performance deterioration in the overall performance.
- Presence of a classifier that performs much better than all of the other available base classifiers may cause degradation in the overall performance.
- Another important factor is the amount of correlation among the incorrect classifications made by each classifier
- **NumTrees**:
  - ➢ Increasing the number of trees will decrease the variance in predictions, improving the model's test-time accuracy.
  - ➢ Training time increases roughly linearly in the number of trees.
- **MaxDepth**: Increasing the depth makes the model more expressive and powerful.
- **SubsamplingRate**: This parameter specifies the size of the dataset used for training each tree in the forest, as a fraction of the size of the original dataset. The default (1.0) is recommended, but decreasing this fraction can speed up training.
- **FeatureSubsetStrategy**: Number of features to use as candidates for splitting at each tree node. The number is specified as a fraction or function of the total number of features. Decreasing this number will speed up training, but can sometimes impact performance if too low.

3.3 Accelerating the classifier - Spark

The distributed framework approach used in this study is shown in Figure 1. The initial step extracts the input features including PSSM profile, Secondary Structure, Hydrophobicity, Polarity, Molecular Volume, Electrostatic Charge, and Solvent Accessibility. Random Forest Algorithm's default feature selection technique, Gini/Permutation is being used for extracting the top features and the decision trees have been constructed using Random Forest algorithm. Apache SPARK is used to accelerate the process and the best model comes out as classifier output with the predicted secondary structures.

A 64-bit Linux platform with two Intel Xeon 2.7 GHz CPUs, 8 cores each, 128 GByte RAM and R 3.1.2 was used for this computation. The runtime of *ranger* 0.2.5, *Random Jungle* 2.1.0, *randomForest* 4.6-10, *randomForestSRC* 1.6.1, *h2o* 3.6.0.8 and *bigrf* 0.1-11 was compared using *microbenchmark* 1.4-2. For visualization, *ggplot2* 1.0.0 was used. Profiling the scripts to get the total time taken is also measured using *Rprof*. The garbage collector was called with the function *gc()* after data preprocessing. R *multicore* package is also used to run randomForest algorithm.

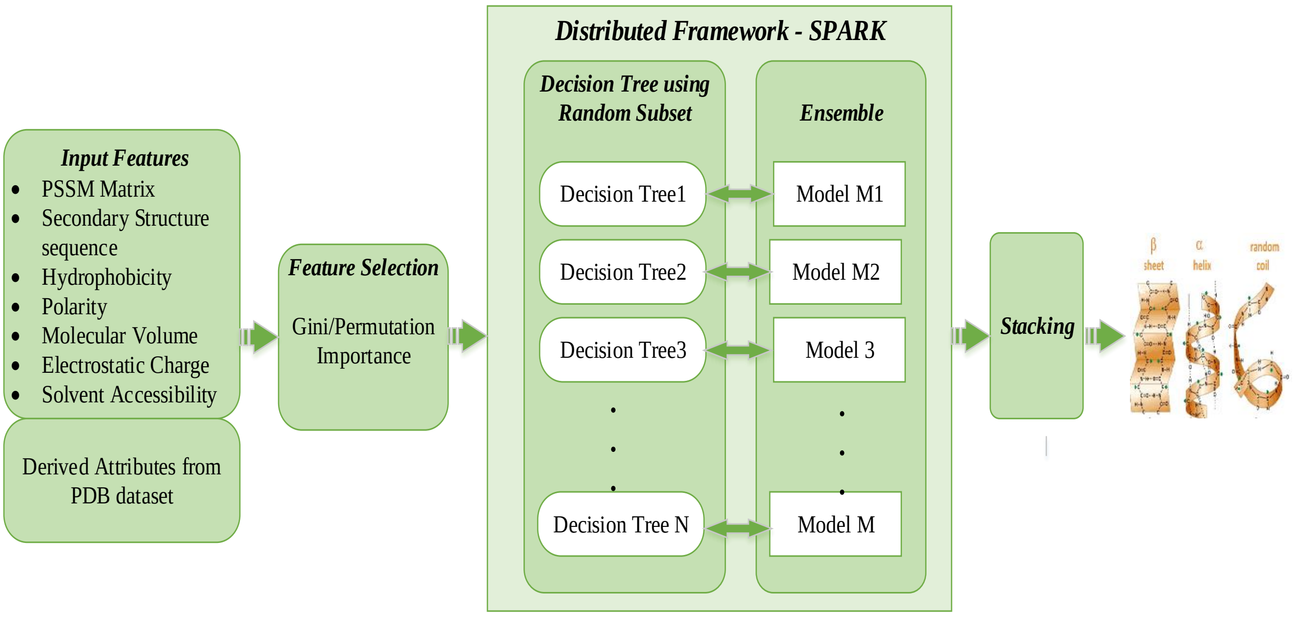

Figure.1 Random forest classifier using distributed SPARK framework

We have decided to further improve the performance by implementing this algorithm in a distributed framework using Spark. ***Apache Spark*** is a new framework for distributed parallel computation can speed up the iterative applications, such as machine learning, when the data is cached in memory [19]. In this research work, we attempt to address this issue by training multiple models using Spark, and combining their output results. This is called Ensemble Learning and can reduce the required primary memory for big data machine learning and potentially speeding up training and classification times. The main challenge is to train multiple individual decision trees over distributed data. Additionally, in Random Forest, decision trees are typically fully grown. So they get large particularly for a large data set. We used ***Apache SPARK Mllib*** along with few customizations to implement our Ensemble model.

4. Performance and accuracy evaluation

The performance of an ensemble mostly depends on the individual performance of the classifiers present in the ensemble. We have used the confusion matrix data for evaluating the classifier accuracy. Plots showing various parameters plotted against accuracy and performance measures.

4.1 Error rate

The number of features used for training/testing affects the performance of learning methods. We have evaluated on how the performance of the random forest changed with respect to the number of features used in training, which ranged from 1 to 125. Figure 2 depicts that the MSE value attains a constant level after adding more number of features and the accuracy of the classifier is directly proportional to the number of features.

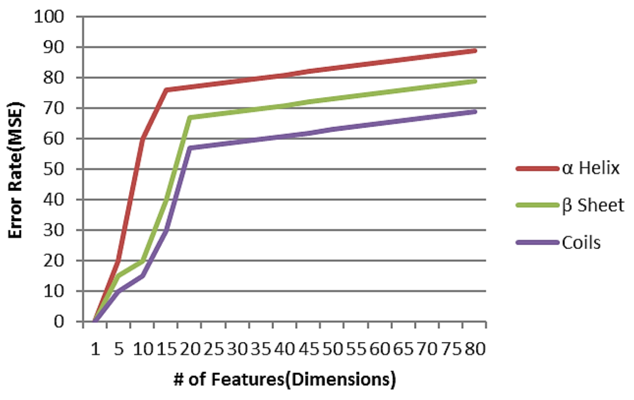

Figure.2 Plots for the error rate for structures Helix, Sheet, and Coils vs. number of features

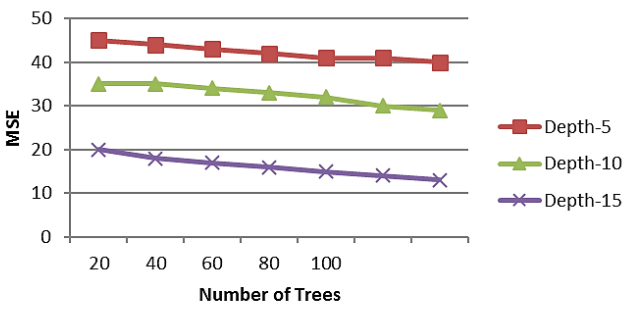

Figure.3 Plots of the error rate for various tree depths

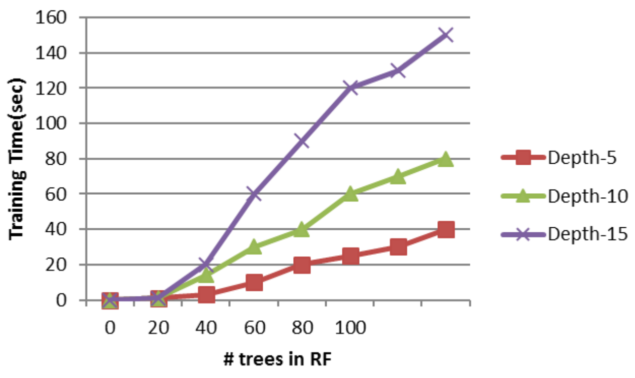

Figure.4 Impact of tree depth vs. time taken for training/testing

The above in Fig.3, for a basis for understanding the MSE, note that the left-most points show the error when using a Random Forest tree of various sizes (of depths 5, 10, or 15, respectively).

4.2 Performance

Performance of the distributed approach was achieved by modifying the various Spark cluster parameters. From Figure 4, it clearly shows that Random Forests with Deep Trees are consuming more processing time, when compared to shallow trees. We have used the depth values as 5, 10, 15 for this study and the respective graphs are shown.

The performance of the distributed approach is increased by adding more nodes. The plot captured against #Instances vs. Running Time is shown in Figure 5. The cluster performance increases rapidly by adding more nodes to the cluster.

4.3 Random forecast vs. decision tree

The constructed Random Forest consisting of 100 decision trees is compared with the single decision tree in terms of the error rate (i.e. percent of incorrectly classified protein structures). The error rate of the random forest classification was 0.65%, which was lower than 1.41% of a single tree.

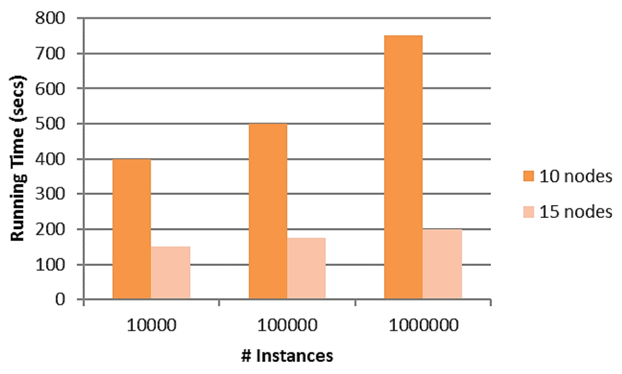

Figure.5 Improvements on increasing the number of cluster nodes

Table 4. Percentage of error rate on random forest and decision tree approach

Algorithm	Error in percentage
Random forest	0.655
Decision tree(single)	1.415

Table 5. Confusion matrix obtained using classifier

CONFUSION MATRIX Proportion of predicted secondary structures				
	Predicted			
		Helix(H)	Sheet(E)	Coil(C)
Actual	Helix(H)	75.80%	10.67%	8.77%
	Sheet(E)	15.57%	78.43%	3.68%
	Coil(C)	14.76%	2.97%	82.74%

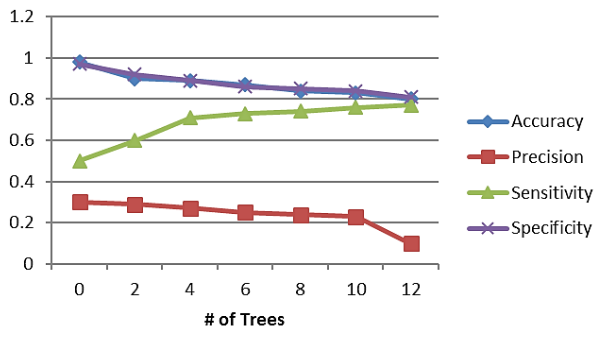

Figure.6 Specificity, Sensitivity, Precision Score are calculated across amino acid residues of all proteins in CB513 dataset

4.4 Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The TP, TN, FP, FN values for the 3 secondary structures obtained using our classifier is shown in Table 5.

The secondary structure Helix has the highest value of 0.912, 0.845, 0.352 for Specificity, Sensitivity, Precision respectively and has shown a high accuracy on CB513 dataset. Plot on showing these metrics are in Fig.6.

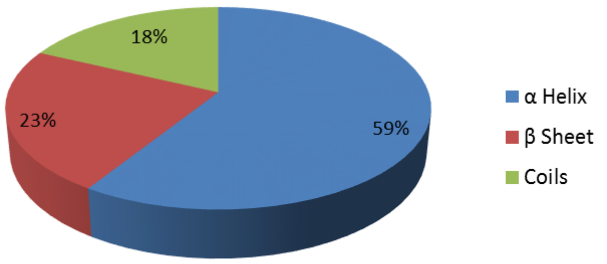

Figure.7 Pie chart showing various distribution of each of the 3 structures

4.5 Secondary structure composition

The composition of Helix, Sheet and Coils spreaded in our dataset is shown in the below pie chart, Figure 7. Among this α-Helix occurrence is about 59% followed by 23% of β Sheet and Coil of 18%.

We have performed an ablation study to discover the key features and parameters involved in our classifier. For this, we have removed and replaced the various components used in this analysis and tested the model performance by keeping different combinations of input features. From the results on CB513 dataset, PSSM along with Physiochemical properties has achieved 69.67% of accuracy and the results are convincing for normalized input. Also the Cluster based approach has increased the performance of up-to 68.20% when compared to sequential processing mode which was only 59.32%.

5. Conclusion

In this paper, we have proposed an Ensemble Based classifier to predict the secondary structure of protein. The relevant input dataset features are automatically extracted using the algorithm itself. We have analyzed the classifier accuracy with varied set of parameters by changing the input dataset features, tree parameters and generated the prediction results. Experiments are carried out using cross-validation tests on RS126 and CB513 benchmark datasets. The research result indicates that the prediction of secondary structures using this proposed algorithm achieved an accuracy of 85.2%, sensitivity of 82.43%, and specificity of 89%. Our results clearly confirm that ensembles are more accurate than other single model machine learning algorithm. This work is also implemented in distributed environment for their efficient computing capacity, and the performance scales well with the number of processors while implemented in distributed framework SPARK.

As a conclusion, Ensemble classifiers together with SPARK will be more considered for its accuracy and performance and enables to train many fully grown decision trees on big data. Future work involves on experimenting different computational approaches and by using other distributed frameworks to improve the accuracy and performance respectively of the results obtained, which is considered as one of the crucial phase in the life cycle of drug discovery. It will be helpful for the recent medicine researcher, which aids in understanding the relation between protein sequence and structure, thereby determine the function for developing drugs and designing novel enzymes and is one of the high focus problems in bioinformatics research today.

References

[1]Z. Wang, F. Zhao, J. Peng, and J. Xu, "Protein 8-class secondary structure prediction using conditional neural fields", *Proteomics*, Vol.11, No.19, pp.3786-92, 2011.

[2]J.J. Ward, L.J McGuffin, B.F. Buxton, and D.T. Jones, "Secondary structure prediction with support vector machines", *Bioinformatics*, Vol.19, No.13, pp.1650-1655, 2003.

[3]H. Bouziane, B. Messabih, and A. Chouarfia, "Effect of simple ensemble methods on protein secondary structure prediction", *Soft Computing*, Vol.19, No.6, pp.1663-78, 2015.

[4]A. Dehzangi, S. Phon-Amnuaisuk, and O. Dehzangi, "Using Random Forest for Protein Fold Prediction Problem: An Empirical Study", *Journal of Information Science and Engineering*, Vol.26, No.6, pp.1941-1956, 2010.

[5]P. Yang, Y. Hwa Yang, B. Zhou, and Y. Zomaya, "A review of ensemble methods in bioinformatics", *Current Bioinformatics*, Vol.5, No.4, pp.296-308, 2010.

[6]A.K. Johal, and R. Singh, "Protein secondary structure prediction using improved support vector machine and neural networks", *International Journal of Engineering and Computer Science*, Vol.3, No.1, pp.3593-3597, 2014.

[7]C. Zhang, H. De Sterck, A. Aboulnaga, H. Djambazian, and R. Sladek, "Case study of scientific data processing on a cloud using Hadoop", *High performance computing systems and applications Springer Berlin Heidelberg*, pp. 400-415, 2010.

[8]M. Rithvik, and G.N, Rao, "A Comparative Study of Methodologies of Protein Secondary Structure", *Computational Intelligence Techniques for Comparative Genomics Springer Singapore*, pp.37-45, 2015.

[9]A. Yaseen, and Y. Li, "Context-based features enhance protein secondary structure prediction accuracy", J*ournal of chemical information and modelling*, Vol.54, No.3, pp.992-1002, 2014.

[10]N. Mossos, D.F. Mejia-Carmona, and I. Tischer, "FS-Tree: Sequential Association Rules and First Applications to Protein Secondary Structure Analysis", *Advances in Computational Biology Springer International Publishing*, pp.189-198, 2014.

[11]W. Qu, H. Sui, B. Yang, and W. Qian, "Improving protein secondary structure prediction using a multi-modal BP method", *Computers in biology and medicine*, Vol.41, No.10, pp.946-59, 2011.

[12]S. Agarwal, P. Agarwal, and D. Mendiratta, "Prediction of Secondary Structure of Protein using Support Vector Machine", *International Journal of Computer Applications®(IJCA)*, pp.0975–8887, 2014.

[13]X. Dencelin, and T. Ramkumar, "Analysis of multilayer perceptron machine learning approach in classifying protein secondary structures", *Biomedical Research*, Vol.27,166-173,2016.

[14]H.M. Berman, T. Battistuz, T.N Bhat, W.F Bluhm, P.E Bourne, K. Burkhardt, Z. Feng, G.L. Gilliland, L. Iype, S. Jain, and P. Fagan, "The protein data bank", *Acta Crystallographica Section D Biological Crystallography*, Vol.58, No.6, pp.899-907, 2002.

[15]D.T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices", *Journal of molecular biology*, Vol.292, No.2, pp.195-202, 1999.

[16]Li D, Li T, P. Cong, W. Xiong, and J. Sun, "A novel structural position-specific scoring matrix for the prediction of protein secondary structures", *Bioinformatics*, Vol.28, No.1, pp.32-399, 2012.

[17]R.Y. Luo, Z.P. Feng, and J.K. Liu, "Prediction of protein structural class by amino acid and polypeptide composition", *European Journal of Biochemistry*, Vol.269, No.17, pp.4219-2425, 2002.

[18]N.C. Toussaint, C. Widmer, O. Kohlbacher, and G. Rätsch, "Exploiting physico-chemical properties in string kernels", *BMC bioinformatics*, Vol.11, No.8, pp.S7, 2010.

[19]M. Szczerba, M.S. Wiewiórka, M.J. Okoniewski, and H. Rybiński, “Scalable Cloud-Based Data Analysis Software Systems for Big Data from Next Generation Sequencing”, *In Big Data Analysis: New Algorithms for a New Society, Springer International Publishing,* pp. 263-283, 2016.